On Generalized Spectral Clustering: Theories and Algorithms

Shota Saito

Submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of
University College London.

Department of Computer Science
University College London

18th October, 2024

I, Shota Saito, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

This thesis considers generalizing spectral graph clustering. Spectral graph clustering exploits the spectrum of graph Laplacian. There is room for generalizations of spectral graph clustering in two directions. The first involves extending the graph Laplacian to the nonlinear p-Laplacian, with the expectation of enhancing performance. The second generalization extends from graphs to hypergraphs, which allows for richer modeling by connecting an arbitrary number of vertices in one edge. Despite recent advancements in these generalizations, there remains to be untapped potential for graph spectral clustering. This thesis addresses this gap by introducing three theoretical frameworks.

Firstly, we propose a unified class of hypergraph p-Laplacians that incorporates existing variants and novel generalizations. Although existing Laplacians have a similar structure, some Laplacians miss some key features. This framework provides a comprehensive foundation for all key features, applying to the entire class of hypergraph p-Laplacians.

Secondly, we consider how to model a hypergraph from vector data. While graph modeling using kernel functions is well-established, an equivalent framework for hypergraph modeling has not been established. We propose such a formulation and establish its theoretical foundations.

Thirdly, we propose a multi-class clustering algorithm leveraging the nonlinearity of p-Laplacian. Spectral clustering via p-Laplacian is difficult since it is difficult to obtain higher eigenvectors. Thus, we take an alternative approach using p-resistance induced by p-Laplacian. We develop a theory on p-resistance for practical use and its application to graph multi-class clustering.

Finally, as a fourth part, we extend our theoretical insights to develop a learning framework for vertex classification tasks, where we present a simple alternative approach to graph neural networks (GNNs). While GNNs are commonly used for this task, they often exhibit

biases towards homophilous information. Instead of overcoming GNNs' limitations, we propose an alternative approach aiming to mitigate these biases.

Impact Statement

Graphs are powerful tools for representing data across a wide range of domains, from social networks to chemical compounds and images. Spectral graph clustering is a common method for machine learning tasks associated with graphs.

This thesis extends spectral graph clustering through multiple generalizations, offering theoretical foundations that open up new avenues for future research. For example, as shown in Chapter 6, the theoretical advancements developed in this work are applied to vertex-with-features problems, an area typically outside the scope of spectral learning.

These generalizations not only provide deeper insights into standard spectral graph methods but also have the potential to influence research beyond the immediate field of graph spectral learning. For example, in Sec. 1.3, we highlight how these generalizations help clarify what is fundamental to spectral graph clustering by identifying properties that hold for both standard and generalized frameworks, such as Courant's min-max theorem. However, some properties, like orthogonality, do not generalize as well to cases like the p-Laplacian, providing valuable insights for future work.

Beyond academia, the methods developed in this thesis could impact real-world applications, such as community detection, biological networks, or recommendation systems. By offering more expressive models through generalized spectral learning, industries working with complex, high-dimensional data could benefit from the enhanced performance and flexibility provided by these methods. However, although our approach shares the applications beyond academia, since this is foundational work towards generalized graph learning and does not target any immediate application, we cannot foresee the shape of positive or negative societal impact that this thesis may have in the future.

UCL Research Paper Declaration Form

1. **1. For a research manuscript that has already been published** (if not yet published, please skip to section 2)**:**

   (a) **What is the title of the manuscript?**

      i. Hypergraph Modeling via Spectral Embedding Connection: Hypergraph Cut, Weighted Kernel k-means, and Heat Kernel [Saito, 2022]

      ii. Generalizing p-Laplacian: Spectral Hypergraph Theory and a Partitioning Algorithm [Saito and Herbster, 2023a]

      iii. Multi-class Graph Clustering via Approximated Effective p-Resistance [Saito and Herbster, 2023b]

   (b) **Please include a link to or doi for the work:**

      i. `https://doi.org/10.1609/aaai.v36i7.20787`

      ii. `https://doi.org/10.1007/s10994-022-06264-y`

      iii. `https://proceedings.mlr.press/v202/saito23a.html`

   (c) **Where was the work published?**

      i. Proceeding of AAAI Conference on Artificial Intelligence 36(7)

      ii. Machine Learning

      iii. Proceeding of International Conference on Machine Learning

   (d) **Who published the work?**

      i. AAAI

      ii. Springer

      iii. Proceedings of Machine Learning Research

(e) **When was the work published?**

i. 2022

ii. 2023

iii. 2023

(f) **List the manuscript's authors in the order they appear on the publication:**

i. Shota Saito

ii. Shota Saito and Mark Herbster

iii. Shota Saito and Mark Herbster

(g) **Was the work peer reviewed?**

i. Yes

ii. Yes

iii. Yes

(h) **Have you retained the copyright?**

i. No

ii. No

iii. No

(i) **Was an earlier form of the manuscript uploaded to a preprint server (e.g. medRxiv)? If 'Yes', please give a link or doi**

i. `https://arxiv.org/abs/2203.09888`

ii. No

iii. `https://arxiv.org/abs/2306.08617`

If 'No', please seek permission from the relevant publisher and check the box next to the below statement:

☑ *I acknowledge permission of the publisher named under 1d to include in this thesis portions of the publication named as included in 1c.*

2. **For a research manuscript prepared for publication but that has not yet been published** (if already published, please skip to section 3)**:**

(a) **What is the current title of the manuscript?**

i. ResTran: A GNN Alternative to Learn A Graph with Features [Saito et al., 2024]

(b) **Has the manuscript been uploaded to a preprint server 'e.g. medRxiv'? If 'Yes', please please give a link or doi:**

i. `https://openreview.net/forum?id=OXaWdXAhjW`

(c) **Where is the work intended to be published?**

i. The manuscript will appear at a non archival workshop "Machine Learning for Genomics Explorations" at International Conference on Learning Representations 2024

(d) **List the manuscript's authors in the intended authorship order:**

i. Shota Saito, Takanori Maehara, Mark Herbster

(e) **Stage of publication:** Appearing at a non-archival venue

3. **For multi-authored work, please give a statement of contribution covering all authors** (if single-author, please skip to section 4)**:** Shota Saito conceives most of the ideas, conducts the experiments, and writes the most of manuscripts. The other author(s) supervise.

4. **In which chapter(s) of your thesis can this material be found?** See Sec. 1.5

e-Signatures confirming that the information above is accurate (this form should be co-signed by the supervisor/ senior author unless this is not appropriate, e.g. if the paper was a single-author work)**:**

Candidate: Shota Saito
Date: 18 Oct, 2024

Supervisor : Mark Herbster
Date: 18 Oct, 2024

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Mark Herbster. I am truly thankful for his unwavering support and encouragement throughout the development of this thesis. His passion for the theory of machine learning has been a constant source of inspiration, and I could not have asked for a better mentor for my research in this field.

I extend my appreciation to Takanori Maehara for his helpful comment and mathematical inspiration. Chapter 6 is based on the joint work with him. I also thank Atsushi Miyauchi, Stephen Pasteris, Maximilian Theissen, Alberto Rumi, and Fabio Vitale for valuable discussions. I had fun working with them, which is not included in this thesis. Also, special thanks are due to James Robinson and Lisa Tse for being under the same "umbrella" of Mark. Additionally, I am grateful for Antonin Schrab, with whom I worked as a teaching assistant for the coursework. I would like to give my thanks to my friend Oki Hayashi for his constant encouragement during my study.

I would like to express my gratitude for my funding. This thesis has been made possible thanks to Huawei, who supported my Ph.D study at UCL. I am particularly grateful for the freedom to pursue my research independently, without any constraints from Huawei. I wish to emphasize that the views expressed in this thesis are solely my own. I would like to acknowledge John Shawe-Taylor for his help to arrange this funding. Additionally, I am also supported by Shigeta Educational Fund.

I thank my family for all the support through my Ph.D study. Finally, I give a big thank you for dedications of my cat to meow. Without his meow, I was not able to write any single line in this thesis.

Contents

List of Figures

List of Tables

List of Commonly Used Symbols

General Symbols

Symbol	Description
x	A real value
$\mathbf{x}$	A vector
x_i	The i-th element of a vector $\mathbf{x}$
X	A matrix
$X^\top$	The transpose of matrix X
X^{-1}	The inverse of regular matrix X
X^+	The pseudoinverse of matrix X
$\mathrm{trace}(X)$	The trace of matrix X
x_{ij}	The ij-th element of matrix X
$\mathbb{N}$	The set of positive integers
$\mathbb{R}$	The set of real numbers
$[k]$	A set of $\{1, 2, \ldots, k\}$ for $k \in \mathbb{N}$
$\mathbf{e}_i$	The i-th coordinate vector (See Eq. (1.1))
$\mathbf{1}$	The all one vector
I	The identity matrix
$\|\mathbf{x}\|$	A general norm for vector $\mathbf{x}$
$\|\mathbf{x}\|_2$	The 2-norm for vector $\mathbf{x}$
$\|\mathbf{x}\|_p$	The p-norm for vector $\mathbf{x}$
$\|\mathbf{x}\|_{\mathbf{r},p}$	The weighted p-norm whose weight is $\mathbf{r}$
$\langle \mathbf{x}, \mathbf{y} \rangle$	A general inner product for $\mathbf{x}, \mathbf{y}$
$\langle \mathbf{x}, \mathbf{y} \rangle_2$	The dot product for $\mathbf{x}, \mathbf{y}$
$\langle \mathbf{x}, \mathbf{y} \rangle_M$	The inner product for $\mathbf{x}, \mathbf{y}$ induced by a positive semi-definite matrix M
$\|M\|_{\mathrm{Fro}}$	The Frobenius norm for matrix M
$\|\|\|M\|\|\|$	A matrix norm for matrix M
$\exp(x)$	e^x, where e is a Napier's constant and x is a real value
$\mathcal{M}$	A manifold

Symbols Used for Graphs

Symbol	Description	Definition
G	Graphs and hypergraphs	Sec. 1.1[1]
V	A set of vertices of G	Sec. 1.1
E	A set of edges of G	Sec. 1.1[2]
n	The number of vertices, i.e. $n := \|V\|$	Sec. 1.1
m	The number of edges, i.e., $m := \|E\|$	Sec. 1.1
A	An $n \times n$ adjacency matrix for graph	Sec. 1.1
a_{ij}	A weight between vertices i and j	Eq. (1.2)
W	An $m \times m$ diagonal matrix for edge weights	Sec. 1.1
$\mathbf{w}$	The m dimensional edge weight vector	Sec. 1.1
w_ℓ	A weight of the ℓ-th edge, where $\ell \in E$.	Sec. 1.1
D	An $n \times n$ diagonal degree matrix	Sec. 1.1
d_i	A degree of the vertex i	Sec. 1.1
C	The incidence matrix	Sec. 1.1
L	The unnormalized graph Laplacian	Eq. (1.3)
λ_i	The i-th smallest eigenvalue for unnormalized graph Laplacian L	Sec. 1.1
$\boldsymbol{\psi}_i$	The eigenvector associated with i-th smallest eigenvalue for unnormalized graph Laplacian L	Sec. 1.1
L_N	The normalized graph Laplacian	Eq. (5.6)
$\lambda_{N,i}$	The i-th smallest eigenvalue for normalized graph Laplacian L_N	Sec. 1.1
$\boldsymbol{\psi}_{N,i}$	The eigenvector associated with i-th smallest eigenvalue for normalized graph Laplacian L_N	Sec. 1.1
k	The number of the clusters	Sec. 1.2
$\mathrm{Cut}(V_1, V_2)$	A cut between two set of vertices V_1 and V_2	Eq. (1.5)
$\mathrm{kCut}(\{V_\ell\}_{\ell=1}^k)$	A k-way cut for $\{V_\ell\}_{\ell=1}^k$	Eq. (1.5)
$\mathrm{RCut}(V_1, V_2)$	A ratio cut between two set of vertices V_1 and V_2	Eq. (1.7)
$\mathrm{kRCut}(\{V_\ell\}_{\ell=1}^k)$	A k-way ratio cut for $\{V_\ell\}_{\ell=1}^k$	Eq. (1.7)
Z	The indicator matrix	Eq. (1.10)
Z_R	The ratio indicator matrix	Eq. (1.10)
$\|\mathbf{x}\|_{G,2}$	A graph 2-seminorm for a vector $\mathbf{x}$	Eq. (1.13)
$\|\mathbf{x}\|_{G,p}$	A graph p-seminorm for a vector $\mathbf{x}$	Eq. (1.23)
Δ_p	The graph p-Laplacian	Eq. (1.23)
$\lambda_{p,i}$	The smallest i-th variational eigenvalue of Δ_p	Eq. (1.26)
$\boldsymbol{\psi}_{p,i}$	The eigenvector associated with i-th variational eigenvalue of Δ_p	Prop. 2.13

[1] This definition of G is for graphs. We overload this symbol for hypergraphs, see Sec. 2.5.1

[2] This definition of E is for graphs. We overload this symbol for hypergraphs, see Sec. 2.5.1

Symbol	Description	Definition
$\mathrm{NCut}(V_1, V_2)$	A normalized cut between two set of vertices V_1 and V_2	Eq. (2.1)
$\mathrm{kNCut}(\{V_\ell\}_{\ell=1}^k)$	A k-way normalized cut for $\{V_\ell\}_{\ell=1}^k$	Eq. (2.2)
$\mathrm{vol}(V)$	The volume for set of the vertices V	Sec. 2.1.1
Z_N	The normalized indicator matrix	Eq. (2.4)
∇	The graph gradient	Def. 2.2
$S_{G,2}(\mathbf{x})$	The Dirichlet energy for $\mathbf{x}$	Eq. (2.15)
div	The graph divergence	Def. (2.3)
$\mathrm{CCut}(U)$	The Cheeger cut between U and $U \backslash V$	Eq. (2.23)
h_2	Cheeger Constant for the graph G	Eq. (2.24)
∇_p	The graph gradient for p-Laplacian	Eq. (2.30)
$S_{G,p}(\mathbf{x})$	The p-Dirichlet energy for $\mathbf{x}$	Eq. (2.31)
$R_{G,p}(\mathbf{x})$	The Rayleigh quotient using graph p-seminorm for $\mathbf{x}$	Eq. (2.36)
$R^{(2)}_{G,p}(\mathbf{x})$	The Rayleigh quotient for the second eigenpairs for graph p-seminorm	Eq. (2.37)
$\gamma(B)$	The Krasnoselskii genus for a set B	Eq. (2.40)
h_k	The k-way Cheeger constant	Eq. (2.44)
$\kappa(\mathbf{x}, \mathbf{x}')$	The kernel function for $\mathbf{x}$ and $\mathbf{x}'$	Eq. (2.47)
K	The graph matrix for κ	Sec. 2.2.2
$\phi(\mathbf{x})$	A feature map	Sec. 2.2
$\mathcal{J}(\{C_\ell\}_{\ell=1}^k)$	The k-means objective for $(\{C_\ell\}_{\ell=1}^k$	Eq. (2.49)
$\mathcal{J}_\phi(\{C_\ell\}_{\ell=1}^k)$	The weighted kernel k-means using ϕ	Eq. (2.52)
Z_M	The indicator matrix for weighted kernel k-means	Eq. (2.54)
$\nabla^{(c)}$	The continuous gradient	Eq. (2.55)
$S^{(c)}_{G,2}(\mathbf{x})$	The continuous Dirichlet energy	Eq. (2.55)
$\Delta^{(c)}$	The continuous Laplace operator	Eq. (2.58)
$\mathcal{V}(M)$	The coordinate spanning set for a positive semidefininte matrix M	Eq. (2.68)
$r_{G,2}(i, j)$	The effective resistance between two vertices i and j for a graph G	Eq. (2.70)
$r_{G,p}(i, j)$	The effective p-resistance between two vertices i and j for a graph G	Eq. (2.74)
L_b	The shifted graph Laplacian	Eq. (6.7)
X_G	The ResTran of the data X	Eq. (6.8)
$r'_G(i, j)$	The extended resistance	Eq. (6.9)
$\mathcal{J}_G(\{V_\ell\}_{\ell=1}^k)$	The k-means objective for ResTran	Eq. (6.10)
$\mathcal{J}_R(\{V_\ell\}_{\ell=1}^k)$	The k-means objective using the effective resistance for the featureless setting	Eq.(6.11)

Symbols Used for Hypergraphs

Symbol	Description	Definition
D_v	The vertex degree matrix for hypergraph	Sec. 2.5.1
D_e	The edge degree matrix for hypergraph	Sec. 2.5.1
W_e	The edge weight matrix for hypergraph	Sec. 2.5.1
H	A hypergraph incidence matrix	Sec. 2.5.1
$\mathrm{Cut}_{ncl}(V_1, V_2)$	The hypergraph cut for edge-normalized clique type	Eq. (2.79)
$\mathrm{kNCut}_{ncl}(\{V_\ell\}_{\ell=1}^k)$	The hypergraph k-way normalized cut for edge-normalized clique type	Eq. (2.85)
$L_{2,ncl}$	The hypergraph Laplacian for edge-normalized clique type	Eq. (2.81)
A_{ncl}	The hypergraph adjacency matrix for edge-normalized clique type	Eq. (2.83)
$\mathrm{Cut}_{cl}(V_1, V_2)$	The hypergraph cut for edge-unnormalized clique type	Eq. (2.80)
$L_{2,cl}$	The hypergraph Laplacian for edge-unnormalized clique type	Eq. (2.82)
A_{cl}	The hypergraph adjacency matrix for edge-unnormalized clique type	Eq. (2.84)
$\mathrm{kNCut}_{cl}(\{V_\ell\}_{\ell=1}^k)$	The hypergraph k-way normalized cut for edge-unnormalized clique type	Eq. (2.86)
$\mathrm{Cut}_s(V_1, V_2)$	The hypergraph cut for star type	Eq. (2.90)
$\mathrm{kNCut}_s(\{V_\ell\}_{\ell=1}^k)$	The hypergraph k-way normalized cut for star type	Eq. (2.86)
$L_{2,s}$	The hypergraph Laplacian for star type	Eq. (2.92)
A_s	The hypergraph adjacency matrix for star type	Eq.(2.94)
$\mathrm{Cut}_{TV}(V_1, V_2)$	The hypergraph cut for total variational type	Eq. (2.98)
$\mathrm{CCut}_{TV}(V_1)$	The hypergraph Cheeger cut for total variational type	Eq. (2.99)
$S_{G,p}^{TV}(\mathbf{x})$	The hypergraph energy of $\mathbf{x}$ for total variational type	Eq. (2.100)
$S_{G,p}^{SUB}(\mathbf{x})$	The hypergraph energy of $\mathbf{x}$ for submoudlar type	Eq. (2.101)

Symbol	Description	Definition
$\nabla_{c,p}$	Hypergraph gradient for the unified framework	Def. 3.1
$c(i,j,e,\mathbf{x})$	The weighting function of the hypergraph gradient	Def. 3.1
$\mathrm{div}_{c,p}$	Hypergraph divergence for the unified framework	Def. 3.2
$\Delta_{c,p}$	Hypergraph p-Laplacian for the unified framework	Def. 3.3
$S_{G,c,p}(\mathbf{x})$	Hypergraph p-energy for the unified framework	Eq. (3.8)
$\lambda_{c,p,\ell}$	The ℓ-th variational eigenvalues of hypergraph p-Laplacian for the unified framework	Sec. 3.2.3
$\boldsymbol{\psi}_{c,p,\ell}$	The eigenvector associated with ℓ-th variational eigenvalues of hypergraph p-Laplacian for the unified framework	Sec. 3.2.3
$R_{G,c,p}(\mathbf{x})$	The Rayleigh quotient for the unified framework	Eq. (3.10)
$\mathrm{CCut}_c(U)$	The Cheeger hypergraph cut of U for the unified framework	Eq. (3.13)
$h_{c,k}$	The Cheeger constant for the unified framework	Eq. (3.15)
$\mathcal{A}$	Adjacency tensor for r-uniform hypergraph	Sec. 4.2
$\kappa^{(r)}$	The biclique kernel for r-uniform hypergraph	Eq. (4.5)
$\mathcal{K}$	The gram tensor for biclique kernel	Sec. 4.3.1
$K^{(r)}$	The gram matrix for the biclique kernel	Sec. 4.3.1

Acronyms and Abbreviations

Abbreviation	Meaning
GNN	Graph Neural Network
NeuralNet	Neural Network
k-NN	k Nearest Neighbours
ResTran	Resistance Transformation
SSL	Semi-supervised Learning
SVM	Support Vector Machine
LP	Label Propagation
GCN	Graph Convolutional Network
GAT	Graph Attention Network
AVAE	Auxiliary Variational Auto Encoder
VAT	Variational Adversarial Training

Chapter 1

Introduction

A graph is a discrete data structure consisting of vertices and edges, where an edge connects two vertices. Graphs serve as powerful tools for representing data in various domains, social networks [Newman, 2006], images [Shi and Malik, 2000], and chemical molecules [Trinajstic, 2018]. Vertex clustering is one of the fundamental problems of graphs in the machine learning area. A well-established method for vertex clustering is spectral clustering, which leverages the spectral properties of a matrix known as the *graph Laplacian* [Chung, 1996]. The use of the graph Laplacian is motivated by its theoretically appealing spectral properties for clustering tasks [von Luxburg, 2007]

Despite its success, there are opportunities for generalizing spectral clustering in several directions. One direction is extending the graph Laplacian to the p-Laplacian, a nonlinear generalization. Introducing parameter p is known to improve the clustering performance [Bühler and Hein, 2009]. The motivation for this generalization can be strengthened by drawing an analogy to physics. In the continuous domain, the Laplace operator in calculus serves as the continuous counterpart to the discrete graph Laplacian [Belkin and Niyogi, 2003]. Similarly, the nonlinear extension of this operator, the p-Laplace operator, serves as the continuous counterpart to the graph p-Laplacian and has been extensively studied [Lindqvist, 2008]. The p-Laplace operator frequently arise in physics, particularly in fluid dynamics, where the parameter p characterizes fluid viscosity. When $p = 2$, the fluid behaves as an "ideal" fluid, unaffected by viscosity, whereas for $p \neq 2$, p represents the viscosity of the fluid [Lê, 2006, Astarita and Marrucci, 1974]. Analogously, in the discrete graph setting, the parameter p is expected to capture key structural characteristics of graphs, much like how it captures the viscosity of fluids. It is intuitive that not all graphs are expected to behave in an "ideal"

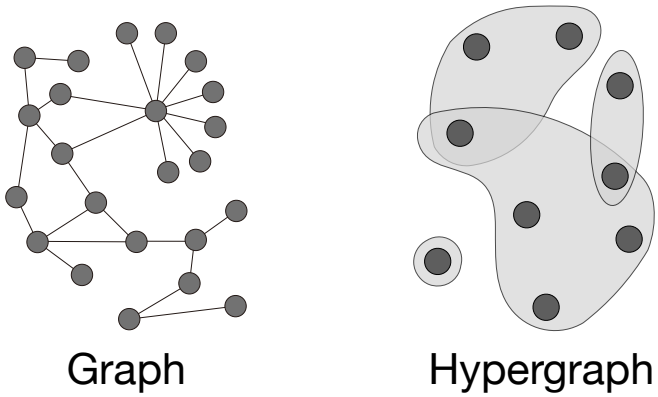


Figure 1.1: Illustration of a graph and a hypergraph. In both figures, circles denote vertices. In the graph (left), edges are represented as lines connecting pairs of vertices. In contrast, the hypergraph (right) uses a gray circle to connect arbitrary sets of vertices, representing edges of hypergraph.

manner, and this nonlinearity provides a valuable degree of flexibility in modeling.

The second direction is broadening the scope from graphs to their generalization hypergraphs. While an edge in a graph connects two vertices, an edge in a hypergraph can connect an arbitrary number of vertices, as illustrated in Fig. 1.1. This allows hypergraphs to provide more expressive models than standard graphs [Berge, 1984]. In real-world applications, hypergraphs have been used to model complex data, such as videos [Huang et al., 2009], web browsing histories [Mobasher et al., 2000], and molecular interactions in cells [Klamt et al., 2009]. Additionally, hypergraphs are known to offer improved performance over standard graphs in such settings [Zhou et al., 2006, Ghoshdastidar and Dukkipati, 2014].

Another direction is generalizing the task; we consider a vertex classification task, with not only looking at a graph but also exploiting "features" over a graph. For instance, a citation network where vertices represent papers and edges represent citations can also incorporate features such as the content of papers alongside the network topology. The standard methods for this problem is graph neural networks (GNNs) [Gori et al., 2005, Kipf and Welling, 2016a, Veličković et al., 2018].

In spite of recent advancements, there are avenues for these generalizations that remain untapped potential of the graph spectral clustering framework. This thesis addresses this gap by introducing several theoretical frameworks.

The first part considers a broader class of hypergraph p-Laplacians. In the past many

different hypergraph Laplacians were proposed, such as star Laplacian [Zhou et al., 2006] and clique Laplacian [Rodriguez, 2002, Saito et al., 2018]. While these prior Laplacians have similar properties, they derive a patchwork of nodal domain theorems, Cheeger inequalities, and partitioning algorithms for some particular cases of hypergraph p-Laplacians. Thus, we propose a unified class of hypergraph p-Laplacians that incorporates existing hypergraph Laplacians as a special case and includes previously unstudied novel generalizations. Both our theory and our partitioning algorithm apply to the complete class.

The second part considers how to model a hypergraph from vector data. While graph modeling using kernel functions is well-established from a weighted kernel k-means view [Dhillon et al., 2004] and a heat kernel view [Belkin and Niyogi, 2003], an equivalent framework for hypergraph modeling is not established. We propose a formulation of hypergraph modeling from vector data and establish its theoretical foundations.

The third part proposes an alternative multi-class clustering algorithm leveraging the nonlinearity inherent in the graph p-Laplacian. The drawback of the graph p-Laplacian is that the third and higher eigenvectors are difficult to obtain [Lindqvist, 2008], hindering multi-class spectral clustering. To exploit the nonlinearity, we take an alternative approach; we are motivated to use the p-resistance induced by the p-Laplacian. We then develop a theory on p-resistance for practical use and its application to graph multi-class clustering.

Finally, the fourth part extends our theoretical insights to develop a learning framework for vertex classification tasks, where presenting a simple alternative approach to GNNs. While GNNs are common for this task, they often exhibit biases towards homophilous information [Li et al., 2018, Oono and Suzuki, 2019]. Instead of overcoming GNNs' limitations, we propose an alternative approach aiming to mitigate these biases.

1.1 Basic Notations and Graph Definition

Before we explain the motivation of this thesis, we introduce the basic notations and graph definitions we use throughout in this thesis.

We define $\mathbb{N}$ as a set of positive integers (i.e., $1, 2, 3, \ldots$), $\mathbb{R}$ as a set of real values, and $\mathbb{R}_+$ is a set of positive real values. For a number $k \in \mathbb{N}$, we write $[k] := \{1, \ldots, k\}$. We write a vector as a small bold letter $\mathbf{a}$ and its i-th element as a_i. If a vector has an index such as $\mathbf{a}_\ell$, we write i-th element of $\mathbf{a}_\ell$ as $(\mathbf{a}_\ell)_i$. For a matrix A, we write ij-th element of A as a_{ij}. Similarly to the vector notation, if we write a matrix with some index, such as A_ℓ, we write

ij-th element of A_ℓ as $(A_\ell)_{ij}$. We denote the i-th column of the matrix A by $A_{\cdot i}$, and i-th row by $A_{i\cdot}$. Let $\mathbf{e}_i$ be the i-th coordinate vector, i.e.,

$$(\mathbf{e}_i)_\ell := \begin{cases} 1 & \text{if } \ell = i \\ 0 & \text{otherwise,} \end{cases} \tag{1.1}$$

and $\mathbf{1}$ as an all-one vector. Also, we write a matrix I as an identity matrix. We also define M^+ as the pseudoinverse of M, and M^{-1} as the inverse of M. Let $A^\top$ denote a transpose of A. Also, we define $\mathrm{trace}(M)$ as the trace of regular matrix M. Let $\|\mathbf{a}\|$ be a norm for a vector $\mathbf{a}$ induced from an inner product $\langle \cdot, \cdot \rangle$. We write the 2-norm for a vector $\mathbf{a}$ as $\|\mathbf{a}\|_2 := (\sum_i a_i^2)^{1/2}$, and this 2-norm is generalized to a p-norm as $\|\mathbf{a}\|_p := (\sum_i a_i^p)^{1/p}$. We define the Frobenius norm for a matrix A as $\|A\|_{\mathrm{Fro}} := (\sum_{ij} a_{ij}^2)^{1/2}$.

Let $G = (V, E, \mathbf{w})$ be a weighted undirected graph, where $V := [n]$ is a set of vertices and $E := [m]$ is a set of edges equipped with a weight vector $\mathbf{w} \in \mathbb{R}^m_+$. The vertices V are also called as *nodes*. In this thesis, we keep using vertices to mention V. An edge connects two vertices, and we do not consider the direction of the edge (*undirected*). We define a *weight matrix* as a diagonal matrix $W \in \mathbb{R}^{m \times m}$ associated with the weight vector $\mathbf{w}$ with ℓ-th diagonal element w_ℓ. In the definition of the incidence matrix $C \in \mathbb{R}^{m \times n}$, $c_{\ell i}$ and $c_{\ell j}$ are set to 1 and -1, respectively, for the edge ℓ connecting vertices i and j ($i > j$), otherwise 0. We represent a graph by an *adjacency matrix* $A \in \mathbb{R}^{n \times n}$;

$$a_{ij} = a_{ji} := \begin{cases} w_\ell & (\text{if the edge } \ell \text{ connects vertices } i \text{ and } j), \\ 0 & (\text{otherwise}). \end{cases} \tag{1.2}$$

the ij-th element and ji-th element of A are w_ℓ if the ℓ-th edge connects vertices i to j, i.e., $a_{ij} = a_{ji} := w_\ell$, and we define $a_{ij} = a_{ji} := 0$ if there is no edge between vertices i and j. Remark that the adjacency matrix A is symmetric by its construction. A *degree* d_i for a vertex i is defined as $d_i := \sum_j a_{ij}$. We define a degree matrix D, a diagonal matrix whose diagonal elements are $D_{ii} := d_i$. Let a *(unnormalized) Graph Laplacian* L be

$$L := D - A. \tag{1.3}$$

The graph Laplacian can also be written as $L = C^\top W C$.

We also define *normalized graph Laplacian* as

$$L_N = D^{-1/2}(D - A)D^{-1/2}. \tag{1.4}$$

We denote i-th smallest eigenvalues of the graph Laplacian by λ_i, and the eigenvector associated with λ_i by $\boldsymbol{\psi}_i$. We also denote i-th smallest eigenvalues of the normalized graph Laplacian by $\lambda_{N,i}$, and the eigenvector associated with $\lambda_{N,i}$ by $\boldsymbol{\psi}_{N,i}$. Note that most of the discussion in this thesis holds for both unnormalized and normalized graph Laplacian in a similar manner. For a more detailed understanding of graph theory basics, refer to [Bapat, 2010].

1.2 Why Spectral Clustering Matters

This section discusses why spectral clustering matters.

A clustering, in general, is the task where we group a set of "similar" data points; data points in the same group are more similar to each other than to a data point in the other group [Bishop and Nasrabadi, 2006, Murphy, 2012]. Clustering is unsupervised learning, i.e., it does not need the labels to learn. Common clustering algorithms in the Euclidean space are hierarchical clustering such as the single-linkage clustering, centroid-based clustering such as k-means, and density-based clustering such as the Gaussian Mixture Model.

Turning to graph clustering, graph cut is one of the simplest methods. We consider the partition of V into two distinct sets, V_1 and $V \backslash V_1$. Such a partition should happen when a "dissimilarity" of two sets is the smallest. In the graph theory language, such a dissimilarity is called as *cut*; a total weight of edges removed. We define the cut for a two-class partition as

$$\mathrm{Cut}(V_1, V \backslash V_1) := \sum_{i \in V_1, j \in V \backslash V_1} a_{ij}. \tag{1.5}$$

We consider the partition of V into k distinct sets, $V_1, \ldots, V_k$, where $V_i \cap V_j = \emptyset$ for $i \neq j$ and $\cup_{i \in [k]} V_i = V$. We generalize the two-way cut to *k-way cut* as

$$\mathrm{kCut}(\{V_\ell\}_{\ell=1}^k) := \sum_{\ell \in [k]} \mathrm{Cut}(V_\ell, V \backslash V_\ell) = \sum_{\ell \in [k]} \sum_{i \in V_\ell, j \in V \backslash V_\ell} a_{ij}, \tag{1.6}$$

which is the sum of the weight edges between different vertex sets. This objective function

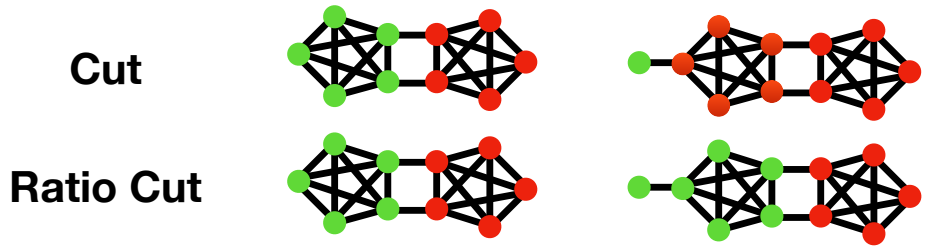


Figure 1.2: Illustration of the cut and ratio cut on two example graphs. The red and green colors represent the clustering results. In the case of the cut, the clustering is sensitive to minor changes, as adding just one vertex can lead to undesirable results. In contrast, the ratio cut provides more robust and desirable clustering outcomes for both graphs

can be minimized in polynomial time [Goldschmidt and Hochbaum, 1994]. Particularly, for the $k = 2$ case, there exists a simple and fast algorithm [Stoer and Wagner, 1997]. However, it is known that in practice, this often leads to the partitioning of one individual vertex from the rest of the graph [von Luxburg, 2007]. This result is not the one we would like to obtain since we expect that the size of each set is reasonably large.

In order to prevent such situations, we consider to penalize if the size of the set is small. For this aim, we define a balanced cut, called *ratio cut*, as

$$\mathrm{RCut}(V_1, V\backslash V_1) := \mathrm{Cut}(V_1, V\backslash V_1)\left(\frac{1}{|V_1|} + \frac{1}{|V\backslash V_1|}\right). \tag{1.7}$$

The difference between the cut and this ratio cut is the "balance term",

$$\left(\frac{1}{|V_1|} + \frac{1}{|V\backslash V_1|}\right), \tag{1.8}$$

by which if $|V_1|$ is small the objective function is penalized. Hence, this ratio cut aims to "balance" the size of the clusters more than the cut, since only looking at this balanced term, the balanced term is minimized when $|V_1| = |V\backslash V_1|$ This balance term prevents the undesirable case which we see in the cut; if we partition a graph into one vertex and the rest of the graph, the "one-vertex-only" cluster penalizes the objective function. In Fig. 1.2, we demonstrate this with two example graphs. On the left, both the cut and ratio cut produce desirable clustering results. However, on the right, when one vertex is added, the cut objective results in a "one-vertex cluster," whereas the ratio cut continues to provide a robust clustering

solution.

We then extend this ratio cut to k-way ratio cut as

$$\mathrm{kRCut}(\{V_\ell\}_{\ell=1}^k) := \sum_{\ell\in[k]} \frac{\mathrm{Cut}(V_\ell, V\backslash V_\ell)}{|V_\ell|}. \tag{1.9}$$

While this ratio cut is known to empirically improve the cut as seen in Fig. 1.2, this ratio cut problem is NP-hard [Wagner and Wagner, 1993]. Thus, we consider to relax the problem as follows.

We first define the indicator matrix $Z \in \mathbb{R}^{n\times k}$ as

$$z_{i\ell} := \begin{cases} 1 & \text{when } i \text{ in } V_\ell \\ 0 & \text{otherwise,} \end{cases} \tag{1.10}$$

and also defined the ratio indicator matrix $Z_R \in \mathbb{R}^{n\times k}$ as

$$Z_R := Z(Z^\top Z)^{-1/2}. \tag{1.11}$$

Note that $Z_R^\top Z_R = I$. Note also that since Z_R can be written as

$$(Z_R)_{i\ell} = \begin{cases} 1/\sqrt{|V_\ell|} & \text{when } i \text{ in } V_\ell \\ 0 & \text{otherwise,} \end{cases} \tag{1.12}$$

the matrix Z_R can be seen as an indicator vector normalized by the square root of size of the cluster. Using Z and Z_R, we can rewrite Eq. (1.9) as

$$\mathrm{kRCut}(\{V_\ell\}_{\ell=1}^k) = \sum_{\ell=1}^k \frac{\|Z_{\cdot\ell}\|_{G,2}^2}{\|Z_{\cdot\ell}\|_2^2}, \text{where } \|\mathbf{x}\|_{G,2}^2 := \sum_{i,j\in V} a_{ij}|x_i - x_j|^2 = \mathbf{x}^\top L\mathbf{x} \tag{1.13}$$

$$= \mathrm{trace}(Z_R^\top L Z_R) \tag{1.14}$$

Note that $\|\cdot\|_{G,2}$ is a seminorm, since $L\mathbf{1} = 0$. The common technique is to relax Z_R from discrete values to real values and use $Z_R^\top Z_R = I$ as constraints. By relaxing Z_R, we see that Eq. (1.13) is connected to eigenvalues problem as follows;

Algorithm 1 Spectral Clustering.

Input: Adjacency matrix A, and the number of clusters k
Compute the graph Laplacian $L = D - A$.
Obtain $\Psi_k := (\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_k)$, the smallest k eigenvectors of L.
Obtain the non-overlapping k sets by treating each of the n rows in Ψ_k as a point in $\mathbb{R}^k$, and run k-means with k clusters
Output: The non-overlapping k sets $V_1, \ldots, V_k$

Proposition 1.1 ("Foundation" of spectral clustering, e.g., von Luxburg [2007])**.**

$$\min_{Z_R \in \mathbb{R}^{n \times k}} \{\mathrm{kRCut}(\{V_\ell\}_{\ell=1}^k) \text{ s.t. } Z_R^\top Z_R = I\} = \sum_{\ell=1}^k \lambda_\ell, \tag{1.15}$$

where λ_ℓ is ℓ-th eigenvalue of L. The solution is given by taking $Z_R = (\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_k)$, where $\boldsymbol{\psi}_\ell$ is ℓ-th eigenvector of L.

This holds due to the Rayleigh-Ritz theorem, where

$$\min_{Z_R \in \mathbb{R}} \{\mathrm{trace}(Z_R^\top L Z_R) \text{ s.t. } Z_R^\top Z_R = I\} = \sum_{\ell=1}^k \lambda_\ell. \tag{1.16}$$

For more details, see Sec. 5.2 in [von Luxburg, 2007]. This proposition serves as a "foundation" of spectral clustering. From this proposition, the eigenproblem of the graph Laplacian L can be seen as a relaxed problem of minimizing the ratio cut. Since the eigenproblem can be solved in polynomial time while minimizing ratio cut is NP-hard, we use the eigenvectors as a relaxed solution of ratio cut. We then recover the clusters from k smallest eigenvectors of L, often by k-means algorithms. We call this procedure as *spectral clustering*, summarized as Alg. 1.

To conclude the discussion above, the balanced cut can be a "good" objective for graph clustering. However, the balanced cut is an NP-hard problem. Spectral clustering is a relaxed solution of the graph cut and thus serves as an approximation of the balanced cut. More details can be found in Chapter 2.

Finally, we briefly mention other graph clustering methods. The cut objective function defined as Eq. (1.6) is well studied, and various cut-based algorithms are proposed; see Aggarwal and Wang [2010]. Similarly to the cut objective function, correlational clustering is established [Bansal et al., 2004]. Another popular method is distance-based clustering, where

we define a distance between vertices over a graph and apply any distance-based clustering methods [Williams et al., 1964, Hennig and Hausdorf, 2006]. Also, there exist statistical inference methods using generative models, such as the planted partition model [Condon and Karp, 2001, McSherry, 2001] and the degree-corrected stochastic block model [Newman, 2016]. The hierarchical clustering is another established method particularly in the community detection community, using a measure of edge-betweenness [Girvan and Newman, 2002] and modularity [Newman, 2006, Blondel et al., 2008]. We remark that although various methods are proposed, sometimes a method turns out to be equivalent to the other methods. Notably, spectral clustering is equivalent to the other popular methods, such as modularity maximization and statistical inference using a planted partition model [Newman, 2013]. This thesis proves that spectral clustering is also equivalent to the distance-based algorithm using the effective resistance as a distance in Sec 6.4.2.1. See Schaeffer [2007] for a more comprehensive survey of graph clustering. Note also that there exists a research line on overlapping community detection, where we are allowed to assign multiple clusters to one vertex. However, in this thesis, we limit our interests to the non-overlapping setting, i.e., we are only allowed to assign one cluster to one vertex. See Xie et al. [2013] for more about overlapping community detection.

1.3 Why Generalizations Matter: A View from Mystical Power of Twoness

This thesis mainly discusses generalizations of spectral clustering. While these generalizations themselves are significant contributions, we now highlight why generalizations matter from a different view, "mystical power of twoness." This view is Eugene L. Lawler's favorite observation: a problem is easy if the number of parameters is two, but if there are more, the problem becomes very hard [Lenstra, 1998]. For example, a two-dimensional matching problem can be solved in polynomial time, but three-dimensional matching problems are NP-hard, which is actually first proven by Lawler. This matching problem is proven by the two-dimensional matching problem, which can be reduced to a polynomially solvable problem while the three-dimensional problem cannot.

While Lawler observed this in the combinatorial optimization realm, we also observe similar phenomena in many areas. For example, the eigenproblems of a tensor are NP-hard,

while the ones for a matrix (2-tensor) are polynomial time problems [Hillar and Lim, 2013]. Another example is a logic problem called a boolean satisfiability problem (SAT); the SAT for three or more literals is NP-hard, while the SAT for two is solvable in polynomial time [Karp, 2010]. An example in physics is that we observe that while it is difficult to obtain a closed-form solution for k-body problems when $k \geq 3$, it is easy to obtain the closed-form solution for the two-body problem [Whittaker, 1964]. Like these examples, we see the "mystical power of twoness" in many areas; the "two" problems are easy, while "non-two" problems are difficult for some reason.

Looking at the spectral clustering in Sec. 1.2, we may say that the standard graph problem is a "2-norm problem for a 2-uniform hypergraph." In the thesis, we generalize this from 2-norm to p-norm or 2-uniform hypergraph to general hypergraph. We argue that generalizing the graph Laplacian offers a better understanding of the standard graph Laplacian. We observe what is essential in the graph Laplacian, i.e., which properties hold under both the $p = 2$ and the general p cases. Furthermore, we seek to discern which properties are unique to the $p = 2$ scenario but do not generalize to other p values. We may see that properties exclusive to $p = 2$ rely on "twoness." Given the "two" problems are easy, these properties make the problems coincidentally easy.

Below, by the example of graph Laplacian and p-Laplacian, the Laplacian's nonlinear generalization, we illustrate how the "twoness" of the graph resides in the graph cut problem using 2-seminorm and its extension to p-seminorm. Particularly, we discuss that while it is easy to obtain k eigenvectors of graph Laplacian, it is hard to do so for p-Laplacian.

We first review the following basic facts on the pair of eigenvectors of graph Laplacian.

Proposition 1.2 (classical)**.** *Let $\boldsymbol{\psi}_i, \boldsymbol{\psi}_j$ be a pair of eigenvectors corresponding to distinct eigenvalues of L. Thus, a pair of eigenvectors of L is orthogonal to each other, i.e.,*

$$\boldsymbol{\psi}_i^\top \boldsymbol{\psi}_j = 0. \tag{1.17}$$

Roughly speaking, a pair of eigenvectors of L is orthogonal to each other because L is symmetric and linear. In other words, without symmetricity and linearity of L, a pair of eigenvectors of L is not orthogonal. Since the proof is straightforward but well demonstrates how the symmetricity and linearity contributes the orthogonality, we provide the proof in Sec. 1.3.1. As seen later, this orthogonality is built on "twoness."

This orthogonality (Prop. 1.2) allows us to further rewrite Eq. (1.15) as follows.

Proposition 1.3. *The following successive sequence can obtain the eigenpair of the graph Laplacian as*

$$\lambda_\ell = \min_{\mathbf{x}\perp\boldsymbol{\psi}_1,\ldots,\boldsymbol{\psi}_{\ell-1}} \frac{\|\mathbf{x}\|_{G,2}^2}{\|\mathbf{x}\|_2^2}, \quad \boldsymbol{\psi}_\ell = \operatorname*{arg\,min}_{\mathbf{x}\perp\boldsymbol{\psi}_1,\ldots,\boldsymbol{\psi}_{\ell-1}} \frac{\|\mathbf{x}\|_{G,2}^2}{\|\mathbf{x}\|_2^2}. \tag{1.18}$$

The alternative sequence is

$$\lambda_\ell = \max_{\mathbf{x}\perp\boldsymbol{\psi}_n,\ldots,\boldsymbol{\psi}_{\ell+1}} \frac{\|\mathbf{x}\|_{G,2}^2}{\|\mathbf{x}\|_2^2}, \quad \boldsymbol{\psi}_\ell = \operatorname*{arg\,max}_{\mathbf{x}\perp\boldsymbol{\psi}_n,\ldots,\boldsymbol{\psi}_{\ell+1}} \frac{\|\mathbf{x}\|_{G,2}^2}{\|\mathbf{x}\|_2^2}. \tag{1.19}$$

This proposition lets us rewrite the k-way ratio cut problem Eq. (1.9) to obtain the relaxed solution recursively as Eq. (1.18). This relation also leads to the efficient algorithm to compute the eigenpair of the graph Laplacian, called Lanczos method [Lanczos, 1950], whose details are in Sec. 10.1 in the established textbook [Golub and Van Loan, 2013].

It is known that we may further generalize Prop. 1.3 as follows.

Proposition 1.4 (Variational Theorem)**.** *The following sequence*

$$\lambda_\ell = \min_{\substack{U\subseteq\mathbb{R}^n \\ \dim(U)=\ell}} \max_{\mathbf{x}\in U} \frac{\|\mathbf{x}\|_{G,2}^2}{\|\mathbf{x}\|_2^2}. \tag{1.20}$$

admits the eigenvalue of the graph Laplacian L. The solution is given by $\boldsymbol{\psi}_\ell$.

See [Struwe, 2000] for the details of this proposition. The difference between Prop. 1.3 and Prop. 1.4 is as follows. The constraints of the optimization in Prop. 1.4 is given as a dimension. On the contrary, the constraints in Prop. 1.3 is given as the pairwise relationship between two eigenvectors; that is, each pair of eigenvectors is orthogonal. In fact, Prop. 1.3 is tighter than Prop. 1.4; using Prop. 1.2, each pair of eigenvectors of L is orthogonal, we may specify U in Eq. (1.20) as

$$U \perp \boldsymbol{\psi}_n, \ldots, \boldsymbol{\psi}_{\ell+1}, \tag{1.21}$$

that reduces Prop. 1.4 to Prop. 1.3.

Now, we consider a natural generalization from 2-seminorm to p-seminorm. Following Bühler and Hein [2009], the natural p-seminorm of the graph seminorm Eq. (1.13) is to define

$$\|\mathbf{x}\|_{G,p}^p := \sum_{i,j\in V} a_{ij}|x_i - x_j|^p. \tag{1.22}$$

We then define the p-Laplacian as

$$(\Delta_p \mathbf{x})_i := \sum_{j\in V} a_{ij}|x_i - x_j|^{p-1}. \tag{1.23}$$

Remark that when $p = 2$, this p-Laplacian reduces to the standard graph Laplacian, i.e.,

$$\Delta_2 \mathbf{x} = L\mathbf{x}. \tag{1.24}$$

Also, the eigenpair of p-Laplacian $(\lambda, \boldsymbol{\psi})$ is defined to satisfy

$$(\Delta_p \boldsymbol{\psi})_i = \lambda|\psi_i|^{p-2}\psi_i. \tag{1.25}$$

The natural idea of spectral clustering is to use the first k eigenvectors of p-Laplacian. In fact, we may obtain in a similar variational manner as Prop. 1.4 as follows.

Proposition 1.5 (Variational Theorem for Δ_p, e.g., [Struwe, 2000])**.** *Let $\gamma(B)$ be a Krasnoselskii genus of a set B, whose formal definition is given later in Eq.* (2.40)*. The following*

$$\lambda_{p,\ell} = \min_{\substack{U\subseteq\mathbb{R}^n \\ \gamma(U)=\ell}} \max_{\mathbf{x}\in U} \frac{\|\mathbf{x}\|_{G,p}^p}{\|\mathbf{x}\|_p^p} \tag{1.26}$$

admits the eigenvalues of Δ_p for $\ell = 1, \ldots, n$. The solution is given by the eigenvector of Δ_p.

This proposition is a generalization of Prop. 1.4; we generalize the dimension in the sequence Eq. (1.20) to Krasnoselskii genus, which is a generalization of dimension. This genus gives "dimensions" to a set residing in not only the Euclidean space but also *any* space. We call the eigenpairs generated from this sequence as *variational eigenpairs*. Note that when $p = 2$, this sequence reduces to Prop. 1.4. Note also that there generally exist the eigenpairs of p-Laplacian other than variational ones. We will discuss the details of this sequence in

Sec. 2.1.4.

Now, we see the natural generalization of the variational theorem from the standard Laplacian (Prop. 1.4) to p-Laplacian (Prop. 1.5). Looking back to the $p = 2$ case, orthogonality conditions make the variational theorem simpler, from Prop. 1.4 to Prop. 1.3. Also, the simpler sequence leads to the Lanczos algorithm, an efficient algorithm to compute the eigenvalues. Thus, a natural question to ask is if we have an even simpler form of Prop. 1.5 for the general p case, like Prop. 1.3? The answer is, unfortunately, no. The reason is that the orthogonality conditions come from linearity, which only holds for $p = 2$ case and hence does not apply to general p-Laplacian. In fact, we do not have any workarounds for this issue at this point; while we know the identifications for the first and second eigenpairs, we do not know how to obtain the third or higher eigenpairs of p-Laplacian [Lindqvist, 2008].

To conclude the examples above, we find that obtaining eigenpairs is practically easy for the standard graph Laplacian but difficult for the general p. The eigenproblem of the graph Laplacian L exhibits a "twoness," which simplifies the eigenproblem, as shown from Prop. 1.4 to Prop. 1.3. This simplification leads to the Lanczos algorithm, an efficient algorithm to obtain k-eigenpairs of the graph Laplacian. On the other hand, for the p-Laplacian, the problem cannot be further simplified from Prop. 1.5 and thus remains difficult.

Now, we observe that generalizations may provide a better understanding of the standard case: what is "foundational," i.e., well-generalized, and what is built on "twoness." In this example, the variational theorem is foundational, while the practical algorithm to obtain higher-order eigenpairs is built on twoness.

1.3.1 Demonstration of Linearity and Symmetricity via Proof of Proposition 1.2

We here provide the proof of Prop. 1.2, which we omitted for the sake of readability. We introduce this proof since it demonstrates how linearity and symmetricity contribute to orthogonality.

Since $\boldsymbol{\psi}_i, \boldsymbol{\psi}_j$ are eigenvectors of L, we have

$$L\boldsymbol{\psi}_i = \lambda_i \boldsymbol{\psi}_i, \tag{1.27}$$

$$L\boldsymbol{\psi}_j = \lambda_j \boldsymbol{\psi}_j. \tag{1.28}$$

We then have

$$\boldsymbol{\psi}_j^\top L \boldsymbol{\psi}_i = \lambda_i \boldsymbol{\psi}_j^\top \boldsymbol{\psi}_i, \tag{1.29}$$

$$\boldsymbol{\psi}_i^\top L \boldsymbol{\psi}_j = \lambda_j \boldsymbol{\psi}_i^\top \boldsymbol{\psi}_j. \tag{1.30}$$

Since L is symmetric and linear, we have

$$(\boldsymbol{\psi}_j^\top L \boldsymbol{\psi}_i)^\top = \boldsymbol{\psi}_i^\top L \boldsymbol{\psi}_j. \tag{1.31}$$

Thus, by subtracting Eq. (1.30) from Eq. (1.29), we have

$$0 = \lambda_i \boldsymbol{\psi}_j^\top \boldsymbol{\psi}_i - \lambda_j \boldsymbol{\psi}_i^\top \boldsymbol{\psi}_j = (\lambda_i - \lambda_j) \boldsymbol{\psi}_i^\top \boldsymbol{\psi}_j. \tag{1.32}$$

From the assumption that $\lambda_i \neq \lambda_j$, we have $\boldsymbol{\psi}_i^\top \boldsymbol{\psi}_j = 0$. Thus, we see that a pair of the eigenvectors corresponding distinct eigenvalues of L are orthogonal. Looking back this explanation, the key for this relationship is Eq. (1.31), which holds because L is symmetric and linear.

1.4 Structure of this Thesis

We discuss the both directions of generalizations; from 2-seminorm to p-seminorm as well as from a graph to hypergpraph. In Chapter 2, we discuss the preliminaries of this thesis. In Chapter 3, we first generalize in both directions, from 2-seminorm to p-seminorm, as well as a graph to hypergraph to grasp an overview. We propose a unified framework that encompasses various existing hypergraph p-Laplacians while also accommodating novel formulations. The chapter presents theoretical properties, including the nodal domain theorem and Cheeger inequality, for this abstract class of hypergraph p-Laplacians. In Chapter 4, we then discuss the hypergraph Laplacian for the $p = 2$ case. This chapter presents a framework for transforming vector data into hypergraphs for spectral clustering. We also provide the theoretical justifications for our representation. In Chapter 5, we then explore the graph p-seminorm realm, reflecting on the limitations both of graph p-Laplacians and hypergraph p-Laplacians, where it is difficult to obtain higher-order eigenvectors. We propose using p-resistance and apply it to multi-class clustering. We then develop an approximation guarantee for p-resistance for practical use. Finally, using the theoretical insights in the previous chapters,

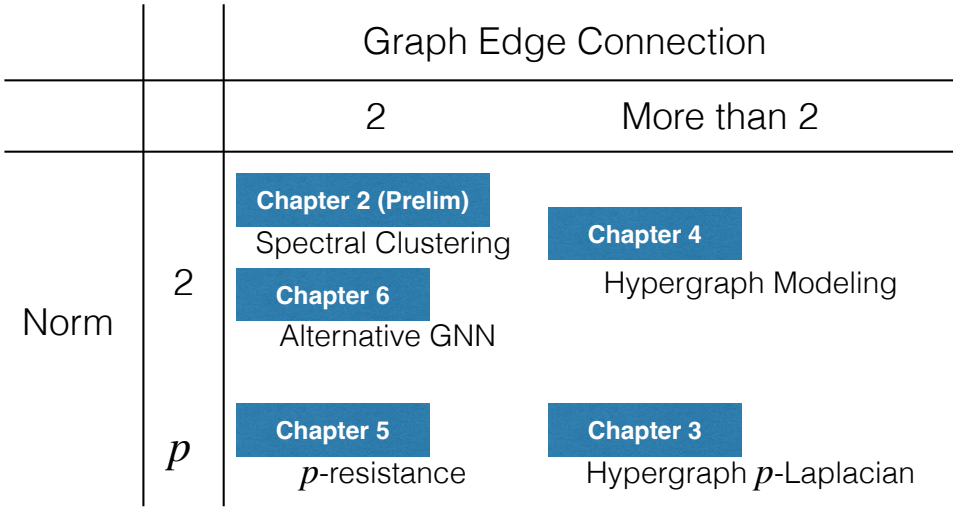


Figure 1.3: Structure of this thesis.

in Chapter 6, we propose a data representation framework for the vertex classification task, which not only considers the topological graph structure but also incorporates features over vertices. This representation serves as an alternative approach to GNNs, aiming to mitigate these biases. We establish such a framework exploiting the theoretical properties of graph Laplacian. Chapter 7 concludes this thesis. In Fig. 1.3, we illustrate how a part of Chapter 2, and from Chapter 3 to Chapter 6 can be structured.

1.5 List of Publications

This thesis is based on the following publications.

- S. Saito and M. Herbster. Generalizing p-Laplacian: spectral hypergraph theory and a partitioning algorithm. Mach. Learn., 112(1):241–280, 2023 (Chapter 3)
- S. Saito. Hypergraph modeling via spectral embedding connection: Hypergraph cut, weighted kernel k-means, and heat kernel. In Proc. AAAI, pages 8141–8149, 2022 (Chapter 4)

- S. Saito and M. Herbster. Multi-class graph clustering via approximated effective p-resistance. In Proc. ICML, pages 29697–29733, 2023 (Chapter. 5)

The following manuscript appears at a non-archival workshop.

- S. Saito, T. Maehara, M. Herbster. ResTran: A GNN Alternative To Learn Graph With Features, MLGenX Workshop of ICLR, 2024 (Chapter 6)

Chapter 2

Preliminaries

This chapter overviews prominent work on spectral learning for graph and hypergraph. Sec. 2.1 extends the discussion from Sec. 1.2 and Sec. 1.3, by reviewing spectral clustering methods for graphs. Sec. 2.2 discusses how to apply spectral clustering for given vector data, instead of the discrete graph. Sec. 2.3 reviews the brief history of the spectral clustering. Sec. 2.4 then explores the analogy between circuit theory and graphs. Next, Sec. 2.5 reviews topics related spectral clustering for hypergraph. Lastly, Sec. 2.6 summarizes this chapter. This chapter introduces topics that are covered across multiple subsequent chapters. In some cases, those subsequent chapters will include additional preliminary discussions of concepts specific to their content.

2.1 Spectral Clustering for Graphs

In Sec. 1.2 and Sec. 1.3, we introduced spectral clustering. We further review additional topics related to spectral clustering for graphs. Sec. 2.1 covers multiple topics, such as normalized cut, Cheeger inequality, spectral connections, as well as p-Laplacian.

2.1.1 Normalized Graph Cut and Spectral Clustering

In Sec. 1.2, we see the connection between ratio graph cut and spectral clustering. In this section, we explain that the similar discussion holds for normalized graph cut and spectral clustering using the normalized graph Laplacian.

Same as Sec. 1.2, we consider partitioning a graph G into two vertices sets V_1,$V_2 \subset V$, $V_1 \cap V_2 = \varnothing$ and $V_1 \cup V_2 = V$. We define the *normalized cut* as

$$\mathrm{NCut}(V_1, V_2) := \mathrm{Cut}(V_1, V_2) \left(\frac{1}{\mathrm{vol}(V_1)} + \frac{1}{\mathrm{vol}(V_2)} \right), \tag{2.1}$$

where $\mathrm{vol}(V) := \sum_{i \in V} d_i$. Minimizing this can partition the vertex set into two subsets. We extend to k-way partitioning problem where we partition into k subsets, $V_i (i = 1, \ldots, k)$, where $V_i \cap V_j = \varnothing$ if $i \neq j$ and $\cup_{i=1}^{k} V_i = V$, formulated as

$$\mathrm{kNCut}(\{V_\ell\}_{\ell=1}^{k}) := \sum_{i=1}^{k} \frac{\mathrm{Cut}(V_\ell, V \backslash V_\ell)}{\mathrm{vol}(V_\ell)}. \tag{2.2}$$

Similarly to the ratio indicator matrix Eq. (1.11), we introduce a normalized indicator matrix Z_N as

$$Z_N := D^{1/2} Z (Z^\top D Z)^{-1/2}, \tag{2.3}$$

Note that $Z_N^\top Z_N = I$. We also note that since Z_N can be written as

$$Z_N = \begin{cases} \sqrt{d_i / \mathrm{vol}(V_j)} & \text{if the vertex } i \text{ belongs to } j \\ 0 & \text{otherwise,} \end{cases} \tag{2.4}$$

the matrix Z_N can be seen as an indicator vector weighted by the degree $\sqrt{d_j}$ and normalized by $\mathrm{vol}(V_j)$ so that $\|(Z_N)_{\cdot j}\|_2 = 1$. Using this indicator matrix, this problem can also be rewritten as

$$\mathrm{kNCut}(\{V_\ell\}_{\ell=1}^{k}) = \mathrm{trace}(Z_N^\top L_N Z_N) \tag{2.5}$$

$$= \sum_{j=1}^{k} \frac{(Z_N)_{\cdot j}^\top L_N (Z_N)_{\cdot j}}{\|(Z_N)_{\cdot j}\|_2^2}. \tag{2.6}$$

Minimizing normalized cut is again a discrete optimization problem, and this is known to be NP-hard [Shi and Malik, 2000]. However, if we relax Z_N into real value, minimizing these becomes an eigenproblem of graph Laplacians L_N, which shows that the graph cut problem can be written as an eigenproblem as follows.

Proposition 2.1 ("Foundation" of normalized spectral clustering, e.g., von Luxburg [2007])**.**

$$\min_{Z_N \in \mathbb{R}^{n \times k}} \{\mathrm{kNCut}(\{V_\ell\}_{\ell=1}^{k}) \text{ s.t. } Z_N^\top Z_N = I\} = \sum_{\ell=1}^{k} \lambda_{N,\ell}, \tag{2.7}$$

Algorithm 2 Normalized Spectral Clustering.

Input: Adjacency matrix A, and the number of clusters k
Compute the graph Laplacian $L_N = I - D^{-1/2}AD^{-1/2}$.
Obtain $\Psi_{N,k} := (\boldsymbol{\psi}_{N,1}, \ldots, \boldsymbol{\psi}_{N,k})$, the smallest k eigenvectors of L_N.
Obtain the non-overlapping k sets by treating each of the n rows in $\Psi_{N,k}$ as a point in $\mathbb{R}^k$, and run k-means with k clusters
Output: The non-overlapping k sets $V_1, \ldots, V_k$

where $\lambda_{N,\ell}$ is ℓ-th eigenvalue of L_N. The solution is given by taking $Z_N = (\boldsymbol{\psi}_{N,1}, \ldots, \boldsymbol{\psi}_{N,k})$, where $\boldsymbol{\psi}_{N,\ell}$ is ℓ-th eigenvector of L_N.

Similarly to the ratio cut discussed as Prop. 1.1 in Sec. 1.2, we see that the eigenvectors of L_N is connected to the normalized graph cut problem. We establish the similar procedure to obtain the clustering result as *normalized spectral clustering*, summarized as Alg. 2.

Now we observe the similar structure of the normalized graph Laplacian to the one of the graph Laplacian. Most of the discussion in this thesis holds both for unnormalized Laplacian and normalized Laplacian, since the most of thesis is based on this spectral clustering discussion.

Finally, only for the normalized Laplacian, the eigenproblem can be rewritten as follows.

$$\min \mathrm{kNCut}(\{V_\ell\}_{\ell=1}^k) = \min_{Z_N} \mathrm{trace}(Z_N^\top L_N Z_N^\top) \text{ s.t. } Z_N^\top Z_N = I, \tag{2.8}$$

$$= \max_{Z} \mathrm{trace}(Z_N^\top D^{-1/2} A D^{-1/2} Z_N) \text{ s.t. } Z_N^\top Z_N = I, \tag{2.9}$$

This rewriting holds since $L_N = I - D^{-1/2}AD^{-1/2}$. Using Eq. (2.9) and relaxing Z_N into real values, Eq. (2.9) can be written as the k largest eigenproblems of $D^{-1/2}AD^{-1/2}$. This type of transition of equation does not hold for unnormalized Laplacian, i.e.,

$$\min_{Z_R \in \mathbb{R}^{n\times k}} \{\mathrm{trace}(Z_R^\top L Z_R^\top) \text{ s.t. } Z_R^\top Z_R = I\} \neq \max_{Z_R \in \mathbb{R}^{n\times k}} \{\mathrm{trace}(Z_R^\top A Z_R) \text{ s.t. } Z_R^\top Z_R = I\}. \tag{2.10}$$

2.1.2 Analogy between Continuous Laplacian and Graph Laplacian

In this section, we establish a differential geometric analogy between the continuous and graph domains, as the graph Laplacian shares several key properties with the continuous Laplace operator. This analogy provides a better understanding of graph Laplacian and lays

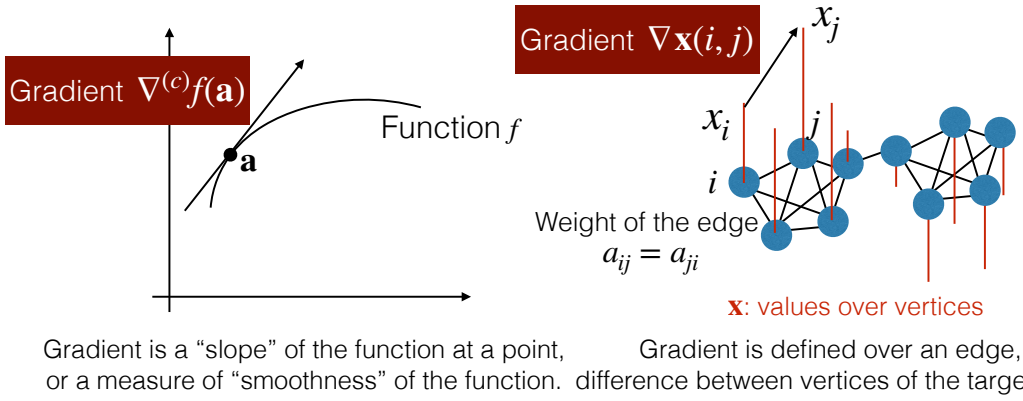


Figure 2.1: Illustration of gradients over the Euclidean space and a graph. On the left, the gradient $\nabla^{(c)}$ is represented as the slope at the point $\mathbf{a}$ in the Euclidean space, where $\nabla^{(c)}$ is a continuous gradient. On the right, the gradient corresponds to the difference between the values of vertices connected by the target edges (i, j). In both cases, the gradient reflects the "smoothness" of the functions.

the foundations for exploring its connection to the heat equations (Sec. 2.2.4) and further generalizations to p-Laplacian (Sec. 2.1.4).

We firstly define $\mathcal{H}(V)$ as a Hilbert space of real-valued functions endowed with the usual inner product

$$\langle \mathbf{x}, \mathbf{x}' \rangle_{\mathcal{H}(V)} \coloneqq \sum_{i \in V} x_i x'_i \tag{2.11}$$

for all $\mathbf{x}, \mathbf{x}' \in \mathcal{H}(V)$. Accordingly, the Hilbert space $\mathcal{H}(E_d)$ is defined with the inner product

$$\langle \mathbf{y}, \mathbf{y}' \rangle_{\mathcal{H}(E_d)} \coloneqq \sum_{\ell \in E_d} s_\ell t_\ell, \tag{2.12}$$

for all $\mathbf{y}, \mathbf{y}' \in \mathcal{H}(E_d)$, where E_d is a set of directed edges, i.e., we distinguish (i, j) and (j, i). We use this space E_d regardless if the underlying graph is undirected or directed. Note that $\mathbf{y}$ and $\mathbf{y}'$ are defined for directed edges.

We shall now define discrete gradient and divergence operators studied in standard graphs, which can be considered graph analogs in both discrete and continuous case [Zhou and Schölkopf, 2006].

Definition 2.2. *The graph unnormalized gradient is an operator* $\nabla \colon \mathcal{H}(V) \to \mathcal{H}(E_d)$ *defined*

by

$$(\nabla \mathbf{x})(i,j) := \sqrt{a_{ij}}\,(x_j - x_i) \tag{2.13}$$

for $(i,j) \in E_d$.

By definition, the gradient operator is linear. Also, from the definition, we have

$$(\nabla \mathbf{x})(i,j) \neq (\nabla \mathbf{x})(i,j) \tag{2.14}$$

for general $\mathbf{x}$, which is why we consider E_d although the graph we consider is undirected. Loosely speaking, the norm of the gradient at a graph vertex quantifies the local smoothness or roughness of the function around that vertex. In Fig. 2.1, we compare the graph gradient with its continuous counterpart, the gradient in Euclidean space, $\nabla^{(c)}$. As shown in Fig. 2.1, both gradients capture the smoothness of their respective function or values.

By using the norm defined by the Hilbert space in Eq.(2.12), we define *Dirichlet sum* of $\mathbf{x} \in \mathcal{H}(V)$ as

$$S_{G,2}(\mathbf{x}) := \|\nabla \mathbf{x}\|_2^2 = \sum_{i,j \in V} |(\nabla \mathbf{x})(i,j)|^2. \tag{2.15}$$

Since the norm of the gradient measures the local smoothness of the function around the vertex in a rough sense, the Dirichlet sum intuitively captures the total smoothness across the entire graph. We should emphasise that $\|\nabla \mathbf{x}\|$ is defined in the space $\mathcal{H}(E_d)$ as $\|\nabla \mathbf{x}\| = \langle \nabla \mathbf{x}, \nabla \mathbf{x} \rangle_{\mathcal{H}(E_d)}^{1/2}$, and satisfies $S_{G,2}(\mathbf{x}) = \|\nabla \mathbf{x}\|_2^2$.

Definition 2.3. *The graph divergence is an operator* $\mathrm{div} : \mathcal{H}(E_d) \to \mathcal{H}(V)$ *which satisfies*

$$\langle \nabla \mathbf{x}, \mathbf{y} \rangle_{\mathcal{H}(E_d)} = \langle \mathbf{x}, -\mathrm{div}\mathbf{y} \rangle_{\mathcal{H}(V)}, \ \forall \mathbf{x} \in \mathcal{H}(V), \forall \mathbf{y} \in \mathcal{H}(E_d). \tag{2.16}$$

Notice that Eq. (2.16) can be regarded as a graph analog of Stokes' Theorem on manifolds. The divergence can now be written in a closed form as follows:

Proposition 2.4. *Let* $\ell \in E_d$ *be an edge* (i,j) *and* $\ell' \in E_d$ *be an edge* (j,i). *The graph*

divergence can be computed as

$$(\mathrm{div}\mathbf{y})(i) = -\sum_{\ell \in E_d \text{ connects } i,j} \sqrt{a_{ij}} y_\ell + \sum_{\ell' \in E \text{ connects } j,i} \sqrt{a_{ji}} y_{\ell'}. \tag{2.17}$$

Recall that w_ℓ is a weight for ℓ-th edge, i.e., $w_\ell = a_{ij}$ if the ℓ-th edge connects vertices i and j. Note that this allows us to use Eq. (2.17) as a definition of the divergence; it satisfies Eq. (2.16), analogously to Stokes' theorem in the continuous case. Note also that divergence is always 0 if $\mathbf{y}$ is undirected.

We next define graph Laplacian as follows.

Definition 2.5. *The Laplacian is an operator* $\Delta_2 \colon \mathcal{H}(V) \to \mathcal{H}(V)$ *defined by*

$$\Delta_2 \mathbf{x} \coloneqq -\mathrm{div}(\nabla \mathbf{x}). \tag{2.18}$$

Observe that this operator is linear; by substituting Eq.(2.13) and Eq.(2.17) into the definition (2.18) the Laplace operator for undirected graph becomes

$$(\Delta_2 \mathbf{x})(i) = -\sum_{j \in V} a_{ij} \left(x_j - x_i\right). \tag{2.19}$$

Note that the first term of Eq. (2.17) vanishes due to the symmetry property of the gradient. The Laplacian in (2.18) can be rewritten as

$$(\Delta_2 \mathbf{x}) = (D - A)\mathbf{x} \tag{2.20}$$

$$= L\mathbf{x}. \tag{2.21}$$

Note that if we change the gradient to

$$(\nabla \mathbf{x})(i,j) \coloneqq \sqrt{a_{ij}} \left(\frac{x_j}{\sqrt{d_j}} - \frac{x_i}{\sqrt{d_i}} \right), \tag{2.22}$$

the corresponding Laplacian is normalized.

2.1.3 Cheeger Inequality

This section reviews the Cheeger inequality for the standard 2-Laplacian case. This Cheeger inequality is one of the established results from the analogy between continuous Laplace operator and discrete Laplacian. The Cheeger inequality for the continuous domain is established, and the Cheeger inequality for discrete domain is particularly useful for spectral clustering because it provides a quality guarantee for the partitioning.

Influenced by the inequality of the eigenvalue of continuous Laplacian called *Cheeger inequality*, there is existing research on the Cheeger inequality in the discrete domain [Alon and Milman, 1985, Lee et al., 2014, Tudisco and Hein, 2018]. This inequality shows the connection between the eigenproblem of Laplacian and a graph cut called as *Cheeger cut* [Alon and Milman, 1985]. This inequality motivates us to use eigenvectors to partition in the following way. While the discrete optimization problem of finding a subset of V that minimizes the Cheeger cut is NP-hard [von Luxburg, 2007], the eigenproblem of the graph Laplacian is not NP-hard. Since the Cheeger inequality gives bounds between eigenvalues of graph Laplacian and optimal Cheeger cut, the Cheeger inequality can guarantee the performance of the eigenproblem compared to the ground truth from the original cut problem. This performance guarantee enables us to use eigenvectors obtained by less computationally expensive eigenproblems instead of the costly ground truth from the discrete cut problem. Moreover, we may say that the Cheeger inequality "connects" Cheeger cut and eigenproblem; the Cheeger inequality shows how much we approximate the original graph cut problem by relaxing this into the real-valued eigenproblem of Laplacian.

We observe the "connection" as follows. We first discuss the unnormalized graph Laplacian L. Let $U \subset V$ be a set and $\overline{U}$ be a complement of U. The unnormalized *Cheeger cut* CCut may be defined as

$$\mathrm{CCut}(U) := \frac{\mathrm{Cut}(U, \overline{U})}{\min(|U|, |\overline{U}|)}, \tag{2.23}$$

The optimal values h_2 for CCut is called as *Cheeger constant*, i.e.,

$$h_2 := \min_{U \subset V} \mathrm{CCut}(U). \tag{2.24}$$

We have the following connection between eigenvalues of unnormalized Laplacian and the

Cheeger constant.

Theorem 2.6 (Cheeger Inequality [Alon and Milman, 1985])**.**

$$\frac{h_2^2}{2\max_i d_i} \leq \lambda_2 \leq 2h_2. \tag{2.25}$$

Thm. 2.6 shows how we approximate the Cheeger constant by relaxing the original discrete cut problem into the real-valued eigenproblem of the graph Laplacian. The Cheeger inequality guarantees the performance of the cut resulting from algorithms using the second eigenvector of Laplacian as follows.

Proposition 2.7 (Chung [2007])**.** *Let $(B, \overline{B})$ be the cut found by the second eigenvector of the Laplacian ψ, i.e.,*

$$B := B_{t'}, \text{ where } t' := \arg\min_t \mathrm{CCut}(B_t), B_t := \{i : (\boldsymbol{\psi}_2)_i \geq t\} \tag{2.26}$$

Then, we have

$$\mathrm{CCut}(B) < \sqrt{2\lambda_2(\max_i d_i)}. \tag{2.27}$$

Plugging the upper bound of Cheeger inequality Eq. (2.25) to this proposition, we observe

$$\mathrm{CCut}(B) < 2\sqrt{h_2(\max_i d_i)}, \tag{2.28}$$

This inequality guarantees the worst case of the performance of spectral clustering. The discussion above motivates us to use spectral methods for graph partitioning problems.

The above discussion naturally generalizes to the normalized setting. The Cheeger cut and Cheeger inequality are also naturally generalized to k-way partitioning, which we will discuss in Sec. 2.1.4.3.

2.1.4 Spectral Clustering via Graph p-Laplacian

This section discusses spectral clustering via graph p-Laplacian. In Sec. 2.1.1, we use the eigenvectors of graph Laplacian for spectral clustering. This eigenproblem is connected to

the Rayleigh quotient to the 2-seminorm induced by the graph Laplacian. The natural idea is to extend this 2-seminorm framework to p-seminorm. This extension is known to improve the performance [Bühler and Hein, 2009]. In the following, we discuss the generalization to p-seminorm.

2.1.4.1 Graph p-Laplacian Definition

This section generalizes graph Laplacian to a nonlinear operator called p-Laplacian.

We now redefine the p-Laplacian using the gradient (Eq. 2.13) and divergence (Eq. (2.16));

Definition 2.8. *An operator $\Delta_p \colon \mathcal{H}(V) \to \mathcal{H}(V)$ is a graph p-Laplacian if*

$$\Delta_p \mathbf{x} := -\mathrm{div}(\|\nabla_p \mathbf{x}\|^{p-2} \nabla_p \mathbf{x}), \tag{2.29}$$

where ∇_p is a similar gradient as Eq. (2.22)*, i.e.,*

$$\nabla_p(\mathbf{x}) = a_{ij}^{1/p}(x_j - x_i). \tag{2.30}$$

This operator is a nonlinear generalization of Laplacian defined (2.18). This nonlinear generalization is also an analog from real domain [Lindqvist, 2008]. In physics, this p-Laplacian is often used in fluid dynamics, where the parameter p characterizes a fluid's viscosity. We define *p-Dirichlet sum* or *p-energy* as

$$S_{G,p}(\mathbf{x}) \coloneqq \|\nabla_p \mathbf{x}\|_p^p = \sum_{i,j \in V} |(\nabla_p \mathbf{x})(i,j)|^p. \tag{2.31}$$

We then compute the p-Laplacian as

$$(\Delta_p \mathbf{x})_i = \sum_{i,j \in V} a_{ij} |x_i - x_j|^{p-1} \mathrm{sgn}(x_i - x_j), \tag{2.32}$$

which is same as Eq. (1.23), and also obtain the same energy as

$$S_{G,p}(\mathbf{x}) = \sum_{i,j \in V} a_{ij} \left| x_j - x_i \right|^p, \tag{2.33}$$

which we saw in Sec. 1.3. Remark that we obtain the normalized graph p-Laplacian by changing the gradient to normalized one.

The following statement about p-Dirichlet sum and p-Laplacian straightforwardly follows. This relation Eq. (2.31) generalizes Eq. (2.15) for 2-Laplacian of the standard graph.

Proposition 2.9.

$$\langle \mathbf{x}, \Delta_p \mathbf{x} \rangle_{\mathcal{H}(V)} = S_{G,p}(\mathbf{x}) \tag{2.34}$$

This proposition shows the connection between the p-Dirichlet sum, gradient, and p-Laplacian.

2.1.4.2 Eigenpairs of Graph p-Laplacian

To start, we recall the definition of eigenpairs of graph p-Laplacian (Eq. (1.25)). The eigenpair $(\lambda, \boldsymbol{\psi})$ of the p-Laplacian Δ_p is defined to satisfy

$$(\Delta_p \boldsymbol{\psi})_i = \lambda |\psi_i|^{p-1} \mathrm{sgn}(\psi_i), \forall i \in V. \tag{2.35}$$

Note that the first eigenpair is $(0, \mathbf{1})$ for a connected graph. The critical values of the Rayleigh quotient characterize the eigenpairs.

Proposition 2.10 (Bühler and Hein [2009])**.** *The eigenpairs of the graph p-Laplacian consist of critical values and corresponding points of the Rayleigh quotient, defined as*

$$R_{G,p}(\mathbf{x}) := \frac{S_{G,p}(\mathbf{x})}{\|\mathbf{x}\|_p^p}. \tag{2.36}$$

From this proposition, we know that $R_{G,p}(a\mathbf{x}) = R_{G,p}(\mathbf{x})$, for $a \in \mathbb{R}$. Therefore, to consider the eigenpairs of the graph p-Laplacian, we can limit our interest to $\mathcal{S}_p := \{\mathbf{x} \mid \|\mathbf{x}\|_p^p = 1\}$, seeing the Rayleigh quotient in Eq. (2.36).

In the sequel, we briefly explain why we can obtain the second eigenpair and why it is difficult to obtain the third or higher eigenpairs of the graph p-Laplacian. We now define the following quotient,

$$R_{G,p}^{(2)}(\mathbf{x}) := \frac{S_{G,p}(\mathbf{x})}{\min_\eta \|\mathbf{x} - \eta \mathbf{1}\|_p^p}. \tag{2.37}$$

This quotient gives the second eigenpair of p-Laplacian.

Proposition 2.11 (Bühler and Hein [2009]). *Let $\boldsymbol{\psi}_{p,2}$ be the second p-eigenvector of Δ_p. The global solution to Eq.* (2.37) *is given by $\boldsymbol{\psi}^*$, that is defined as*

$$\boldsymbol{\psi}^* := \boldsymbol{\psi}_{p,2} + \eta^* \mathbf{1}, \quad \text{where} \quad \eta^* := \arg\min_{\eta} \|\boldsymbol{\psi}_{p,2} - \eta \mathbf{1}\|_p^p. \tag{2.38}$$

This proposition shows that we have an exact identification for the second p-eigenpair; minimizing Eq. (2.37) gives the second p-eigenpair of Δ_p. However, we have not known the exact identification for the third or higher eigenpair of p-Laplacian [Lindqvist, 2008] yet.

While we do not know the identification such as Eq. (2.37) for the higher order eigenpairs, the next question is if there is a characterization of eigenpairs of the graph p-Laplacian, such as "orthogonality" for the $p = 2$ case. Recall that when $p = 2$ the eigenvectors of graph Laplacian are orthogonal to each other. By using orthogonality and the Rayleigh-Ritz theorem, we we can obtain the full eigenvectors of the graph 2-Laplacian as follows.

Proposition 2.12 (Rayleigh-Ritz). *Let $\boldsymbol{\psi}_1, \dots \boldsymbol{\psi}_{k-1}$ be eigenvectors of the graph Laplacian L. Then the k-th eigenvector $\boldsymbol{\psi}_k$ is given as*

$$\boldsymbol{\psi}_k = \arg\min_{\mathbf{x}} R_{G,2}(\mathbf{x}) \quad \text{s.t. } \boldsymbol{\psi}_k \perp \boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_{k-1}. \tag{2.39}$$

This proposition is the simplest form of Courant's min-max theorem and is also called the *variational theorem*. This proposition further characterizes the eigenvectors of the graph Laplacian from Prop. 2.10. We can easily obtain the higher order eigenvectors by this proposition and the sequence Eq. (2.39). This orthogonality constraint of eigenvectors comes from the nature of L^2 space induced from the graph 2-seminorm. However, we lose this sense of orthogonality if we expand to p-seminorm since we lose the inner product structure in the L^p space, as we saw in Sec. 1.3.

In this context, the following further generalizes the "orthogonality" to graph p-Laplacian. To characterize the eigenpairs of graph p-Laplacian, we use *Krasnoselskii genus* γ for a set B;

$$\gamma(B) = \begin{cases} 0 \text{ if } B = \emptyset \\ \inf\{k \in \mathbf{Z}^+ \mid \exists \text{odd continuous } h : B \to \mathbf{R}^k \backslash \{0\}\} \\ \infty \text{ when no such } h \text{ exists } \forall j \in \mathbf{Z}^+ \end{cases} \tag{2.40}$$

This genus is a generalized concept of the dimension; this genus defines a dimension-like value for any set B. Notably, this set B can be a subset of not only the Euclidean space, but also any spaces such as manifold. Using this genus, we can characterize the eigenpairs;

Proposition 2.13 (Tudisco and Hein [2018])**.** *Consider the set of subsets* $\mathcal{F}_k(\mathcal{S}_p) := \{B \subset \mathcal{S}_p \mid B = -B, \text{ closed }, \gamma(B) \geq k\}$. *The sequence defined as*

$$\lambda_{p,k} := \min_{B \subset \mathcal{F}_k(\mathcal{S}_p)} \max_{\mathbf{x} \in B} R_{G,p}(\mathbf{x}) \tag{2.41}$$

admits a critical point of $R_{G,p}(\mathbf{x})$. *Moreover, the pair of* $\lambda_{p,k}$ *and the vector* $\boldsymbol{\psi}_{p,k}$ *such that* $\lambda_{p,k} = R_{G,p}(\boldsymbol{\psi}_{p,k})$ *constitutes an eigenpair of* Δ_p.

This proposition is the generalized Courant's min-max theorem (Prop. 2.12); when $p = 2$, this proposition corresponds to Prop. 2.12. This proposition is also called *variational theorem*. The eigenvectors obtained by the sequence Eq. (2.41) are called *variational eigenvectors*. In this proposition, the space $\mathcal{F}_k(\mathcal{S}_p)$ serves as a generalized orthogonal k-dimensional space. Moreover, the sequence Eq. (2.41) may serve as a method to obtain eigenpairs successively.

However, we have two issues with the practical use of the sequence Eq. (2.41). The first problem is that due to the abstract characterization of the Krasnoselskii genus, we do not know how we can *numerically* apply this genus to obtain the higher eigenvectors. When $p = 2$, this abstract characterization can be translated into the concrete and "numerically computable" characterization, "orthogonality". However, in the current form of the Krasnoselskii genus given as Eq. (2.40), at this point, we do not know how to obtain this genus numerically. Secondly, similarly to the continuous p-Laplacian theory [Lindqvist, 2008], we do not know in which condition this sequence yields exhaustive eigenpairs. For the tree (and the disconnected forest) case, the sequence Eq. (2.41) exhausts all the spectra [Deidda et al., 2022, Zhang, 2021]. On the other hand, for the complete graph case, it is shown that there are other eigenpairs than ones yielded by the sequence Eq. (2.41) [Amghibech, 2003]. Despite these extensive studies, we are yet to understand in which conditions this sequence exhausts all the spectra of p-Laplacian. Thus, for a general graph, we do not know if the variational eigenvalues are the same as the Rayleigh quotient would do in Prop. 2.10.

To conclude the discussion above, while we know the identification for the second eigenpair of the graph p-Laplacian (Eq. (2.37)), we have three open problems as follows;

1. We do not know the identification of the third or higher eigenpairs.
2. We do not know in what condition of a graph the sequence Eq. (2.41) exhausts the spectra of the graph p-Laplacian
3. We do not know how to obtain the Krasnoselskii genus numerically.

2.1.4.3 Cheeger Inequalities for graph p-Laplacian

This section discusses Cheeger inequalities for the graph p-Laplacian. The Cheeger inequality theoretically supports using the variational eigenvectors of p-Laplacian from the Cheeger cut point of view.

We start our discussion from a 2-way Cheeger cut. Recall that in Eq. (2.23) and in Eq. (2.24), we defined a Cheeger cut for a subset $U \subset V$ for 2-way cut and Cheeger constant h_2 as

$$C(U) = \frac{\mathrm{Cut}(U, \overline{U})}{\min(|U|, |\overline{U}|)} \tag{2.42}$$

$$h_2 = \min_U C(U). \tag{2.43}$$

By recursively using this Cheeger cut, here we define the multi-class Cheeger cut, which we call *k-way Cheeger constant* as

$$h_k := \min_{\{V_i\}_{i=1,\cdots,k}} \max_{j \in \{1,\ldots,k\}} C(V_j). \tag{2.44}$$

This k-way Cheeger constant can be seen as the smallest k-way Cheeger cut. To obtain the k-way Cheeger cut is known to be NP-hard. However, relaxing into the real-value would ease this problem; this Cheeger cut can be approximated by the variational eigenvalues of the graph p-Laplacian by Cheeger inequality.

Before we discuss the Cheeger inequality, we need a setup of the nodal domain.

Definition 2.14 (nodal domain)**.** *A nodal domain of* $\mathbf{x}$ *over a graph* G *is a maximally connected subgraph* A *of a graph* G *such that for* $\mathbf{x} \in \mathbb{R}^n$ *where* A *is either* $\{i \mid x_i > 0\}$ *or* $\{i \mid x_i < 0\}$.

With this idea, a nodal domain of $\mathbf{x}$ can be seen as a "partition" of the graph by the sign of $\mathbf{x}$. We may obtain the non-trivial bound of the number of the nodal domains of the variational eigenvectors is bounded; see Tudisco and Hein [2018].

This nodal domain is used in the Cheeger inequality for the variational eigenvectors of graph p-Laplacian as follows.

Proposition 2.15 (Tudisco and Hein [2018]). *Let $(\lambda_{p,k}, \boldsymbol{\psi}_{p,k})$ be a k-th eigenpair of Δ_p, obtained by the sequence Eq.* (2.41). *Let also m_k be the number of nodal domains of $\boldsymbol{\psi}_{p,k}$. Then,*

$$\left(\max_i \frac{d_i}{2}\right)^{-(p-1)} \left(\frac{h_{m_k}}{p}\right)^p \leq \lambda_{p,k} \leq 2^{p-1} h_k$$

Recall that d_i is a degree of a vertex i. This proposition provides how much the variational eigenvalues approximate the Cheeger constant. Thus, this proposition motivates using the higher variational eigenvectors of the graph p-Laplacian for multi-way Cheeger Cut since the eigenvalues can serve as an approximation of the k-way Cheeger constant. We finally remark that the discussion here can naturally generalizes to the normalized setting.

2.1.4.4 Limitation of Multi-class Spectral Clustering using p-Laplacian

For two-class spectral clustering, we are ready to use the second eigenvector of the graph p-Laplacian. The reason is that we are theoretically motivated (Prop. 2.15) and we can numerically obtain the second eigenvector by Prop. 2.11. On the other hand, for multi-class spectral clustering, while we are still theoretically motivated for the use of the higher-order eigenpairs by the same proposition, we do not have a numerical way to obtain the higher eigenpairs due to the three open problems discussed in Sec. 2.1.4.2. As we see in this section, these open problems are key if we want to apply the graph p-Laplacian to multi-class clustering. Hence, without solving these open problems, we cannot say that it is theoretically guaranteed to use the graph p-Laplacian for multi-class clustering. However, these problems remain open not only in the graph domain but also in the continuous domain, which has a longer history and broader research communities. Thus, the limitation of spectral clustering using p-Laplacian is that it is practically difficult to do multi-class graph clustering using p-Laplacian at this point.

2.1.4.5 Existing Work for Multi-class Spectral Clustering Using p-Laplacian

So far, we discuss limitations for spectral clustering using the graph p-Laplacian. This section discusses how the existing works "bypass" this limitation. In a rough sense, there are two ways to materialize the multi-class clustering using graph p-Laplacian; i) recursively bisectioning and ii) the use of the approximated orthogonality.

For i), Bühler and Hein [2009] proposed a multi-class clustering, which recursively bisections a graph by using Prop. 2.11. Thus, Bühler and Hein [2009] partitions a subgraph when we partition further than two, which does not exploit the full structure of a graph. In fact, the theoretical nature of recursive bisectioning using spectral clustering is also less well-understood compared to the multi-way one, even for the case of $p = 2$ [Verma and Meila, 2003]. Note that, in contrast, for k-way spectral clustering using p-Laplacian, we always have a bound via the Cheeger inequality (Prop. 2.15).

The methods in line with ii), such as [Ding et al., 2019, Luo et al., 2010, Pasadakis et al., 2022], assume that the k eigenvectors of the graph p-Laplacian are close to ones of the graph 2-Laplacian, since the Rayleigh quotients $R_{G,p}$ and $R_{G,2}$ are similar. Using this assumption, these methods use optimization methods for the Rayleigh quotients $R_{G,p}$ with the initial conditions as the first k-eigenvectors of the graph 2-Laplacian. By these initial conditions, we expect that the obtained k-eigenvectors are "close" to the first k-eigenvectors of the graph 2-Laplacian, and thus we expect that the obtained k-eigenvectors are the first k-eigenvectors from the Rayleigh quotient $R_{G,p}$. These methods exploit approximated orthogonality proven in Luo et al. [2010] of eigenvectors of graph p-Laplacian in order to achieve better algorithms. However, this assumption might be too strong, especially for the very large p or very small p, i.e., p close to 1. Moreover, even if this assumption may be reasonable, we do not know if the first k eigenvectors from Rayleigh quotient are the same set of vectors with the first k-eigenvectors obtained by the sequence Eq. (2.41) as we discussed in Sec. 2.1.4.2. Now, recall that the Cheeger inequality guarantees the quality of the cut for the latter, the eigenvectors obtained by Eq. (2.41). Thus, at this point, we do not know if this assumption is suitable for multi-class clustering.

2.1.4.6 Another Graph p-Laplacian: Vertex-wise Graph p-Laplacian

We remark that we have a different way to define graph p-energy and corresponding graph p-seminorm in many literature [Bougleux et al., 2007, 2009, Calder, 2018, Elmoataz et al., 2008, Singaraju et al., 2009, Zhou and Schölkopf, 2006]. In this line, the p-energy is defined in "vertex-wise" way, which is written as

$$S_{G,p}^{VW}(\mathbf{x}) = \sum_{i \in V} \left(\sum_{j \in V} a_{ij} |x_i - x_j|^2 \right)^{p/2}, \tag{2.45}$$

and the p-Laplacian is defined as

$$\Delta_{G,p}^{VW} \mathbf{x} := \frac{\partial}{\partial \mathbf{x}} S_{G,p}^{VW}(\mathbf{x}). \tag{2.46}$$

This definition sums the vertex-wise energy. For more details of the difference, see Saito et al. [2018]. For the corresponding p-Laplacian, we do not have an exact identification of higher eigenpairs either. Moreover, for the corresponding seminorm, we have not theoretical characteristics yet, such as Cheeger inequality discussed in Sec. 2.1.4.3. Thus, for this graph p-Laplacian, we have less understanding than the graph p-Laplacian induced from the p-energy we used in this thesis.

2.2 Graph Spectral Connection

This section explores the connection between the graph cut problem, weighted kernel k-means, and the heat kernel. Previously, spectral clustering was only applied to discrete graphs. Here, we extend the discussion to vector data by introducing a method to construct a graph from vector data. We provide theoretical justifications for this method through "spectral connections" from both the weighted kernel k-means perspective (Sec. 2.2.3) and the heat kernel framework (Sec. 2.2.4). It is important to note that the discussion in this section only applies to the normalized graph Laplacian.

2.2.1 Spectral Clustering via Kernel Function

From the graph cut discussion, the most common embedding algorithm is as follows. Given a vector data $\mathbf{x}_1, \ldots, \mathbf{x}_n \in X$,

1. Using a kernel function κ

$$\kappa(\mathbf{x}, \mathbf{x}') := \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle, \tag{2.47}$$

where ϕ is a feature map, we form a graph whose adjacency matrix is as

$$a_{ij} := \kappa(\mathbf{x}_i, \mathbf{x}_j) \tag{2.48}$$

2. We then apply spectral clustering (Alg. 1 and Alg. 2) to the graph whose adjacency matrix is A.

While this algorithm "models" vector data into a graph by a kernel function without any justifications, next Sec. 2.2.3 justifies this algorithm. Sec. 2.2.3 shows that this algorithm is equivalent to solving the weighted kernel k-means problem. Sec. 2.2.4 established a connection between spectral clustering heat kernel to justify the Gaussian kernel.

2.2.2 The Standard k-means and Weighted Kernel k-means

Since this section is built on the k-means formulation, we briefly review this topic.

Consider to partition the data points into $\{C_\ell\}_{\ell=1}^k$. The standard *k-means* algorithm is to partition data points in Euclidean space by minimizing the sum of the square of the distance between each data point belonging to the cluster and its centroid. The standard k-means is to minimize the following objective function.

$$\mathcal{J}(\{C_\ell\}_{\ell=1}^k) := \sum_{\ell \in [k]} \sum_{i \in C_\ell} \|\mathbf{x}_i - \mathbf{m}_\ell\|_2^2 \quad \mathbf{m}_\ell := \sum_{j \in C_\ell} \mathbf{x}_j / |C_\ell|. \tag{2.49}$$

Note that $\mathbf{m}_j$ is the average within the cluster V_j and serves as the centroid. Minimizing $\mathcal{J}(\{C_\ell\}_{\ell=1}^k)$ is NP-hard [Mahajan et al., 2012]. The approximated *discrete* solution is obtained by EM-type algorithms [Bishop, 2007].

This k-means is generalized to the weighted and kernel setting. The weighted kernel k-means algorithm partitions data points in feature space and conducts the k-means. Let ϕ be a feature map. We define the *weighted kernel k-means* objective as

$$\mathcal{J}_\phi(\{C_\ell\}_{\ell=1}^k) := \sum_{\ell \in [k]} \sum_{i \in C_\ell} \theta(\mathbf{x}_i) \|\phi(\mathbf{x}_i) - \mathbf{m}_{\phi,\ell}\|_2^2, \tag{2.50}$$

$$\text{where } \mathbf{m}_{\phi,\ell} := \sum_{j \in C_\ell} \theta(\mathbf{x}_j)\phi(\mathbf{x}_j) / \sum_{j \in C_\ell} \theta(\mathbf{x}_j), \tag{2.51}$$

$\|\cdot\|_2$ is a norm induced from the dot product[1], $\theta(\mathbf{x}_i)$ is a weight at $\mathbf{x}_i$ and $\mathbf{m}_{\phi,j}$ serves as a weighted mean of the cluster C_j.

Minimizing $\mathcal{J}_\phi(\{C_\ell\}_{\ell=1}^k)$ is also NP-hard. In practice, we may apply the same EM-type algorithm as the standard k-means case to obtain the approximate discrete solution. However, the EM-type algorithm does not practically work when the feature map ϕ maps $\mathbf{x}$ to the infinite dimensional space. Thus, we consider to approximate the solution in a different way, which we call *approximated relax* solution.

To consider the approximated relax solution, we can further rewrite $\mathcal{J}_\phi(\{C_\ell\}_{\ell=1}^k)$ as follows.

$$\mathcal{J}_\phi(\{C_\ell\}_{\ell=1}^k) = \text{trace}\Theta K \Theta - \text{trace} Z_M^\top \Theta^{1/2} K \Theta^{1/2} Z_M, \tag{2.52}$$

where Θ is a diagonal matrix whose i-th element is $\theta(\mathbf{x}_i)$ and K is a gram matrix formed by the dot product kernel, and indicator matrix $Z_M \in \mathbb{R}^{n \times k}$ as

$$Z_M := \Theta^{1/2} Z (Z^\top \Theta Z)^{-1/2}. \tag{2.53}$$

Note that $Z_M^\top Z_M = I$ and

$$(Z_M)_{i\ell} = \begin{cases} \sqrt{\theta(\mathbf{x}_i) / \sum_{\mathbf{x}_j \in C_\ell} \theta(\mathbf{x}_j)} & (\mathbf{x}_i \in C_\ell) \\ 0 & (\mathbf{x}_i \notin C_\ell) \end{cases} \tag{2.54}$$

We consider to minimize Eq. (2.52) with respect to the variable Z_M. Since the first term of Eq. (2.52) is fixed with respect to the variable Z_M, we want to maximize the second term of Eq. (2.52) in order to minimize the k-means objective function. We then relax Z_M from discrete values to real values. Exploiting the linear algebra theory, maximizing $\text{trace}(Z_M^\top \Theta^{1/2} K \Theta^{1/2} Z_M)$ can be realized by taking Z_M as the largest k eigenvectors of $\Theta^{1/2} K \Theta^{1/2}$. We shall call this relaxed Z_M as an *approximated relaxed* solution of the weighted kernel k-means. We refer to Dhillon et al. [2004] for more details.

[1] Sec. 4.5.1.1 shows that the discussion in this section holds for *any* kernel.

2.2.3 Weighted Kernel k-means and Spectral Clustering.

This section reviews the connection between spectral clustering and the weighted kernel k-means. While these share the same purpose in clustering, the formulations seem quite different. However, these formulations are linked via the same trace maximization problem. This discussion leads to embedding vector data into graphs by considering kernels for clustering. We review an established justification by Dhillon et al. [2004], which develops the following strategy based on the discussion above.

1. Using vectors transformed by a feature map $\phi(\mathbf{x}_i)$ to the weighted kernel k-means.

2. Showing a connection from this weighted kernel k-means to the spectral clustering.

By this, we can ground the use of a feature map to spectral clustering through k-means lens. [Dhillon et al., 2004] shows the following claim.

Proposition 2.16 (Dhillon et al. [2004])**.** *Consider a graph $a_{ij} = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ and its degree d_i. We apply k-way normalized cut (Eq.* (2.2)*) to this graph A. We substitute a weight $w(\mathbf{x}_i) = 1/d_i$ to the weighted kernel k-means $\mathcal{J}_\phi(\{V_\ell\}_{\ell=1}^k)$ (Eq.* (2.50)*). If we relax Z_M and Z_N into real values with orthogonal constraints, we obtain*

$$\min_{Z_M \in \mathbb{R}^{n \times k}} \{\mathcal{J}_\phi(\{V_\ell\}_{\ell=1}^k) \text{ s.t. } Z_M^\top Z_M = I\} = \min_{Z_N \in \mathbb{R}^{n \times k}} \{\mathrm{kNCut}(\{V_\ell\}_{\ell=1}^k) \text{ s.t. } Z_N^\top Z_N = I\}$$

Above all, modeling vector data into a graph by the dot product kernel is justified; if we conduct spectral clustering to a graph formed by the dot product kernel, this is equivalent to the weighted kernel k-means with a particular weight in a relaxed sense. For more details, see [Dhillon et al., 2004, von Luxburg, 2007].

2.2.4 Heat Kernel and Spectral Clustering

Heat kernel is closely related to the energy minimization problem using the Laplace operator, while the graph cut also can be seen the energy minimization problem using graph Laplacian. This section explains that these two problems are connected.

We now discuss that the continuous Laplace operator is related to the energy minimization problem. Assume a compact differentiable d-dimensional manifold $\mathcal{M}$ isometrically

embedded into $\mathbb{R}^N$, a variable $x \in \mathcal{M}$, and a measure μ. We consider a problem to obtain a function $f : \mathcal{M}^{r/2} \to \mathbb{R}$, that minimizes the energy as

$$\min_f S_2^{(c)}(f) = \|\nabla^{(c)} f\|^2 \text{ s.t. } \|f\|^2 = 1, \tag{2.55}$$

where operators with superscripted by (c) are the standard continuous calculus ones. From a physics point of view, we can see $S_2^{(c)}(f)$ as energy and the problem as an energy minimization problem. This problem often appears in physics, as well as machine learning. For more details, we refer to Sec. 4.5.2 of this thesis and [Courant and Hilbert, 1962, Belkin and Niyogi, 2003]. This problem often appears when we want to know a profile that minimizes energy, such as velocity profile in fluid dynamics [Courant and Hilbert, 1962]. In machine learning, this problem can be considered a clustering problem. The operator ∇f can be seen to measure how close each data point is when we embed data from a manifold to the Euclidean space. Then, this problem Eq. (2.55) can be thought of as finding suitable mapping f best preserving locality over all data points. More on this way of thinking, refer to Sec.3 in [Belkin and Niyogi, 2003]. Note that we can rewrite the energy using Laplace operator as

$$S_2^{(c)}(f) = \|\nabla^{(c)} f\|^2 = \langle \nabla^{(c)} f, \nabla^{(c)} f \rangle = \langle \Delta^{(c)} f, f \rangle \tag{2.56}$$

The third equality follows from the Stokes theorem. We introduce the additional constraints $\langle f, c\mathbf{1} \rangle = 0$ for the original minimization problem Eq. (4.40) in order to avoid the trivial solution to this problem, which is $f = c\mathbf{1}$. Thus, we reformulate the problem Eq. (4.40) as

$$\min \langle \Delta^{(c)} f, f \rangle \text{ s.t. } \|f\|^2 = 1, \langle f, c\mathbf{1} \rangle = 0. \tag{2.57}$$

See more discussions in [Belkin and Niyogi, 2003].

We now discuss heat equation and heat kernel, which are useful tools for analyzing $\Delta^{(c)} f$. We pay attention to the term Δf because this term is a main "actor" of the energy minimization problem in Eq. (2.56). Consider a variable $\mathbf{x} \in \mathcal{M}$. The heat equation on $\mathcal{M}$ is as

$$\left(\frac{\partial}{\partial t} + \Delta^{(c)} \right) U(t, \mathbf{x}) = 0, \;\; U(0, \mathbf{x}) = f(\mathbf{x}) \tag{2.58}$$

The solution is given to satisfy

$$U = \int H_t(\mathbf{x}, \mathbf{y}) U(0, \mathbf{y}) d\mu(\mathbf{y}) \tag{2.59}$$

where H_t is a *heat kernel*. A well-known example of a heat kernel is the Gaussian kernel as

$$G_t(\mathbf{x}, \mathbf{y}) = \frac{1}{(4\pi t)^{d/2}} \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|_2^2}{4t}\right), \tag{2.60}$$

which gives a solution to one variable Eq. (2.58) when $\mathcal{M} = \mathbb{R}^n$. However, obtaining a concrete form of heat kernel for a general manifold is difficult.

We give a solution in an asymptotic case when $t \to 0$. Locally, we can approximate $H_t(\mathbf{x}, \mathbf{y}) = G_t(\mathbf{x}, \mathbf{y})$ when t and $\|\mathbf{x} - \mathbf{y}\|$ are small [Rosenberg and Steven, 1997, Belkin and Niyogi, 2003]. Together with

$$\lim_{t \to 0} \int_{\mathcal{M}} d\mu(y) G_t(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) = f(\mathbf{x}) \tag{2.61}$$

$$\lim_{t \to 0} \int_{\mathcal{M}} d\mu(\mathbf{y}) G_t(\mathbf{x}, \mathbf{y}) = 1, \tag{2.62}$$

for small t and discrete values $\mathbf{x}_1, \ldots, \mathbf{x}_n$ instead of continuous value we can approximate as

$$\Delta^{(c)} f(\mathbf{x}_i) \approx \sum_j G_t(\mathbf{x}_i, \mathbf{x}_j) f(\mathbf{x}_i) - \sum_j G_t(\mathbf{x}_i, \mathbf{x}_j) f(\mathbf{x}_j). \tag{2.63}$$

The right-hand side of Eq. (2.63) is equal to the graph Laplacian L for a graph whose adjacency matrix is a gram matrix of the Gaussian Kernel. Following Eq. (2.63), we can relate the original energy minimization problem and graph cut problem (Eq. (2.5)) as

$$\|\Delta^{(c)} f\|^2 \approx Y^\top L Y \tag{2.64}$$

with proper constraints. By properly introducing "normalizing" constraints, the continuous energy minimization problem corresponds to a 2-way normalized cut problem.

This discussion can justify the embedding by a kernel function for spectral clustering. The reason is that the graph cut problem for a graph made from a Gaussian kernel can be seen as an approximated continuous energy problem of an asymptotic case of the heat equation.

Therefore, embedding by the kernel for spectral clustering is a discrete analog of the energy minimization problem.

Finally, we remark that the approximation becomes exact when a number of randomly generated data is infinite (See Thm.3.1 in [Belkin and Niyogi, 2005]). For more discussion of this, we refer to [Belkin and Niyogi, 2003, 2005].

2.3 History of Spectral Clustering and Spectral Connection

In this section, we review the history of the spectral clustering and spectral connection.

Donath and Hoffman [1972, 1973] firstly suggested the eigenvectors of adjacency matrices for the partitioning purpose. Independently, Fiedler [1973, 1975a,b] first reported that the connectivity of the graph is related to the second eigenvalue of the graph Laplacian. For this historical reason, the second eigenvector of Laplacian is sometimes called as *Fiedler vector*, and the associated eigenvalue is called *Fiedler value*. Slightly after this, Barnes [1982] and Barnes and Hoffman [1984] rediscovered spectral clustering as a linear programming problem. Simultaneously, an analog between differential geometry and graph Laplacian was established; notably, the Fiedler value was connected to Cheeger constant [Alon and Milman, 1985, Alon, 1986]. Then, the eigenvectors for clustering were experimentally demonstrated in various literature, such as [Pothen et al., 1990, Barnard and Simon, 1994]. Around the same time, the Ratio Cut [Hagen and Kahng, 1992] is established and then extended to multi-way ratio cut [Chan et al., 1994]. These are generalized to the normalized cut in [Shi and Malik, 2000, Yu and Shi, 2003], which is empirically shown to improve the ratio cut.

In the machine learning community, spectral clustering became popular by the normalized cut [Shi and Malik, 2000], followed by many seminal works such as [Ng et al., 2001, Meilă and Shi, 2001, Bach and Jordan, 2003, Joachims, 2003]. Researchers in the machine learning community were interested in the setting where we have vector data, formulate a graph from the vector data, and then apply spectral clustering, as we will see in the next section. The graph modeling formulation from vector data has been long explored, especially in the computer vision vein such as [Scott and Longuet-Higgins, 1990, Weiss, 1999]. Then, Belkin and Niyogi [2003] established the relationship between the continuous Laplace operator and the graph Laplacian if we form a graph using a Gaussian kernel. Then, it is proven that the graph Laplacian converges to the continuous Laplace operator when we draw an infinite number of data points and formulate a graph Laplacian using Gaussian [Lafon, 2004, Belkin

and Niyogi, 2005, Belkin et al., 2006, Belkin and Niyogi, 2007, 2008, Trillos and Slepčev, 2018]. Further details are discovered by the seminal work, such as [von Luxburg et al., 2004, Giné and Koltchinskii, 2006, Hein, 2006, Hein et al., 2007, von Luxburg et al., 2008]. Also, the connection between kernel k-means and spectral clustering is discussed [Zha et al., 2001, Dhillon et al., 2004].

Finally, the recent developments involve the consistency of spectral clustering for certain statistical graph models [Rohe et al., 2011], such as stochastic block model [Holland et al., 1983]. This area actively evolves including stronger results than the consistency discussed by Rohe et al. [2011] for the stochastic block model [Fishkind et al., 2013, Abbe et al., 2015, Lei and Rinaldo, 2015, Sarkar and Bickel, 2015, Su et al., 2019], consistency for its generalized model called degree-corrected stochastic block model [Qin and Rohe, 2013], and further refinements of the spectral clustering [Joseph and Yu, 2016].

For more details of early history of spectral clustering, see [Spielman and Teng, 1996, von Luxburg, 2007] and for spectral clustering on stochastic block model see [Abbe, 2018].

2.4 Graph Analogy to Circuit: Resistance and p-Resistance

This section introduces an analogy between the circuit theory and a graph. In this analogy, the effective resistance between any two vertices is shown to be a distance measure, which makes effective resistance useful in machine learning. To further explore this analogy, we first define the notion of coordinate spanning set for a matrix. We then examine an analog using the coordinate spanning set for the graph Laplacian. We will see that the resistance is induced by the energy $S_{G,2}(\mathbf{x})$ (Eq. (2.15)) and graph Laplacian.

2.4.1 Coordinate Spanning Set

This section sets up the notation of coordinate spanning set, which is a convenient tool for an analog between the circuit theory and a graph.

A symmetric matrix $M \in \mathbb{R}^{n\times n}$ is a positive semidefinite (PSD) when a quadratic form of M is nonnegative, i.e.,

$$\mathbf{x}^\top M\mathbf{x} \geq 0, \forall \mathbf{x} \in \mathbb{R}^n. \tag{2.65}$$

When quadratic form is strictly positive, we shall call positive definite (PD). A PSD matrix M induces a semi-inner product as $\langle \mathbf{x}, \mathbf{y}\rangle_M := \mathbf{x}^\top M\mathbf{y}$. This inner product induces a semi-norm,

as

$$\|\mathbf{x}\|_M := \langle \mathbf{x}, \mathbf{x} \rangle_M \tag{2.66}$$

The reproduced kernel associated with the above semi-inner product is M^+ since

$$\langle \mathbf{u}, \mathbf{v}_i \rangle_M = \mathbf{u}^\top M M^+ \mathbf{e}_i = u_i, \quad \forall \mathbf{v}_i \in \mathcal{V}(M),\ \mathbf{u} \in \mathcal{H}(M). \tag{2.67}$$

We define the coordinate spanning set

$$\mathcal{V}(M) := \{v_i := M^+ \mathbf{e}_i : i = 1, \ldots, n\} \tag{2.68}$$

and let $\mathcal{H}(M) := \mathrm{span}(\mathcal{V}(M))$. This $\mathcal{H}(M)$ is a *Hilbert space* induced by inner product $\langle \cdot, \cdot \rangle_M$.

The set $\mathcal{V}$ acts as "coordinates" for $\mathcal{H}$, that is, if $\mathbf{w} \in \mathcal{H}$ we have $w_i = \mathbf{e}_i^\top M^+ M \mathbf{w} = \langle \mathbf{e}_i, M^+ \mathbf{e}_i \rangle_M$. Note that the vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ are not necessarily orthonormal. We also remark that this coordinate property is simply the reproducing kernel property for kernel M^+ [Aronszajn, 1950, Shawe-Taylor and Cristianini, 2004].

2.4.2 Graph Effective Resistance and p-Resistance

An analogy is established between graph and electric circuit [Doyle and Snell, 1984]. In this analog, a vertex is a point at a circuit, and an edge is a resistor with resistance $1/a_{ij}$. A flow over a graph mapped to a current, and a distribution over V as $\mathbf{x}$ is seen as a potential at each vertex point. The elemental circuit formula for the resistance is

$$\text{resistance} = \frac{\text{voltage}^2}{\text{energy}}. \tag{2.69}$$

From this formula, the effective resistance between two vertices is defined as the inverse of the energy induced by a unit voltage between two vertices [Kirchhoff, 1847].

In this analog, the energy $S_{G,2}(\mathbf{x})$ for a vector over vertices $\mathbf{x} \in \mathbb{R}^n$ is defined over the potential $\mathbf{x}$ as Eq. (2.15). We define effective 2-resistance $r_{G,2}(i, j)$ between two vertices

$i, j \in V$ as

$$r_{G,2}(i,j) := \frac{1}{\min_{\mathbf{x}}\{S_{G,2}(\mathbf{x}) \text{ s.t. } x_i - x_j = 1\}} \tag{2.70}$$

The coordinate spanning set using graph Laplacian (Laplacian Coordinate) plays a role. For the definition of the coordinate, we obtain the Laplacian coordinate by putting $M = L$. Recall that the graph Laplacian is symmetric PSD. Using the coordinate spanning set $\mathcal{V}(L)$, we may rewrite 2-resistance[2] as

$$r_{G,2}(i,j) = \|L^{+}\mathbf{e}_i - L^{+}\mathbf{e}_j\|_L^2 \tag{2.71}$$

$$= \|\mathbf{v}_i - \mathbf{v}_j\|_L^2, \quad \mathbf{v}_i, \mathbf{v}_j \in \mathcal{V}(L). \tag{2.72}$$

From the definition, $r_{G,2}(i,i) = 0$ and $r_{G,2}(i,j) = r_{G,2}(j,i)$. Remark that we may further write the resistance using the 2-norm as

$$r_{G,2}(i,j) = \|L^{+1/2}\mathbf{e}_i - L^{+1/2}\mathbf{e}_j\|_2^2 \tag{2.73}$$

Using p-energy $S_{G,p}$, these energy and effective resistance are extended to p-resistance $r_{G,p}$ as

$$r_{G,p}(i,j) := \frac{1}{\min_{\mathbf{x}}\{S_{G,p}(\mathbf{x}) \text{ s.t. } x_i - x_j = 1\}}. \tag{2.74}$$

The triangle inequality [Herbster, 2010] holds for the p-resistance, that is for $a, b, c \in V$,

$$r_{G,p}^{1/(p-1)}(a,b) \leq r_{G,p}^{1/(p-1)}(a,c) + r_{G,p}^{1/(p-1)}(c,b). \tag{2.75}$$

With $r_{G,p}^{1/(p-1)}$, the graph G is a metric space. Particularly, when $p = 2$, 2-resistance defines a metric between $\mathbf{v}_i, \mathbf{v}_j \in \mathcal{V}(L)$. More properties on p-energy and p-resistance, see [Herbster and Lever, 2009, Alamgir and Luxburg, 2011].

[2] In the rest of this thesis, we abbreviate effective p-resistance as p-resistance.

2.4.3 A Variant of p-Resistance

While we use p-resistance as in Eq. (2.74), we mention that there is a different variant on the p-resistance $r^A_{G,p}$ proposed by Alamgir and Luxburg [2011], that is defined as

$$r^A_{G,p}(i,j) = \frac{1}{\min_{\mathbf{x}} \sum_{ij} a_{ij}^{1/(p-1)} |x_i - x_j|^{p/(p-1)} \text{ s.t. } x_i - x_j = 1} \tag{2.76}$$

$$= \frac{1}{\min_{\mathbf{x}} \sum_{ij} a_{ij}^{q-1} |x_i - x_j|^{q} \text{ s.t. } x_i - x_j = 1}. \tag{2.77}$$

Despite this slight change of the definition, p-resistance in [Herbster and Lever, 2009] and [Alamgir and Luxburg, 2011] shares almost the same properties. However, we note that in [Alamgir and Luxburg, 2011], the parameter p works in the opposite way; if we mean large, p [Alamgir and Luxburg, 2011] means smaller p and vice-versa.

2.5 Hypergraph Laplacians and p-Laplacians

This section introduces hypergraph Laplacians and p-Laplacians, a generalization of graph Laplacian and p-Laplacian previously discussed in Sec. 2.1. First, Sec. 2.5.1 presents the notations used for hypergraphs. Then, Sec. 2.5.2 introduces the existing hypergraph Laplacians and p-Laplacians. Lastly, Sec 2.5.3 provides a brief overview of the historical development of hypergraph Laplacians and p-Laplacians.

2.5.1 Hypergraph Notation

This section introduces hypergraph notation.

We begin with standard definitions and notation for hypergraphs. Hypergraph is defined to generalize a graph; particularly, the edge is generalized to connect arbitrary number of vertices. A *hypergraph* is defined as $G = (V, E, \mathbf{w}, \boldsymbol{\mu})$ with the following symbol definitions. We define $V := [n]$ as a *vertex* set. We also define E as an *edge* set, whose element $e \in E$ is a tuple of vertices , e.g.,

$$e = (i_1, \ldots, i_k), \quad i_1, \ldots, i_k \in V. \tag{2.78}$$

A hypergraph is *undirected* when this tuple is unordered, and *directed* when the tuple is ordered. When all the edge contains the same number of vertices, we call *uniform*. Denote

by $\mathbf{w}$ a vector $\{w(e)\}_{e\in E}$ where $w\colon E \to \mathbb{R}^+$ maps each edge with a *weight*, and let $\boldsymbol{\mu}$ be a vector $\{\mu_i\}_{i\in V}$, where $\mu\colon V \to \mathbb{R}^+$ maps each vertex with a *vertex weight*. A hypergraph is *connected* if there is a path between every pair of vertices. If an edge contains the same vertex multiple times, we call that this edge has a *self-loop*. In what follows, we assume that the hypergraph G is connected and undirected unless noted. We define the *degree of a vertex* $i \in V$ as $d_i = \sum_{e\in E: i\in e} w(e)$, while *the degree of an edge* $e \in E$ is defined as $|e|$.

For the benefit of the representation of hypergraph, we define various matrices. *Degree matrices for vertex* D_v is a diagonal matrix whose diagonal elements are the degree of vertices. Also, the *degree matrices for edges* $(D_e)_{ii}$ is a diagonal matrix whose diagonal element is the number of vertices contained in the i-th edge. Let W_e be a diagonal $|E| \times |E|$ matrix, whose diagonal elements are the weight of edge e. Let $H \in \mathbb{R}^{|V|\times|E|}$ be an *incidence matrix*, whose element $h(v, e) = \sqrt{\rho_{v,e}}$ if a vertex v is connected to an edge e, and 0 otherwise, where $\rho_{v,e}$ counts how many times the edge e contains the vertex v, e.g., if edge is $e = (v, v, v_1, v_2)$ for 4 uniform hypergraphs, $\rho_{v,e} = 2$.

For more details of basics of hypergraph, we refer to [Berge, 1984].

2.5.2 Hypergraph Spectral Clustering via Laplacians and p-Laplacians

This section introduces existing methods for hypergraph spectral clustering. Since the generalization from graph to hypergraph is not straight forward, we have a variety of such methods, whereas graph spectral clustering is an established area.

Hypergraph Spectral clustering also uses the hypergraph Laplacian and p-Laplacians, simiarlry to the graph spectral clustering. This section discusses the existing hypergraph Laplacians and p-Laplacians. Roughly speaking, we have two established ways of the generalization from graph to hypergraph. One established formulation to deal with hypergraphs is to reduce hypergraphs to graphs. The other way is a submodular approach. In the following we discuss these.

2.5.2.1 Graph Contraction Approaches

In this section, we introduce the graph contraction hypergraph Laplacians. Agarwal et al. [2006] classified this graph contraction hypergraph Laplacians into two categories; one reduction is *clique expansion* [Rodriguez, 2002, Saito et al., 2018] and the other reduction is *star expansion* [Zhou et al., 2006]. Note that the some of the existing Laplacian are not

further explored for their generalization to p-Laplacian.

Clique Expansion (CLIQUE). This approach constructs a graph where a clique replaces every pair of vertices in an original edge of a hypergraph. In this approach, we consider two type of approach, one is edge-normalized [Saito et al., 2018], and the other is edge-unnormalized [Rodriguez, 2002]. We consider the following two hypergraph cuts for both settings.

$$\text{(edge-unnormalized)} \quad \mathrm{Cut}_{ncl}(V_1, V_2) := \sum_{e \in E} w(e) \frac{|e \cap V_1||e \cap V_2|}{m-1}, \tag{2.79}$$

$$\text{(edge-unnormalized)} \quad \mathrm{Cut}_{cl}(V_1, V_2) := \sum_{e \in E} w(e)|e \cap V_1||e \cap V_2|. \tag{2.80}$$

If we do not consider the "balance" like the ratio and normalized cut cases (see Sec. 2.1 and Sec. 5.3), we may have unbalanced results. Thus, we consider these cuts in the balanced approach, usually in the normalized cut approach. The hypergraph normalized cut is actually rewritten by the quadratic form of the following Laplacians, similarly to the graph case. For each case, we have two hypergraph Laplacians, hypergraph clique 2-Laplacian normalized by a degree of edge $L_{2,ncl}$ (CLIQUE E-N), and clique 2-Laplacian but edge-unnormalized $L_{2,cl}$ (CLIQUE E-UN) as

$$\text{(edge-normalized)} \quad L_{2,ncl} := I - D_v^{-1/2} A_{ncl} D_v^{-1/2}, \tag{2.81}$$

$$\text{(edge-unnormalized)} \quad L_{2,cl} := I - D_{cl}^{-1/2} A_{cl} D_{cl}^{-1/2}, \tag{2.82}$$

where

$$\text{(edge-normalized)} \quad (A_{ncl})_{ij} := \sum_{(i,j) \in e} w(e)/(|e| - 1) \tag{2.83}$$

$$\text{(edge-unnormalized)} \quad (A_{cl})_{ij} := \sum_{i,j \in e} w(e), \quad (D_{cl})_{ii} := \sum_{i \in e} w(e). \tag{2.84}$$

Note that this can be seen as hypergraph contraction to graph, represented by A_{cl}, and $L_{2,cl}$ is a standard 2-Laplacian induced by A_{cl}.

We define the normalized hypergraph cut problem as

$$\text{(edge-normalized)} \quad \mathrm{kNCut}_{ncl}(\{V_\ell\}_{\ell=1}^k) := \sum_{\ell=1}^k \frac{\mathrm{Cut}_{ncl}(V_\ell, V\backslash V_\ell)}{\mathrm{vol}(V_\ell)} \tag{2.85}$$

$$\text{(edge-unnormalized)} \quad \mathrm{kNCut}_{cl}(\{V_\ell\}_{\ell=1}^k) := \sum_{\ell=1}^k \frac{\mathrm{Cut}_{cl}(V_\ell, V\backslash V_\ell)}{\mathrm{vol}(V_\ell)} \tag{2.86}$$

where $\mathrm{vol}(V_\ell) := \sum_{i\in V_\ell} d_i$ Using these, the k-way normalized hypergraph cut for is rewritten as

$$\begin{aligned}\text{(edge-normalized)} \quad & \min \mathrm{kNCut}_{ncl}(\{V_\ell\}_{\ell=1}^k) \\ & = \min Z_N^\top D_v^{-1/2} L_{ncl} D_{ncl}^{-1/2} Z_N \text{ s.t. } Z_N^\top Z_N = 1 \end{aligned} \tag{2.87}$$

$$\begin{aligned}\text{(edge-unnormalized)} \quad & \min \mathrm{kNCut}_{cl}(\{V_\ell\}_{\ell=1}^k) \\ & = \min Z_N^\top D_{cl}^{-1/2} L_{cl} D_{cl}^{-1/2} Z_N \text{ s.t. } Z_N^\top Z_N = 1, \end{aligned} \tag{2.88}$$

where the normalized incidence matrix Z_N (Eq. (2.4)) defined for the contracted graph A_{ncl} and A_{cl} and the partitioning $\{V_\ell\}_{\ell=1}^k$. For both approaches, it is NP-hard to optimize the cut objectives discretely. Thus, as a spectral clustering, we obtain the eigenvectors of graph Laplacian matrices, $L_{2,ncl}$ and $L_{2,cl}$, similarly to the graph case like Alg. 4.

Note that Saito et al. [2018] extends this 2-Laplacian to p-Laplacian in a vertex-wise way (See Eq. (2.45)) in a differential geometry way. In Saito et al. [2018], the p-energy $S_{G,p}^{VW}(\mathbf{x})$ is defined as

$$S_{G,p}^{VW}(\mathbf{x}) \coloneqq \sum_{i\in V} \left(\sum_{e\in E: e_{[1]}=i} \frac{w(e)}{(|e|-1)} \sum_{j\in e} \left(\frac{x_j}{\mu_j^{1/p}} - \frac{x_i}{\mu_i^{1/p}} \right)^2 \right)^{p/2}. \tag{2.89}$$

The detail of this formulation is discussed in Sec. 3.5.

Star Expansion (STAR). This way constructs a graph by making a new vertex for every edge to form a star [Zhou et al., 2006]. The hypergraph cut for this approach is defined as

$$\mathrm{Cut}_s(V_1, V_2) := \sum_{e\in E} w(e) \frac{|e\cap V_1||e\cap V_2|}{|e|} \tag{2.90}$$

$$= \sum_{e\in E} \sum_{j_1, j_2 \in e; j_1\in V_1, j_2 \in V_2} \frac{w(e)}{|e|}. \tag{2.91}$$

Similar to the clique approach, we define the Hypergraph 2-Laplacian for star $L_{2,s}$ corresponding this cut can be written as

$$L_{2,s} := I - D_v^{-1/2} A_s D_v^{-1/2} \tag{2.92}$$

$$\tag{2.93}$$

where

$$A_s := H W_e D_e^{-1} H^\top. \tag{2.94}$$

We remark that this view is also hypergraph contraction to graph, represented by adjacency matrix A_s. Note also that this Laplacian can be seen as the standard Laplacian if we consider the hypergraph as a graph, except for the coefficient 1/2. This coefficient difference comes from the nature of this view, as discussed in [Saito et al., 2018].

We define k-way normalized hypergraph cut for star expansion problem as

$$\mathrm{kNCut}_s(\{V_i\}_{i=1}^k) := \sum_{i=1}^k \frac{\mathrm{Cut}_s(V_i, V \backslash V_i)}{\mathrm{vol}(V_i)} \tag{2.95}$$

We can rewrite the minimization problem of Eq. (2.95) as

$$\min_{\{V_i\}_{i=1}^k} \mathrm{kNCut}(\{V_i\}_{i=1}^k) = \min_{Z_N} \mathrm{trace} Z_N^\top D_v^{-1/2} L_s D_v^{-1/2} Z_N \text{ s.t.} Z_N^\top Z_N = I \tag{2.96}$$

$$= \max_{Z_N} \mathrm{trace} Z_N^\top D_v^{-1/2} A_s D_v^{-1/2} Z_N \text{ s.t.} Z_N^\top Z_N = I. \tag{2.97}$$

Again, we obtain the k-th eigenvectors of L_s as a hypergraph spectral clustering.

Note on Graph Contraction Approach for Uniform Hypergraphs. We firstly introduce the contraction way for *any* hypergraph. We then discuss all of these, two clique-ways and star expansion way are equivalent for r-uniform hypergraphs. Recall the hypergraph notations in Sec. 2.5.2. The star adjacency matrix is A_s, and two clique ways, edge-normalized way by [Saito et al., 2018] as A_{nc} and edge-unnormalized way by [Rodriguez, 2002] as A_c. For r-uniform hypergraphs, the cuts are rewritten as eigenproblem of Laplacians $L_{2,cl} := D_{cl} - A_{cl}$ and $L_{ncl} := D_v - A_{ncl}$, but $L_{2,ncl} = (r-1)L_{2,s} = rL_{2,ncl}$ for an r-uniform hypergraph. Thus, the results of the spectral clustering are the same for these because the only difference

between these is the constant coefficient, which does not affect the eigenproblem results.

2.5.2.2 Submodular Approaches

This section introduces the other way of the hypergraph Laplacian. This approach first established by Hein et al. [2013] as a generalization of total variational approach of the graph Laplacian, that becomes the submodular problem. Later year, [Li and Milenkovic, 2018] extends the approach by Hein et al. [2013] to general submodular functions.

Total Variation (TV/SUB). The total variation (TV) approach for hypergraph has been considered in a different context than the other two [Hein et al., 2013]. The hypergraph cut for this approach is defined as

$$\mathrm{Cut}_{TV}(V_1, V\backslash V_1) = \sum_{e\in E, e\cap V_1\neq\varnothing, e\cap V\backslash V_1\neq\varnothing} w(e), \tag{2.98}$$

and then balanced in a Cheeger way as

$$\mathrm{CCut}_{TV}(V_1) = \frac{\sum_{e\in E, e\cap V_1\neq\varnothing, e\cap V\backslash V_1\neq\varnothing} w(e)}{\min(\mathrm{vol}(V_1), \mathrm{vol}(V\backslash V_1))}. \tag{2.99}$$

Using the submodular gradient decent by Hein et al. [2013], this can be approximately solvable.

The Hein et al. [2013] did not explicitly propose the Laplacian. Instead, Hein et al. [2013] discussed the energy form of this total variation in p-seminorm manner, a regularizer, is defined as

$$S^{TV}_{G,p}(\mathbf{x}) := \sum_{e\in E} w(e) \max_{i,j\in e} |x_i - x_j|^p, \tag{2.100}$$

which we shall call unnormalized total variation(TV V-UN), since this is not normalized by degrees. Note that Hein et al. [2013] did not discuss the algorithms for general $p > 1$.

Submodular p-Laplacian (SUB) The TV p-Laplacian is actually incorporated by the submodular p-Laplacian [Li and Milenkovic, 2018]. The extensive study by Li and Milenkovic [2018] considers hypergraph p-Laplacian in the context of a submodular function, which we refer to as SUB. This approach generalizes the energy of total variation Eq. (2.100) as follows.

For a submodular function $F : 2^{|e|} \to [0, 1]$, associated with edge e, the submodular energy;

$$S^{SUB}_{G,p}(\mathbf{x}) := \sum_{e \in E} (w(e) \max_{S \subset V}(F(S)) \left(\sum_{\ell=1}^{|e|-1} F(S_\ell) \left(\frac{x_{i_{\ell+1}}}{\mu_{i_{\ell+1}}^{1/p}} - \frac{x_{i_\ell}}{\mu_{i_\ell}^{1/p}} \right) \right)^p, \tag{2.101}$$

by reordering vertices in e as $x_{i_{|e|}} \geq x_{i_{|e|-1}} \geq \ldots \geq x_{i_1}$, where $S_\ell := \{i_j \in V\}_{j=1}^{\ell}$. Note that this form is one form of the Lovász extension. By taking $F(S_i) = 1$ for all i, we obtain TV energy. The p-Laplacian for this submodular p-Laplacian [Li and Milenkovic, 2018] is associated as

$$\langle \mathbf{x}, \Delta_p \mathbf{x} \rangle := S^{SUB}_{G,p}(\mathbf{x}). \tag{2.102}$$

Li and Milenkovic [2018] proposed the algorithms to obtain the second eigenvector of this hypergraph p-Laplacian for the $p = 1$ and $p = 2$ cases using the submodular technique as well.

2.5.3 Brief History of Hypergraph Laplacians and p-Laplacians

This section briefly discusses the history of the hypergraph Laplacians and p-Laplacians.

To deal with hypergraphs, one major formulation of hypergraph Laplacian is to reduce hypergraphs to graphs. Until around 2010, hypergraph 2-Laplacians are proposed in various literature in this line of research. Agarwal et al. [2006] showed that the various hypergraph 2-Laplacians in the line of reduction can be classified into two categories. One category is a star hypergraph 2-Laplacian. The star Laplacian contracts a hypergraph into a graph by composing the hyperedge into graph edges by making a star for each vertex. The various work such as [Zien et al., 1999, Li and Solé, 1996, Zhou et al., 2006] can be seen as in this star contraction way. The other category is a clique hypergraph 2-Laplacian. The clique hypergraph 2-Laplacian contracts hypergraphs into graphs by expanding the hyperedge into graph cliques. This Laplacian includes hypergraph 2-Laplacian includes hypergraph Laplacian by Rodriguez [2002] as well as Bolla [1993], and Gibson et al. [2000]. When expanding the hyperedge into the clique, Saito et al. [2018] normalized the weight of the edge by the number of vertices to which the hyperedge connects. Among the hypergraph Laplacians above, Saito et al. [2018] hypergraph 2-Laplacian is the one that corresponds to the graph Laplacian when we consider a graph as a hypergraph.

In 2013, Hein et al. [2013] proposed the regularizer using p-seminorm. Although Hein et al. [2013] defined the regularizer that involves p-seminorm, they never mentioned "p-Laplacian" in their paper. The first hypergraph p-Laplacian is introduced by Saito et al. [2018]. This hypergraph p-Laplacian generalizes the vertex-wise graph Laplacian Eq. (2.45). However, this vertex-wise graph Laplacian does not enjoy nice theoretical properties that graph p-Laplacian has, such as nodal domain theorem and Cheeger inequality as the original graph one does not. Slightly after hypergraph p-Laplacian by Saito et al. [2018], Li and Milenkovic [2018] defines the hypergraph p-Laplacian by generalizing Hein et al. [2013]. Contrary to hypergraph p-Laplacian by Saito et al. [2018], this hypergraph p-Laplacian satisfies nodal domain theorem and Cheeger inequality.

2.6 Summary

In combination with Sec. 1.2 and Sec. 1.3, this chapter provides a comprehensive review of topics related to this thesis.

Sec. 2.1 covers spectral clustering for graphs. In Sec 2.2, we introduced a method for modeling graphs from vector data using "spectral connections." In Sec. 2.3, we reviewed the history of spectral clustering as well as spectral connection. In Sec. 2.4, we also explored the analogy between circuits and graphs, focusing on effective resistance. Lastly, Sec. 2.5 introduced the hypergraph Laplacian and p-Laplacian as generalizations of their graph counterparts in Sec. 2.1.

These notions and techniques will be frequently used in subsequent chapters, forming a foundation for the discussions to come.

Chapter 3

Generalizing p-Laplacian: Spectral Hypergraph Theory and a Partitioning Algorithm

For hypergraph clustering, various methods have been proposed to define hypergraph p-Laplacians in the literature. This work proposes a general framework for an abstract class of hypergraph p-Laplacians from a differential-geometric view. This class includes previously proposed hypergraph p-Laplacians and also includes previously unstudied novel generalizations. For this abstract class, we extend current spectral theory by providing an extension of nodal domain theory for the eigenvectors of our hypergraph p-Laplacian. We use this nodal domain theory to provide bounds on the eigenvalues via a higher-order Cheeger inequality. Following our extension of spectral theory, we propose a novel hypergraph partitioning algorithm for our generalized p-Laplacian. Our empirical study shows that our algorithm outperforms spectral methods based on existing p-Laplacians.

3.1 Introduction

This chapter considers generalized hypergraph p-Laplacians. As we see in Chapter 1 and Sec. 2.5, hypergraphs generalize graphs and serve as a natural representation of multi-relational data. However, although a hypergraph is a natural data representation, generalization from graph Laplacian to hypergraph Laplacian is not straightforward. Thus, as we see in Sec. 2.5, multiple such generalizations have been proposed in the literature [Agarwal et al., 2006, Saito et al., 2018, Hein et al., 2013, Li and Milenkovic, 2018].

While these hypergraph Laplacians are proposed from a different view points, these share a similar structural foundation between energy and the p-Laplacian. However, despite sharing a similar structural foundation, some of these Laplacians miss key features, as shown in Table 3.1. This raises the question: Do these missing features result from fundamental limitations in the existing models despite their structural similarities? We argue that the answer is no; to address this, this work aims to construct a theoretical structure to bring these similar but disparate models into one unified framework.

In our unified framework, we define an abstract class of hypergraph p-Laplacians that incorporates a number of previously proposed hypergraph p-Laplacians as well as previously unstudied novel hypergraph p-Laplacians. This framework builds on a limited special case previously proposed in [Saito et al., 2018]. The overall framework is inspired by a differential-geometric analogy from the continuous to the discrete domain. Exploiting the differential-geometric connection, we provide a generalized nodal domain theorem (see Thm. 3.13) and a generalized Cheeger inequality (see Thm. 3.14 and Cor. 3.15). These provide a theoretical justification and bounds for using the eigenvectors of a hypergraph p-Laplacian to perform partitioning. Exploiting these theoretical results, we provide an algorithm for finding an approximation to the second eigenvector. We empirically demonstrate that our algorithm outperforms a variety of existing hypergraph p-Laplacian based methods.

We highlight five salient contributions of this work.

1. From a differential-geometric perspective, we define an abstract class of p-Laplacians of hypergraphs that can incorporate previously proposed p-Laplacians as well as novel unstudied p-Laplacians.
2. We provide theoretical results for our abstract class of p-Laplacians, such as the Nodal domain theorem, the Cheeger inequality, and a bound on the relationship between the minimum Cheeger cut and the second p-eigenvalue of the p-Laplacian.
3. Exploiting these theoretical results, we propose a convergent hypergraph partitioning algorithm with respect to our abstract class of hypergraph p-Laplacians.
4. We demonstrate empirically that our method can improve the performance of the existing p-Laplacians on the standard benchmark for hypergraph clustering research.
5. Based on our theoretical and empirical observations, we provide guidance on the choice

Table 3.1: Comparison table for existing methods and ours. STAR is studied in [Zhou et al., 2006], and unnormalized CLIQUE is studied in [Rodriguez, 2002] and edge-normalized CLIQUE is in [Saito et al., 2018]. The TV is first proposed in [Hein et al., 2013] and generalized to a submodular hypergraph [Li and Milenkovic, 2018]. The relation between Laplacian and energy serves as a foundation (See Prop. 3.4). See the main text for details.

	STAR	CLIQUE	TV	Submodular	Ours
Energy and Laplacian relation	✓	✓	✓	✓	✓
p-Laplacian		✓	✓	✓	✓
Nodal domain theorem				✓	✓
Cheeger inequality				✓	✓
Clustering algorithm for fixed p	✓	✓	✓	✓	✓
Clustering algorithm for any $p > 1$		✓			✓

of p-Laplacian.

We remark that in the literature, there are a number of special case results [Zhou et al., 2006, Agarwal et al., 2006, Saito et al., 2018, Hein et al., 2013]. These prior results derive a patchwork of key features, such as nodal domain theorems, Cheeger inequalities, and partitioning algorithms for some particular cases of hypergraph p-Laplacians, as shown in Table 3.1. The advantage of the approach here is that we define an abstract class of hypergraph p-Laplacians, and both our theory and our partitioning algorithm apply to the complete class. Finally, we provide guidance on selecting a particular value of p for hypergraph p-Laplacians. *All proofs are in Appendix.*

3.2 Hypergraph p-Laplacian

This section proposes a hypergraph p-Laplacian and discusses its p-eigenpairs. Here, we generalize graph differential geometric notions such as gradient, divergence, Laplacian, and cut objects to the general hypergraph setting.

3.2.1 Differential Operators: Gradient $\nabla_{c,p}$, Divergence $\mathrm{div}_{c,p}$ and p-Laplacian $\Delta_{c,p}$

In this section, we aim to extend various differential operators proposed in [Saito et al., 2018] to an abstract class of p-Laplacians. We consider differential operations space over the set of vertices and the set of the directed edges E_d made from edges of hypergraph E.

We firstly introduce the following two inner product spaces $\mathcal{H}(V)$ and $\mathcal{H}(E_d)$ of real-valued functions over the vertex set and the directed edge set respectively,

$$\langle \mathbf{f}, \mathbf{g} \rangle_{\mathcal{H}(V)} \coloneqq \sum_{i \in V} f_i g_i \tag{3.1}$$

$$\langle \mathbf{s}, \mathbf{t} \rangle_{\mathcal{H}(E_d)} \coloneqq \sum_{e \in E_d} \frac{s(e)t(e)}{|e|!}. \tag{3.2}$$

We next define three operators on these spaces; *gradient* $\nabla_{c,p}$:$\mathcal{H}(V) \to \mathcal{H}(E_d)$, *divergence* $\mathrm{div}_{c,p} : \mathcal{H}(E_d) \to \mathcal{H}(V)$, and *$p$-Laplacian* $\Delta_{c,p} : \mathcal{H}(V) \to \mathcal{H}(E_d)$. These operators are discrete geometric analogs to comparable operators in continuous differential geometry. In the continuous domain, for the second differentiable function f, the p-Laplace operator is defined as $\Delta_p^{(c)} f := \mathrm{div}^{(c)}(\|f\|^{p-2}\nabla^{(c)} f)$, where operators with superscripted by (c) are the standard continuous calculus ones. In the following, we would like to establish a differential-geometric framework in a generalized discrete setting analogous to the continuous one to define an abstract class of p-Laplacians. The operators of divergence and p-Laplacian were introduced in the graph setting [Zhou and Schölkopf, 2005, Grady, 2006] and generalized to the hypergraph setting in [Saito et al., 2018], whereas a similar formulation of gradient was given graph and hypergraph settings [Zhou and Schölkopf, 2005, Saito et al., 2018]. The definition that we propose below broadly generalizes all previous definitions. We define and discuss its interpretation below.

We propose to define the hypergraph-gradient as follows. The definition below is the generalization of the definition of gradient over hypergraphs proposed in Saito et al. [2018].

Definition 3.1. *Let $\nabla_{c,p}$ be an operator $\nabla_{c,p} : \mathcal{H}(V) \to \mathcal{H}(E_d)$. A hypergraph-gradient $\nabla_{c,p}$ is*

$$(\nabla_{c,p}\mathbf{x})(e) \coloneqq \sum_{i,j \in e} w^{\frac{1}{p}}(e) c^{\frac{1}{p}}(i,j,e,\mathbf{x}) \left(\frac{x_j}{\mu_j^{1/p}} - \frac{x_i}{\mu_i^{1/p}} \right),$$

where the operator $\nabla_{c,p}$ and the function $c(i,j,e,\mathbf{x})$ satisfies the following three conditions for all $e \in E_d$ and vertices $i, j \in e$;

$$|\nabla_{c,p}(\alpha \mathbf{x})| = |\alpha \nabla_{c,p}(\mathbf{x})| \tag{3.3}$$

$$\sum_{i,j \in e} c(i,j,e,\mathbf{x}) = c'(e) \tag{3.4}$$

$$\left.\frac{\partial c^{1/p}(i,j,e,\mathbf{x})}{\partial \mathbf{x}}\right|_i = 0, \forall j \in e \quad \text{and} \quad \left.\frac{\partial c^{1/p}(u,v,e,\mathbf{x})}{\partial \mathbf{x}}\right|_j = 0, \forall i \in e \tag{3.5}$$

This hypergraph-gradient can be intuitively interpreted as follows. The term $x_j/\mu_j^{1/p} - x_i/\mu_i^{1/p}$ can be interpreted as "roughness" (normalized by $\boldsymbol{\mu}$) between two vertices. The hypergraph-gradient is a sum of all possible combinations of this term for the edge e. Hence, the hypergraph-gradient can be intuitively seen as the roughness in one edge, similar to the continuous gradient $\nabla^{(c)}$ and the standard graph case discussed in Sec. 2.1.2.

The definition of the hypergraph-gradient function depends on a "weighting" function $c(i,j,e,\mathbf{x})$. This weighting function can be seen as a coefficient of the difference between every pair of vertices. Varying $c(i,j,e,\mathbf{x})$ allows us to model different types of hypergraph expansions including but not limited to the star [Zhou et al., 2006] or clique expansions [Saito et al., 2018] (see Table 3.2 for details), i.e., the function c enables the following our generalized p-Laplacian framework to be abstract.

We leave a few remarks on equations of gradient Def. 3.1. First, the gradient operator $\nabla_{c,p}$ and the function c has three conditions described as Eq. (3.3), Eq. (3.4), and Eq. (3.5). Eq. (3.3) requires the operator $\nabla_{c,p}$ to be either homogeneous or absolute homogeneous. Eq. (3.4) requires that the summation of the function over all the pairs of vertices at an edge e is independent of $\mathbf{x}$. Eq. (3.5) enforces that the function $c^{1/p}$ is independent of $\mathbf{x}$ once we fix one edge and one vertex in the edge. In the following, when c is not differentiable, we consider subdifferential instead of derivative. We will later discuss more details in Sec. 3.2.5. We normalize $\mathbf{x}$ by vertex weights $\boldsymbol{\mu}$. We call the vertex weights *unnormalized* when $\mu_i = 1$ and *normalized* when $\mu_i = d_i$ for all $i \in V$. We observe that the existing unnormalized p-Laplacian such as [Hein et al., 2013] $\mu_i = 1$ for all $i \in V$ and normalized 2-Laplacian [Zhou et al., 2006, Saito et al., 2018] when $\mu_i = d_i$ for all $i \in V$.

The following definition of a divergence operator is inspired by an analogy to the continuous setting.

Definition 3.2. *A hypergraph divergence is an operator* $\mathrm{div}_{c,p} : \mathcal{H}(E_d) \to \mathcal{H}(V)$ *which*

satisfies

$$\langle \mathbf{y}, \nabla_{c,p}\mathbf{x}\rangle_{\mathcal{H}(E_d)} = \langle \mathbf{x}, -\mathrm{div}_{c,p}\mathbf{y}\rangle_{\mathcal{H}(V)}, \quad \forall \mathbf{x} \in \mathcal{H}(V), \forall \mathbf{y} \in \mathcal{H}(E_d). \tag{3.6}$$

Note that Def. 3.2 is an analog to the continuous Stokes' Theorem. Also, we can check that div is unique. Intuitively, divergence counts the net flow defined by ϕ on the vertex, similar to the intuition in the continuous domain.

Finally, we propose to define p-Laplacian.

Definition 3.3. *An operator $\Delta_{c,p} : \mathcal{H}(V) \to \mathcal{H}(V)$ is a hypergraph p-Laplacian if*

$$\Delta_{c,p}\mathbf{x} := -\mathrm{div}_{c,p}(\|\nabla_{c,p}\mathbf{x}\|_p^{p-2}\nabla_{c,p}\mathbf{x}) \tag{3.7}$$

.

3.2.2 p-Dirichlet Sum and p-Laplacian

This section defines the p-Dirichlet sum, which can be interpreted as energy over the hypergraph. Also, we discuss relations between the p-Dirichlet sum and the p-Laplacian. Lastly, we discuss how these relations are the foundation of graph partitioning.

Using the norm defined by the Hilbert space in Eq.(3.2), we define *p-Dirichlet sum* of $\mathbf{x} \in H(V)$ as

$$S_{G,c,p}(\mathbf{x}) := \|\nabla_{c,p}\mathbf{x}\|_p^p = \sum_{e\in E_d} \frac{|(\nabla_{c,p}\mathbf{x})(e)|^p}{|e|!}, \tag{3.8}$$

which measures roughness of $\mathbf{x}$ over the hypergraph. Hence, it is natural to interpret the p-Dirichlet sum as an energy over a hypergraph. Later we will use this energy as the objective function of the hypergraph partitioning.

For the p-Dirichlet sum and p-Laplacian, the following relationships hold;

Proposition 3.4. $S_{G,c,p}(\mathbf{x}) = \langle \mathbf{x}, \Delta_{c,p}\mathbf{x}\rangle_{\mathcal{H}(V)}$.

Proposition 3.5. $\left.\dfrac{\partial S_{G,c,p}(\mathbf{x})}{\partial \mathbf{x}}\right|_i = p\Delta_{c,p}\mathbf{x}(i)$.

These relations are important both in the continuous and discrete domains. In the continuous domain, the analog of these relations is fundamental for an important problem on

p-Laplacian, called *Dirichlet Principle* [Courant and Hilbert, 1962]; the Dirichlet energy is minimized when the Laplace equation is satisfied. For the clustering in the discrete domain, we minimize the p-Dirichlet sum. To do so, we consider a problem similar to the Laplace equation, which is the eigenproblem of Laplacian. The discussion above shows how we see an analogy between continuous differential and discrete geometry, as illustrated in Sec. 2.1.2.

We often see the properties of Prop. 3.4 and Prop. 3.5 in the graph p-Laplacian [Bühler and Hein, 2009, Bougleux et al., 2009]. Moreover, also in the hypergraph context, without a defining differential geometric setup, we see these properties in the existing hypergraph Laplacians, as seen in Table 3.1 [Zhou et al., 2006, Saito et al., 2018, Hein et al., 2013, Li and Milenkovic, 2018]. Hence, it is natural to expect that all of the hypergraph Laplacians have a similar structure in this sense. However, as we see in Table 3.1, some Laplacians miss some features; particularly, some Laplacians miss the useful nodal domain theorem and Cheeger inequality (discussed in Sec. 3.3). Note that these results are "borrowed" from the continuous differential geometry. One of the benefits of our abstract Laplacian is to give comprehensive analyses to all Laplacians defined from the gradient Def. 3.1.

3.2.3 p-Eigenproblem of p-Laplacian

Next, we discuss the eigenproblem of this p-Laplacian. Since a p-Laplace operator is nonlinear, we introduce the standard generalization of eigenpair for p-Laplacian (see for examples of [Tudisco and Hein, 2018]).

Definition 3.6. *Let $\xi_p(x) \coloneqq |x|^{p-1}\mathrm{sgn}(x)$. For $p > 1$, a hypergraph p-eigenpair, which is a pair of p-eigenvalue $\lambda \in \mathbb{R}$ and p-eigenvector $\boldsymbol{\psi} \in \mathcal{H}(V)$ of $\Delta_{c,p}$, is defined by*

$$(\Delta_{c,p}\boldsymbol{\psi})_i = \lambda\xi_p(\psi_i), \forall i \in V. \tag{3.9}$$

The standard Laplacian shows the connection between its eigenpair and Rayleigh quotient from the matrix theory and the continuous analysis. To obtain p-eigenpair, we consider the following Rayleigh quotient:

Proposition 3.7. *Consider the Rayleigh quotient for our abstract class of p-Laplacians as*

$$R_{G,c,p}(\mathbf{x}) \coloneqq \frac{S_{G,c,p}(\mathbf{x})}{\|\mathbf{x}\|_p^p}. \tag{3.10}$$

The function $R_{G,c,p}$ has a critical point at $\boldsymbol{\psi}^*$ if and only if $\boldsymbol{\psi}^*$ is p-eigenvector of $\Delta_{c,p}$. The corresponding p-eigenvalue λ^* is given as $\lambda_p^* = R_{G,c,p}(\boldsymbol{\psi}^*)$. Moreover, the first p-eigenvalue is 0, whose p-eigenvector is $M^{1/p}\mathbf{1}$, where M is a $|V| \times |V|$ diagonal matrix whose diagonal element is μ_v.

For the standard Laplacians, the first p-eigenvector is $\mathbf{1}$ for unnormalized and $D_v^{1/2}\mathbf{1}$ for normalized case.

3.2.4 Variational Hypergraph p-Laplacians

We firstly remark that the definition of p-eigenpair (Def. 3.6) leads to existing an infinite number of p-eigenpairs, similarly to the continuous case [Binding and Rynne, 2008]. Now, we discuss the subset of eigenpairs, similarly to the graph p-Laplacian case as seen in Sec. 2.1.4.

We move our discussion to a property of multiplicity of first p-eigenvalues.

Proposition 3.8. *Suppose that hypergraph G is a union of k independent and connected hypergraphs G_i ($i = 1, \ldots, k$), i.e, $G = \bigcup_{i=1}^{k} G_i$ where $G_j \cap G_l = \varnothing, \text{for}, j \neq l$. Then, k equals to the multiplicity of eigenvalue 0 of $\Delta_{c,p}$.*

The following corollary follows from this proposition.

Corollary 3.9. *The second p-eigenvalue of $\Delta_{c,p}$ is greater than 0, if a hypergraph G is connected.*

To analyze critical point of Eq. (3.10), Index theory [Struwe, 2000] is useful. We use Krasnoselskii genus defined in Eq. (2.40), which we use for the discussion of graph p-Laplacian (See Sec. 2.1.4). Since $R_{G,c,p}(\alpha\mathbf{x}) = R_{G,c,p}(\mathbf{x})$ by Cor. 3.20, to consider the p-eigenpair of p-Laplacian, we can limit our interest to $\mathcal{S}_p \coloneqq \{\mathbf{x} \mid \|\mathbf{x}\|_p^p = 1\}$. From the results in discrete case [Chang, 2016, Tudisco and Hein, 2018, Li and Milenkovic, 2018] and continuous case [Lindqvist, 2008], we obtain the following proposition, which is a generalized Rayleigh min-max theorem.

Proposition 3.10. *Consider the set of subsets $\mathcal{F}_k(\mathcal{S}_p) = \{B \subset \mathcal{S}_p \mid B = -B, \text{ closed }, \gamma(B) \geq k\}$. The sequence defined as*

$$\lambda_{c,p,k} := \min_{B \subset \mathcal{F}_k(\mathcal{S}_p^n)} \max_{\mathbf{x} \in B} R_{G,c,p}(\mathbf{x}) \tag{3.11}$$

Table 3.2: The relationship between a function $c(i, j, e, \mathbf{x})$ in hypergraph-gradients and Laplacians. We denote $e_{[1]}$ by the first vertex of an edge e. Also, $F : 2^{|e|} \to [0, 1]$ is a submodular function, and we use rearranged vertices i_ℓ so that $x_{i_{|e|}} \geq \ldots \geq x_{i_1}$. See Sec. 2.5.2 and Sec. 3.2.5 for the details and all the notations.

Type	$c(i, j, e, \mathbf{x})$
CLIQUE E-N	$1/(\lvert e\rvert - 1)$ when $e_{[1]} = i$ otherwise 0
CLIQUE E-UN	1 when $e_{[1]} = i$ otherwise 0
STAR	$1/\lvert e\rvert$ when $e_{[1]} = i$ otherwise 0
TV	1 when $(i, j) = \arg\max_{i,j \in V} \lvert x_j/\mu_j^{1/p} - x_i/\mu_i^{1/p}\rvert$ otherwise 0
SUB	$F(S_\ell)(\max_{S \subset e}(F(S)))^{1/p}$ when $i = i_\ell$ and $j = i_{\ell+1}$ otherwise 0

gives a critical point of $R_{G,c,p}(\mathbf{x})$. Moreover, the pair of $\lambda_{c,p,k}$ and the vector $\boldsymbol{\psi}_{c,p,k}$ such that $\lambda_{c,p,k} = R_{G,c,p}(\boldsymbol{\psi}_{c,p,k})$ constitutes an eigenpair of $\Delta_{c,p}$.

Similarly to the Rayleigh minmax theorem and the graph p-Laplacian case discussed in Sec. 2.1.4, this proposition yields the sequence of p-eigenpairs. Moreover, for a standard Laplacian of standard graph, this reduces into Rayleigh min-max theorem. However, similarly to the continuous p-Laplacian theory [Lindqvist, 2008], we do not know if this sequence yields exhaustive p-eigenpairs. We call the p-eigenpairs obtained by the sequence in Prop. 3.10 as *variational* p-eigenpairs.

3.2.5 p-Laplacians and Related Regularizers

This section shows that various hypergraph Laplacians in Sec. 2.5.2 and related regularizers can be seen as a special case of our framework.

The hypergraph Laplacians discussed in Sec. 2.5.2 can be seen as a special case of our abstract Laplacian, defined by Def. 3.3, followed by hypergraph-gradient (Def. 3.1) and hypergraph-divergence (Def. 3.2). Table 3.2 summarizes the corresponding function $c(i, j, e, \mathbf{x})$ in the definition of hypergraph-gradient.

Proposition 3.11. *The weighting functions c in Table 3.2 correspond to the respective hypergraph p-Laplacians in Sec. 2.5. Also, the weighting functions satisfy the conditions of Def. 3.1.*

Edge normalized and unnormalized clique 2-Laplacians in Table 3.2 are 2-Laplacians proposed by Saito et al. [2018] and Rodriguez [2002], respectively. Star 2-Laplacian in

Table 3.2 is equal to the Laplacian proposed by [Zhou et al., 2006]. The regularizer of unnormalized TV p-Laplacian in Table 3.2 corresponds to one by [Hein et al., 2013]. The Submodular hypergraph p-Laplacian (SUB) is proposed by [Li and Milenkovic, 2018].

Finally, for the family of total variation, we here propose a normalized TV p-Laplacian, whose p-energy as

$$S^{TV}_{G,p}(\mathbf{x}) := \sum_{e \in E} w(e) \max_{i,j \in e} \left| \frac{x_i}{d_i} - \frac{x_j}{d_j} \right|^p . \tag{3.12}$$

3.3 Properties of Variational p-Eigenpair of p-Laplacian

This section discusses the properties of the p-eigenproblem of our hypergraph p-Laplacian. Hence, we aim to establish the theoretical background of spectral clustering using p-Laplacian, such as the nodal domain theorem and the Cheeger inequality. The nodal domain theorem is about the bounds of the number of nodal domains, which can be seen as a "division". Using this nodal domain, the Cheeger inequality shows how much p-eigenproblem can approximate a minimal graph cut.

3.3.1 Nodal Domain Theorem of the p-Laplacian

This section aims to extend the classical nodal domain theorem to our framework. The nodal domain theorem in the discrete domain is developed analogously from Courant's nodal domain theorem in the continuous domain [Courant and Hilbert, 1962]. In the continuous case, a *nodal domain* is defined as a region for a function where a sign does not change. Therefore, a nodal domain marks the natural division of regions of real values. The nodal domain theorem shows a connection between eigenvectors of Laplacian and nodal domains; the theorem describes the bounds of the number of nodal domains of eigenvectors of Laplacian [Lindqvist, 2008]. The same idea can be established in the discrete domain, i.e., a nodal domain is a connected sub-hypergraph where a sign of p-eigenvector does not change. This nodal domain can be seen as a "partition" by the p-eigenvector in the discrete domain. The next question is "can we obtain a similar bound to the number of this nodal domain?"

We begin with the definition of a nodal domain for a hypergraph.

Definition 3.12. *A nodal domain is a maximally connected subgraph G' of hypergraph G such that G' is either $\{i \mid x_i > 0\}$ or $\{i \mid x_i < 0\}$ for $\mathbf{x} \in H(V)$.*

Next, with this definition, we discuss the nodal domain theorem for our hypergraph p-Laplacian. The nodal domain theorem for graph Laplacian has been proven in Fiedler [1975b], generalized to graph p-Laplacian by Tudisco and Hein [2018], and extended to a particular type of hypergraph p-Laplacian Li and Milenkovic [2018]. In this line of research, we extend these nodal domain theorems to our abstract class of hypergraph p-Laplacians as follows;

Theorem 3.13. *Let* $0 = \lambda_{c,p,1} < \lambda_{c,p,2} \leq \ldots \leq \lambda_{c,p,k-1} < \lambda_{c,p,k} = \ldots = \lambda_{c,p,k+r-1} < \lambda_{c,p,k+r} \leq \ldots$*, be variational eigenvalues of* $\Delta_{c,p}$*, and* $\boldsymbol{\psi}_{c,p,k}$ *is an associated variational eigenvector with* $\lambda_{c,p,k}$*. Then* $\boldsymbol{\psi}_k$ *induces at most* $k + r - 1$ *nodal domains.*

As seen in Thm. 3.13, the nodal domain theorem studies the structure of p-eigenvectors of p-Laplacian; Thm. 3.13 shows the bound on the number of nodal domains of p-eigenvectors. The number of nodal domains matters to *Cheeger inequality*, which is a theoretical justification for spectral methods via our p-Laplacian. We will discuss this Cheeger inequality next.

3.3.2 k-way Cheeger Inequality

This section establishes the k-way Cheeger inequality for our hypergraph p-Laplacian. As we saw in Sec. 2.1.3, the 2-way Cheeger inequality serves as the connection between Cheeger cut and eigenproblem. Moreover, the inequality gives a performance guarantee of the relaxed graph partitioning problem. We want to establish such a connection between the Cheeger cut and p-eigenproblem of our hypergraph p-Laplacian. For this purpose, we aim to generalize this Cheeger inequality to our hypergraph p-Laplacians to achieve spectral partitioning via our p-Laplacian.

We start our discussion from a 2-way Cheeger cut. Let $U \subset V$ be a set and $\overline{U}$ be a complement of A. The generalized Cheeger cut may be defined as

$$\mathrm{CCut}_c(U) := \frac{\mathrm{Cut}_c(U, \overline{U})}{\min(\mathrm{vol}(U), \mathrm{vol}(\overline{U}))}, \quad \text{where } \mathrm{vol}(U) = \sum_{i \in U} \mu_i \tag{3.13}$$

$$\mathrm{Cut}_c(U, \overline{U}) := \sum_{e: i,j \in e, i \in U, j \in \overline{U}} w(e) c'(e), \quad c'(e) := \sum_{i,j \in e} c(i, j, e, \mathbf{x}). \tag{3.14}$$

We call the optimal cut

$$h_{c,2} := \min_{U \subset V} \mathrm{CCut}_c(U)$$

as *Cheeger constant*. Considering the standard graph, this generalized Cheeger cut becomes the standard Cheeger cut discussed in Sec. 2.1.3. We shall extend this generalized 2-way Cheeger cut to k-way Cheeger cut. Consider disjoint partitioning of V into k sets $\{V_i\}_{i=1,\cdots,k}$. Then, we define the k-way Cheeger constant as

$$h_{c,k} \coloneqq \min_{\{V_i\}_{i=1,\cdots,k}} \max_{j\in\{1,\ldots,k\}} \mathrm{CCut}_c(V_j). \tag{3.15}$$

Similarly to the previous studies [Tudisco and Hein, 2018, Li and Milenkovic, 2018], we establish k-way Cheeger inequality for our p-Laplacian as follows.

Theorem 3.14. *Let $(\lambda_{c,p,k}, \boldsymbol{\psi}_{c,p,k})$ be a variational p-eigenpair of $\Delta_{c,p}$, $m_{c,k}$ be the number of nodal domains of $\boldsymbol{\psi}_{c,p,k}$. Then,*

$$\left(\max_{i\in V}\frac{d_i}{\mu_i}\right)^{-(p-1)}\left(\frac{h_{c,m_k}}{p}\right)^p \le \lambda_{c,p,k} \le \min(k, \max_{e\in E}|e|)^{p-1}h_{c,k}.$$

Corollary 3.15. *Let $(B, \overline{B})$ be the cut found by the second eigenvector of the p-Laplacian ψ, such that $(\{i : x_i \ge t\}, \{i : x_i < t\})$ minimizing Cheeger cut. Then,*

$$\left(\frac{1}{\max_i d_i/\mu_i}\right)^{p-1}\left(\frac{\mathrm{CCut}_c(B)}{p}\right)^p < 2h_{c,2} \tag{3.16}$$

Thm. 3.14 is an extension of the graph Cheeger inequality in terms of three perspectives; graph to hypergraph, 2-way to k-way, and the standard 2-Laplacian to our abstract class of p-Laplacians. Following Thm. 3.14, Cor. 3.15 is the bound of the relationship between the cut obtained by the second p-eigenvector of our abstract class of p-Laplacians and the generalized Cheeger constant. Similarly to the classical case in Sec. 2.1.3, Thm. 3.14 shows how we approximate the k-way Cheeger constant by relaxing discrete k-way cut problem into p-eigenproblem of $\Delta_{c,p}$; Thm. 3.14 gives the upper and lower bounds of the optimal cut using k-th p-eigenvalue. Moreover, Cor. 3.15 gives a guarantee for the worst case of a 2-way cut obtained by p-eigenvector. These bounds can be said to guarantee the performance of the cut resulting from spectral methods via p-eigenvectors of our p-Laplacian. Hence, Thm. 3.14 and Cor. 3.15 motivate us to use spectral methods via our p-Laplacian for the hypergraph partitioning problem instead of the costly discrete original cut problem. This inequality gives the tightest bound when $p \to 1$. Since the original cut problem is NP-hard, the eigenproblem

is also an NP-hard problem in this asymptotic case. Moreover, considering the standard graph 2-Laplacian, this inequality can be reduced to the classical Cheeger inequality. Also, when $k = 2$, this inequality is for h_2, which is a 2-way Cheeger cut for graphs. Therefore, in the next section, we focus on constructing a spectral algorithm for a 2-way partitioning.

Finally, we remark that the discussion on k-way Cheeger cut is a generalization of the standard graph 2-way Cheeger inequality of 2-Laplacian [Alon and Milman, 1985, Alon, 1986], k-way Cheeger inequality of 2-Laplacian [Lee et al., 2014], k-way Cheeger inequality of graph p-Laplacian [Tudisco and Hein, 2018], and k-way Cheeger inequality of p-Laplacian of submodular hypergraph [Li and Milenkovic, 2018] cases. We also note that the proofs for the nodal domain theorem (Thm. 3.13) and the Cheeger inequality (Thm. 3.14) are a natural generalization of the previous studies such as [Tudisco and Hein, 2018] and [Li and Milenkovic, 2018]. Rather than introducing new techniques in the proofs, the focus of this work is that we generalize the hypergraph p-Laplacian as much as possible where these structures preserve in order to provide a unified framework.

3.4 Hypergraph Partitioning via p-Laplacian

Sec. 3.3 shows the guarantee of performance of the eigenproblem instead of the NP-hard discrete Cheeger cut problem. Therefore, this section establishes our partitioning algorithm, exploiting p-eigenpairs of our hypergraph p-Laplacian.

We first discuss a property of p-eigenvectors of $\Delta_{c,p}$. For the p-Laplacian eigenproblem, since the p-Laplacian is nonlinear, p-eigenvectors are not necessarily orthogonal. However, we still want a relationship between p-eigenvectors. For this motivation, instead of the orthogonality, [Luo et al., 2010] proposed p-orthogonality as follows.

Definition 3.16 ([Luo et al., 2010])**.** *Let $\mathbf{\Xi}_p(\mathbf{x})$ be a vector, whose v-th element is $\xi_p(x_i)$. We call $\mathbf{x} \neq 0$ and $\mathbf{x}' \neq 0$ as p-orthogonal if $\mathbf{\Xi}_p(\mathbf{x})^\top \mathbf{\Xi}_p(\mathbf{x}') = 0$.*

In order to analyze this p-orthogonality of our abstract class of p-Laplacians, we recall the Taylor expansion, which is often used for approximating functions in physics. For example, in the motion of a pendulum, if we approximate functions with respect to the amplitude of the angular of the pendulum by Taylor expansion, the motion equation is approximated by a simple harmonic motion [Courant and Hilbert, 1962]. The Taylor expansion leads an infinite

differentiable function $f(x)$ to write as

$$f(x) = f(a_{f,x}) + \sum_{n=1}^{\infty} \frac{f^{(n)}(x - a_{f,x})}{n!}, \tag{3.17}$$

where $a_{f,x}$ is a constant, and $f^{(n)}$ is a n-th derivative of f. This Taylor expansion is often used to approximate the function. If we consider approximating the function by the first order, the remainder (the second or higher terms) can be seen as the approximation error. For two functions f and g, if the error term can be written as the sum of the second or higher terms, i.e.,

$$f(x) = g(x) + o_2, \quad \text{where} \quad o_2 = \sum_{n_f+n_g \geq 2, n_f, n_g \in \mathbb{N}} \beta_{f,g,n_f,n_g} (x - a_{f,x})^{n_f} (x - a_{g,x})^{n_g}, \tag{3.18}$$

and β_{f,g,n_f,n_g} is a coefficient, then we call the function f is equal to g up to the second order of Taylor expansion. Using this notion and p-orthogonality, we obtain the following;

Theorem 3.17. *Let $(\boldsymbol{\psi}_c, \lambda_c^{\psi})$ and $(\boldsymbol{\psi}'_c, \lambda_c^{\psi'})$ be the p-eigenpairs of $\Delta_{c,p}$. The p-eigenvectors $\boldsymbol{\psi}_c$ and $\boldsymbol{\psi}'_c$ are p-orthogonal up to the second order of Taylor expansion with the vertex if λ_c^{ψ} and $\lambda_c^{\psi'}$ are not equal up to the second order of Taylor expansion.*

Note that for this theorem the p-eigenpairs are not necessarily variational. Thm. 3.17 tells us that two p-eigenvectors are approximated p-orthogonal, up to the second-order of Taylor expansion.

We move our discussion to the second p-eigenpair by considering the Rayleigh quotient. In the graph, p-Laplacian [Bühler and Hein, 2009] and the clique p-Laplacian case [Saito et al., 2018] and also the continuous case [Lindqvist, 2008], the global minimum of a variant of Rayleigh quotient gives the second p-eigenpair. Similarly to these works, we propose to define the following quotient as

$$R^{(2)}_{G,c,p}(\mathbf{x}) \coloneqq \frac{S_{G,c,p}(\mathbf{x})}{\min_{\eta} \|\mathbf{x} - \eta \boldsymbol{\psi}_{c,1}\|_p^p}, \tag{3.19}$$

where $\boldsymbol{\psi}_{c,1}$ is the first p-eigenvector. The following theorem supports this quotient.

Theorem 3.18. *The global solution of Eq.* (3.19) *is given by* $\boldsymbol{\psi}_c^*$*, where*

$$\boldsymbol{\psi}_c^* = \boldsymbol{\psi}_{c,p,2} + \eta^* \boldsymbol{\psi}_{c,p,1}, \text{where } \eta^* = \arg\min_{\eta} \|\boldsymbol{\psi}_{c,p,2} - \eta \boldsymbol{\psi}_{c,p,1}\|_p^p, \tag{3.20}$$

where $\boldsymbol{\psi}_{c,p,2}$ *is the second* p*-eigenvector.*

This theorem shows that we have an exact identification for the second p-eigenpair; minimizing Eq. (3.19) gives the second p-eigenpair of $\Delta_{c,p}$. Note that the second p-eigenpair of $\Delta_{c,p}$ is the second variational eigenpair of $\Delta_{c,p}$. However, the major disadvantage is that Eq. (3.19) is not convex and hence difficult to obtain the global optimum; optimization algorithms applied to Eq. (3.19) would give the local optimum instead of the global optimum.

Therefore, we next consider a strategy to get a better local optimum for a 2-way hypergraph partitioning. The idea to obtain a better optimum is using the *exact* p-orthogonality as a constraint, instead of the constraint "p-orthogonal up the second order", which each pair of p-eigenvectors must obey (Thm. 3.17). The reason why we use this strategy is as follows. Due to the non-convexity of Eq. (3.19), the solution obtained by an optimization algorithm can be a local optimum. However, this local optimum is not guaranteed to be p-orthogonal up to the second-order to the first p-eigenvector $\boldsymbol{\psi}_{c,p,1}$ while $\boldsymbol{\psi}_{c,p,2}$ is so. To gain a better optimal solution, we exploit Thm. 3.17, and we want a constraint that enforces the solution to be p-orthogonal up the second-order to $\boldsymbol{\psi}_{c,p,1}$. However, it is difficult to work directly with this constraint "p-orthogonal up the second order". To ease this difficulty, we propose to use an exact p-orthogonal condition as a constraint. Thanks to Thm. 3.17, this exact constraint can be seen as an approximated condition by the second order of Taylor expansion. We borrow this approximation idea from physics; it is common to approximate physical phenomena by the second order of Taylor expansion, such as the explained motion of a pendulum case. Following this discussion, we incorporate the exact p-orthogonality as a constraint. Then, we consider the optimization problem as,

$$\min_{\mathbf{x}} J(\mathbf{x}) = R^{(2)}_{G,c,p}(\mathbf{x}) \quad \text{s.t.} \quad \boldsymbol{\Xi}_p(\mathbf{x})^\top \boldsymbol{\Xi}_p(\boldsymbol{\psi}_{c,p,1}) = 0. \tag{3.21}$$

Since $R^{(2)}_{G,c,p}(\alpha_c \mathbf{x}) = R^{(2)}_{G,c,p}(\mathbf{x})$ for $\alpha \neq 0$, we can arbitrarily change the scale of ψ to

Algorithm 3 Natural Gradient Algorithms for Eigenvectors of p-Laplacian.

Input: hypergraph-gradient, and p
  Initialize $\mathbf{x}$
  while the change is not sufficiently small **do**
    $\mathbf{x}' \leftarrow \frac{\partial J'}{\partial \mathbf{x}} - \mathbf{x}(\frac{\partial J'}{\partial \mathbf{x}})^\top \mathbf{x}$
    $\mathbf{x} \leftarrow \mathbf{x} - \alpha \mathbf{x}'$
  end while
Output: Second p-eigenvector ψ of $\Delta_{c,p}$

$R^{(2)}_{G,c,p}(\mathbf{x})$. Hence, we add the scale constraints to Eq. (3.21) as

$$J'(\mathbf{x}) = R^{(2)}_{G,c,p}(\Xi_p^{-1}(\mathbf{x})) \quad \text{s.t.} \quad \mathbf{x}^\top \boldsymbol{\psi}_{c,p,1} = 0, \|\mathbf{x}\|_2^2 = 1, \tag{3.22}$$

which gives the same global minimum solution as Eq. (3.21). To solve Eq. (3.22), we propose to apply natural gradient algorithm [Amari, 1998] as shown in Algorithm 3, similarly to [Luo et al., 2010]. If we use a simple gradient method as $\partial J'/\partial \psi$, the orthogonal condition does not hold for each update. Instead of using this for the update of Algorithm 3, we use $\frac{\partial J'}{\partial \mathbf{x}} - \mathbf{x}(\frac{\partial J'}{\partial \mathbf{x}})^\top \mathbf{x}$ so that we can preserve the orthogonal condition in Eq.(3.22) [Luo et al., 2010]. The convergence of this algorithm is also guaranteed [Luo et al., 2010]. Finally, we discuss the computational time. It takes at most $O(|e|^2)$ to compute one gradient. For energy, we need to directed edge, where we compute $|e|$ times for one edge by taking the symmetricity. Thus, the complexity time is $O(\sum_{e \in E} |e|^3)$ for one iteration.

3.5 Related Work

This section compares related hypergraph 2-Laplacians and p-Laplacians and partitioning algorithms. This section is complementary to Sec. 3.2.5. While Sec. 3.2.5 defines the related p-Laplacians, this section focuses on discussing the context and explaining the difference between ours and existing ones.

One major hypergraph Laplacian is from a clique expansion way (CLIQUE). The unweighted setting edge-unnormalized 2-Laplacian was proposed in [Rodriguez, 2002] (CLIQUE E-UN). This 2-Laplacian and the Laplacians proposed in other studies [Zien et al., 1999, Bolla, 1993, Gibson et al., 2000] are theoretically equivalent [Agarwal et al., 2006]. In this line of research, 2-Laplacian from a differential geometry viewpoint is proposed [Saito et al., 2018]. When $p = 2$, this Laplacian also can be explained by the clique expansion way

but normalized by a degree of edge (CLIQUE E-N). Moreover, this p-Laplacian is proposed based on forming vertex-wise energy (CLIQUE E-N-VW) [Saito et al., 2018], while ours is edge-wise energy. In Saito et al. [2018], the p-energy $S^{VW}_{G,p}(\mathbf{x})$ is defined using the norm of the hypergraph-gradient is defined $\nabla^{VW}_p \mathbf{x}$ at vertex i (originally for edge-normalized clique gradient) as

$$S^{VW}_{G,p}(\mathbf{x}) \coloneqq \sum_{i \in V} \|(\nabla^{VW}_p \mathbf{x})(i)\|^p, \text{where } \|(\nabla^{VW}_p \mathbf{x})(i)\| \coloneqq \left(\sum_{e \in E: e_{[1]}=i} \frac{|\nabla^{VW}_p \mathbf{x}(e)|^2}{|e|!} \right)^{\frac{1}{2}}. \tag{3.23}$$

This idea comes from the definition of the energy around the vertex i as $\|\nabla \mathbf{x}(i)\|$ and obtains total energy by summing up those energies over all vertices. Note that if we assume the standard graph, p-Laplacian in [Saito et al., 2018] corresponds to a series of graph studies [Zhou and Schölkopf, 2005, Bougleux et al., 2009], which also assume vertex-wise energy, as discussed in Sec. 2.1.4.6. On the other hand, ours corresponds to the graph p-Laplacian, which assumes edge-wise energy [Bühler and Hein, 2009, Tudisco and Hein, 2018], as discussed in Sec. 2.1.4.1. Hence, our work does not incorporate p-Laplacian proposed in [Saito et al., 2018] since the p-Dirichlet sum setting is different. Remark that when $p = 2$, our model incorporates CLIQUE E-N-VW by using c in Table 3.2. However, Saito et al. [2018] did not give theoretical analyses such as the nodal domain theorem or the Cheeger inequality. Moreover, [Saito et al., 2018] did not give a specific partitioning algorithm exploiting characteristics of p-Laplacian such as p-orthogonality. Hence, we need a general-purpose optimization method for the p-eigenproblem. However, such methods do not always leverage the characteristics of p-Laplacian, which could lead to better performance in terms of space, time, and accuracy.

Another line of research is in a star expansion way, shown in Section 3.2.5. Zhou et al. [2006] proposed 2-Laplacian based on a lazy random walk view. Agarwal et al. [2006] shows that this 2-Laplacian is theoretically equivalent to Laplacians by studies of [Zien et al., 1999, Li and Solé, 1996], also further discussed in [Ghoshdastidar and Dukkipati, 2017a].

Other Laplacian is from a total variation way and subsequent submodular way (TV/SUB). A regularization framework for $p \geq 1$ is proposed in [Hein et al., 2013] with hypergraph partitioning algorithm for $p = 1$, and further explored in [Chan et al., 2018]. This idea is

extended to a submodular hypergraph [Li and Milenkovic, 2018]. A submodular hypergraph has an objective energy function using one form of Lovász expansion of a submodular function. Moreover, SUB incorporates the inhomogeneous cut proposed by [Li and Milenkovic, 2017], where weights can vary when we partition the edge. Along with this new class of hypergraph cut, Li and Milenkovic [2018] proposed partitioning algorithms for $p = 1$ and $p = 2$. Seeing the definition (Eq. (2.101)), submodular p-Laplacian describes a broad class hypergraph p-Laplacian using the submodular function. We also mention that the $p = 2$ case for submodular cut objective functions is discussed in [Yoshida, 2019] using the general form of Lovász extension. Moreover, a series of research [Veldt et al., 2020, Benson et al., 2020] directly defines objective function using a submodular function instead of Lovász extension. While submodular models seem flexible, ours are more versatile since we do not assume submodularity. The submodular p-Laplacian is a special case of ours as long as the conditions in Def. 3.1 are satisfied. Additionally, our algorithm can address arbitrary p, while algorithms from [Hein et al., 2013] and Li and Milenkovic [2018] focused on the specific p ($p = 1$ or $p = 2$).

We remark that our framework can address existing 2-Laplacian from CLIQUE and STAR, and TV/SUB p-Laplacian. Moreover, our partitioning algorithm can work for arbitrary $p > 1$, while those existing algorithms focus on specific p or use a general-purpose optimization algorithm without theoretical analyses. We also note that our framework can define a new p-Laplacian, which is (but not limited to) normalized TV, shown in Section 3.2.5. However, we need to recognize that it is out of the scope of our work to incorporate CLIQUE E-N-VW p-Laplacian. Moreover, since our framework is based on the relationships of Prop. 3.4 and Prop. 3.5, our framework does not incorporate a tensor modeled Laplacian for uniform hypergraph, where all edges connect the same number of vertices [Cooper and Dutle, 2012, Hu and Qi, 2012, Qi, 2013, Hu and Qi, 2015, Chen et al., 2017, Ghoshdastidar and Dukkipati, 2017b, Chang et al., 2020]. We cannot incorporate CLIQUE E-N-VW p-Laplacian and tensors into our model because our model is based upon the energy formed as Eq. (3.8), while energies for those two are differently defined. We note that the difference in the aims between tensor modeled Laplacians and our framework is as follows. In contrast, the tensor modeled Laplacians are the tensor operation; our framework focuses on the contraction made by the energy Eq. (3.8).

Table 3.3: Summary of the dataset used in the experiment. All the dataset has two classes. The parameter δ is the average edge degree parameter $\delta := \sum_{e\in E} |e|/|E|$, and $\tau := \sum_{e\in E} |e|/|V||E|$ is the average ratio of the number of vertices connected by each edge to the total number of vertices, which we can recognize as "density" of a hypergraph.

	mushroom	chess	cancer	congress	news(1,2)	news(3,4)
$\lvert V\rvert$	8124	3196	699	435	8124	8188
$\lvert E\rvert$	112	73	90	48	100	100
$\sum_{e\in E} \lvert e\rvert$	170604	115056	6291	6960	31066	34382
δ	1523.25	1576.11	69.9	145	310.66	343.82
τ	0.18	0.49	0.10	0.33	0.038	0.042

Lastly, we comment on p-Laplacians in the continuous domain. The continuous p-Laplacian has a longer history than the discrete domain. The Dirichlet energy is defined similarly to Eq. (3.8), and the variation of the energy would give the Laplace equation [Courant and Hilbert, 1962]. The energy is minimized when the Laplace equation is satisfied. This framework extended to arbitrary p-norm, such as [Binding and Rynne, 2008], and was theoretically analyzed in many ways, such as nodal domain theorem and Cheeger inequality. We remark that in the continuous case, we can identify the second p-eigenpair similarly to Eq.(3.19), but no exact identification for the third or higher has been found yet [Lindqvist, 2008]. For a more comprehensive study, we refer to [Lindqvist, 2008] and [Struwe, 2000].

3.6 Experiments

Our experiments aim to evaluate our approximation algorithm (Alg. 3) as a function of p and the particular type of hypergraph Laplacians (STAR, CLIQUE, and TV/SUB).

Objective of the Experiments. The objective of the experiments is to see if our algorithm (Alg. 3) improves on the existing methods introduced in Sec. 3.5. Alg. 3 has two key "levers"; the choice of the parameter p and the choice of hypergraph Laplacian, i.e., the function c in the gradient (Def. 3.1). On the one hand, in the previous works discussed in Sec. 3.5, the algorithms for hypergraph p-Laplacians were designed for a particular p (e.g., $p = 1, 2$) or applied to all $p > 1$ without theoretical justifications. On the contrary, Alg. 3 for our abstract class of hypergraph p-Laplacians works for all $p > 1$ with theoretical justification. Therefore, we provide experiments for a wide range of hypergraph Laplacians for $p > 1$ in comparison to existing algorithms.

Datasets. We build a hypergraph using the method for categorical datasets introduced

Table 3.4: The experimental results for hypergraph partitioning for our methods and existing ones. We applied our algorithm 3 for $p > 1$ to five geometry of the hypergraph Laplacians (CLIQUE E-N, CLIQUE E-UN, STAR, TV V-UN, TV V-N). We compared these to the existing fixed p algorithms for the five hypergraph Laplacians; CLIQUE E-N for $p = 2$ by Saito et al. [2018], CLIQUE E-UN for $p = 2$ is by Rodriguez [2002], STAR $p = 2$ is by [Zhou et al., 2006], and TV V-UN and TV V-N for $p = 1$ is by [Hein et al., 2013]. Moreover, for CLIQUE E-N, we also compared with the algorithm for $p > 1$ (CLIQUE E-N-VW) by Saito et al. [2018]. Thus, we compare five instantiations of ours with six existing ones. We compare the performance by error. Performance with ours is marked with # in the left-most column. For free-parameter $p > 1$, we give the value of p giving the smallest error in the column p next to the dataset. The superscripted * means fixed-parameter. The See the main text for more discussion.

	mushroom	p	chess	p	cancer	p	congress	p	news(1,2)	p	news(3,4)	p
# CLIQUE E-N p	0.1118 ± 0.0033	1.1	0.4765 ± 0.0011	2.1	$\mathbf{0.0243} \pm 0.0000$	1.7	$\mathbf{0.1172} \pm 0.0010$	1.6	$\mathbf{0.2390} \pm 0.0021$	1.1	$\mathbf{0.2146} \pm 0.0000$	1.1
$p = 2$	0.3156 ± 0.0021	2.0^*	0.4781 ± 0.0008	2.0^*	0.0286 ± 0.0000	2.0^*	0.1241 ± 0.0000	2.0^*	0.2906 ± 0.0000	2.0^*	0.2173 ± 0.0000	2.0^*
CLIQUE E-N-VW p	0.2329 ± 0.0012	1.0	$\mathbf{0.2847} \pm 0.0007$	2.6	$\mathbf{0.0243} \pm 0.0000$	1.5	0.1195 ± 0.0010	2.5	0.2442 ± 0.0000	3.0	0.2167 ± 0.0000	1.1
# CLIQUE E-UN p	0.1211 ± 0.0010	1.1	0.4768 ± 0.0000	2.6	0.0358 ± 0.0000	2.2	0.1195 ± 0.0000	1.4	0.2493 ± 0.0023	1.7	0.3272 ± 0.0000	1.1
$p = 2$	0.4791 ± 0.0000	2.0^*	0.4772 ± 0.0000	2.0^*	0.1659 ± 0.0000	2.0^*	0.1379 ± 0.0000	2.0^*	0.2511 ± 0.0015	2.0^*	0.3447 ± 0.0020	2.0^*
# STAR p	0.1113 ± 0.0021	1.1	0.4759 ± 0.0000	1.3	0.0286 ± 0.0000	1.1	0.1195 ± 0.0010	2.3	0.2411 ± 0.0000	2.4	0.2162 ± 0.0030	1.7
$p = 2$	0.3156 ± 0.0000	2.0^*	0.4781 ± 0.0000	2.0^*	0.0300 ± 0.0000	2.0^*	0.1333 ± 0.0000	2.0^*	0.2477 ± 0.0011	2.0^*	0.2169 ± 0.0000	2.0^*
# TV V-UN p	$\mathbf{0.1083} \pm 0.0000$	1.1	0.4659 ± 0.0011	2.6	0.1960 ± 0.0007	1.1	0.1862 ± 0.0021	1.2	0.2701 ± 0.0025	1.1	0.3274 ± 0.0000	1.7
$p = 1$	0.1349 ± 0.0000	1.0^*	0.4778 ± 0.0014	1.0^*	0.3362 ± 0.0000	1.0^*	0.3034 ± 0.0010	1.0^*	0.3072 ± 0.0027	1.0^*	0.4313 ± 0.0019	1.0^*
# TV V-N p	$\mathbf{0.1083} \pm 0.0000$	1.1	0.4643 ± 0.0000	2.7	0.2489 ± 0.0013	1.6	0.1839 ± 0.0021	1.5	0.2672 ± 0.0033	1.1	0.3275 ± 0.0017	1.4
$p = 1$	0.1088 ± 0.0000	1.0^*	0.4997 ± 0.0000	1.0^*	0.4692 ± 0.0010	1.0^*	0.3034 ± 0.0009	1.0^*	0.2983 ± 0.0020	1.0^*	0.4544 ± 0.0012	1.0^*

Table 3.5: The Computational time for Experiments in seconds. We took from the best performing p. The randomness are involved due to the 10 times of postprocessing by k-means.

	cancer	chess	mushroom	news(1,2)	news(3,4)	congress
# CLIQUE E-N p	3.58± 0.00	12.32± 0.00	16.45± 0.01	39.12± 0.00	42.55± 0.00	1.89± 0.00
$p = 2$	1.71± 0.00	4.83± 0.00	5.88± 0.00	15.36± 0.00	17.44± 0.00	0.77± 0.00
CLIQUE E-N-VW p	4.53± 0.00	12.93± 0.00	17.25± 0.00	34.18± 0.00	43.97± 0.00	1.64± 0.00
# CLIQUE E-UN p	3.77± 0.00	12.51± 0.00	16.12± 0.00	38.12± 0.00	45.91± 0.00	1.99± 0.00
$p = 2$	1.72± 0.00	12.31± 0.01	5.88± 0.00	15.37± 0.00	17.21± 0.00	0.77± 0.00
# STAR p	3.89± 0.00	12.87± 0.00	16.85± 0.00	33.14± 0.00	44.75± 0.00	1.91± 0.00
$p = 2$	1.71± 0.01	12.32± 0.00	5.88± 0.00	15.37± 0.00	17.44± 0.00	0.79± 0.00
# TV V-UN p	4.32± 0.00	11.34± 0.00	18.55± 0.00	35.01± 0.00	42.11± 0.00	2.11± 0.00
$p = 1$	0.05± 0.00	0.13± 0.00	0.21± 0.00	0.97± 0.00	1.31± 0.00	0.01± 0.00
# TV V-N p	4.31± 0.01	11.89± 0.00	17.91± 0.00	36.33± 0.00	43.86± 0.00	2.44± 0.00
$p = 1$	0.06± 0.00	0.15± 0.00	0.19± 0.00	0.88± 0.00	1.45± 0.00	0.02± 0.00

Table 3.6: The detailed results of hypergraph partitioning using our hypergraph p-Laplacian for various p. The randomness are due to the postprocessing.

		best	$p = 1.1$	$p = 1.5$	$p = 2$	$p = 2.5$	$p = 3.0$
mushroom	CLIQUE E-N	0.1188 ± 0.0033 ($p = 1.1$)	0.1188 ± 0.0033	0.4719 ± 0.0021	0.4791 ± 0.0021	0.4796 ± 0.0031	0.4796 ± 0.0024
	CLIQUE E-UN	0.1211 ± 0.0010 ($p = 1.1$)	0.1211 ± 0.0010	0.1967 ± 0.0011	0.1460 ± 0.0000	0.4793 ± 0.0030	0.4788 ± 0.0021
	STAR	0.1113 ± 0.0021 ($p = 1.1$)	0.1113 ± 0.0021	0.1278 ± 0.0000	0.2189 ± 0.0000	0.4796 ± 0.0021	0.4791 ± 0.0038
	TV V-UN	0.1083 ± 0.0000 ($p = 1.1$)	0.1083 ± 0.0000	0.4796 ± 0.0000	0.4796 ± 0.0010	0.4796 ± 0.0023	0.4798 ± 0.0038
	TV V-N	0.1083 ± 0.0000 ($p = 1.1$)	0.1083 ± 0.0000	0.4798 ± 0.0000	0.4796 ± 0.0010	0.4446 ± 0.0012	0.4796 ± 0.0038
chess	CLIQUE E-N	0.4765 ± 0.0011 ($p = 2.1$)	0.4768 ± 0.0010	0.4981 ± 0.0000	0.4765 ± 0.0008	0.4778 ± 0.0000	0.4781 ± 0.0038
	CLIQUE E-UN	0.4768 ± 0.0000 ($p = 2.2$)	0.4950 ± 0.0000	0.4950 ± 0.0000	0.4772 ± 0.0000	0.4775 ± 0.000i	0.4775 ± 0.0038
	STAR	0.4759 ± 0.0000 ($p = 1.3$)	0.4865 ± 0.0000	0.4781 ± 0.0000	0.4781 ± 0.0000	0.4781 ± 0.0010	0.4778 ± 0.0034
	TV V-UN	0.4659 ± 0.0011 ($p = 2.6$)	0.4966 ± 0.0020	0.4740 ± 0.0000	0.4775 ± 0.0013	0.4778 ± 0.0011	0.4784 ± 0.0025
	TV V-N	0.4643 ± 0.0000 ($p = 2.7$)	0.4912 ± 0.0000	0.4734 ± 0.0000	0.4778 ± 0.0015	0.4775 ± 0.0000	0.4671 ± 0.0031
cancer	CLIQUE E-N	0.0243 ± 0.0000 ($p = 1.7$)	0.0272 ± 0.0000	0.0253 ± 0.0010	0.0286 ± 0.0000	0.1288 ± 0.0000	0.1531 ± 0.0000
	CLIQUE E-UN	0.0358 ± 0.0000 ($p = 2.2$)	0.0801 ± 0.0000	0.0930 ± 0.0000	0.2117 ± 0.0000	0.3462 ± 0.0000	0.3433 ± 0.0020
	STAR	0.0286 ± 0.0000 ($p = 1.1$)	0.0286 ± 0.0000	0.0286 ± 0.0000	0.0300 ± 0.0010	0.3433 ± 0.0033	0.3448 ± 0.0000
	TV V-UN	0.1960 ± 0.0007 ($p = 1.1$)	0.1960 ± 0.0007	0.2632 ± 0.0000	0.2546 ± 0.0013	0.3419 ± 0.0020	0.3433 ± 0.0000
	TV V-N	0.2489 ± 0.0013 ($p = 1.6$)	0.4612 ± 0.0021	0.3090 ± 0.0000	0.2661 ± 0.0021	0.3433 ± 0.0015	0.3462 ± 0.0000
congress	CLIQUE E-N	0.1172 ± 0.0010 ($p = 1.5$)	0.1264 ± 0.0010	0.1172 ± 0.0000	0.1241 ± 0.0000	0.1448 ± 0.0020	0.3287 ± 0.0030
	CLIQUE E-UN	0.1195 ± 0.0000 ($p = 1.5$)	0.1218 ± 0.0000	0.1287 ± 0.0000	0.1379 ± 0.0000	0.1241 ± 0.0000	0.3885 ± 0.0030
	STAR	0.1195 ± 0.0000 ($p = 2.3$)	0.1287 ± 0.0000	0.1310 ± 0.0000	0.1333 ± 0.0000	0.1241 ± 0.0000	0.1333 ± 0.0000
	TV V-UN	0.1862 ± 0.0021 ($p = 1.2$)	0.1839 ± 0.0000	0.2598 ± 0.0000	0.2828 ± 0.0000	0.3218 ± 0.0000	0.3241 ± 0.0000
	TV V-N	0.1839 ± 0.0021($p = 1.5$)	0.1908 ± 0.0022	0.1839 ± 0.0021	0.2736 ± 0.0021	0.2966 ± 0.0023	0.2529 ± 0.0000
news(1,2)	CLIQUE E-N	0.2390 ± 0.0021 ($p = 1.1$)	0.2390 ± 0.0021	0.2587 ± 0.0010	0.2906 ± 0.0000	0.2400 ± 0.0030	0.2490 ± 0.0000
	CLIQUE E-UN	0.2493 ± 0.0023 ($p = 1.7$)	0.2507 ± 0.0000	0.2516 ± 0.0023	0.2511 ± 0.0015	0.4330 ± 0.0050	0.4330 ± 0.0040
	STAR	0.2411 ± 0.0000 ($p = 2.4$)	0.2477 ± 0.0000	0.4324 ± 0.0000	0.2477 ± 0.0011	0.4323 ± 0.0032	0.4330 ± 0.0040
	TV V-UN	0.2701 ± 0.0025 ($p = 1.1$)	0.2701 ± 0.0025	0.4356 ± 0.0020	0.3015 ± 0.0012	0.3016 ± 0.0033	0.3017 ± 0.0021
	TV V-N	0.2672 ± 0.0033 ($p = 1.5$)	0.2672 ± 0.0011	0.4334 ± 0.0033	0.3029 ± 0.0000	0.3015 ± 0.0048	0.3028 ± 0.0032
news(3,4)	CLIQUE E-N	0.2146 ± 0.0000 ($p = 1.1$)	0.2146 ± 0.0000	0.2167 ± 0.0000	0.2173 ± 0.0000	0.2186 ± 0.0030	0.2174 ± 0.0041
	CLIQUE E-UN	0.3272 ± 0.0000 ($p = 1.1$)	0.3272 ± 0.0000	0.3507 ± 0.0000	0.3447 ± 0.0020	0.3481 ± 0.0038	0.3474 ± 0.0020
	STAR	0.2162 ± 0.0030 ($p = 1.7$)	0.2183 ± 0.0010	0.2356 ± 0.0030	0.2169 ± 0.0000	0.2170 ± 0.0000	0.2170 ± 0.0030
	TV V-UN	0.2701 ± 0.0000 ($p = 1.4$)	0.4378 ± 0.0000	0.3274 ± 0.0000	0.3274 ± 0.0013	0.4305 ± 0.0021	0.4340 ± 0.0050
	TV V-N	0.3275 ± 0.0017 ($p = 1.4$)	0.4544 ± 0.0000	0.3274 ± 0.0017	0.3274 ± 0.0014	0.4340 ± 0.0023	0.4305 ± 0.0045

in [Zhou et al., 2006], which is the community standard benchmark for the hypergraph spectral clustering community. Each instance in the dataset consists of $|E|$ categories. The vertices of the hypergraph are the instances. The edges are defined by the attribute values. Each attribute value within a given category defines an edge where each vertex in the edge corresponds to those instances that share the same attribute value. All edges are given weight one. Our experiment is performed on the datasets mushroom, cancer, chess, and congress from the UCI repository [Dua and Graff, 2019], and two datasets created from 20newsgroups [*1] (for short, "news") with two classes (1,2) and (3,4). All of these datasets were used in the previous studies [Zhou et al., 2006, Hein et al., 2013, Saito et al., 2018]. We summarize the datasets in Table 3.3. We provide two characterizations of the dataset. that is

$$\delta := \sum_{e \in E} \frac{|e|}{|E|} \tag{3.24}$$

$$\tau := \sum_{e \in E} \frac{|e|}{|V||E|} \tag{3.25}$$

The value δ is the average edge degree. Furthermore, τ is the average ratio of the number of vertices connected by each edge to the total number of vertices, which we can recognize as the "density" of a hypergraph.

Experimental Setup. From the dataset, we build hypergraphs, and in Table 3.4 we compare 11 instantiations of hypergraph p-Laplacians as discussed below. We apply Alg. 3 to five existing hypergraph Laplacians (CLIQUE E-N, CLIQUE E-UN, STAR, TV V-UN, and TV V-N) for $p > 1$. We compare these to the existing fixed p algorithms for a particular type of Laplacians. Moreover, since CLIQUE E-N has a partitioning algorithm using a particular hypergraph p-Laplacian (CLIQUE E-N-VW by [Saito et al., 2018], see Sec. 3.5 for the definition), we also compare to this. Hence, we compare five instantiations of ours with six previous algorithms;

- Alg. 3 for all $p > 1$ is applied to the five geometries:

  1. CLIQUE E-N: $p > 1$
  2. CLIQUE E-UN: $p > 1$

[1]We used the tiny version of the original 20newsgroups available at `https://cs.nyu.edu/~roweis/data/20news_w100.mat`.

3. STAR: $p > 1$
4. TV V-UN: $p > 1$
5. TV V-N: $p > 1$

- Comparison as six existing algorithms:

1. CLIQUE E-N: $p = 2$ [Saito et al., 2018]
2. CLIQUE E-N-VW: $p > 1$ [Saito et al., 2018]
3. CLIQUE E-UN: $p = 2$ [Rodriguez, 2002]
4. STAR: $p = 2$ [Zhou et al., 2006]
5. TV V-UN: $p = 1$ [Hein et al., 2013]
6. TV V-N: $p = 1$ [Hein et al., 2013]

Note that there is a variety of submodular functions for SUB that can be considered, but we made TV by [Hein et al., 2013] as a representative of the SUB group, since TV is the simplest form of SUB. For CLIQUE E-N-VW ($p \in [1, 3]$), we conducted experiments using the same setting as [Saito et al., 2018] as the setting matches to ours. For our methods we used $p \in \{1.1, 1.2, \ldots, 3.0\}$; we limited ourselves to $p \leq 3$ since the Cheeger Inequality (Thm. 3.14), is progressively looser for larger p values. For the free-parameter experiments, we set the starting condition of our algorithm to the solution of the corresponding fixed-parameter Laplacian. We used the step size $\alpha = 0.01\|\mathbf{x}\|_1^1/\|\mathbf{x}'\|_1^1$ as done in [Luo et al., 2010]. For all methods, a second eigenvector was computed, and we used the k-means objective to determine the "split point" on the eigenvector (as was also done in [Zhou et al., 2006, Saito et al., 2018]). We evaluated the performance of our algorithms via their error rate, i.e., (# of errors)/(# of data), as used in [Zhou et al., 2006, Hein et al., 2013, Saito et al., 2018]. We ran our experiment on mac mini with Intel i7 and 32GiB RAM. The implementation is available at `https://github.com/ShotaSAITO/generalized-hypergraph-laplacian`.

Overall Results. The results are summarized in Table 3.4. First, looking into our algorithm (Alg. 3) vs. fixed-parameter algorithms (existing methods, see the performances associated with * in Table 3.3) for five geometries, we see that our methods consistently demonstrate improved performance from existing fixed-parameter methods. We also remark that among for CLIQUE E-N, ours consistently outperforms CLIQUE E-N-VW, except chess.

Further Discussion. A natural question to ask for our algorithm is "which hypergraph Laplacian and which p is suitable?". To observe this, we provide a result table on the detailed behavior of p in Table 3.6. A further look into our abstract class of p-Laplacians can answer this question; the experimental result reveals how the choice of p and type of the hypergraph Laplacian are connected to the underlying parameters δ (average edge degree) and τ (density) of the datasets. Although the experiments are preliminary, there seem to be consistent trends that provide guidance on a range of p and the type of Laplacian to consider. Further, the experimental guidance is supported by the theory given earlier in this chapter.

Our observation is that the density parameter (τ) is related to the range of p while the average edge degree parameter (δ) is connected to the hypergraph Laplacian. The density parameter (τ) indicates the natural range for p. The dataset chess is significantly denser (large τ) than the other datasets. The table indicates that while large p tends to work better for the chess dataset, the tendency is that small p improves on large p for the non-chess datasets. To understand this, we consider the trade-off between the Cheeger inequality (see Thm. (3.14)) and the p-Dirichlet sum. The Cheeger inequality is tighter for smaller p; hence, the relaxed objective becomes closer to the discrete objective. On the other hand, if we examine the p-Dirichlet sum (see Eq. (3.8)), one may observe that it is a p-norm to p-th power of the hypergraph-gradient. The dimensionality of hypergraph-gradient scales with the graph density (τ). Hence in the dense case, a relatively larger p is needed to induce the same magnitude of change in the p-Dirichlet sum, which is connected to the second p-eigenvector via Rayleigh quotient (see Eq. (3.10)). The analogous phenomena connecting the choice of p to density have been observed in a standard graph such as online graph transduction [Herbster and Lever, 2009]. Turning to the average edge degree parameter (δ), we observe the following indications that suggest how to choose the Laplacian as a function of δ. There we see on the large δ dataset (chess and mushroom) that all TV methods outperform STAR and CLIQUE methods of our p-Laplacian whereas for the other smaller δ datasets all STAR and clique methods outperform all TV methods. We have provided some guidance on the choice of Laplacian and the range of p based on the density τ and average edge degree δ of the graph.

We have two different energy forms for clique, vertex-wise (proposed by [Saito et al., 2018]) and edge-wise (this work). The vertex-wise energy outperforms our edge-wise energy Laplacian for chess, but for the mushroom, cancer, and congress dataset, ours outperforms

vertex-wise Laplacian. This difference may arise because the chess dataset is better suited to a vertex-based energy model, whereas the structure of the mushroom dataset might favor an edge-based approach. Vertex-wise energy captures information more sensitively in dense graphs by amplifying the sum of the neighborhood information, while edge-wise energy amplifies each edge individually, which can lead to redundancy when many edges are similar.

We further observe a different behavior than the semi-supervised learning in [Alamgir and Luxburg, 2011, Slepcev and Thorpe, 2019] using the same energy Eq. (3.8) in the standard graph setting. These works deal with the case of semi-supervised learning using p-Laplacians of a graph with an asymptotically large number of vertices. In this case, the problem does not degenerate into the trivial one when p is large, while the problem does so when p is small. However, from these experimental results, we observed a different behavior; small p also works when τ is small, as we discussed. This might be because there is a structural difference in using the p-Laplacian between semi-supervised and unsupervised learning.

Computational Time. The computational times are summarized in Table 3.5. For our method, the computational times across different instances do not vary significantly. However, compared to existing work, our method is generally slower. In the case of $p = 2$, both the existing clique and star methods have a time complexity of $O(n^3)$, while ours takes $O(\sum_{e\in E} |e|^3)$. Although both are roughly in the same computational class for our datasets, n^3 is expected to be larger than $\sum_{e\in E} |e|^3$. However, the experiments show that our method is slower because it involves gradient descent, requiring multiple iterations to converge and increasing the overall computation time. For $p = 1$, the TV method by Hein et al. [2013] is faster than our approach. This is due to the ability of TV method to exploit the sparse structure of the graph; this results in a time complexity of $O\left(\sum_{e\in E} |e| \log(|e|)\right)$, which is faster than $O(\sum_{e\in E} |e|^3)$ for the datasets used in our experiments. It is worth noting that the TV method is optimized specifically for the $p = 1$ case. Developing a similarly optimized algorithm for our unified hypergraph p-Laplacians for the general $p > 1$ remains an area of future work.

3.7 Summary

This chapter has considered hypergraph spectral clustering. We have proposed a general framework for hypergraph p-Laplacian and provided theoretical results for our p-Laplacian. We also have proposed a convergent hypergraph partitioning algorithm with respect to our abstract class of p-Laplacian exploiting theoretical results. Our experiment has shown that our

algorithm outperforms the existing spectral clustering algorithms for hypergraph Laplacians. Also, we have shown practical guidance on the choice of p-Laplacian. There are several future directions of Chapter 3. First, as discussed in the experimental section, it would be nice if we have a better general algorithm that can exploit the sparsity when the graph is sparse, like $p = 1$ case in Hein et al. [2013]. Furthermore, while we conduct our experiment on a real dataset, it would be interesting to conceive an illustrative toy dataset where some hypergraph Laplacian works better than others or where some p works while $p = 2$ does not.

Appendices for Chapter 3

In the following sections, we provide the omitted proofs, additional discussions, and additional experimental result for Chapter 3.

3.A Proof of Proposition 3.4

For the convenience of the other proofs, we start our discussion from Prop. 3.4 and Prop. 3.5. Prop. 3.4 can be shown by

$$\begin{aligned}\langle \mathbf{x}, \Delta_{c,p}\mathbf{x}\rangle_{\mathcal{H}(V)} &= \langle \mathbf{x}, -\mathrm{div}_{c,p}\|\nabla_{c,p}\mathbf{x}\|^{p-2}\nabla_{c,p}\mathbf{x}\rangle_{\mathcal{H}(V)} \\ &= \langle \nabla_{c,p}\mathbf{x}, \|\nabla_{c,p}\mathbf{x}\|^{p-2}\nabla_{c,p}\mathbf{x}\rangle_{\mathcal{H}(E)} \\ &= \sum_{e\in E}\|\nabla_{c,p}\mathbf{x}\|^{p-2}\frac{(\nabla_{c,p}\mathbf{x})^2(e)}{|e|!} \\ &= \|\nabla_{c,p}\mathbf{x}(e)\|_p^p = S_{G,c,p}(\mathbf{x}). \end{aligned} \tag{3.26}$$

3.B Proof of Proposition 3.5

Prop. 3.5 can be shown by the following. By differentiating $S_{G,c,p}(\mathbf{x})$ by $\mathbf{x}$ at the vertex i, we obtain

$$\begin{aligned}\frac{\partial}{\partial \mathbf{x}}S_{G,c,p}(\mathbf{x})\bigg|_i &= \frac{\partial}{\partial \mathbf{x}}\sum_{e\in E}\frac{1}{|e|!}|(\nabla_{c,p}\mathbf{x})(e)|^p\bigg|_i \\ &= \sum_{e\colon i\in e} p\frac{1}{|e|!}|(\nabla_{c,p}\mathbf{x})(e)|^{p-1}\,\frac{\partial}{\partial \mathbf{x}}|\nabla_{c,p}\mathbf{x}(e)|\bigg|_i \\ &= p\sum_{e\colon i\in e}\frac{1}{|e|!}|(\nabla_{c,p}\mathbf{x})(e)|^{p-1}\mathrm{sgn}((\nabla_{c,p}\mathbf{x})(e))\frac{w^{1/p}(e)}{\mu_i^{1/p}} \\ &\times\left(\sum_{i\in e;\{i,j\}\subseteq e}c^{1/p}(j,i,e,\mathbf{x}) - \sum_{i\in e;\{i,j\}\subseteq e}c^{1/p}(i,j,e,\mathbf{x})\right)\end{aligned} \tag{3.27}$$

By using Eq. (3.27), we consider the following equation.

$$\begin{aligned}&\left\langle \mathbf{x}, \frac{1}{p}\frac{\partial}{\partial \mathbf{x}}S_{G,c,p}(\mathbf{x})\right\rangle_{\mathcal{H}(V)} \\ &= \sum_{i\in V}x_i\sum_{e:\in e}\frac{1}{|e|!}|(\nabla_{c,p}\mathbf{x})(e)|^{p-1}\mathrm{sgn}(\nabla_{c,p}\psi(e))\frac{w^{1/p}(e)}{\mu_i^{1/p}}\end{aligned}$$

$$\times \left(\sum_{i \in e; \{i,j\} \subseteq e} c^{1/p}(j, i, e, \mathbf{x}) - \sum_{u \in e; \{i,j\} \subseteq e} c^{1/p}(i, j, e, \mathbf{x}) \right) \tag{3.28}$$

As the summation in Eq. (3.28) runs over all vertices $v \in V$, we can reconstruct all pairs of vertices in each edge. Therefore,

$$\begin{aligned} &\sum_{i \in V} x_i \sum_{e: i \in e} \operatorname{sgn}(\nabla_{c,p}\mathbf{x}(e)) \frac{w^{1/p}(e)}{\mu_i^{1/p}} \left(\sum_{j \in e; \{i,j\} \subseteq e} c^{1/p}(j, i, e, \mathbf{x}) - \sum_{u \in e; \{j,i\} \subseteq e} c^{1/p}(i, j, e, \mathbf{x}) \right) \\ &= \sum_{e \in E_d} |(\nabla_{c,p}\mathbf{x})(e)| \end{aligned} \tag{3.29}$$

From this and Eq. (3.28), we obtain

$$\begin{aligned} \left\langle \mathbf{x}, \frac{1}{p} \frac{\partial}{\partial \mathbf{x}} S_{G,c,p}(\mathbf{x}) \right\rangle_{\mathcal{H}(V)} &= \sum_e \frac{1}{|e|!} |(\nabla_{c,p}\mathbf{x})(e)|^p \\ &= \|\nabla_{c,p}\psi\|_p^p \\ &= S_{G,c,p}(\psi) \end{aligned} \tag{3.30}$$

We then can show that

$$\frac{1}{p} \frac{\partial}{\partial \mathbf{x}} S_{G,c,p}(\mathbf{x}) = \Delta_{c,p}\mathbf{x} \tag{3.31}$$

3.C Proof of Proposition 3.7

To start, we show that the following basic properties of hypergraph-gradient easily follow from the definition.

Proposition 3.19. *The hypergraph-gradient has the following properties.*

$$(\nabla c_{c,p} M^{1/p})\mathbf{1}(e) = \mathbf{0}, \forall e \in E, \tag{3.32}$$

$$(\nabla_{c,p}\mathbf{0})(e) = \mathbf{0}, \forall e \in E, \tag{3.33}$$

$$\partial^2 \nabla_{c,p}\mathbf{x} / \partial^2 \mathbf{x}\big|_v = 0, \forall e \in E \tag{3.34}$$

Also, hypergraph-gradient is not 0 except Eq. (3.32) *and Eq.* (3.33)*.*

These properties directly follow from the definition of the hypergraph-gradient.

By differentiating Eq. (3.10) by ψ, we can obtain the condition for critical points of Eq. (3.10) as follows;

$$\Delta_{c,p}\mathbf{x} - \frac{S_{G,c,p}(\mathbf{x})}{\|\mathbf{x}\|_p^p}\xi_p(\mathbf{x}) = 0 \tag{3.35}$$

By Def. 3.6, we can immediately show that ψ is an eigenvector of $\Delta_{c,p}$. Moreover, the eigenvalue λ can be obtained by $S_{G,c,p}(\mathbf{x})/\|\mathbf{x}\|_p^p$. The last statement can be shown immediately by the definition.

By the definition of Rayleigh quotient, we immediately have the following property.

Corollary 3.20. *We have* $R_{G,c,p}(\alpha\mathbf{x}) = R_{G,c,p}(\mathbf{x})$ *for* $\alpha \in \mathbb{R}, \alpha \neq 0$.

For the first p-eigenvector, we compute p-Laplacian by differentiating by ψ, that is

$$p\Delta_{c,p}\mathbf{x} = \frac{\partial}{\partial \mathbf{x}} S_{G,c,p}(\mathbf{x}). \tag{3.36}$$

Then, we obtain

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{x}} p\Delta_{c,p}\mathbf{x} &= \frac{\partial}{\partial \mathbf{x}} S_{G,c,p}(\mathbf{x}) \\
&= \frac{\partial}{\partial \mathbf{x}} \|(\nabla_{c,p}\mathbf{x})(e)\|_p^p \\
&= \frac{\partial}{\partial \mathbf{x}} \Big(\sum_e \frac{|\nabla_{c,p}(\mathbf{x})(e)|^p}{|e|!} \Big) \\
&= p \sum_e \frac{|(\nabla_{c,p}\mathbf{x}(e)|^{p-1}}{|e|!} \times \frac{1}{|e|!} \frac{\partial}{\partial \mathbf{x}} (\nabla_{c,p}\mathbf{x})(e)
\end{aligned} \tag{3.37}$$

From Eq. (3.34), the derivative of hypergraph-gradient is independent of $\mathbf{x}$. Therefore, from Eq. (3.37), the p-Laplacian $\Delta_{c,p}\mathbf{x}$ equals to 0 if $(\nabla_{c,p}\mathbf{x})(e) = 0, \forall e \in E$. This means that $\Delta_{c,p}\mathbf{0} = 0$. Also, $\Delta_{c,p} M^{1/p}\mathbf{1} = 0$.

As the p-eigenvalue is equal or greater than 0 from Prop. 3.7, the first p-eigenvector is $M^{1/p}\mathbf{1}$, associated with the first p-eigenvalue 0.

The following corollary follows.

Corollary 3.21.

$$\frac{\partial}{\partial \mathbf{x}} \Delta_{c,p} \mathbf{x}|_{\mathbf{x}=M^{1/p}\mathbf{1}} = 0. \tag{3.38}$$

Proof. Similarly to the proof of p-eigenvector, we compute

$$\begin{aligned} p\frac{\partial}{\partial \mathbf{x}} \Delta_{c,p} \mathbf{x} &= \frac{\partial^2}{\partial^2 \mathbf{x}} S_{G,c,p}(\mathbf{x}) \\ &= \frac{\partial}{\partial \mathbf{x}} p \sum_e \frac{|\nabla_{c,p}\mathbf{x}(e)|^{p-1}}{|e|!} \times \frac{1}{|e|!}\frac{\partial}{\partial \mathbf{x}} \nabla_{c,p}\mathbf{x}(e) \\ &= p(p-1)\sum_e \frac{|\nabla_{c,p}\mathbf{x}(e)|^{p-2}}{|e|!} \times \left(\frac{1}{|e|!}\frac{\partial}{\partial \mathbf{x}} \nabla_{c,p}\mathbf{x}(e)\right)^2 \\ &\quad + p\sum_e \frac{|\nabla_{c,p}\mathbf{x}(e)|^{p-1}}{|e|!} \times \frac{1}{|e|!}\frac{\partial^2}{\partial^2 \mathbf{x}} (\nabla_{c,p}\mathbf{x})(e) \end{aligned} \tag{3.39}$$

As the second derivative of hypergraph-gradient is 0 from Eq.(3.34), $\Delta_{c,p}\mathbf{x} = 0$ if $(\nabla_{c,p}\mathbf{x})(e) = 0, \forall e \in E$. □

3.D Proof of Proposition 3.8

As we observe $R_{G,c,p}(\mathbf{x}) \geq 0$ from the definition, we can show that all the p-eigenvalues are non-negative. We denote by $\mathbf{1}_{G_i}$ a vector whose size of vector is the number of vertices of G and fill 1 to the elements corresponds to G_i and else 0. By using this notation, we show that $\Delta_{c,p}(M^{-1/p}\mathbf{1}_{G_i}) = 0$ for all $i = 1, \ldots, k$, which shows that those vectors are p-eigenvector and corresponding p-eigenvalues are 0. From the definition of p-Laplacian, those are the only p-eigenvectors whose p-eigenvalues are 0. The above shows that the multiplicity of p-eigenvalues of 0 equals to the number of independent components.

3.E Proof of Proposition 3.10

We start the proof by introducing a classical notion of *locally Lipschitz.*

Definition 3.22. *A function $g : \mathcal{S}_p \to \mathbb{R}$ is locally Lipschitz when for each $\mathbf{x} \in \mathcal{S}_p$, there exists a neighborhood $\mathcal{N}_\mathbf{x}$ of $\mathbf{x}$ and a constant C depending on $\mathcal{N}_\mathbf{x}$ such that $|g(\mathbf{x}') - g(\mathbf{x})| \leq C\|\mathbf{x}' - \mathbf{x}\|_2$ for all $\mathbf{x}' \in \mathcal{S}_p \cap \mathcal{N}_\mathbf{x}$.*

Here we obtain the following observation.

Lemma 3.23. *$R_{G,c,p}(\mathbf{x})|_{\mathcal{S}_p}$ is locally Lipschitz.*

Proof.

$$\begin{aligned}
&\left|\frac{S_{G,c,p}(\mathbf{x}')}{\|\mathbf{x}'\|_p^p} - \frac{S_{G,c,p}(\mathbf{x})}{\|\mathbf{x}\|_p^p}\right| = \left|\frac{S_{G,c,p}(\mathbf{x}')\|\mathbf{x}\|_p^p - S_{G,c,p}(\mathbf{x})\|\mathbf{x}'\|_p^p}{\|\mathbf{x}'\|_p^p\|\mathbf{x}\|_p^p}\right| \\
&\leq \frac{\sup_{\mathbf{x}'\in\mathcal{N}_x\cap\mathcal{S}_p} S_{G,p}(\mathbf{x}')}{\inf_{\mathbf{x}'\in\mathcal{N}_\mathbf{x}\cap\mathcal{S}_p}\|\mathbf{x}'\|_p^{2p}}|\|\mathbf{x}'\|_p^p - \|\mathbf{x}\|_p^p| \\
&\leq \frac{\sup_{\mathbf{x}'\in\mathcal{N}_b fx\cap\mathcal{S}_p} S_{G,c,p}(\mathbf{x}')}{\inf_{\mathbf{x}'\in\mathcal{N}_\mathbf{x}\cap\mathcal{S}_p}\|\mathbf{x}'\|_p^{2p}}(\sup_{\mathbf{x}'\in\mathcal{N}_\mathbf{x}\cap\mathcal{S}_p} p\|\mathbf{x}'\|_p^p)|\|\mathbf{x}'\|_p - \|\mathbf{x}\|_p| \\
&\leq \frac{\sup_{\mathbf{x}'\in\mathcal{N}_\mathbf{x}\cap\mathcal{S}_p} S_{G,c,p}(\mathbf{x}')}{\inf_{\mathbf{x}'\in\mathcal{N}_\mathbf{x}\cap\mathcal{S}_p}\|\mathbf{x}'\|_p^{2p}}(\sup_{\mathbf{x}'\in\mathcal{N}_\mathbf{x}\cap\mathcal{S}_p} p\|\mathbf{x}'\|_p^p)\|\mathbf{x}'-\mathbf{x}\|_p \qquad (3.40)
\end{aligned}$$

From Eq.(3.40), if we choose $\mathcal{N}_\mathbf{x}$ as the space where $\|\mathbf{x}'-\mathbf{x}\|_p < \|\mathbf{x}'-\mathbf{x}\|_2$, e.g., $|\mathbf{x}'-\mathbf{x}| < 1$, then we can conclude $R_{G,p}(\mathbf{x})|_{\mathcal{S}_p}$ is locally Lipschitz. □

Next, we introduce the classical result of Lunsternik-Schinirelman theorem.

Theorem 3.24 (Struwe [2000])**.** *Suppose a function $g : \mathcal{S}_p \to \mathbb{R}$ is a locally Lipschitz, then*

$$\lambda_k = \min_{B\subset\mathcal{F}_k(\mathcal{S}_p^n)} \max_{\mathbf{x}\in B} g(\mathbf{x}) \qquad (3.41)$$

yields a sequence of critical values of g.

By applying Thm. 3.24 to $R_{G,p}$, we can show that the sequence in Eq. (3.10) yields a critical values of $R_{G,c,p}$, which are p-eigenvalues of p-Laplacian.

3.F Proof of Proposition 3.11

This section provides the proof of the Prop. 3.11. We divide the proof into two parts; one is that weighting function corresponds to hypergraph Laplacians, and the other is the weighting function satisfies the conditions.

3.F.1 Proof of the Hypergraph p-Laplacians

We discuss how Table. 3.2 connects to the existing Laplacians, by splitting the discussion by clique, star, and total variation Laplacians.

We note that all the functions in Table 3.2 satisfies the conditions of Def. 3.1, which is discussed in Section 3.F.2.

3.F.1.1 Clique Laplacians

The hypergraph-gradient for edge-normalized Laplacian is given as

$$(\nabla_{c,p}\mathbf{x})(e) = \frac{w^{1/p}(e)}{(|e|-1)^{1/p}} \sum_{\ell=1}^{|e|} \left(\frac{x_{i_\ell}}{\mu_{i_\ell}^{1/p}} - \frac{x_{i_1}}{\mu^{1/p}(x_{i_1})} \right), \tag{3.42}$$

and edge-unnormalized Laplacian can be written as

$$(\nabla_{c,p}\mathbf{x})(e) = w^{1/p}(e) \sum_{\ell=1}^{|e|} \left(\frac{x_{i_\ell}}{\mu_{i_\ell}^{1/p}} - \frac{x_{i_1}}{\mu_{i_1}^{1/p}} \right). \tag{3.43}$$

To show that the function c for clique can induce existing clique Laplacian, we only consider when $p = 2$, and $\mu_i = d_i$ for all $i \in V$. The following proposition directly shows that Saito's 2-Laplacian is our 2-Laplacian for clique setting.

Proposition 3.25 (Saito et al. [2018])**.** *Let e be $e = \{i_1, \cdot, i_{|e|}\}$. Then if we choose hypergraph-gradient operator $\nabla_{c,p} \colon \mathcal{H}(V) \to \mathcal{H}(E_d)$ for [Saito et al., 2018] as*

$$(\nabla_{c,p}\mathbf{x})(e) \coloneqq \frac{\sqrt{w(e)}}{\sqrt{|e|-1}} \sum_{\ell=1}^{|e|} \left(\frac{x_{i_\ell}}{\sqrt{d_{i_\ell}}} - \frac{x_{i_1}}{\sqrt{d_{i_1}}} \right). \tag{3.44}$$

The induced 2-Laplacian correspond to 2-Laplacians proposed by [Saito et al., 2018]. If we choose the same hypergraph-gradient but omitted denominator $\sqrt{|e|-1}$, then the induced 2-Laplacian corresponds to Rodriguez's Laplacian.

This also shows that Rodriguez 2-Laplacian is our edge-unnormalized clique 2-Laplacian. The 2-Laplacians L are given as

$$L = I - D^{-1/2} A D^{-1/2}, \tag{3.45}$$

where for an edge-normalized setting A is a matrix whose element is $a_{ij} = \sum_{i,j \in e} w(e)/(|e|-1)$ and $D = D_v$ and for an edge-unnormalized setting $a_{ij} = \sum_{i,j \in e} w(e)$ and D is a diagonal matrix whose element $d_{ii} = \sum_{i \in e} w(e)$.

In [Saito et al., 2018], they used the different energy setup for p-Laplacian, as discussed in Section 3.5. However, when $p = 2$, Saito's 2-Laplacian matches our clique normalized

2-Laplacian. Actually, in the case of $p = 2$, we obtain

$$\begin{aligned} S^S_{G,2}(\mathbf{x}) &= \sum_{v\in V}\sum_{e\in E: e[1]=v} \frac{1}{|e|!}|(\nabla_{c,p}\mathbf{x})(e)|^2 \\ &= \sum_{e\in E} \frac{1}{|e|!}|(\nabla_{c,p}\mathbf{x})(e)|^2 \\ &= S_{G,c,2}(\mathbf{x}) \end{aligned} \tag{3.46}$$

Therefore, given [Saito et al., 2018] has the same structure of different geometry, we can directly apply their proof to our setting in the case of $p = 2$.

3.F.1.2 Star Laplacian

The given hypergraph-gradient for star Laplacian can be written as

$$(\nabla_{c,p}\mathbf{x})(e) = \frac{w^{1/p}(e)}{(|e|)^{1/p}} \sum_{\ell=1}^{|e|} \left(\frac{x_{i_\ell}}{\mu_{i_\ell}^{1/p}} - \frac{x_{i_1}}{\mu_{i_1}^{1/p}} \right). \tag{3.47}$$

Here, we show that this hypergraph 2-Laplacian also can be seen from the same framework. Similarly to the clique Laplacians, the following proposition follows.

Proposition 3.26. *Let e be $\{i_1, \ldots, i_{|e|}\} = e$. Then if we choose hypergraph-gradient operator $\nabla_{c,p}\colon \mathcal{H}(V) \to \mathcal{H}(E_d)$ for as*

$$(\nabla_{c,p}\mathbf{x})(e) \coloneqq \frac{\sqrt{w(e)}}{\sqrt{|e|}} \sum_{\ell=1}^{|e|} \left(\frac{x_{i_\ell}}{\sqrt{d_{i_\ell}}} - \frac{x_{i_1}}{\sqrt{d_{i_1}}} \right), \tag{3.48}$$

this induces star expansion 2-Laplacian.

We can compute 2-Laplacian in the same manner as [Saito et al., 2018], with a slight change of denominator of hypergraph-gradient from $\sqrt{|e|-1}$ to $\sqrt{|e|}$. The 2-Laplacian induced from the hypergraph-gradient Eq. (3.48) can be computed as

$$L = I - D_v^{-1/2} A_s D_v^{-1/2}, \tag{3.49}$$

where A_s is a matrix whose element $(A_s)_{ij} = \sum_{i,j\in e} w(e)/|e|$. We can show that Eq. (3.49) satisfies the condition of Laplacian, in the same manner as the proof for Prop. 9 in [Saito

et al., 2018].

3.F.1.3 Total Variation and Submodular Laplacian

The hypergraph-gradient for total variation is written as

$$(\nabla_{c,p}\mathbf{x})(e) = w^{1/p}(e) \max_{i,j\in e} \left(\frac{x_j}{\mu_j^{1/p}} - \frac{x_i}{\mu_i^{1/p}} \right). \tag{3.50}$$

We show that the total variation method in [Hein et al., 2013] can be seen as a special case of our framework.

Proposition 3.27. *Let $\mu_i = 1$ for all $i \in V$. The Total Variation Regularizer defined as*

$$S_{G,c,p}(\mathbf{x}) = w(e) \left(\max_{i,j\in e} \left| \frac{x_i}{\mu_i^{1/p}} - \frac{x_k}{\mu_i^{1/p}} \right| \right)^p \tag{3.51}$$

is p-Dirichlet Sum, if we choose hypergraph-gradient as Eq. (3.50)*.*

This is obvious from the definition of p-Dirichlet energy by Eq. (3.8), which is called *regularizer* in [Hein et al., 2013].

The hypergraph-gradient for SUB is for

$$\nabla_{c,p}\psi(e) = (w(e) \max_{S\subset e}(F(S)))^{1/p} \sum_{i=1}^{|e|} F(S_i) \left(\frac{\psi(u_{i+1})}{\mu^{1/p}(u_{i+1})} - \frac{\psi(u_i)}{\mu^{1/p}(u_i)} \right). \tag{3.52}$$

By definition, the energy can be written as

$$S_{G,c,p}(\mathbf{x}) = \sum_{e\in E} w(e) \max_{S\subset e}(F(S)) \left(\sum_{\ell=1}^{|e|} F(S_\ell) \left(\frac{x_{i_{\ell+1}}}{\mu_{i_{\ell+1}}^{1/p}} - \frac{x_{i_\ell}}{\mu_{i_\ell}^{1/p}} \right) \right)^p, \tag{3.53}$$

which is Eq. (2.101).

3.F.2 Proof of Conditions

In this section, we discuss the conditions of the operator ∇ and the function $c(u, v, e, \psi)$ by drawing examples. We use the examples listed in Table 3.2, which is mainly discussed in Section 3.2.5.

The first condition

$$\nabla_{c,p}(\alpha\mathbf{x}) = \alpha\nabla_{c,p}\mathbf{x} \quad \text{or} \quad \nabla_{c,p}(\alpha\mathbf{x}) = |\alpha|\nabla_{c,p}\mathbf{x} \tag{3.54}$$

forces the operator $\nabla_{c,p}$ to be homogeneous or absolute homogeneous. For the examples in Table 3.2, this condition for cliques and STAR is also obviously satisfied since the functions c for cliques and STAR are independent of ψ. For the total variation, we obtain

$$\begin{aligned}
c(i,j,e,\alpha\mathbf{x}) &= \begin{cases} 1 & ((i,j) = \arg\max_{i,j\in V} \left|\alpha\frac{x_j}{\mu_j} - \alpha\frac{x_i}{\mu_j}\right|) \\ 0 & (otherwise) \end{cases} \\
&= \begin{cases} 1 & ((i,j) = \arg\max_{i,j} |\alpha| \left|\frac{x_j}{\mu_j} - \frac{x_i}{\mu_i}\right|) \\ 0 & (otherwise) \end{cases} \\
&= \begin{cases} 1 & ((i,j) = \arg\max_{i,j} \left|\frac{x_j}{\mu_j} - \frac{x_i}{\mu_i}\right|) \\ 0 & (otherwise) \end{cases} \\
&= c(i,j,e,\mathbf{x}),
\end{aligned} \tag{3.55}$$

which therefore satisfies the condition Eq. (3.54). For SUB, we compute

$$\begin{aligned}
(\nabla_{c,p}\alpha\mathbf{x})(e) &= \sum_{i,j\in e} w^{\frac{1}{p}}(e) c^{\frac{1}{p}}(i,j,e,\alpha\mathbf{x}) \alpha \left(\frac{x_j}{\mu_j^{1/p}} - \frac{x_i}{\mu_i^{1/p}}\right) \\
&= \sum_{i=1}^{|e|} \alpha F(S_i) \max_{S\subset e}(F(S)) \left(-\frac{x_{i+1}}{\mu_{i+1}^{1/p}} + \frac{x_i}{\mu_i^{1/p}}\right) \\
&= -\alpha\nabla_{c,p}\psi(e),
\end{aligned} \tag{3.56}$$

and therefore this satisfies the condition.

Secondly, we discuss the first condition, which is

$$\sum_{i,j\in e} c(i,j,e,\mathbf{x}) = c'(e). \tag{3.57}$$

The condition Eq. (3.57) wants the summation of the function over all the pairs of vertices at an edge e to be independent of ψ. For the examples in Table 3.2, this condition for cliques and STAR is satisfied since the functions c for cliques and STAR are independent of ψ. For the total variation, the function c depends on ψ. However, since the function c for total variation can be written as

$$c(i,j,e,\mathbf{x}) = \begin{cases} 1 & ((i,j) = \arg\max_{i,j\in V} |x_j - x_i|) \\ 0 & (otherwise), \end{cases} \tag{3.58}$$

then the summation is

$$\sum_{i,j\in e} c(i,j,e,\mathbf{x}) = 1. \tag{3.59}$$

Therefore, the summation of $c(i,j,e,\mathbf{x})$ over i,j is independent of $\mathbf{x}$ although $c(i,j,e,\mathbf{x})$ is dependent on $\mathbf{x}$. To see SUB, we observe that

$$\sum_{i,j\in V} c(i,j,e,\mathbf{x}) = \sum_{i=1}^{|e|} \max_{S\subset e}(F(S))F(S_i) = c(e) \tag{3.60}$$

which is independent of the order of $\mathbf{x}$ and the particular vertices i,j.

The third condition is

$$\partial c^{\frac{1}{p}}(i,j,e,\mathbf{x})/\partial \mathbf{x} \mid_{i:i\in e} = 0, \forall e \in E, j \in e, \tag{3.61}$$

$$\partial c^{\frac{1}{p}}(i,j,e,\mathbf{x})/\partial \mathbf{x} \mid_{j:j\in e} = 0, \forall e \in E, i \in e \tag{3.62}$$

which means the function $c(i,j,e,\mathbf{x})$ is constant once we fix $e \in E$ and one vertex in the edge. From this condition, we obtain

$$\begin{aligned} &\frac{\partial}{\partial \psi} c^{\frac{1}{p}}(i,j,e,\mathbf{x})\mathbf{x}\Big|_{j:\, j\in e} \\ &= \frac{\partial}{\partial \mathbf{x}} c^{\frac{1}{p}}(i,j,e,\mathbf{x})\Big|_{j:\, j\in e} \mathbf{x} + c^{1/p}(i,j,e,\mathbf{x}) \\ &= c^{1/p}(i,j,e,\mathbf{x}), \forall e \in E, i \in e. \end{aligned} \tag{3.63}$$

Similarly, we can prove the condition of i. This implies that the function c works as a coefficient for $\mathbf{x}$ once we fix j and e, and c is independent of $\mathbf{x}$. Note that if c is not differentiable, then we simply change to subdifferentiation as

$$\partial c^{\frac{1}{p}}(i,j,e,\mathbf{x}) \mid_{j:j\in e} \ni 0, \forall e \in E, i \in e, \tag{3.64}$$

$$\partial c^{\frac{1}{p}}(i,j,e,\mathbf{x}) \mid_{i:i\in e} \ni 0, \forall e \in E, j \in e. \tag{3.65}$$

For examples in Table 3.2, this condition is also obviously satisfied, since the functions c for cliques and star are independent of $\mathbf{x}$. Although the function c for total variation depends on $\mathbf{x}$, the function c is constant once we fix one vertex and one edge e. Moreover, this implies that the function c satisfies the condition Eq. (3.62).

3.G Proof of Theorem 3.13

Most of the proof can be done in the similar manner as graph [Tudisco and Hein, 2018, Li and Milenkovic, 2018], while we need to change from the the graph p-Laplacian in [Tudisco and Hein, 2018] and hypergraph p-Laplacian in [Li and Milenkovic, 2018] to our framework p-Laplacian. We firstly denote by $(\mathbf{x}|_A)_i$ for a set $A \subset V$ as

$$(\mathbf{x}|_A)_i = \begin{cases} x_i & i \in A \\ 0 & i \notin A \end{cases} \tag{3.66}$$

We start to prove the following lemma to prove Thm. 3.13.

Lemma 3.28. *For a set $A \subset V$,*

$$\left\langle \frac{\partial}{\partial \mathbf{x}} (\nabla_{c,p}\mathbf{x})(e), \mathbf{x}|_A \right\rangle = (\nabla_{c,p}\mathbf{x}|_A)(e). \tag{3.67}$$

Proof. Since hypergraph-gradient is a first degree polynomial of $\mathbf{x}$ from Def. 3.2, for $c_i \in \mathbb{R}$ we can write hypergraph-gradient as

$$(\nabla_{c,p}\mathbf{x})(e) = \sum_v c_i x_i. \tag{3.68}$$

By Eq. (3.68),

$$\left\langle \frac{\partial}{\partial \mathbf{x}} \nabla_{c,p}\mathbf{x}(e), \mathbf{x}|_A \right\rangle = \sum_v c_{v,\mathbf{x}}(\mathbf{x}|_A)_i = (\nabla_{c,p}\mathbf{x}|_A)(e), \tag{3.69}$$

which ends the proof. □

We move to prove the following lemma.

Lemma 3.29. *Denote $\lambda_{c,p}, \boldsymbol{\psi}_{c,p}$ be a p-eigenpair of p-Laplacian $\Delta_{c,p}$. Let $A_1(\boldsymbol{\psi}_{c,p}), \ldots, A_m(\boldsymbol{\psi}_{c,p})$ is a nodal domains induced from $\boldsymbol{\psi}_{c,p} \in H(V)$, and $\boldsymbol{\psi}'_{c,p}$ be the vector $\boldsymbol{\psi}'_{c,p} \in F(\boldsymbol{\psi}_{c,p})$, where $F(\psi)$ is a nodal space induced from ψ. Then, $S_{G,c,p}(\boldsymbol{\psi}'_{c,p}) \leq \lambda_{c,p}\|\boldsymbol{\psi}'_{c,p}\|_p^p$*

Proof. We consider the vector $\mathbf{f} = \sum_{i=1}^m \alpha_i \boldsymbol{\psi}_{c,p}|_{A_i}$, where α_i is a constant. From the definition of nodal domains, each edge e intersects at most two nodal domains with different signs. Therefore, $\boldsymbol{\psi}_{c,p}|_{e\cap A_i} = \boldsymbol{\psi}_{c,p}|_e$ for any $e \in E$ and for any nodal domain A_i, and $\boldsymbol{\psi}_{c,p}|_{e\cap A_i} = \operatorname{sgn}(\alpha_i)y|_e$ for any $e \in E$ and for any nodal domain A_i. We divide edges into two classes according to the number of nodal domains intersected by each edge as follows.

$$E_1 = \{e \mid : |\{e \cap A_i\}| \leq 1\} \tag{3.70}$$

$$E_2 = \{e \mid : |\{e \cap A_i\}| = 2\} \tag{3.71}$$

Note that $E_1 \cup E_2 = E$. Then, since $\nabla_{c,p}\boldsymbol{\psi}_{c,p}(e) = \nabla_{c,p}\boldsymbol{\psi}_{c,p}|_{A_i}(e)$ if $A_i \cap e = \emptyset$ and $\nabla_{c,p}\boldsymbol{\psi}_{c,p}(e) = 0$ for those i such that $A_i \cap e = \emptyset$ and simpler version of Hölder's inequality $(\sum_{i=1}^n |x_i|)^p \leq n^{p-1}\sum_{i=1}^n |x_i|^p$, we have

$$\begin{aligned} S_{G,c,p}(\mathbf{f}) &= \sum_{e\in E} |\nabla_{c,p}\mathbf{f}(e)|^p \\ &= \sum_{e\in E_1}\sum_i |\alpha_i|^p |\nabla_{c,p}\boldsymbol{\psi}_{c,p}|_{A_i}(e)||\nabla_{c,p}\boldsymbol{\psi}_{c,p}(e)|^{p-1} + \sum_{e\in E_2}\left(\sum_i |\alpha||\nabla_{c,p}\boldsymbol{\psi}_{c,p}|_{A_i}(e)|\right)^p \end{aligned} \tag{3.72}$$

Moreover, we have

$$\lambda_{c,p}\|\mathbf{f}\|_p^p = \sum_i |\alpha_i|^p \lambda_{c,p}\|\boldsymbol{\psi}_{c,p}|_{A_i}\|_p^p$$

$$
\begin{aligned}
&= \sum_i |\alpha_i|^p \langle \psi|_{A_i}, \Delta_{c,p}\boldsymbol{\psi}_{c,p}\rangle_{H(V)} \\
&= \sum_i |\alpha_i|^p \sum_v \boldsymbol{\psi}_{c,p}|_{A_i}(v)\Delta_{c,p}\psi_{c,p}(v) \\
&= \sum_i |\alpha_i|^p \sum_v \psi|_{A_i}(v) \sum_{e\colon v\in e} \frac{1}{|e|!}|\nabla_{c,p}\boldsymbol{\psi}_{c,p}(e)|^{p-1}\frac{\partial}{\partial \boldsymbol{\psi}_{c,p}}\nabla_{c,p}\boldsymbol{\psi}_{c,p}(e) \\
&= \sum_i |\alpha_i|^p \sum_v \sum_{e\colon v\in e} \frac{1}{|e|!}|\nabla_{c,p}\boldsymbol{\psi}_{c,p}(e)|^{p-1}\frac{\partial}{\partial \boldsymbol{\psi}_{c,p}}\nabla_{c,p}\boldsymbol{\psi}_{c,p}(e)\boldsymbol{\psi}_{c,p}|_{A_i}(v) \\
&= \sum_i |\alpha_i|^p \sum_e \nabla_{c,p}\boldsymbol{\psi}_{c,p}|_{A_i}(e)|\nabla_{c,p}\boldsymbol{\psi}_{c,p}(e)|^{p-1} \qquad (3.73)
\end{aligned}
$$

From Eq. (3.72) and Eq. (3.73), we have

$$
\begin{aligned}
&S_{G,c,p}(\mathbf{f}) - \lambda_{c,p}\|\mathbf{f}\|_p^p \\
&= \sum_{e\in E_2}\left(\left(\sum_i |\alpha||\nabla_{c,p}\boldsymbol{\psi}_{c,p}|_{A_i}(e)|\right)^p - \left(\sum_i |\alpha_i|^p \sum_e \nabla_{c,p}\boldsymbol{\psi}_{c,p}|_{A_i}(e)|\nabla\boldsymbol{\psi}_{c,p}(e)|^{p-1}\right)\right) \\
&= \sum_{e\in E_2}\left(|\alpha_{i_1}|\nabla_{c,p}\boldsymbol{\psi}_{c,p}|_{A_{i_1}}(e) + |\alpha_{i_2}|\nabla_{c,p}\boldsymbol{\psi}_{c,p}|_{A_{i_2}}(e)\right)^p \\
&+ \left(|\alpha_{i_1}|^p\nabla_{c,p}\boldsymbol{\psi}_{c,p}|_{A_{i_1}}(e) + |\alpha_{i_2}|^p\nabla_{c,p}\boldsymbol{\psi}_{c,p}|_{A_{i_2}}(e)\right)\left(\nabla_{c,p}\boldsymbol{\psi}_{c,p}|_{A_{i_1}} + \nabla_{c,p}\boldsymbol{\psi}_{c,p}|_{A_{i_2}}\right) \\
&\geq 0. \qquad (3.74)
\end{aligned}
$$

The last inequality follows from Lemma 3.7 in [Tudisco and Hein, 2018]. □

Now we prove Thm. 3.13. Suppose that $\lambda_{c,p,k}$ has multiplicity r and associated eigenvector $\boldsymbol{\psi}_{c,p,k}$. As functions $\boldsymbol{\psi}_{c,p,k}|_{A_1}, \ldots, \boldsymbol{\psi}_{c,p,k}|_{A_m}$ are linear independent of the definition of a nodal domain, $\gamma(F\cap\mathcal{S}_p) \leq m$. From Lemma 3.29, for all $\mathbf{x}' \in F\cap\mathcal{S}_p$ we have $R_{G,c,p}(\mathbf{x}') \leq R_{G,c,p}(\boldsymbol{\psi}_{c,p,k}) = \lambda_{c,p,k}$. Also, $F\cap\mathcal{S}_p \in \mathcal{F}_m(\mathcal{S}_p)$. Hence,

$$
\lambda_{c,p,m} = \min_{A\in\mathcal{F}_m(\mathcal{S}_p)}\max_{\mathbf{x}'\in A} R_{G,c,p}(\mathbf{x}') \leq \max_{\mathbf{x}\in F\cap\mathcal{S}_p} R_{G,c,p}(\mathbf{x}) \leq \lambda_{c,p,k}. \qquad (3.75)
$$

From Eq. (3.75) $\lambda_{c,p,m} \geq \lambda_{c,p,k}$ and $m \geq k + r - 1$. This concludes the proof.

3.H Proof of Theorem 3.14

We begin our discussion by the following lemma.

Lemma 3.30. *Let* $\mathcal{A} = \mathrm{Span}(\mathbf{1}_{V_1} \ldots, \mathbf{1}_{V_k})$. *Choose a vector* $\mathbf{x} \in \mathcal{A} \cap \mathcal{S}_p$ *and suppose that it can be written as* $\|\mathbf{x}\|_p^p = 1$ *and* $\mathbf{x} = \sum_i \alpha_i M^{1/p} \mathbf{1}_{V_i}$. *Then,*

$$\sum_i |\alpha_i|^p \mathrm{vol}(V_i) = 1. \tag{3.76}$$

Proof. As $\mathcal{A}$ is a k-way partition, $V_i \cap V_j = \emptyset$. Therefore, we obtain

$$\begin{aligned} 1 &= \|\mathbf{x}\|_p^p \\ &= \sum_i \|\alpha_i M^{1/p} \mathbf{1}_{V_i}\|_p^p \\ &= \sum_i |\alpha_i|^p \mathrm{vol}(V_i) \end{aligned} \tag{3.77}$$

□

Firstly, we prove the upper bound of the inequality. Without loss of generality, we limit our interest to $\|\mathbf{x}\|_p^p = 1$. If we set $\mathbf{x} = \sum_i \alpha_i M^{1/p} \mathbf{1}_{V_i}$, then we obtain

$$\begin{aligned} R_{G,c,p}(\mathbf{x}) &= \sum_e \|\nabla_{c,p} \mathbf{x}\|_p^p \\ &= \sum_e \sum_i \alpha_i^p \|\nabla_{c,p}(M^{1/p} \mathbf{1}_{V_i})\|_p^p \\ &\leq \sum_e \min(k, |e|)^{p-1} \sum_i |\alpha|^p \|\nabla_{c,p}(M^{1/p} \mathbf{1}_{V_i})\|_p^p \\ &\leq \min(\max |e|, k) \sum_i |\alpha|^p \sum_e \|\nabla_{c,p}(M^{1/p} \mathbf{1}_{V_i})\|_p^p \end{aligned} \tag{3.78}$$

From $|\nabla_{c,p} M^{1/p} \mathbf{1}_{V_i}|_p^p \leq \sum_{e \in V_i \overline{V_i}} c(e) w(e) = \mathrm{Cut}_c(V_i, \overline{V_i})$ and Lemma 3.30, Eq. (3.78) can be tightened as

$$\begin{aligned} R_{G,c,p}(\mathbf{x}) &\leq \min(\max |e|, k) \sum_i |\alpha|^p \mathrm{Cut}_c(V_i, \overline{V_i}) \\ &\leq \min(\max |e|, k) \frac{\sum_i |\alpha|^p \mathrm{Cut}_c(V_i, \overline{V_i})}{\sum_i |\alpha|^p \mathrm{vol}(V_i)} \\ &\leq \min(\max |e|, k) h_{c,k}. \end{aligned} \tag{3.79}$$

Next, we prove the lower bound. The proof of the lower bound depends on the following lemma.

Lemma 3.31. *For any vectors* $\mathbf{x} \in H(V)$*, and* $p \geq 1$*, there exists* $\varsigma \geq 0$ *such that* $B(x, \varsigma) = \{i : x_i > \varsigma\}$ *satisfies*

$$R_{G,c,p}(M^{1/p}\mathbf{x}) \geq (\rho)^{p-1} \left(\frac{c(V)}{p}\right)^p, \tag{3.80}$$

where $\rho \coloneqq \max_i(d_i/\mu_i)$.

Proof. We denote by $\mathbf{x}^p$ element-wise p-th power. Then, we obtain

$$\begin{aligned}
S_{G,c,1}(M\psi^p) &= \sum \|\nabla_{c,p}(M\mathbf{x}^p)\| \\
&= \sum_e \sum_{i,j\in e} w(e)c(i,j,e,\mathbf{x})|x_i^p - x_j^p| \\
&\leq \sum_e \sum_{i,j} w(e)pc(i,j,e,\mathbf{x})\,|x_i - x_j| \max(x_i,x_j)^{p-1} \\
&\leq p\left(\left(\sum_e \sum_{i,j} w(e)pc(i,j,e,\mathbf{x})|x_j - x_i|\right)^p\right)^{\frac{1}{p}} \\
&\times \left(\sum_e \sum_{i,j\in e} \max(x_i,x_j)^p\right)^{\frac{p-1}{p}} \\
&\leq pS_{G,p}^{\frac{1}{p}}(M^{1/p}\mathbf{x}) \left(\sum_{i\in V} d_i x_j^p\right)^{\frac{p-1}{p}} \\
&\leq p\rho^{1-1/p} R_{G,c,p}^{\frac{1}{p}}(\mathbf{x}) \|M^{1/p}\mathbf{x}\|_p^{p-1}
\end{aligned} \tag{3.81}$$

Moreover, we get

$$\begin{aligned}
S_{G,c,1}(M\mathbf{x}^p) &= \sum_{e\in E} \sum_{i,j\in e} w(e)c(i,j,e,\psi)\left|x_i^p - x_j^p\right| \\
&= \sum_{e\in E} \sum_{i,j\in e; x_i^p - x_j^p > 0} w(e)c(i,j,e,\mathbf{x}) \int_{x_j}^{x_i} d\varsigma \\
&= \int_0^\infty \sum_{e\in E} w(e) \sum_{i,j\in e:\, x_i^p - x_j^p > \varsigma} c(i,j,e,\mathbf{x}) d\varsigma.
\end{aligned} \tag{3.82}$$

From Eq.(3.82), the left hand side of Eq. (3.80) can be rewritten as

$$\begin{aligned}\frac{S_{G,c,1}(M\mathbf{x}^p)}{\|M\mathbf{x}\|_p^p} &= \frac{\int_0^\infty \sum_{e\in E}\sum_{i,j\in e; x_i^p - x_j^p>\varsigma} w(e)c(i,j,e,\mathbf{x})d\varsigma}{\int_0^\infty \sum_{x_j^p>\varsigma}\mu(j:x_j^p>\varsigma)d\varsigma} \\ &\geq \inf_{\varsigma>0}\frac{\sum_{e\in E} w(e)\sum_{i,j\in e; x_i^p-x_j^p>\varsigma} c(i,j,e,\mathbf{x})}{\mu(i:x_i^p>\varsigma)} \\ &= \inf_{\varsigma>0}\frac{\mathrm{Cut}_c(\{i:x_i^p>\varsigma\},\overline{\{i:x_i^p>\varsigma\}})}{\mathrm{vol}(\{i:x_i^p>\varsigma\})} \\ &= \inf_{\varsigma>0} c(\{i:x_i^p>\varsigma\}) \end{aligned} \tag{3.83}$$

Hence, the following inequality holds for the set $B^* = \{i : x_i^p > \varsigma^*\}$ where ς^* is the minimizer.

$$\begin{aligned} R_{G,c,p}(M^{1/p}\mathbf{x}) &= \frac{S_{G,c,p}(M^{1/p}\mathbf{x})}{\|M^{1/p}\mathbf{x}\|_p^p} \\ &\geq \left(\frac{S_{G,c,1}(M\mathbf{x}^p)}{\|M^{1/p}\mathbf{x}\|_p^p}\frac{1}{p^p\rho^{p-1}}\right) \\ &= \left(\frac{1}{\rho}\right)^{p-1}\left(\frac{c(B^*)}{p}\right)^p \end{aligned} \tag{3.84}$$

This concludes Lemma 3.31. □

Suppose $\lambda_{c,p,k}$ has a corresponding eigenvector ψ_c that induces the strong nodal domains $A_1, A_2, \ldots, A_m$. From Lemma 3.29, we have $\lambda \leq R_{G,c,p}(M^{1/p}\mathbf{1}_{A_i})$. Moreover, from Lemma 3.31, for any $i(\ell = 1,\ldots,m)$, there exists a set $B_\ell \subset A_\ell$ such that

$$R_{G,c,p}(M^{1/p}\mathbf{1}_{A_i}) \leq \left(\frac{1}{\rho}\right)^{p-1}\left(\frac{c(B_i))}{p}\right)^p \tag{3.85}$$

Therefore,

$$\begin{aligned}\lambda_{c,p,k} &\geq R_{G,c,p}(M^{1/p}\mathbf{1}_{A_i}) \\ &\geq \max_i \left(\frac{1}{\rho}\right)^{p-1}\left(\frac{c(B_i))}{p}\right)^p \\ &\geq \min_{B_i}\max_i \left(\frac{1}{\rho}\right)^{p-1}\left(\frac{c(B_i))}{p}\right)^p \end{aligned}$$

$$\geq \left(\frac{1}{\rho}\right)^{p-1} \left(\frac{h_{c,m}}{p}\right)^{p} \tag{3.86}$$

3.I Proof of Corollary 3.15

Let us start with the following lemma.

Lemma 3.32. *Let $\mathbf{x} \in H(V)$ be orthogonal to $\mathbf{1}$. Then there is $\mathbf{x}' \in H(V)$ and for all $i \in V, x'_j \geq 0$ with at most $|V|/2$ non-zero entries such that*

$$R_{G,c,p}(M^{1/p}\mathbf{x}') \leq R_{G,c,p}(M^{1/p}\mathbf{x}). \tag{3.87}$$

Moreover, $\forall t$ satisfying $0 < t \leq \max_v x'_i$, the set $B = \{i : x'_i \geq t\}$ is one of the set obtained by $\mathbf{x}$, such that $(\{i : x_i \geq t\}, \{i : x_i < t\})$ minimizing Cheeger cut.

Proof. Firstly, we observe that

$$R_{G,c,p}(M^{1/p}(\mathbf{x} + c\mathbf{1})) \geq R_{G,c,p}(\mathbf{x}), \tag{3.88}$$

since $S_{G,c,p}(M^{1/p}(\mathbf{x} + c\mathbf{1})) = S_{G,c,p}(M^{1/p}\mathbf{x})$ and $\|M^{1/p}(\mathbf{x} + c\mathbf{1})\| = \|M^{1/p}(\mathbf{x})\| + \|cM^{1/p}\mathbf{1}\| \geq \|M^{1/p}\mathbf{x}\|$.

Let ς be the median value of $\mathbf{x}$, and set $\mathbf{x}_\varsigma \coloneqq \mathbf{x} - \varsigma\mathbf{1}$. Then $R_{G,c,p}(\mathbf{x}_\varsigma) < R_{G,c,p}(\mathbf{x}_\varsigma)$, and median of $\mathbf{x}_\varsigma$ is zero, which means $\mathbf{x}_m$ has at most $|V|/2$ positive entities and at most $|V|/2$ negative entities. We decompose ψ_m as follows;

$$\mathbf{x}_\varsigma = \mathbf{x}_{\varsigma+} - \mathbf{x}_{\varsigma-}, \tag{3.89}$$

where $(\mathbf{x}_{\varsigma+})_i = (\mathbf{x}_\varsigma)_i$ if $(\mathbf{x}_\varsigma)_i$ is positive and otherwise set 0, and $(\mathbf{x}_{\varsigma-})_i = -(\mathbf{x}_\varsigma)_i$ if $(\mathbf{x}_\varsigma)_i$ is negative and otherwise set 0. We remark that $\mathbf{x}_{\varsigma+}$ and $\mathbf{x}_{\varsigma-}$ are non-negative, orthogonal to each other, and have at most $|V|/2$ non-zero entities. The cut defined by the set $\{i : (\mathbf{x}_{\varsigma+})_i \geq t\}$ for $t \in \mathbb{R}$ is one of the cut obtained by $\mathbf{x}$, such that $(\{i : x_i \geq t\}, \{i : x_i < t\})$ minimizing Cheeger cut, since we can obtain the same cut by considering

$$(\{i : x_i \geq t + m\}, \{i : x_i < t + m\}). \tag{3.90}$$

Similarly, cut defined by the set $\{i : (\mathbf{x}_{m-})_i \geq t\}$ for $t \in \mathbb{R}$ is one of the cut obtained by $\mathbf{x}$, such that $(\{i : x_i \geq t\}, \{i : x_i < t\})$ minimizing Cheeger cut.

We move on to show that at least one of $\mathbf{x}_{\varsigma+}$ or $\mathbf{x}_{\varsigma-}$ has Rayleigh quotient equal to or smaller than Rayleigh quotient of $\mathbf{x}_{\varsigma}$, by showing the following

$$\begin{aligned} R_{G,c,p}(M^{1/p}\mathbf{x}_{\varsigma}) &= \frac{S_{G,c,p}(M^{1/p}\mathbf{x}_{\varsigma})}{\|M^{1/p}\mathbf{x}_{\varsigma}\|^p} \\ &\geq \frac{S_{G,c,p}(M^{1/p}\mathbf{x}_{\varsigma+}) + S_{G,c,p}(M^{1/p}\mathbf{x}_{\varsigma-})}{\|M^{1/p}\mathbf{x}_{\varsigma+}\|^p + \|M^{1/p}\mathbf{x}_{\varsigma-}\|^p} \\ &= \frac{R_{G,c,p}(M^{1/p}\mathbf{x}_{\varsigma+})\|M^{1/p}\mathbf{x}_{\varsigma+}\| + R_{G,c,p}(M^{1/p}\mathbf{x}_{\varsigma-})\|M^{1/p}\mathbf{x}_{\varsigma-}\|}{\|M^{1/p}\mathbf{x}_{\varsigma+}\|^p + \|M^{1/p}\mathbf{x}_{\varsigma-}\|^p} \\ &\geq \min(R_{G,c,p}(M^{1/p}\mathbf{x}_{\varsigma+}), R_{G,c,p}(M^{1/p}\mathbf{x}_{m-})) \end{aligned} \tag{3.91}$$

This concludes the proof. □

By combining lemma 3.31 and lemma 3.32, we can say a stronger statement than Cor. 3.15.

Corollary 3.33. *Let $\mathbf{x} \in H(V)$ be orthogonal to $M^{1/p}\mathbf{1}$, and let $(B, \overline{B})$ be the cut found by ψ, such that $(\{i : x_i \geq t\}, \{i : x_i < t\})$ minimizing Cheeger cut. Then*

$$R_{G,c,p}(M^{1/p}\mathbf{x}) \geq \rho^{p-1}\left(\frac{c(B)}{p}\right)^p \tag{3.92}$$

3.J Proof of Theorem 3.17

By definition, for all $i \in V$, we have

$$(\Delta_{c,p}\boldsymbol{\psi}_c)_i = \lambda_c^{\boldsymbol{\psi}_c}\xi_p(\psi_{c,i}) \tag{3.93}$$

$$(\Delta_{c,p}\boldsymbol{\psi}'_c)_i = \lambda_c^{\boldsymbol{\psi}'_c}\xi_p(\psi'_{c,i}), \tag{3.94}$$

where $\lambda_{c,\mathbf{x}}$ and $\lambda_{c,\mathbf{x}'}$ are eigenvalues associated with $\mathbf{x}$ and $\mathbf{x}'$, respectively. Then, for all $i \in V$ we obtain

$$(\Delta_{c,p}\boldsymbol{\psi}_c)_i\xi_p(\psi'_{c,i}) = \lambda_c^{\boldsymbol{\psi}}\xi_p(\psi_{c,i})\xi_p(\psi'_{c,i}) \tag{3.95}$$

$$(\Delta_{c,p}\boldsymbol{\psi}'_c)_i\xi_p(\psi_{c,i}) = \lambda_c^{\boldsymbol{\psi}'_c}\xi_p(\psi'_{c,i})\xi_p(\psi_{c,i}). \tag{3.96}$$

By summing up over all $v \in V$ and taking difference of both side of Eq. (3.95) and Eq. (3.96), we compute

$$\begin{aligned}&(\lambda_c^{\boldsymbol{\psi}} - \lambda_c^{\boldsymbol{\psi}'}) \sum_{i \in V} \xi_p(\psi_{c,i}) \xi_p(\psi'_{c,i}) \\ &= \sum_{i \in V} (\Delta_{c,p} \boldsymbol{\psi}_c)_i \xi_p(\psi'_{c,i}) - (\Delta_{c,p} \boldsymbol{\psi}'_c)_i \xi_p(\psi_{c,i}) \end{aligned} \tag{3.97}$$

By applying Taylor expansion at $a_{\Delta_{c,p}\boldsymbol{\psi}} M^{1/p}\mathbf{1}$ to $\Delta_{c,p}\boldsymbol{\psi}$, at $a_{\Delta_{c,p}\boldsymbol{\psi}'} M^{1/p}\mathbf{1}$ to $\Delta_{c,p}\boldsymbol{\psi}'$ at $a_{\xi(\boldsymbol{\psi})}$ to $\xi_p(\boldsymbol{\psi})$, and at $a_{\xi(\boldsymbol{\psi}')}$ to $\xi_p(\boldsymbol{\psi}')$ in right hand side of Eq. (3.97), we obtain

$$\begin{aligned}(\lambda^{\psi} - \lambda^{\psi'}) \sum_{i \in V} \xi_p(\psi_{c,i}) \xi_p(\psi'_{c,i}) = &\sum_{i \in V} (\Delta_{c,p} (a_{\Delta_{c,p}\boldsymbol{\psi}_c} M^{1/p}\mathbf{1})_i \\ &+ \Delta'_{c,p} (a_{\Delta_{c,p}\boldsymbol{\psi}_c} M^{1/p}\mathbf{1})_i (\psi_{c,i} - a_{\Delta_{c,p}\psi} \mu_i^{1/p}) + o_{2,\Delta_{c,p},\boldsymbol{\psi}_c}) \\ &\times (\xi_p(a_{\xi(\boldsymbol{\psi}'_c)}) + \xi'_p(a_{\xi(\psi')})(\psi_{c,i} - a_{\xi_p(\boldsymbol{\psi}'_c)}) + o_{2,\xi_p(\boldsymbol{\psi}'_c)}) \\ &- \sum_{i \in V} (\Delta_{c,p} (a_{\Delta_{c,p}\boldsymbol{\psi}'_c} M^{1/p}\mathbf{1})_i \\ &+ \Delta'_{c,p} (a_{\Delta_{c,p}\boldsymbol{\psi}'_c} M^{1/p}\mathbf{1})_i (\psi_{c,i} - c_{\Delta_{c,p}\boldsymbol{\psi}'_c)} + o_{2,\Delta_{c,p}\boldsymbol{\psi}'_c}) \\ &\times (\xi_p(a_{\xi(\psi)}) + \xi'_p(a_{\xi(\psi)})(\psi(v) - a_{\xi_p(\psi)}) + o_{2,\xi_p(\psi)}) \end{aligned} \tag{3.98}$$

By Prop. 3.7 and Cor. 3.21, Eq. (3.98) is

$$(\lambda_c^{\boldsymbol{\psi}} - \lambda_c^{\boldsymbol{\psi}'}) \sum_{i \in V} \xi_p(\psi_{c,i}) \xi_p(\psi'_{c,i}) = o_2,$$

which concludes the proof.

3.K Proof of Theorem 3.18

Before we start a proof, let us motivate the discussion by considering $p = 2$. Let $\boldsymbol{\psi}_{c,1}$ be a first eigenvector either $M^{1/p}\mathbf{1}$, then $\langle \boldsymbol{\psi}_{c,1}, \boldsymbol{\psi}_{c,2} \rangle = 0$ for the second eigenvector $\boldsymbol{\psi}_{c,2}$, and we observe

$$\|\boldsymbol{\psi}_{c,2}\|_2^2 = \left\| \boldsymbol{\psi}_{c,2} - \left(\frac{\langle \boldsymbol{\psi}_{c,1}, \boldsymbol{\psi}_{c,2} \rangle}{|V|} \right) \boldsymbol{\psi}_{c,1} \right\|_2^2 = \min_{\eta \in \mathbb{R}} \|\boldsymbol{\psi}_{c,2} - \eta \boldsymbol{\psi}_{c,1}\|_2^2$$

Using this equation, the Rayleigh quotient to get the second eigenvector ψ_2 can be written as

$$\boldsymbol{\psi}_{c,2} = \underset{\mathbf{x}\in H(V)}{\arg\min} \frac{S_{G,c,2}(\mathbf{x})}{\min_\eta \|\mathbf{x} - \eta\boldsymbol{\psi}_{c,1}\|_2^2}. \tag{3.99}$$

This inspires Eq. (3.19).

Let us start our proof by proving $\lambda_{c,p,2} \geq \inf_{\mathbf{x}} R_{G,c,p}(\mathbf{x})$. Let $\boldsymbol{\psi}_{c,2}$ be a p-eigenvector corresponding to the second p-eigenvalue $\lambda_{c,p,2}$. As $\sum_i \xi_p)(\boldsymbol{\psi}_{c,p,2})_i) = \mathbf{1}^\top \delta\boldsymbol{\psi}_{c,p,2} = 0$, the norm $\|\boldsymbol{\psi}_{c,p,2} - \eta\mathbf{1}\|$ is convex in c is minimized when $c = 0$. Moreover, $\lambda_{c,p,2} = R_{G,c,p}(\boldsymbol{\psi}_{c,2}) = S_{G,c,p}(\boldsymbol{\psi}_{c,2})/\|\boldsymbol{\psi}_{c,2}\|_p^p = S_{G,p,c}(\boldsymbol{\psi}_{c,p,2})/\min_\eta \|\boldsymbol{\psi}_{c,p,2} - \eta\|_p^p = R^{(2)}_{G,c,p}(\boldsymbol{\psi}_{c,p,2})$. Hence, $\lambda_{c,p,2} \geq \inf R^{(2)}_{G,c,p}(\boldsymbol{\psi}_{c,p,2})$.

Second, we prove $\lambda_{c,p,2} \leq \inf_{\mathbf{x}} R_{G,c,p}(\mathbf{x})$. From the definition of $R^{(2)}_{G,c,p}(\mathbf{x})$, we can easily check that $R^{(2)}_{G,c,p}(a\mathbf{x} + b) = R^{(2)}_{G,c,p}(\mathbf{x})$. Let $\mathbf{x}^* = \arg\min R^{(2)}_{G,c,p}(\mathbf{x})$, and consider the space $A = \{a\mathbf{x}^* + b\}$. As $\mathbf{x}^* \neq c\mathbf{1}$, $\gamma(A \cap \mathcal{S}_p) = 2$. From Proposition 3.7, we obtain

$$\begin{aligned}
\lambda_{c,p,2} &\leq \max_{\mathbf{x}\in A\cap\mathcal{S}_p} S_{G,c,p}(\mathbf{x}) \\
&= \max_{a,b} \frac{S_{G,c,p}(a\mathbf{x}^* + b)}{\|a\mathbf{x}^* + b\|_p^p} \\
&= \max_{a,b} \frac{S_{G,c,p}(a\mathbf{x}^*)}{\|a\mathbf{x}^* + b\|_p^p} \\
&= \max_{a} \frac{S_{G,c,p}(a\mathbf{x}^*)}{\min_b \|a\mathbf{x}^* + b\|_p^p} \\
&= R^{(2)}_{G,c,p}(\mathbf{x}^*)
\end{aligned} \tag{3.100}$$

As we have $\lambda_{c,p,2} \geq \inf R^{(2)}_{G,c,p}(\mathbf{x}^*)$ and $\lambda_{c,p,2} \leq \inf R^{(2)}_{G,c,p}(\mathbf{x}^*)$, we obtain $\lambda_{c,p,2} = R^{(2)}_{G,c,p}(\mathbf{x}^*)$.

Since the global minimum is $\lambda_{c,p,2}$, then

$$\begin{aligned}
\lambda_{c,p,2} &= R_{G,c,p}(\boldsymbol{\psi}_{c,p,2}) \\
&= \frac{S_{G,c,p}(\boldsymbol{\psi}_{c,p,2})}{\|\boldsymbol{\psi}_{c,p,2}\|_p^p} \\
&= \frac{S_{G,c,p}(\mathbf{x}^* + \eta^*\boldsymbol{\psi}_{c,p,1})}{\min_\eta \|\mathbf{x}^* - \eta\boldsymbol{\psi}_{c,p,1}\|} \\
&= \frac{S_{G,c,p}(\boldsymbol{\psi}_{c,p,2})}{\min_\eta \|\boldsymbol{\psi}_{c,p,2} - \eta\boldsymbol{\psi}_{c,p,1}\|} \\
&= R^{(2)}_{G,c,p}(\boldsymbol{\psi}^*).
\end{aligned} \tag{3.101}$$

Note that we use $S_{G,c,p}(\boldsymbol{\psi}+\boldsymbol{\psi}_{c,p,1}) = S_{G,c,p}(\boldsymbol{\psi})$. Therefore, we can see $\boldsymbol{\psi}^* = \boldsymbol{\psi}_{c,p,2}+\eta^*\boldsymbol{\psi}_{c,p,1}$, where $\eta = \arg\min_\eta \|\boldsymbol{\psi}_{c,p,2} - \eta\boldsymbol{\psi}_{c,p,1}\|_p^p$.

Chapter 4

Hypergraph Modeling via Spectral Embedding Connection

This chapter proposes a theoretical framework of multi-way similarity to model vector data into hypergraphs for clustering via spectral embedding. For graph cut based spectral clustering, it is common to model vector data into graph by modeling pairwise similarities using kernel function, which has a theoretical connection to graph cut, as discussed in Sec. 2.2. For problems where using multi-way similarities are more suitable than pairwise ones, it is natural to model as a hypergraph. However, although the hypergraph cut is well-studied, there is not yet established a hypergraph cut based framework to model multi-way similarity. In this chapter, we formulate multi-way similarities by exploiting the theoretical foundation of kernel function. We show a theoretical connection between our formulation and hypergraph cut in two ways, generalizing both weighted kernel k-means and the heat kernel, by which we justify our formulation. We also provide a fast algorithm for spectral clustering. Our algorithm empirically shows better performance than existing graphs and other heuristic modeling methods.

4.1 Introduction

This chapter considers modeling a hypergraph from vector data. In Chapter 3, we discussed hypergraph p-Laplacian. In the experiments in Chapter 3, we use categorical data, which is a discrete dataset and thus can be naturally considered as a hypergraph. On the other hand, we were not able to model a hypergraph from vector data, as opposed to the standard graph case where we can easily model a graph from vector data. This chapter discusses such a model to

form a hypergraph. Note that since we miss the hypergraph modeling method from vector data even for $p = 2$ and uniform hypergraph case, we focus on these settings.

As discussed in Sec. 2.2, spectral clustering is useful not only for graph data but also for vector data. We model vector data as a graph by forming a vertex from each data point and an edge from the pairwise similarity of each pair of data points [Goyal and Ferrara, 2018]. One popular modeling method uses kernel functions. The kernel has been theoretically justified via weighted kernel k-means [Dhillon et al., 2004] (See Sec. 2.2.3) and via heat kernel [Belkin and Niyogi, 2003] (See Sec. 2.2.4).

Hypergraphs, generalization of graphs, and hence are suitable to model data that have multi-way relationships (see Sec. 3.2 and Chapter 3). Recall that cut-based spectral clustering for hypergraph has also been established [Zhou et al., 2006, Hein et al., 2013]. Therefore, from the discussion on graphs, it is natural to model vector data as hypergraphs for clustering. However, while heuristic modeling as hypergraphs have been done in several domains [Govindu, 2005, Sun et al., 2017, Yu et al., 2018], we are yet to have a modeling framework that is theoretically connected to hypergraph cut problems.

This chapter proposes a hypergraph modeling and its spectral embedding framework for clustering, which we theoretically connect to the established hypergraph cut problems. This framework models vector data as an even order r-uniform hypergraphs, all of whose edges connect m vertices. For this purpose, we propose a *biclique kernel*, which formulates multi-way similarity, by exploiting the kernel function's ability to model similarity but in a way where we expand from pairs to multiplets. We give a theoretical foundation to biclique kernel; a biclique kernel is equivalent to semi-definite even-order tensor (Thm. 4.1). We show that biclique kernel is theoretically connected to the established hypergraph cut problems proposed by [Zhou et al., 2006, Saito et al., 2018, Ghoshdastidar and Dukkipati, 2015] via two problems, weighted kernel k-means and heat kernels. We provide a spectral clustering algorithm for our formulation, which is faster than existing ones ($O(n^3)$ vs. $O(n^r)$, where n is the number of data points). This speed-up allows us to model as an arbitrarily higher-order hypergraphs in a reasonable computational time. We numerically demonstrate that our algorithm outperforms the existing graph and heuristic modeling methods. Our empirical study also shows that by increasing order of a hypergraph, the performance is gained until a certain point but slightly drops from there. To our knowledge, it is first time to obtain

the behavior of performance of spectral clustering using higher-order (say, $r \geq 8$) uniform hypergraph.

Our contributions are as follows; i) We provide a formulation to model vector data as an even order r-uniform hypergraph. ii) We show that our formulation is theoretically linked to the established hypergraph cuts in two ways, weighted kernel k-means and heat kernel. iii) We provide a fast spectral clustering algorithm. iv) We numerically show that our method outperforms the standard graph ones and existing heuristic modeling ones.

All proofs are in Appendix.

4.2 Half Symmetric Tensors, Semi-Definiteness, and Uniform Hypergraph

We consider the uniform hypergraph, and introduce that uniform hypergraph can be written using tensors.

We define an r-order tensor as $\mathcal{A} \in \mathbb{R}^{n_1 \times \ldots \times n_r}$, whose $(i_1, i_2, \ldots, i_r)$-th element is $a_{i_1 i_2 \ldots i_r} \in \mathbb{R}$. If all the dimensions of an r-order tensor $\mathcal{A}$ are identical, i.e., $n_1 = \ldots = n_r = n$, we call this tensor as *cubical*. Letting $\mathfrak{S}_r$ be a set of permutations σ on $\{1, ..., r\}$, an even r-order cubical tensor is called as *half-symmetric* if for every elements

$$\mathcal{A}_{i_{\sigma(1)} \ldots i_{\sigma(r/2)} i_{r/2+\sigma'(1)} \ldots i_{r/2+\sigma'(r/2)}} = \mathcal{A}_{i_{r/2+\sigma'(1)} \ldots i_{r/2+\sigma'(r/2)} i_{\sigma(1)} \ldots i_{\sigma(r/2)}}, \forall \sigma, \sigma' \in \mathfrak{S}_{\frac{r}{2}}, \tag{4.1}$$

As an example of half-symmetricity, we consider 4-order half-symmetric cubical tensor $\mathcal{A}$. Let us think about the element (1,2,3,4) of half-symmetric tensor $\mathcal{A}$. Then

$$\mathcal{A}_{1234} = \mathcal{A}_{2134} = \mathcal{A}_{1243} = \mathcal{A}_{2143} = \mathcal{A}_{3412} = \mathcal{A}_{4312} = \mathcal{A}_{3421} = \mathcal{A}_{4321}.$$

More general, for elements (i_1, i_2, i_3, i_4) where $i_j \neq i_l$ if $j \neq l$, the tensor $\mathcal{A}$ is

$$\mathcal{A}_{i_1 i_2 i_3 i_4} = \mathcal{A}_{i_2 i_1 i_3 i_4} = \mathcal{A}_{i_1 i_2 i_4 i_3} = \mathcal{A}_{i_2 i_1 i_4 i_3} = \mathcal{A}_{i_3 i_4 i_1 i_2} = \mathcal{A}_{i_4 i_3 i_1 i_2} = \mathcal{A}_{i_3 i_4 i_2 i_1} = \mathcal{A}_{i_4 i_3 i_2 i_1}.$$

Then, the natural question to ask is what happens if the elements contain the same index. Let us think about the element (1,1,2,3) of half-symmetric tensor $\mathcal{A}$. Then

$$\mathcal{A}_{1123} = \mathcal{A}_{1132} = \mathcal{A}_{2311} = \mathcal{A}_{3211}.$$

However, if we think about the elements containing the same index but in a different "half", e.g., (1,2,1,3), then

$$\mathcal{A}_{1213} = \mathcal{A}_{2113} = \mathcal{A}_{1231} = \mathcal{A}_{2131}$$
$$\mathcal{A}_{1312} = \mathcal{A}_{3112} = \mathcal{A}_{1321} = \mathcal{A}_{3121}.$$

Needless to say, we do not have such a permutation of the index for the "diagonal" elements, where all the indices are the same number, e.g., $(1,1,1,1)$ and $\mathcal{A}_{1111}$.

In the following, we assume a half-symmetric even-order cubical tensor. We define the *mode-$\mathfrak{k}$ product* of $\mathcal{A} \in \mathbb{R}^{n_1 \times \ldots \times n_r}$ and a vector $\mathbf{x} \in \mathbb{R}^{n_\mathfrak{k}}$ as $\mathcal{A} \times_\mathfrak{k} \mathbf{x} \in \mathbb{R}^{n_1 \times \ldots \times n_{\mathfrak{k}-1} \times 1 \times n_{\mathfrak{k}+1} \times \ldots \times n_r}$, whose element is

$$(\mathcal{A} \times_\mathfrak{k} \mathbf{x})_{i_1 \ldots i_{\mathfrak{k}-1} 1 i_{\mathfrak{k}+1} \ldots i_r} := \sum_{i_\mathfrak{k}=1}^{n_\mathfrak{k}} \mathcal{A}_{i_1 \ldots i_\mathfrak{k} \ldots i_r} x_{i_\mathfrak{k}} \tag{4.2}$$

We define a *contracted matrix* $A^{(r)}$ for a half-symmetric even r-order cubical tensor $\mathcal{A}$ as

$$A^{(r)} := \mathcal{A} \times_2 \mathbf{1} \times_3 \cdots \times_{\frac{r}{2}-1} \mathbf{1} \times_{\frac{r}{2}+1} \mathbf{1} \cdots \times_r \mathbf{1} \tag{4.3}$$

Note that $A^{(r)}$ is symmetric. For details, see [Lim, 2005, Qi, 2005, De Lathauwer et al., 2000].

Similarly to the matrix case, we define the semi-definiteness of even-order tensors. An even r-order cubical tensor $\mathcal{A}$ is *semi-definite* if

$$\mathcal{A} \times_1 \mathbf{x} \ldots \times_m \mathbf{x} = \sum_{i_1 \ldots i_m} \mathcal{A}_{i_1 \ldots i_m} \mathbf{x}_{i_1} \ldots \mathbf{x}_{i_m} \geq 0. \tag{4.4}$$

Note that our definition of semi-definiteness is not our own and follows the existing work such as [Qi, 2005, Hu and Qi, 2012, Hillar and Lim, 2013]. Note also that the semi-definiteness can be applied only to even order tensors since no odd-order tensors satisfy this semi-definiteness for the following reason. Let us assume a tensor $\mathcal{A}$ is 3-order cubical tensor. From this definition, polynomial for $\mathbf{x}$ formed from odd order tensor can take both positive and negative values, such as

$$\mathcal{A} \times_1 -\mathbf{x} \times_2 -\mathbf{x} \times_3 -\mathbf{x} = -\mathcal{A} \times_1 \mathbf{x} \times_2 \mathbf{x} \times_3 \mathbf{x}.$$

This means that, the polynomial Eq. (4.4) for $\mathbf{x}$ and for $-\mathbf{x}$ take different signs. However, the semi-definiteness requires the polynomial Eq. (4.4) to be positive for all vectors, including $\mathbf{x}$ and $-\mathbf{x}$. Therefore, there is no odd-order semi-definite tensors for this definition. For more discussion on tensor semi-definiteness, see Sec. 11 in [Hillar and Lim, 2013] and [Qi, 2005].

An r-uniform hypergraph can be represented by an r-order cubical tensor. Recall that we call hypergraph is *uniform*, when all the edge contains the same number of vertices,. We define an *adjacency tensor* $\mathcal{A}$ for uniform hypergraph, where we assign the weight of edge $e = \{i_1, \ldots, i_r\}$ to $(i_1, \ldots, i_m)$-th element of r-order cubical tensor. A uniform hypergraph is *half-undirected* when its adjacency tensor is half-symmetric. Note that a uniform hypergraph is half-undirected if undirected. The following assumes that a hypergraph G is uniform, connected, half-undirected, and has self-loops unless noted.

Since this thesis focuses on clustering, we remark on using this representation for the clustering. In terms of clustering for half-undirected uniform hypergraph, which is mainly discussed in Chapter 5, these three different methods produce the same result (see Sec. 2.5.2.1). When we discuss half-directed uniform hypergraph, for a matrix representation, we use the star method, which contracts a hypergraph into a graph by forming $A_s := HW_eH^\top/r$.

4.3 Formulation of Multi-way Similarity

This section proposes a formulation of multi-way similarity and discusses its properties. Looking back at a pairwise similarity, kernel functions are a convenient tool to model a similarity. However, kernel functions consider pairwise similarities, not multi-way similarities. The idea to construct a multi-way similarity framework is that we take the benefits of the kernel framework's modeling ability, but at the same time, we expand to multiplets from pairs.

4.3.1 Biclique Kernel and Tensor Semi-definitness

This section formulates multi-way similarity as a *biclique kernel* and discusses its semi-definite property. For two sets of $r/2$ variables, $\{\mathbf{x}_{i.}\}$ and $\{\mathbf{t}_{l.}\}$, $\mathbf{x}_{i.}, \mathbf{t}_{l.} \in \mathbf{X}$, $\mathbf{X} \subseteq \mathbb{R}^d$, we formulate even r multi-way similarity function $\kappa^{(r)}(\mathbf{x}_{i_1}, .., \mathbf{x}_{i_{r/2}}, \mathbf{t}_{l_1}, .., \mathbf{t}_{l_{r/2}}) : \mathbf{X}^{r/2} \times \mathbf{X}^{r/2} \to \mathbb{R}$ as

$$\kappa^{(r)}(\{\mathbf{x}_{i.}\}, \{\mathbf{t}_{l.}\}) := \sum_{\gamma=1}^{r/2} \sum_{\nu=1}^{r/2} \kappa(\mathbf{x}_{i_\gamma}, \mathbf{t}_{l_\nu}), \tag{4.5}$$

where $\kappa : \mathbf{X} \times \mathbf{X} \to \mathbb{R}$ is a standard kernel. We call κ as a *base kernel*. By construction, $\kappa^{(r)}$ is also a kernel. Therefore, we call $\kappa^{(r)}$ as *biclique kernel*. Let $\mathcal{K}$ be a *gram tensor* of $\kappa^{(r)}$, i.e., an r-order cubical tensor formed by Eq. (4.5), whose $(i_1, \ldots, i_r)$-th element is $\kappa^{(r)}(\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_r})$. Note that $\mathcal{K}$ is half-symmetric due to the construction of $\kappa^{(r)}$. Seeing Eq. (4.5), we can obtain arbitrary even m order biclique kernel from a standard kernel function κ. For example, the biclique kernel using Gaussian kernel for $m = 4$ is as

$$\begin{aligned}
&\kappa^{(4)}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{t}_1, \mathbf{t}_2) \\
&= \kappa(\mathbf{x}_1, \mathbf{t}_1) + \kappa(\mathbf{x}_1, \mathbf{t}_2) + \kappa(\mathbf{x}_2, \mathbf{t}_1) + \kappa(\mathbf{x}_2, \mathbf{t}_2) \\
&= \exp(-\gamma \|\mathbf{x}_1 - \mathbf{t}_1\|_2^2) + \exp(-\gamma \|\mathbf{x}_1 - \mathbf{t}_2\|_2^2) \\
&\quad + \exp(-\gamma \|\mathbf{x}_2 - \mathbf{t}_1\|_2^2) + \exp(-\gamma \|\mathbf{x}_2 - \mathbf{t}_2\|_2^2).
\end{aligned} \tag{4.6}$$

The biclique kernels are connected to the semi-definite even order tensors, which serves as a theoretical ground of the biclique kernel. For the standard kernel, a gram matrix for a kernel function is equivalent to a semi-definite matrix [Shawe-Taylor and Cristianini, 2004]. This characteristic is one of the theoretical foundations of kernel function. Here, we establish a generalization of this characteristics for the gram tensor $\mathcal{K}$. For this semi-definiteness of tensors, the following theorem for a tensor formed by a biclique kernel holds.

Theorem 4.1. *Given a function* $\kappa^{(r)} : \mathbf{X}^{r/2} \times \mathbf{X}^{r/2} \to \mathbb{R}$ *defined by* $\kappa^{(r)}(\{\mathbf{x}_{i\cdot}\}, \{\mathbf{t}_{l\cdot}\}) = \sum_{\gamma,\nu} \kappa(\mathbf{x}_{i_\gamma}, \mathbf{t}_{l_\nu})$*, where* κ *is a function* $\kappa : \mathbf{X} \times \mathbf{X} \to \mathbb{R}$*, then* κ *can be decomposed as* $\kappa(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$ *if and only if* $\kappa^{(r)}$ *is half-symmetric and has the* m*-order tensor semi-definite property.*

This theorem gives a theoretical foundation of the biclique kernel. Thm. 4.1 shows that a half-symmetric even-order semi-definite tensor and a biclique kernel are equivalent, which is similar to the foundations of the standard kernel function.

4.3.2 Contraction of Biclique Kernel

Despite of the nice property of Thm. 4.1, tensors are practically hard to work with. Many tensor problems of generalized common operations in matrix are NP-hard [Hillar and Lim, 2013], such as computing eigenvalues. This motivates us to explore a practically easy while

theoretical guaranteed way to deal with biclique kernel. This section argues that a contracted matrix of a gram tensor can address this issue.

We consider a contracted matrix $K^{(r)}$ (defined in Eq. (4.3)) of a gram tensor $\mathcal{K}$ of the biclique kernel $\kappa^{(r)}$. We call this contracted matrix $K^{(r)}$ as a *gram matrix* of $\kappa^{(r)}$. In the following, we see this gram matrix is more computationally efficient while equivalent to the original biclique kernel. We first observe the following lemma and corollary by contracting a gram tensor into a gram matrix.

Lemma 4.2. *Assume $\kappa(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle_\kappa$ is a base kernel of the biclique kernel $\kappa^{(r)}$. Let $\phi_i := \phi(\mathbf{x}_i)$, and $\Phi := \sum_{l=1}^{n} \phi_l / n$. The gram matrix $K^{(r)}$ of $\kappa^{(r)}$ is equal to a gram matrix formed by a kernel $\kappa' : X \times X \to \mathbb{R}$ as*

$$\kappa'(\mathbf{x}_i, \mathbf{x}_j) := n^{r-2} \Big\langle \phi_i + \frac{r-2}{2}\Phi, \phi_j + \frac{r-2}{2}\Phi \Big\rangle_\kappa. \tag{4.7}$$

Corollary 4.3. *The gram matrix $K^{(r)}$ is semi-definite.*

From this lemma, we observe that $K^{(r)}$ is more computationally efficient than $\mathcal{K}$ for the following reason. Computing Eq. (4.7), we can rewrite $K^{(r)}$ by using the gram matrix K of the base kernel κ as

$$K^{(r)}_{ij} = n^{r-2} \left(K_{ij} + \frac{r-2}{2n}(\delta_i + \delta_j) + \frac{(r-2)^2}{4n^2}\rho \right) \tag{4.8}$$

where δ_i is the sum of i-th row of K and ρ is a sum of all the elements of K, i.e., $\rho = \sum_{i,j} K_{ij}$. Since we can pre-compute δ_i, δ_j and ρ from K in $O(n^2)$, the overall computational time for $K^{(r)}$ is $O(n^2)$, whereas $O(n^r)$ if we naively form $K^{(r)}$ from the original tensor and Eq. (4.3). Note that if we see $K^{(r)}$ as a graph, its degree matrix is equal to a degree matrix D_v of a hypergraph formed by $\mathcal{K}$. Using this lemma, we obtain the following proposition about equivalence of $\mathcal{K}$ and $K^{(r)}$.

Proposition 4.4. *There exists only one kernel κ' from a biclique kernel $\kappa^{(r)}$. Also, we can compose only one biclique kernel $\kappa^{(r)}$ from a kernel κ' and even-order r.*

This proposition shows that a biclique kernel $\kappa^{(r)}$ and a set of a kernel function κ' and even order m are equivalent. Therefore, Prop. 4.4 is a theoretical guarantee to use a

Algorithm 4 Spectral clustering for hypergraph modeled by generalized kernel.

Input: Data $\mathbf{X}$, κ, and r

Compute K from the base kernel kernel κ from data X.

Construct a gram matrix $K^{(r)}$ of the biclique kernel $\kappa^{(r)}$ from K by using Eq. (4.8).

Compute degree matrix D_v from $K^{(r)}$ and obtain top k-eigenvectors of $D_v^{-1/2} K^{(r)} D_v^{-1/2}$.

Conduct k-means to the obtained top k-eigenvectors

Output: The clustering result.

computationally cheaper gram matrix $K^{(r)}$ instead of a computationally expensive gram tensor $\mathcal{K}$.

4.4 Proposed Algorithm

We propose an algorithm for clustering vector data via modeling as an even r-uniform hypergraph and using hypergraph cut. The overall algorithm is shown in Alg. 4. The core of our algorithm is that we model vector data as a hypergraph by our biclique kernel (Eq. (4.5)) and use hypergraph spectral clustering (Prop. 4.8). To do this efficiently, we firstly compute $K^{(r)}$ using Eq. (4.8) (the first and second step of Alg. 4) and then conduct spectral clustering (the third step). The fourth step uses a simple k-means algorithm for obtained eigenvectors to decide the split points, same as the previous studies [Zhou et al., 2006, Ghoshdastidar and Dukkipati, 2015]. The overall computation time of Alg. 4 is $O(n^3)$, since it takes $O(n^2)$ to compute K as well as $K^{(r)}$, and takes $O(n^3)$ to compute eigenvectors, which is equivalent to the standard graph spectral methods. Alg. 4 is faster than spectral algorithms naively using Eq. (2.97) Zhou et al. [2006], Saito et al. [2018] and Eq. (4.27) Ghoshdastidar and Dukkipati [2015] for a hypergraph formed by $\mathcal{K}$. Both of these cost $O(n^r)$ to compute $\mathcal{K}$ and $K^{(r)}$, while ours takes overall $O(n^3)$. This reduction allows us to model as an arbitrary even r-uniform hypergraphs in a reasonable computation time, e.g., for a 20-uniform hypergraph $O(n^3)$ vs. $O(n^{20})$. Therefore, Alg. 4 is as scalable as the standard graph methods in terms of n, and more scalable than the existing hypergraph methods in terms of m.

4.5 Justification for Biclique Kernel

We discussed the proposed algorithm via biclique kernel. The question is, what are theoretical justifications for Alg. 4? At this point, it seems ad-hoc to model vector data as a hypergraph via biclique kernel for spectral clustering since we do so without any justifications. To justify Alg. 4, next two sections connect Alg. 4 to the weighted kernel k-means and explain Alg. 4

with Gaussian-type biclique kernel from a heat kernel view.

4.5.1 Weighted Kernel k-means and Spectral Clustering

The graph cut and the standard kernel have a connection through a trace maximization problem via weight kernel k-means [Dhillon et al., 2004], as seen in Sec. 2.2. This section explores a similar connection between our biclique kernel and the hypergraph cuts. To do so, we first revisit the connection for the standard case and give an alternative way of connection for *any* kernel, instead of the dot product kernel originally discussed in [Dhillon et al., 2004]. This way is a kernel function approach instead of an explicit feature map approach done in [Dhillon et al., 2004]. We generalize this way of the graph case to our biclique kernel setting. We show that this generalized weighted kernel k-means objective for our biclique kernel is equivalent to the established cut in Prop. 4.8, which we see as a justification of Alg. 4.

4.5.1.1 Revisiting Spectral Connection

This section revisits the claim in [Dhillon et al., 2004] that weighted kernel k-means and graph cuts are connected. We here give an alternative way of connection. This alternative way allows us to handle *any* inner product kernels, while the original in [Dhillon et al., 2004] only assumes the dot product kernel. We define clusters by C_ℓ, a partitioning of points as $\{C_\ell\}_{\ell=1}^k$, and the weighted kernel k-means objective for this as

$$J_\phi(\{C_\ell\}_{\ell=1}^k) := \sum_{\ell\in[k]} \sum_{\mathbf{x}_i\in C_\ell} \theta(\mathbf{x}_i)\|\phi(\mathbf{x}_i) - \mathbf{m}_\ell\|^2, \tag{4.9}$$

where $\mathbf{m}_\ell$ is a weighted mean, which is defined as

$$\mathbf{m}_\ell := \sum_{\mathbf{x}_j\in C_\ell} \frac{\theta(\mathbf{x}_j)\phi(\mathbf{x}_j)}{s_\ell},\ s_\ell := \sum_{\mathbf{x}_j\in C_\ell} \theta(\mathbf{x}_j), \tag{4.10}$$

and $\|\cdot\|$ is a norm induced by *any* inner product forming a kernel function $\kappa(\mathbf{x},\mathbf{y}) = \langle\phi(\mathbf{x}),\phi(\mathbf{y})\rangle$. Let $\kappa_{ij} := \kappa(\mathbf{x}_i,\mathbf{x}_j), \phi_i := \phi(\mathbf{x}_i)$, and $\theta_i := \theta(\mathbf{x}_i)$. Using the kernel κ and its gram matrix K we can rewrite Eq. (4.9) as

$$J_\phi(\{C_\ell\}_{\ell=1}^k) = \sum_{\ell\in[k]} \sum_{i\in C_\ell} \theta_i(\|\phi_i\|^2 - 2\langle\phi_i,\mathbf{m}_\ell\rangle + \|\mathbf{m}_\ell\|^2)$$

$$= \sum_{\ell\in[k]}\sum_{i\in C_\ell}\Big(\theta_i\kappa_{ii} - 2\theta_i\sum_{j\in C_\ell}\frac{\theta_j}{s_\ell}\kappa_{ij} + w_i\sum_{r,t\in C_\ell}\frac{\theta_r\theta_t}{s_\ell^2}\kappa_{rt}\Big)$$

$$= \sum_{\ell\in[k]}\sum_{i\in C_\ell}\theta_i\kappa_{ii} - \sum_{\ell\in[k]}\sum_{r,t\in C_\ell}\frac{\theta_r\theta_t\kappa_{rt}}{s_\ell} \tag{4.11}$$

$$= \mathrm{trace}\Theta^{1/2}K\Theta^{1/2} - \mathrm{trace}Y\Theta^{1/2}K\Theta^{1/2}Y, \tag{4.12}$$

where

$$Y_{i\ell} := \begin{cases} \sqrt{\theta(\mathbf{x}_i)/s_\ell} & (\mathbf{x}_i \in C_\ell) \\ 0 & (\text{otherwise}), \end{cases} \tag{4.13}$$

and Θ is a diagonal matrix whose diagonal element is θ_i. To minimize Eq. (4.12), we want to maximize the second term because the first term is constant w.r.t. the partitioning variable Y. Since $Y^\top Y = I$, maximizing the second term is taking the top k eigenvectors of $\Theta^{1/2}K\Theta^{1/2}$. Taking K as a graph and Θ as inverse of the degree matrix, Eq. (4.12) becomes the relaxed graph cut problem. This gives an alternative way to connect the weighted kernel k-means and the graph cut.

4.5.1.2 Spectral Connection for Multi-way Similarity

This section aims to establish a connection between our formulation of multi-way similarity and the hypergraph cut problem, similarly to the graph one. To do so, we first generalize a weighted kernel k-means for our biclique kernel. Looking at Eq. (4.11), the objective function of weighted kernel k-means uses the kernel function κ directly. Therefore, we consider generalizing by replacing κ in Eq. (4.11) to our biclique kernel. This discussion leads us to define an objective function for weighted kernel k-means for multi-way similarity as follows:

$$J'(\{C_\ell\}_{\ell=1}^k) := \sum_{\ell\in[k]}\sum_{i\in C_\ell}\sum_{\{i.\}\subset C_\ell}\theta_i\kappa^{(r)}(i,i.,i,i.) - \sum_{\ell\in[k]}\sum_{i,j\in\pi_\ell}\sum_{\{i.\},\{j.\}\subset\pi_\ell}\frac{\theta_i\theta_j\kappa^{(r)}(i,i.,j,j.)}{s_\ell}, \tag{4.14}$$

where we write i instead of $\mathbf{x}_i$, and write $i_\cdot$ instead of $\{\mathbf{x}_{i_\cdot}\}$, a set of $r/2-1$ variables. Seeing the way we form the gram matrix $K^{(r)}$ of $\kappa^{(r)}$ (Eq. (4.3)), we can rewrite Eq. (4.14) as

$$\begin{aligned} J'(\{C_\ell\}_{\ell=1}^k) &= \sum_{\ell\in[k]}\sum_{\mathbf{x}\in C_\ell}\theta_i K^{(r)}_{ii} - \sum_{\ell\in[k]}\sum_{i,j\in C_\ell}\frac{\theta_i\theta_j K^{(r)}_{ij}}{s_\ell} \\ &= \mathrm{trace}\Theta^{\frac{1}{2}}K^{(r)}\Theta^{\frac{1}{2}} - \mathrm{trace}Y\Theta^{\frac{1}{2}}K^{(r)}\Theta^{\frac{1}{2}}Y, \end{aligned} \tag{4.15}$$

where Y is defined as Eq. (4.13) and $K^{(r)}$ is a gram matrix of biclique kernel $\kappa^{(r)}$. Similarly to the graph case, Eq. (4.15) can be solved by taking top k eigenvectors of $\Theta^{1/2}K^{(r)}\Theta^{1/2}$.

This discussion draws a connection between hypergraph cut and biclique kernel, and justifies Alg. 4. Recall that a gram matrix $K^{(r)}$ is obtained by a contraction of a gram tensor $\mathcal{K}$. Taking a gram matrix $K^{(r)}$ as a contracted matrix from the adjacency tensor of r-uniform hypergraph and $\Theta = D_v^{-1}$, where D_v is its degree matrix, Eq. (4.15) is equivalent to the hypergraph cut problem (Prop 4.8 and Eq. (2.91)). Thus, the hypergraph cut problem for a hypergraph formed by $\kappa^{(r)}$ is equivalent to the weighted kernel k-means objective function for $\kappa^{(r)}$ (Eq. (4.14)) with a particular weight. This discussion justifies Alg. 4, since Alg. 4 turns out to be equivalent to a generalization of weighted kernel k-means for $\kappa^{(r)}$. Note that since we form $\mathcal{K}$ by $\kappa^{(r)}$, elements of $\mathcal{K}$ can be negative. This contradicts the assumption that all the weight of an edge is positive. However, this can be practically resolved in a way that does not affect topological structures, e.g., by adding the same constant to all the data points. Finally, we remark that we can rewrite Eq. (4.14) as an Eq. (4.9)-style objective function. Let

$$\phi'_i := n^{\frac{r-2}{2}}\left(\phi_i + \frac{r-2}{2}\sum_{i'=1}^n \frac{\phi_{i'}}{n}\right). \tag{4.16}$$

Observing Eq. (4.15), we can rewrite Eq. (4.14) as

$$J'(\{C_\ell\}_{\ell=1}^k) = \sum_{\ell\in[k]}\sum_{i\in C_\ell}\theta_i\|\psi'_i - \mathbf{m}'_\ell\|^2, \tag{4.17}$$

where

$$\mathbf{m}'_\ell := \sum_{j\in C_\ell}\frac{\theta_j\phi'_j}{s_\ell}, \quad s_\ell := \sum_{j\in\pi_\ell}\theta_j$$

4.5.2 Heat Kernels and Spectral Clustering

This section establishes a connection between heat kernel and biclique kernel to justify Alg 4. In the graph case, for a graph made from a gram matrix of Gaussian kernel formed by randomly generated data, the cut of this graph can be seen as an analog of the asymptotic case of an energy minimization problem of the single variable heat equation using Gaussian kernel as a heat kernel [Belkin and Niyogi, 2003]. It is also shown that the graph Laplacian converges to the continuous Laplace operator with infinite number of data points [Belkin and Niyogi, 2005]. We formulate a multivariate heat equation, to which we can similarly connect our biclique kernel. We show that the hypergraph cut problem converges to an asymptotic case of the energy minimization problem of this heat equation using our biclique kernel as heat kernel if the number of data points is infinite.

We define a discrete Laplacian $L^{(r)}_{t,n}$ for $r/2$ variables $\{\mathbf{x}_{i.}\} \in \mathbf{X}^{r/2}$, $\mathbf{X} \subset \mathbb{R}^d$ and a function $f : \mathbf{X}^{r/2} \to \mathbb{R}$ which is "decomposable" by a single variable function f' as $f(\{\mathbf{x}_{i.}\}) = \sum_{\mu=1}^{r/2} f'(\mathbf{x}_{i_\mu})$, $f' : \mathbf{X} \to \mathbb{R}$ as

$$L^{(r)}_{t,n} f(\{\mathbf{x}_{i.}\}) := -\sum_{\{\mathbf{y}_{i.}\}} H^{(r)}_t(\{\mathbf{x}_{i.}\}, \{\mathbf{y}_{i.}\}) f(\{\mathbf{y}_{i.}\}) + \sum_{\{\mathbf{y}_{i.}\}} \frac{H^{(r)}_t(\{\mathbf{x}_{i.}\}, \{\mathbf{y}_{i.}\}) f(\{\mathbf{x}_{i.}\})}{r/2} \tag{4.18}$$

where $H^{(r)}_t$ is a biclique kernel formed as

$$H^{(r)}_t(\{\mathbf{x}_{i.}\}, \{\mathbf{y}_{i.}\}) := \sum_{\gamma,\nu=1}^{r/2,r/2} G_t(\mathbf{x}_{i_\gamma}, \mathbf{y}_{i_\nu}), \text{ where } G_t(\mathbf{x}, \mathbf{y}) := \frac{\exp\left(-\|\mathbf{x}-\mathbf{y}\|^2/4t\right)}{(4\pi t)^{d/2}}.$$

Note that G_t is a Gaussian kernel. Note also that the coefficient $r/2$ in Eq. (4.18) comes from approximation of heat equation. Also, define an energy as

$$S_2(H^{(r)}_t, f) := \sum_{\{\mathbf{x}_{i.}\}, \{\mathbf{y}_{i.}\}} L^{(r)}_{t,n} f(\{\mathbf{x}_{i.}\}) f(\{\mathbf{y}_{i.}\}) \tag{4.19}$$

It is straightforward to think minimizing this energy; however, this leads to the trivial solution

$f = \mathbf{1}$. To avoid this trivial solution, we now consider

$$\min S_2(H_t^{(r)}, f) \text{ s.t. } \|f\|^2 = 1, \langle f, c\mathbf{1}\rangle = 0. \tag{4.20}$$

For this formulation, we claim as follows.

Proposition 4.5. *Minimizing Eq.* (4.20) *is equivalent to the hypergraph normalized cut Eq.* (2.91) *for a hypergraph formed by formed by* $H_t^{(r)}$

We consider to relate discrete operator $L_{t,n}^{(r)}$ to continuous Laplace operator. Let us begin with the Laplace operator. Similarly to the previous sections, if (c) is superscripted that operator is continuous one. Assume a compact differentiable d-dimensional manifold $\mathcal{M}$ isometrically embedded into $\mathbb{R}^N$, a set of $r/2$ variables $\{\mathbf{x}_i\}_{i=1}^{r/2}$, $\mathbf{x}_i \in \mathcal{M}$, abbreviated as $\{x.\}$, and a measure μ. Consider a problem to obtain a function $f : \mathcal{M}^{r/2} \to \mathbb{R}$, such that

$$f = \arg\min S_2^{(c)}(f) \text{ s.t. } \|f\|^2 = 1, \text{ where } S_2^{(c)}(f) := \|\nabla^{(c)} f\|^2, f(\{\mathbf{x}.\}) := \sum_i f'(\mathbf{x}_i), \tag{4.21}$$

and f' is a single variable function $f' : \mathcal{M} \to \mathbb{R}$. From this formulation, the function f in this problem can be described as "decomposable" by f', similarly to Eq. (4.18). In physics analogy, we can recognize $S_2(f)$ as energy, and the problem as an energy minimization problem. This problem often appears where we want to know a profile minimizing energy, such as velocity profile in fluid dynamics [Courant and Hilbert, 1962]. In machine learning, ∇f can be seen to measure how close each data point is when we embed data from a manifold to the Euclidean space. Then, this problem can be thought to find a suitable mapping f best preserving locality, and hence as a clustering algorithm [Belkin and Niyogi, 2003].

By using Stokes' theorem, $\|\nabla^{(c)} f\|^2 = \langle \Delta^{(c)} f, f\rangle$, which rewrites this energy minimization problem as

$$\min(S_2^{(c)}(f) = \langle \Delta^{(c)} f, f\rangle) \text{ s.t. } \|f\|^2 = 1, \langle f, c1\rangle = 0, \tag{4.22}$$

where c is constant. Since Laplace operator $\Delta^{(c)}$ is semi-definite and $\|f\|^2 = 1$ in constraint, the minimizer of Eq. (4.22) is given as an eigenfunction of $\Delta^{(c)} f$. The first eigenfunction is a constant function that maps variables $\mathbf{x}_i \in \mathcal{M}$ to one point. To avoid this, we introduce the

second constraint since the second eigenfunction is orthogonal to the first, which is constant.

We now formulate a multivariate heat equation to analyze Δf. For even r and $r/2$ variables $x_i \in \mathcal{M} \subset \mathbb{R}^d$, consider the following heat equation on a manifold $\mathcal{M}^{r/2}$ as

$$\left(\frac{\partial}{\partial t} + \Delta^{(c)}\right) U(t, \{\mathbf{x}_\cdot\}) = 0,\ U(0, \{\mathbf{x}_\cdot\}) = f(\{\mathbf{x}_\cdot\}),\ \text{where } f(\{\mathbf{x}_{i_\cdot}\}) = \sum_{\mu=1}^{r/2} f'(\mathbf{x}_{i_\mu}). \tag{4.23}$$

and f is "decomposable" in the same sense as Eq. (4.18) and Eq. (4.21). Eq. (4.23) governs an $r/2$ variables system, which evolves by $r/2$ variables interacting with each other but the initial conditions f' only depend on one variable. The solution is given as to satisfy

$$U = \int H_t(\{\mathbf{x}_\cdot\}, \{\mathbf{y}_\cdot\}) U(0, \{\mathbf{y}_\cdot\}) d\mu(\mathbf{y}_*) \quad \text{where} \quad d\mu(\mathbf{y}_*) := \prod_{i=1}^{r/2} d\mu(\mathbf{y}_i) \tag{4.24}$$

and H_t is a *heat kernel*. The $d\mu(\mathbf{y}_*)$ corresponds to the decomposable functions as in Eq. (4.18) and Eq. (4.21). For the heat kernel, a well-known example of heat kernel is Gaussian, which gives a solution to one variable Eq. (4.23) when $\mathcal{M} = \mathbb{R}^n$. However, it is difficult to obtain a concrete form of heat kernel for a general manifold. For details of heat kernel, refer to [Rosenberg and Steven, 1997]. For $H_t^{(r)}$, we claim as follows.

Proposition 4.6. *$H_t^{(r)}$ is a heat kernel.*

From this proposition, we can say that there exists a heat equation on manifolds $\mathcal{M}'$ and $\mathcal{M}''$, where $\mathcal{M}' = \mathcal{M}''^{r/2}$, whose solution is given as Eq. (4.24) using $H_t = H_t^{(r)}$. In the following, we consider the heat equation on this manifold $\mathcal{M}'$.

Using Eq. (4.24), we can relate the energy minimization problem to hypergraph cut and justify Alg. 4. The energy minimization problem Eq.(4.22) in the Euclidean space can be approximated as

$$S_2^{(c)}(f) = \langle \Delta^{(c)} f, f \rangle \approx \frac{1}{t} S_2(H_t^{(r)}, f), \tag{4.25}$$

with proper constraints in Eq. (4.22) (see Appendix 4.F.2 for details). As discussed when we defined discrete Laplacian (Eq. (4.18)), the fourth term $S_2(H_t^{(r)}, f)$ is equivalent to the 2-way hypergraph cut problem using a hypergraph formed by a biclique kernel $H_t^{(r)}$ if properly

treating constraints. Hence, the energy minimization problem Eq.(4.22) can be seen as a continuous analog to the hypergraph spectral clustering. This discussion supports our biclique kernel with Gaussian kernel and Alg. 4, since Alg. 4 with the Gaussian-type biclique kernel can be thought as an approximation of energy minimization problem Eq. (4.22). The key observation is that taking a different m corresponds to taking a different manifold satisfying heat equation Eq. (4.23). This is because the biclique kernel $H^{(r)}$ is a different heat kernel for each m, and each heat kernel has a manifold, on which Eq. (4.23) holds. This key observation gives an intuitive insight; choosing better m corresponds to choosing a manifold $\mathcal{M}'$ to which the given data space $\mathbf{X}$ fits better. We conclude this section by theoretically formulating the above discussion in the following theorem.

Theorem 4.7. *Let $\mathcal{M}' = \mathcal{M}^{r/2}$ be a manifold, on which Eq.* (4.23) *satisfies with solutions using $H_t^{(r)}$. Let the data points $\mathbf{x}_1, \cdots \mathbf{x}_n$ be sampled from a uniform distribution on a manifold $\mathcal{M}$, and $f \in C^\infty(\mathcal{M}')$. Putting $t_n = n^{-1/(2+\alpha)}$, where $\alpha > 0$, there exists a constant C such that*

$$\lim_{n\to\infty} C(nt_n)^{-1} L_{n,t_n}^{(r)} f(\{\mathbf{x}_{i.}\}) = \Delta^{(c)} f(\{\mathbf{x}_{i.}\}) \text{ in probability.}$$

This theorem theoretically supports the discussion in this section; if we have infinite number of data, Eq. (4.18) converges to the continuous Laplace operator and approximation in Eq. (4.25) becomes exact. Note that this theorem is a multivariate version of the result in [Belkin and Niyogi, 2005].

4.5.3 Summary of Generalizations From Graph to Hypergraph

This section relates to the discussion of graph here to the discussion in this chapter. Similarly to Sec. 2.2, we justify our biclique kernel by generalizing spectral connection in the graph in Sec. 4.5.1 and Sec. 4.5.2. We summarize the relationship of spectral connection in graph (Sec. 2.2) and in hypergraph (this chapter) in Table 4.1.

4.6 Related Work

This section reviews the related work of graph and hypergraph modeling.

Justifications of Graph Modeling. There are several approaches for justification of graph modeling via kernel function. Existing work shows the theoretical connection to the

Table 4.1: List of objective functions of r-uniform hypergraph spectral connection and the corresponding pairwise ones. If we model by the kernels as listed, the k-way cut, the weighted kernel k-means with a particular weight, energy minimization problem using Laplace operator are equivalent to the spectral clustering. Details are discussed in the main text.

	Graph (pairwise)	Hypergraph (multi-way)
Kernel	$\kappa(\psi(\mathbf{x}_1), \psi(\mathbf{x}_2)) := \langle \psi(\mathbf{x}_1), \psi(\mathbf{x}_2) \rangle$	$\kappa^{(r)}(\{\mathbf{x}_{i.}\}, \{\mathbf{t}_{i.}\}) := \sum_{\gamma,\nu} \kappa(\mathbf{x}_{i_\gamma}, \mathbf{t}_{j_\nu})$
k-way cut	$\sum_{j=1}^{k} \sum_{i_1 \in V_j, i_2 \in V \setminus V_j} w_{i_1 i_2}$	$\sum_{i=1}^{k} \sum_{e \in E} \sum_{j_1, j_2 \in e; j_1 \in V_j, j_2 \in V_i \setminus V} w(e).$
Spectral clustering	The top k largest eigenvectors of graph adjacency matrix A / gram matrix K	Top k largest eigenvectors of hypergraph adjacency matrix A / gram matrix $K^{(r)}$.
Kernel k-means	$\sum_{j=1}^{k} \sum_{\mathbf{x}_i \in \pi_j} \theta(\mathbf{x}_i) \|\psi(\mathbf{x}_i) - \mathbf{m}_j\|^2$	$\sum_{\mathbf{x}_i \in \pi_j} \theta'(\mathbf{x}_i) \|\psi'(\mathbf{x}_i) - \mathbf{m}'_j\|^2$
Heat Kernel	$\langle \Delta^{(c)} f, f \rangle$, s.t., $\langle f, c1 \rangle$ where f obeys $\left(\frac{\partial}{\partial t} + \Delta^{(c)}\right) U(t,x) = 0,$ $U(0,x) = f(x)$	$\langle \Delta^{(c)} f, f \rangle$, s.t., $\langle f, c1 \rangle$ where f obeys $\left(\frac{\partial}{\partial t} + \Delta^{(c)}\right) U(t, \{x.\}) = 0,$ $U(t, \{x.\}) = f(\{x.\}) = \sum_{i}^{r/2} f'(x_i)$

graph cut from the weighted kernel k-means [Dhillon et al., 2004], energy minimization problem via continuous heat kernel [Belkin and Niyogi, 2003], and kernel PCA [Bengio et al., 2004]. Our approach follows the first two.

Hypergraph Cut for Any Hypergraphs. In Sec. 2.5.2, we summarized the established hypergraph cuts. Recall that a study on hypergraph cut has three approaches. One way is a graph reduction way [Agarwal et al., 2006], which also works for non-uniform hypergraphs. There are two variants of this; star [Zhou et al., 2006] and clique [Rodriguez, 2002, Saito et al., 2018]. The other ways are submodular approach [Hein et al., 2013, Li and Milenkovic, 2018]. Note also that there is another line called inhomogeneous ways [Li and Milenkovic, 2017, Veldt et al., 2020, Liu et al., 2021]. Our approach follows star and clique ways as well as tensor and its graph reduction approach of [Ghoshdastidar and Dukkipati, 2015].

Hypergraph Cut for Uniform Hypergraphs. For uniform hypergraph, there is tensor modeling for uniform hypergraph [Hu and Qi, 2012, Chen et al., 2017, Chang et al., 2020]. For these approach, the eigenvector of the tensors is considered. Furthermore, there is a line of the research where the contraction of the tensors and k-way partitioning problem is considered

[Ghoshdastidar and Dukkipati, 2014, 2015, 2017b], which we refer as *GD*. Slightly changing from GD, we form an adjacency matrix A_g as a contracted matrix of $\mathcal{A}$, i.e.,

$$A_g := \mathcal{A} \times_3 \mathbf{1} \cdots \times_r \mathbf{1}. \tag{4.26}$$

A change from GD is the "position" of mode-$\natural$ products, i.e., GD defines a contraction as $\mathcal{A} \times_3 \mathbf{1} \ldots \times_r \mathbf{1}$. The reason for this change is that we want a contraction of half-undirected hypergraph to be symmetric, and Eq. (4.3) gives the one for half-undirected hypergraph. On the other hand, this change does not affect the result in GD since GD assumes undirected hypergraph and symmetric tensor and hence contraction does not change by the position of mode-$\natural$ products. The clustering algorithm of GD is to solve the eigenproblem as

$$\max \mathrm{trace} Z_N^\top D_v^{-1/2} A_g D_v^{-1/2} Z_N, \mathrm{s.t.} Z_N^\top Z_N = I. \tag{4.27}$$

We here show the connection between these two algorithms through the following proposition.

Proposition 4.8. *For half-symmetric uniform hypergraphs, Eq.* (4.27) *and Eq.* (2.97) *are equivalent.*

Therefore, we may say that we also follow this tensor contraction approach.

Hypergraph Modeling. Comparing to the production of hypergraph cut objectives as above, ways of modeling as hypergraphs have received less attention. There are various studies to model vector data as hypergraphs by heuristic ways [Govindu, 2005, Sun et al., 2017, Yu et al., 2018]. All of these cost $O(n^r)$, comparing $O(n^3)$ to ours. However, to our knowledge, no studies developed a hypergraph cut-based framework to model vector data as hypergraphs.

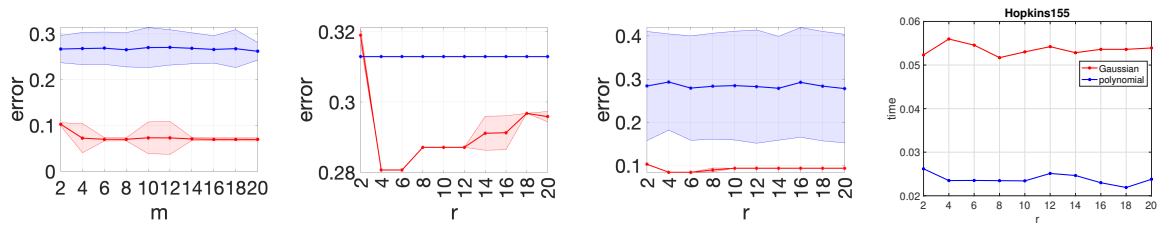
Heat Equations for Hypergraph. We remark that, for hypergraph connection, Whang et al. [2020] considers weighted kernel k-means, but they consider a naive connection between reduced contracted graphs and the standard kernel. Also, Louis [2015] and Ikeda et al. [2018] consider discrete heat equation, which is connected to random walk. However, those three are different to ours since they do not intend to formulate multi-way relationships.

Table 4.2: Dataset Summary. Since Hopkins 155 contains 155 different videos, we report the sum of the data points and average dimensions of videos.

	iris	spine	ovarian	hopkins155
# of class	3	3	2	2/3
size	150	310	216	45850
dimension	4	6	100	89.32

Table 4.3: Experimental results. The standard deviation is from randomness involved in the fourth step of Alg. 4. The kernel for $r = 2$ means that we use the standard kernel. GD stands for the method used in [Ghoshdastidar and Dukkipati, 2015]. Gaussian AS stands for Gaussian formed by affine subspace. Gaussian d^{H-2} is a method discussed in [Li and Milenkovic, 2017]. Polynomial Y stands for a method proposed by [Yu et al., 2018]. Since Hopkins155 is the average performance of 155 datasets, this only shows the average. Details are in the main text.

Kernel and Method	iris	spine	Ovarian	Hopkins155
Gaussian ($r = 2$)	0.1027 ± 0.0033	0.3191 ± 0.0025	0.1315 ± 0.0023	0.1600
Gaussian Ours ($r \geq 4$)	$\mathbf{0.0693} \pm 0.0033$	$\mathbf{0.2807} \pm 0.0000$	$\mathbf{0.0841} \pm 0.0000$	**0.1112**
Gaussian GD	0.0737 ± 0.0318	0.3000 ± 0.0000	0.1806 ± 0.0000	0.1465
Gaussian AS	0.2267 ± 0.0000	0.2839 ± 0.0000	0.1690 ± 0.0023	0.1294
Gaussian d^{H-2}	0.2407 ± 0.0662	0.3195 ± 0.0078	0.3317 ± 0.0892	0.1490
Polynomial ($r = 2$)	0.2922 ± 0.0746	0.3183 ± 0.0295	0.2043 ± 0.0780	0.2278
Polynomial Ours ($r \geq 4$)	0.2719 ± 0.0383	0.3142 ± 0.0452	0.1898 ± 0.0794	0.2258
Polynomial GD	0.4359 ± 0.0546	0.3219 ± 0.0050	0.2817 ± 0.1201	0.2934
Polynomial Y	0.3227 ± 0.0199	0.3828 ± 0.0754	0.4399 ± 0.0093	0.2654



(a) Iris (b) Spine (c) Ovarian (d) Hopkins155

Figure 4.1: Experimental results. Red shows the result for Gaussian and blue shows for polynomial. The shade shows the standard deviation of the fourth step of Alg. 4. Since Hopkins155 is the average performance of 155 datasets, this only shows the average.

4.7 Experiments

This section numerically demonstrates the performance of our Alg. 4 using our formulation of multi-way similarity with biclique kernel.

Objective of the Experiments. These experiments evaluate our modeling by comparing the standard kernel and other heuristic hypergraph modelings. To focus on this purpose, we varied the modelings and kept fixed the cut objective function as Eq. (2.91).

Datasets. Our experiments were performed on classification datasets, iris and spine from the UCI repository, and ovarian cancer data [Petricoin III et al., 2002]. We also used Hopkins155 dataset [Tron and Vidal, 2007], which contains 155 motion segmentation datasets. These type of vector classification datasets were used in the previous studies, such as [Ghoshdastidar and Dukkipati, 2015]. Note that we used vector data since the primary aim of the research in this chapter is to model a hypergraph from vector data. As discussed later, since the experiments need to take $O(n^3)$ time complexity and need to run multiple times on various parameters, we need to restrict ourselves to small to medium datasets. The dataset is summarized in Table 4.2.

Experimental Setting. We used Gaussian kernel ($\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2)$) and polynomial kernel ($\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\sum_i x_i x_j + c)^d$) as a base kernel to form a biclique kernel $\kappa^{(r)}$, and conduct Alg. 4. We used $r = 2, 4..., 20$. For comparison, we employed the following types of modeling. Note that we restrict hypergraph comparison methods to be $m = 3$ to make the comparison fair in terms of computational time. By this, all of the comparisons and ours equally cost $O(n^3)$, which is equivalent to Alg. 4. For the first of the comparison method, we used $r = 2$, the standard graph method, for both kernels as a baseline. Secondly, we used ad-hoc modeling used in the experiment of [Ghoshdastidar and Dukkipati, 2015] for both kernels, which is, $\mathcal{A}_{ijk} := \max(\kappa(\mathbf{x}_i, \mathbf{x}_j), \kappa(\mathbf{x}_j, \mathbf{x}_k), \kappa(\mathbf{x}_k, \mathbf{x}_i))$. Third, we employed Gaussian-type modeling used in various papers such as [Govindu, 2005, Li and Milenkovic, 2017], which is the mean Euclidean distance to the optimal fitted affine subspace. More formally, $\mathcal{A}_{ijk} := \exp(-\gamma\lambda_1)$, where λ_1 is the smallest eigenvalue of $\mathbf{X}_{ijk}^\top \mathbf{X}_{ijk}$, and $\mathbf{X}_{ijk} := (\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$. Fourth, we used Gaussian-type modeling used in [Li and Milenkovic, 2017], which is referred to as d^{H-2}. The d^{H-2} is a Euclidean distance between v and the affine subspace generated by $e/\{i\}$, for all $I \in e$, and sum this up for all $I \in e$. Lastly, for polynomial, we used a generalized dot product form [Yu et al., 2018], which is $\mathcal{A}_{ijk} := \sum_l x_{il} x_{jl} x_{kl}$. Note that all the hypergraph comparison methods work for any uniform hypergraph. We can say that we compare five Gaussian-type methods (ours, baseline, [Ghoshdastidar and Dukkipati, 2015], affine subspace, and d^{H-2}) and four polynomial-type

methods (ours, baseline, [Ghoshdastidar and Dukkipati, 2015], and [Yu et al., 2018]). For the comparisons, we also used the spectral clustering as Eq. (2.97), and conduct the fourth step of Alg. 4. We used a free parameter $\gamma \in \{10^{-3}, 10^{-2}, ..., 10^{5}\}$ for Gaussian, and $d \in \{1, 3, ..., 9\}$ and $c = 0, 1$ for polynomial. Since the fourth step of Alg. 4 involves randomness at k-means, we repeated this step 100 times. We evaluated our performance on error rate, i.e., (# of mis-clustered data points)/(# of data points), same as the previous studies [Zhou et al., 2006, Li and Milenkovic, 2017]. We report average errors and standard deviations caused from the fourth step except for Hopkins155. Since Hopkins155 contains 155 tasks and standard deviations vary by each task, we only report an average error of 155 tasks, similar to the previous studies [Ghoshdastidar and Dukkipati, 2014, 2017b]. Our experimental code is available at github[1]. We want to mention that our experiments was run with Matlab on Mac Mini with Intel i7 Processor and 32GiB RAM.

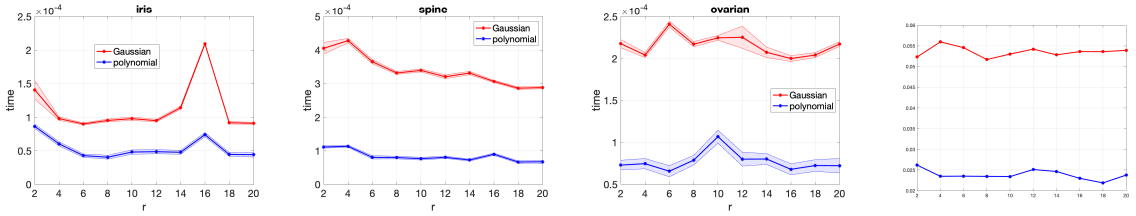
Overall Results. We summarize the results in Table 4.3 and Fig. 4.1. From Table 4.3, we see that ours with Gaussian kernel outperforms the other methods at all the datasets. Ours with polynomial kernel also outperforms other polynomial methods. Note also that the Gaussian kernel is generally better than the polynomial kernel in our experiments; this is expected since the Gaussian kernel is theoretically known to approximate function very well in the standard setting, known as universality, while the polynomial kernel is known to be less expressive than the Gaussian [Micchelli et al., 2006]. Additionally, for most cases in Fig. 4.1, if we increase m, results are improved until a certain point but slightly drop from there. This corresponds to the intuition; multi-way relations could be too "multi" beyond a certain point: Too many relations could work as noise to separate the data. To our knowledge, it is first time to obtain insights on behaviors of higher-order (say, $r \geq 8$) uniform hypergraph on spectral clustering. Moreover, for Gaussian methods, the variance for ours is smaller than one for the others. This means that our methods offer more separated modeling.

Computational Time. Although all the comparisons and ours equally cost $O(n^3)$, we provide the runtime of our experiment in Table 4.4 and Fig. 4.2. Ours are faster than the hypergraph comparisons. This comes from the difference in construction of hypergraphs. All methods have roughly two parts; i) construction of hypergraph and ii) spectral clustering, and ii) costs $O(n^3)$ in all the methods. However, for the construction of hypergraphs, while

[1] `https://github.com/ShotaSAITO/HypergraphModeling`

Table 4.4: Runtime Summary (unit:secs). Here we use E notation, e.g., E-06 $= 10^{-6}$. GD stands for the method used in [Ghoshdastidar and Dukkipati, 2015]. Gaussian AS stands for Gaussian formed by affine subspace. Gaussian d^{H-2} is a method discussed in [Li and Milenkovic, 2017]. Polynomial Y stands for a method proposed by [Yu et al., 2018]. For Hopkins155, we sum up all the computational time, and report the time which produced the best error result summarized in Table 4.3.

Kernel and Method	iris	spine	ovarian	Hopkins155
Gaussian ($r = 2$)	6.99E-05±3.54E-06	4.05E-04±1.11E-09	2.18E-04±7.29E-10	4.82E-02
Gaussian Ours ($r \geq 4$)	6.05E-05±2.90E-06	4.28E-04±1.13E-09	2.41E-04±6.56E-10	5.22E-02
Gaussian GD	1.57E-03±4.44E-06	1.54E-02±3.15E-06	4.31E-03±1.45E-06	3.69E+00
Gaussian AS	9.16E-04±1.06E-05	7.14E-03±5.51E-06	2.70E-02±6.11E-06	4.12E+01
Gaussian (d^{H-2})	6.12E-02±2.54E-06	2.15E-01±1.08E-06	1.01E-01±3.15E-06	1.22E+01
Polynomial ($r = 2$)	3.94E-05±6.98E-06	1.11E-04±2.18E-09	7.29E-05±4.82E-07	2.69E-02
Polynomial Ours ($r \geq 4$)	3.11E-05±3.73E-06	6.73E-05±2.17E-09	1.07E-04±4.96E-05	2.20E-02
Polynomial GD	1.57E-03±4.44E-06	1.54E-02±3.15E-06	4.31E-03±1.45E-06	3.69E+00
Polynomial Y	9.16E-04±1.06E-05	7.14E-03±5.51E-06	2.70E-02±6.11E-06	4.12E+01



(a) Iris (b) Spine (c) Ovarian (d) Hopkins155

Figure 4.2: Runtime for our method. Red shows the result for Gaussian and blue shows for polynomial. The shade shows the standard deviation of the fourth step of Alg. 4. Since Hopkins155 is the sum of runtime of different 155 datasets, this only shows the average.

ours costs $O(n^2)$, the other comparisons cost $O(n^3)$ to construct. This difference induces the time difference. The actual running time of our method does not change the time very much when we increase the order of the hypergraph r. This supports our claim – no matter which r we take, the overall computational time does not depend on r. On the other hand, for the comparison methods, if we increase m we expect the actual running time to increase since the comparisons cost $O(n^r)$ to compute.

4.8 Summary

To conclude, we have provided a hypergraph modeling method, and a faster spectral clustering algorithm ($O(n^3)$ for ours while $O(n^r)$ for existing ones) that is connected to the hypergraph cut problems proposed by [Zhou et al., 2006, Ghoshdastidar and Dukkipati, 2015, Saito et al.,

2018]. A future direction would be to explore other constructions of multi-way similarity which can connect to other uniform and non-uniform hypergraph cuts not having kernel characteristics, such as Laplacian tensor ways [Chen et al., 2017, Chang et al., 2020], total variation and its submodular extension [Hein et al., 2013, Yoshida, 2019]. Also, it would be interesting to study more on connections between this work and a general splitting functions of inhomogeneous cut [Li and Milenkovic, 2017, Chodrow et al., 2021], e.g., to see which class of splitting functions can be connected to the biclique kernel. The limitation of our work is that we cannot apply our formulation to an odd-order uniform hypergraph. The reason for this limitation is that our biclique kernel is equivalent to half-symmetric semi-definite even-order tensor while odd-order semi-definiteness is indefinite as discussed.

Appendices for Chapter 4

In the following sections, we provide the omitted proofs, additional discussions, and additional experimental result for Chapter 4.

4.A Proof of Theorem 4.1

We start with the 'only if' direction. Following the definition of semi-definiteness of tensors,

$$\begin{aligned}
\mathcal{K} \times_1 \mathbf{v} \ldots \times_r \mathbf{v} &= \sum_{i_1 \ldots i_r = 1}^{n} v_{i_1} \ldots v_{i_r} \kappa^{(r)}(\{\mathbf{x}_{i_\mu}\}_{\mu=1}^{r/2}, \{\mathbf{x}_{i_\mu}\}_{\gamma=r/2+1}^{r}) \\
&= \sum_{i_1 \ldots i_r} \sum_{j_1, j_2 = 1}^{r/2} v_{i_1} \ldots v_{i_{r/2}} \kappa(\mathbf{x}_{i_{j_1}}, \mathbf{x}_{i_{i_{r/2+j_2}}}) v_{i_{r/2+1}} \ldots v_{i_r} \\
&= \left\langle \sum_{i_1 \ldots i_{r/2}, j_1} v_{i_1} \ldots v_{i_{r/2}} \phi(\mathbf{x}_{ij_1}), \sum_{i_{r/2+1} \ldots i_r, i_{j_2}} v_{i_{r/2+1}} \ldots v_{i_r} \phi(\mathbf{x}_{i_{r/2+j_2}}) \right\rangle \\
&= \left\| \sum_{i_1 \ldots i_{r/2}, j_1} v_{i_1} \ldots v_{i_{r/2}} \phi(\mathbf{x}_{i_{j_1}}) \right\|^2 \geq 0.
\end{aligned}$$

This shows that a gram tensor $\mathcal{K}$ is semidefinite.

We now move to prove 'if' direction. We construct the space such as

$$\mathscr{F} = \left\{ \sum_{i_1, \ldots, i_{r/2} = 1}^{l} \alpha_{i_1} \alpha_{i_2} \ldots \alpha_{i_{r/2}} \kappa(\{\mathbf{x}_i\}_{i=1}^{r/2}, \cdot) \;\middle|\; l \in \mathbb{N}, \mathbf{x}_i \in X, \alpha_{i.} \in \mathbb{R}, i = 1, \ldots, l \right\}.$$

We emphasize that the element of the set $\mathscr{F}$ is a function that takes $r/2$ arguments. Note that we have used $\cdot$ to indicate the position of the argument of the function. Let the function $f, g \in \mathscr{F}$ as

$$\begin{aligned}
f(\{\mathbf{x}_i\}_{i=1}^{r/2}) &= \sum_{i.=1}^{l} \alpha_{i_1} \alpha_{i_2} \ldots \alpha_{i_{r/2}} \kappa(\{\mathbf{t}_i\}_{i=1}^{r/2}, \{\mathbf{x}_i\}_{i=1}^{r/2}) \\
g(\{\mathbf{x}_i\}_{i=1}^{r/2}) &= \sum_{l.=1}^{n} \beta_{l_1} \beta_{l_2} \ldots \beta_{l_{r/2}} \kappa(\{\mathbf{z}_i\}_{i=1}^{r/2}, \{\mathbf{x}_i\}_{i=1}^{r/2})
\end{aligned}$$

We now introduce inner product $\langle f, g \rangle$ as follows;

$$\begin{aligned}\langle f, g \rangle &:= \sum_{i.=1}^{l} \sum_{j.=1}^{n} \alpha_{i_1} \alpha_{i_2} \dots \alpha_{i_{r/2}} \beta_{l_1} \beta_{l_2} \dots \beta_{l_{r/2}} \kappa(\{\mathbf{t}_i\}_{i=1}^{r/2}, \{\mathbf{z}_i\}_{i=1}^{r/2}) \\ &= \sum_{i.=1}^{l} \alpha_{i_1} \alpha_{i_2} \dots \alpha_{i_{r/2}} g(\{\mathbf{t}_i\}_{i=1}^{r/2}) \\ &= \sum_{l.=1}^{l} \beta_{l_1} \beta_{l_2} \dots \beta_{l_{r/2}} f(\{\mathbf{z}_i\}_{i=1}^{r/2})\end{aligned} \tag{4.28}$$

where the second equation follows from the definition. We remark that since the assumption that κ is semi-definite,

$$\langle f, f \rangle = \sum_{i_1, \dots, i_m} \alpha_{i_1} \dots \alpha_{i_m} \kappa(\{\mathbf{x}_i\}_{i=1}^{r/2}, \{\mathbf{x}_i\}_{i=r/2+1}^{r}) \geq 0.$$

Similarly to the standard kernel, the biclique kernel has reproducing property. The reproducing property follows from Eq. (4.28) if we take $g = \kappa(\{\mathbf{x}_i\}_{i=1}^{r/2}, \cdot)$, and we do the operation defined as Eq. (4.28) on the function f,

$$\begin{aligned}\langle f, \kappa(\{\mathbf{x}_i\}_{i=1}^{r/2}, \cdot) \rangle &= \sum_{i.=1}^{l} \alpha_{i_1} \alpha_{i_2} \dots \alpha_{i_{r/2}} \kappa(\{\mathbf{t}_i\}_{i=1}^{r/2}, \{\mathbf{x}_i\}_{i=1}^{r/2}) \\ &= f(\{\mathbf{x}_i\}_{i=1}^{r/2}).\end{aligned}$$

We call this property as *reproducing property*.

To conclude the proof, it remains to show separability and completeness. Since κ is also kernel, separability follows for the same reasoning as [Shawe-Taylor and Cristianini, 2004]. For completeness, we consider a fixed input $\{\mathbf{x}_i\}_{i=1}^{r/2}$ and a Cauchy sequence $(f_n)_{n=1}^{\infty}$. From the Cauchy-Schwarz inequality, we obtain

$$\begin{aligned}\left(f_n(\{\mathbf{x}_i\}_{i=1}^{r/2}) - f_m(\{\mathbf{x}_i\}_{i=1}^{r/2})\right)^2 &= \left\langle f_n - f_m, \kappa(\{\mathbf{x}_i\}_{i=1}^{r/2}, \cdot) \right\rangle^2 \\ &\leq \|f_n - f_m\|^2 \kappa(\{\mathbf{x}_i\}_{i=1}^{r/2}, \{\mathbf{x}_i\}_{i=1}^{r/2}).\end{aligned}$$

This means that the Cauchy sequence $(f_n)_{n=1}^{\infty}$ is bounded, and has a limit. We define such a

limit

$$g(\{\mathbf{x}_i\}_{i=1}^{r/2}) = \lim_{n\to\infty} f_n(\{\mathbf{x}_i\}_{i=1}^{r/2}),$$

and include all such limit functions in $\mathcal{F}$. Then, we obtain the Hilbert space $\mathcal{F}_\kappa$ associated with kernel κ.

While we so far have the feature space, we need to specify the image of an input $\{\mathbf{x}_i\}_{i=1}^{r/2}$ under the mapping ψ.

$$\psi(\{\mathbf{x}_i\}_{i=1}^{r/2}) = \kappa(\{\mathbf{x}_i\}_{i=1}^{r/2}, \cdot) \in \mathcal{F}_\kappa.$$

Then, inner product between an element of $\mathcal{F}_\kappa$ and the image of an input $\{\mathbf{x}_i\}_{i=1}^{r/2}$ is

$$\langle f, \psi(\{\mathbf{x}_i\}_{i=1}^{r/2})\rangle = \langle f, \kappa(\{\mathbf{x}_i\}_{i=1}^{r/2}, \cdot)\rangle = f(\{\mathbf{x}_i\}_{i=1}^{r/2}).$$

This is what we need. Furthermore, the inner product is strict since if $\|f\| = 0$, then $\forall x$ we have

$$f = \langle f, \psi(\{\mathbf{x}_i\}_{i=1}^{r/2})\rangle \leq \|f\|\|\psi(\{\mathbf{x}_i\}_{i=1}^{r/2})\| = 0.$$

4.B Proof of Lemma 4.2

We begin the proof by computing a gram matrix $K^{(r)}$ of $\kappa^{(r)}$. First, to avoid confusion in proof, we omit the variables $\mathbf{t}_{\cdots}$ in the definition. We rewrite our kernel $\kappa^{(r)}$ for the two sets of $r/2$ variables $\{\mathbf{x}_{i_\cdot}\}$ and $\{\mathbf{x}_{i_{r/2+\cdot}}\}$ as

$$\kappa^{(r)}(\{x_{i_\mu}\}_{\mu=1}^{r/2}, \{x_{i_\mu}\}_{\gamma=r/2+1}^{r}) = \sum_{j_1,j_2=1}^{r/2} \kappa(\mathbf{x}_{i_{j_1}}, \mathbf{x}_{i_{i_{r/2+j_2}}}), \tag{4.29}$$

instead of $\{\mathbf{x}_{i_\cdot}\}$ and $\{\mathbf{t}_{l_\cdot}\}$ in the original definition Eq. (4.5). This writing change does not change the definition, but just rewrites the variables.

To save the space, we introduce the abbreviation as

$$\phi_i := \phi(\mathbf{x}_i)$$

$$\phi_{ij} := \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle.$$

Let $\kappa^{(r)}$ for two $r/2$ variables $\{\mathbf{x}_i, \mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{r/2-1}}\}$, $\{\mathbf{x}_j, \mathbf{x}_{x_{r/2}}, \ldots, \mathbf{x}_{r-2}\}$ For $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, we can compute the gram matrix as

$$\begin{aligned}
&K_{ij}^{(r)} \\
&= \sum_{i_\cdot, j_\cdot} \kappa^{(r)}(\{\mathbf{x}_i\}, \cup\{\mathbf{x}_{i_\cdot}\}, \{\mathbf{x}_j\} \cup \{\mathbf{x}_{r/2+j_\cdot}\}) && (4.30) \\
&= \sum_{i_\cdot, j_\cdot} \psi_{ij} + \underbrace{\phi_{ij_1} + \ldots + \phi_{ij_{r/2-1}}}_{r/2-1 \text{ terms}} + \underbrace{\phi_{ji_1} + \ldots + \phi_{ji_{r/2-1}}}_{r/2-1 \text{ terms}} + \underbrace{\phi_{i_1 j_1} + \phi_{i_1 j_2} + \ldots + \phi_{i_{r/2-1} j_{r/2-1}}}_{(r/2-1)\times(r/2-1) \text{ terms}} && (4.31) \\
&= n^{r-2}\phi_{ij} + n^{r-3}\frac{r-2}{2}\sum_{l=1}^{n}(\phi_{il} + \phi_{jl}) + n^{m-4}\left(\frac{r-2}{2}\right)^2 \sum_{l,k=1}^{n} \phi_{lk} && (4.32) \\
&= n^{r-2}\langle \phi_i + \frac{r-2}{2}\sum_{l=1}^{n}\frac{\phi_l}{n}, \phi_j + \frac{r-2}{2}\sum_{l=1}^{n}\frac{\phi_l}{n} \rangle. && (4.33)
\end{aligned}$$

Note that $i_\cdot$ and $j_\cdot$ runs from 1 to n. Then,

$$\sum_{i_\cdot, j_\cdot} \phi_{ij_r} = n^{r-3}\sum_{l} \phi_{il} \tag{4.34}$$

$$\sum_{i_\cdot, j_\cdot} \phi_{ji_r} = n^{r-3}\sum_{l} \phi_{jl}, \tag{4.35}$$

for all $r = 1, \ldots, r/2 - 1$,

$$\sum_{i_\cdot, j_\cdot} \phi_{i_r j_s} = n^{r-4}\sum_{l,k} \phi_{lk} \tag{4.36}$$

for all $r, s = 1, \ldots, r/2 - 1$, and

$$\sum_{i_\cdot, j_\cdot} \phi_{ij} = n^{r-2}\phi_{ij} \tag{4.37}$$

Using this we obtain Eq. (4.32). By Eq. (4.33), we now prove Lemma. 4.2. We remark that Eq. (4.32) is equivalent to Eq. (4.8), since $\phi_{ij} = K_{ij}$ by definition. Since $\kappa^{'(r)}$ is kernel by Eq. (4.33), $K^{(r)}$ is semi-definite, which concludes Cor. 4.3.

4.C Proof of Proposition 4.4

Actually, Prop. 4.4 follows directly from the proof of Lemma 4.2. We apply Lemma 4.2 to prove Prop. 4.4.

It is clear that there exists unique $\kappa^{'(r)}$ from $\kappa^{(r)}$, by seeing the way of composition. We move on to show that there exists unique $\kappa^{(r)}$ from $\kappa^{'(r)}$ and m. Since $K^{(r)}$ is a kernel by Lemma 4.2, this concludes that $K^{(r)}$ is semi-definite. For a semi-definite matrix for $A^{(r)}$, we have a decomposition ψ' by the standard Mercer's theorem. Then, we have

$$\phi'_i = n^{\frac{r-2}{2}}\phi_i + n^{\frac{r-2}{4}}\frac{r-2}{2}\sum_{l=1}^{n}\phi_l,$$

where ψ is a feature map for biclique kernel for $\mathcal{A}$. By the construction, we can rewrite ψ by ψ' by solving the linear equation as

$$\phi' = C\phi,$$

where $\phi' := (\phi'_1, \ldots, \phi'_n)^\top$, $\phi := (\phi_1, \ldots, \phi_n)^\top$, and

$$C := n^{\frac{r-4}{2}}\begin{pmatrix} n+\frac{r-2}{2} & \frac{r-2}{2} & \cdots & \frac{r-2}{2} \\ \frac{r-2}{2} & n+\frac{r-2}{2} & \cdots & \frac{r-2}{2} \\ \vdots & & \ddots & \vdots \\ \frac{r-2}{2} & \cdots & \frac{r-2}{2} & n+\frac{r-2}{2} \end{pmatrix}.$$

By construction, C is a full rank matrix, and therefore we can write as

$$\phi = C^{-1}\phi'. \tag{4.38}$$

Therefore, this concludes the proof.

4.D Proof of Proposition 4.5

We start to define a matrix $L^{(r)}$ to satisfy

$$L^{(r)}f'(\mathbf{x}_1) = \sum_{\{\mathbf{x}_{i.}\}_{\mu=2}^{r/2}} L^{(r)}_{t,n} f(\{\mathbf{x}_{i.}\}). \tag{4.39}$$

Then, we can rewrite the minimization problem as

$$\begin{aligned} S_2^{(r)}(f) &:= \sum_{\{\mathbf{x}_{i.}\},\{\mathbf{y}_{i.}\}} L_{t,n}^{(r)} f(\{\mathbf{x}_{i.}\}) f(\{\mathbf{y}_{i.}\}) \\ &= \sum_{\mathbf{x},\mathbf{y}} L^{(r)} f'(\mathbf{x}) f'(\mathbf{y}) \end{aligned} \tag{4.40}$$

Let A be a contracted matrix from a gram tensor $\mathcal{H}$ for the biclique kernel $H_t^{(r)}$, and D be a degree matrix of A. By construction of L^m, we can rewrite

$$L^{(r)} = \frac{D}{r/2} - A.$$

Now we consider to introduce normalizing constrains. This is justified since the continuous counterpart of energy minimization problem Eq. (4.22) we can introduce such a constraints by properly choosing the measure μ for inner product. Now, we consider Eq. (4.40). Introducing the normalizing constraints, we write as

$$\begin{aligned} \min S_2^{(r)}(f) &= \min f^\top L^{(r)} f \\ &= \min f^\top D^{-1/2} \left(\frac{D}{r/2} - A \right) D^{-1/2} f \\ &= \max f^\top D^{-1/2} A D^{-1/2} f, \text{ s.t. } f^\top f = 1. \end{aligned}$$

This corresponds to the hypergraph cut problem for the hypergraph formed by $H_t^{(r)}$.

4.E Proof of Proposition 4.6

Since H is a heat kernel $H : \mathcal{M} \times \mathcal{M} \times (0, \infty) \to \mathbb{R}$, we can decompose into

$$H_t(\mathbf{x}, \mathbf{y}) = \sum_i \exp(-\lambda_i t) \phi_i(\mathbf{x}) \phi_i(\mathbf{y}) \tag{4.41}$$

Using the fact that H_t is a heat kernel, we can prove that the kernel $H_t^{(r)}$ is a heat kernel $\mathcal{M}^{r/2} \times \mathcal{M}^{r/2} \times (0, \infty) \to \mathbb{R}$, since we can rewrite $H_t^{(r)}$ as

$$H_t^{(r)}(\{\mathbf{x}.\}, \{\mathbf{y}.\}) = \sum_i \exp(-\lambda_i t)(\phi_i(\mathbf{x}_1) + \dots \phi_i(\mathbf{x}_{r/2}))(\phi_i(\mathbf{y}_1) + \dots \phi_i(\mathbf{y}_{r/2})). \tag{4.42}$$

Similarly to Eq. (4.42), this heat kernel $H^{'(r)}$ is also a heat kernel since we can rewrite this as

$$\begin{aligned} &H^{'(r)}(\mathbf{x},\mathbf{y}) \\ &= \sum_i \exp(-\lambda_i t)\left(\phi(x) + (r/2-1)\int_{\mathcal{M}'}\phi(\mathbf{x})d\mathbf{x}\right)\left(\phi(\mathbf{y}) + (r/2-1)\int_{\mathcal{M}'}\phi(\mathbf{y})d\mathbf{y}\right). \end{aligned} \tag{4.43}$$

4.F Proof of Theorem 4.7

This section provides the proof of Thm. 4.7. Also, we offer the differnt approach of the theorem by directly approximating Eq. (4.25).

4.F.1 Main Proof

The strategy to prove Thm. 4.7 is using Hoeffding's inequality. We start by reviewing Hoeffding's inequality.

Lemma 4.9 (Hoeffding)**.** *Let $X_1, \ldots, X_n$ be independent identically distributed random variables, such that $|X_i| \leq K$. Then*

$$P\left(\left|\sum_i \frac{X_i}{n} - E(X_i)\right| > \epsilon\right) < 2\exp\left(-\frac{\epsilon^2 n}{2K^2}\right) \tag{4.44}$$

To prove the theorem for $L_{tn}^{(r)}$, we evaluate the equation in Eq. (4.18). We define the operator $L_t^{(r)} : L^2(\mathcal{M}) \to L^2(\mathcal{M})$ as

$$\begin{aligned} L_t^{(r)} f(\{\mathbf{x}_{i.}\}) := &\int_{\mathcal{M}'} d\mu(y_*)\frac{H_t^{(r)}(\{\mathbf{x}_{i.}\},\{\mathbf{y}_{i.}\})}{r/2} f(\{\mathbf{x}_{i.}\} \\ &- \int_{\mathcal{M}'} d\mu(y_*) H_t^{(r)}(\{\mathbf{x}_{i.}\},\{\mathbf{y}_{i.}\}) f(\{\mathbf{y}_{i.}\}) \end{aligned} \tag{4.45}$$

We remark that $L_t^{(r)}$ is the empirical average of n independent random variables with the expectation

$$E(L_{nt}^{(r)} f(\{\mathbf{x}_{i.}\})) = L_t^{(r)} f(\{\mathbf{x}_{i.}\}).$$

Applying Hoeffding inequality (Lemma 4.9), we obtain

$$P\left(\left|\frac{1}{t}\frac{L_n^t f(\{\mathbf{x}_{i.}\})}{n} - L_t^{(r)} f(\{\mathbf{x}_{i.}\})\right| > \epsilon\right) \leq \exp\left(-\frac{\epsilon^2 n t^2}{2}\right).$$

If we choose t as a function of n, letting $t = t_n = n^{-1/(2+\alpha)}$ where $\alpha > 0$ to the equation Thm. 4.7, we can obtain for any $\epsilon > 0$,

$$\begin{aligned}\lim_{n\to\infty} P\left(\left|\frac{1}{t_n}\frac{L_{nt}^{(r)}}{n} f(\{\mathbf{x}_{i.}\}) - L_t^{(r)} f(\{\mathbf{x}_{i.}\})\right| > \epsilon\right) &\leq \lim_{n\to\infty} \exp\left(-\frac{\epsilon^2 n t_n^2}{2}\right) \\ &\leq \lim_{n\to\infty} \exp\left(-\frac{\epsilon^2 n (n^{-\frac{1}{2+\alpha}})^2}{2}\right) \\ &= 0.\end{aligned}$$

With the discussion in the proof of Prop. 4.10, we can see that

$$\lim_t L_t^{(r)} f(\{\mathbf{x}_{i.}\}) = \Delta^{(c)} f(\{\mathbf{x}_{i.}\}).$$

If $n \to \infty$ then, $t \to 0$. The above discussion in all leads the conclusion,

$$\lim_{n\to\infty} \frac{C}{n t_n} L_n^{t_n} f(\{\mathbf{x}_{.}\}) = \Delta^{(c)} f(\{\mathbf{x}_{.}\}).$$

4.F.2 Detailed Steps of Approximation Eq. (4.25)

Although Thm. 4.7 justifies the approximation, this section endevors to directly approximate asymptotic heat equation Eq. (4.25). In this subsection about approximation, we approximate $\mathcal{M}' \approx \mathbb{R}^n$, and $d\mu(y_.) \approx dy$, similar to [Belkin and Niyogi, 2003].

We start with expanding the heat equation as done in the main text.

$$\begin{aligned}\lim_{t\to 0} \Delta^{(c)} U(t, \{\mathbf{x}_i\}_{i=1}^{r/2}) &= \Delta^{(c)} f(\{\mathbf{x}_i\}_{i=1}^{r/2}) \\ &= -\lim_{t\to 0} \frac{\partial}{\partial t} \int_{\mathcal{M}'} d\mu(\mathbf{y}_*) H_t(\{\mathbf{x}_i\}_{i=1}^{r/2}, \{\mathbf{y}_.\}) U(0, \{\mathbf{y}_.\})\end{aligned}$$

Therefore, for small t, we obtain

$$
\begin{aligned}
&-\lim_{t\to 0}\frac{\partial}{\partial t}\int_{\mathcal{M}'} d\mu(\mathbf{y}_*)H_t(\{\mathbf{x}_i\}_{i=1}^{r/2},\{\mathbf{y}_\cdot\})U(0,\{\mathbf{y}_\cdot\}) \\
&\approx -\frac{1}{t}\left(\int_{\mathcal{M}'} d\mu(\mathbf{y}_*)H_t(\{\mathbf{x}_i\}_{i=1}^{r/2},\{\mathbf{y}_\cdot\})f(\{\mathbf{y}_\cdot\}) - \int_{\mathcal{M}'} d\mu(\mathbf{y}_*)H_0(\{\mathbf{x}_i\}_{i=1}^{r/2},\{\mathbf{y}_\cdot\})f(\{\mathbf{y}_\cdot\})\right),
\end{aligned}
\tag{4.46}
$$

following from the definition of partial differentiation.

In the following, we consider $H_t = H_t^{(r)}$ since we are interested in this case. Note that the heat kernel $H_t^{(r)}$ is a biclique kernel whose base kernel is Gaussian kernel G_t. To further consider Eq. (4.46), we next examine a solution to Eq. (4.23) in the asymptotic case $t \to 0$. This allows us to have Δf, which we want to analyze in Eq. (4.22). We provide an analysis on the solution to Eq. (4.23) when $t \to 0$.

Proposition 4.10. *In the Euclidean space, i.e., $\mathcal{M}' = \mathbb{R}^n$ and $d\mu(\mathbf{y}_i) = d\mathbf{y}_i$, a given a function $f : \mathcal{M}' \to \mathbb{R}$ and the constraints in Eq.* (4.22)*, then*

$$
\lim_{t\to 0}\int_{\mathcal{M}'} d\mu(\mathbf{y}_*)H_t^{(r)}(\{\mathbf{x}_\cdot\},\{\mathbf{y}_\cdot\})f(\{\mathbf{y}_\cdot\}) = \lim_{t\to 0}\int_{\mathcal{M}'} d\mu(\mathbf{y}_*)\frac{H_t^{(r)}(\{\mathbf{x}_\cdot\},\{\mathbf{y}_\cdot\})}{r/2}f(\{\mathbf{x}_\cdot\})
\tag{4.47}
$$

This proposition is a generalized version of Gaussian kernel features in the following sense. For a single variable Gaussian Kernel, we have

$$
\lim_{t\to 0}\int_{\mathcal{M}'} G_t(\mathbf{x},\mathbf{y})f(\mathbf{y})d\mu(\mathbf{y}) = f(\mathbf{x}), \tag{4.48}
$$

$$
\lim_{t\to 0}\int_{\mathcal{M}'} G_t(\mathbf{x},\mathbf{y})d\mu(\mathbf{y}) = 1. \tag{4.49}
$$

Combining these two, we obtain

$$
\begin{aligned}
\lim_{t\to 0}\int_{\mathcal{M}} G_t(\mathbf{x},\mathbf{y})f(\mathbf{y})d\mu(\mathbf{y}) &= f(\mathbf{x}) \\
&= \lim_{t\to 0}\int_{\mathcal{M}} G_t(\mathbf{x},\mathbf{y})f(\mathbf{x})d\mu(\mathbf{y}).
\end{aligned}
\tag{4.50}
$$

Note the difference between the variables of $\mathbf{y}$ of f on the left-hand side and $\mathbf{x}$ of f on the right-hand side. Prop. 4.10 is a generalized version of this relationship Eq. (4.50).

Using this relation Eq. (4.50), for small t, we can further rewrite as

$$\begin{aligned}\int_{\mathcal{M}} G_{t\to 0}(\mathbf{x},\mathbf{y})f(\mathbf{y})d\mu(\mathbf{y}) &= \lim_{t\to 0}\int_{\mathcal{M}} G_t(\mathbf{x},\mathbf{y})f(\mathbf{y})d\mu(\mathbf{y}) \\ &= f(\mathbf{x}) \\ &= \lim_{t\to 0}\int_{\mathcal{M}} G_t(\mathbf{x},\mathbf{y})f(\mathbf{x})d\mu(\mathbf{y}) \\ &= \int_{\mathcal{M}} G_{t\to 0}(\mathbf{x},\mathbf{y})f(\mathbf{x})d\mu(\mathbf{y}) \\ &\approx \int_{\mathcal{M}} G_t(\mathbf{x},\mathbf{y})f(\mathbf{x})d\mu(\mathbf{y}).\end{aligned}$$

It is too rough to approximate the left-hand side in the form of the right-hand side, i.e.,

$$\lim_{t\to 0}\int_{\mathcal{M}} G_t(\mathbf{x},\mathbf{y})f(\mathbf{y})d\mu(\mathbf{y}) \not\approx \int_{\mathcal{M}} G_t(\mathbf{x},\mathbf{y})f(\mathbf{y})d\mu(\mathbf{y})$$

The reason is that since we have the variable y, which we take integral in both the Gaussian and the function, we cannot see the approximated shape if we increase the value of t, even if t is very small. The right-hand side overcomes this problem. Using this relation, a single variable version of Eq. (4.46) is further approximated.

Now, using the same strategy for Eq. (4.47) in Prop. 4.10, we have an approximation for small t as

$$\begin{aligned}\int_{\mathcal{M}'} d\mu(\mathbf{y}_*)H^{(r)}_{t\to 0}(\{\mathbf{x}_\cdot\},\{\mathbf{y}_\cdot\})f(\{\mathbf{y}_\cdot\}) &= \lim_{t\to 0}\int_{\mathcal{M}'} d\mu(\mathbf{y}_*)H^{(r)}_{t}(\{\mathbf{x}_\cdot\},\{\mathbf{y}_\cdot\})f(\{\mathbf{y}_\cdot\}) \\ &= \lim_{t\to 0}\int_{\mathcal{M}'} d\mu(\mathbf{y}_*)\frac{H^{(r)}_{t}(\{\mathbf{x}_\cdot\},\{\mathbf{y}_\cdot\})}{r/2}f(\{\mathbf{x}_\cdot\}) \\ &= \int_{\mathcal{M}'} d\mu(\mathbf{y}_*)\frac{H^{(r)}_{t\to 0}(\{\mathbf{x}_\cdot\},\{\mathbf{y}_\cdot\})}{r/2}f(\{\mathbf{x}_\cdot\}) \\ &\approx \int_{\mathcal{M}'} d\mu(\mathbf{y}_*)\frac{H^{(r)}_{t}(\{\mathbf{x}_\cdot\},\{\mathbf{y}_\cdot\})}{r/2}f(\{\mathbf{x}_\cdot\}) \qquad (4.51)\end{aligned}$$

This relation further rewrites Eq. (4.46), as

$$\begin{aligned}&-\lim_{t\to 0}\frac{\partial}{\partial t}\int_{\mathcal{M}'} d\mu(\mathbf{y}_*)H_t(\{\mathbf{x}_i\}_{i=1}^{r/2},\{\mathbf{y}_\cdot\})U(0,\{\mathbf{y}_\cdot\}) \\ &\approx -\frac{1}{t}\left(\int_{\mathcal{M}'} d\mu(\mathbf{y}_*)H_t(\{\mathbf{x}_i\}_{i=1}^{r/2},\{\mathbf{y}_\cdot\})f(\{\mathbf{y}_\cdot\}) - \int_{\mathcal{M}'} d\mu(\mathbf{y}_*)H_0(\{\mathbf{x}_i\}_{i=1}^{r/2},\{\mathbf{y}_\cdot\})f(\{\mathbf{y}_\cdot\})\right)\end{aligned}$$

$$\approx -\frac{1}{t}\left(\int_{\mathcal{M}'} d\mu(\mathbf{y}_*) H_t(\{\mathbf{x}_\cdot\},\{\mathbf{y}_\cdot\}) f(\{\mathbf{y}_\cdot\}) - \int_{\mathcal{M}'} d\mu(\mathbf{y}_*) \frac{H_t^{(r)}(\{\mathbf{x}_\cdot\},\{\mathbf{y}_\cdot\})}{r/2} f(\{\mathbf{x}_\cdot\})\right). \tag{4.52}$$

To obtain the third approximation, we apply Eq. (4.51) to the second term of the second equation. We also use the fact that $\{\mathbf{x}_\cdot\}$ is an abbreviated form of $r/2$ continuous variables $\{\mathbf{x}_i\}_{i=1}^{r/2}$.

Consider discrete data points in $\mathcal{M}'$ instead of the continuous variables, Eq. (4.46), which is approximated as Eq. (4.52), is further approximated by

$$\frac{1}{t}\left(-\sum_{\{\mathbf{y}_{i\cdot}\}} H_t^{(r)}(\{\mathbf{x}_{i\cdot}\},\{\mathbf{y}_{i\cdot}\}) f(\{\mathbf{y}_{i\cdot}\}) + \sum_{\{\mathbf{y}_{i\cdot}\}} \frac{H_t^{(r)}(\{\mathbf{x}_{i\cdot}\},\{\mathbf{y}_{i\cdot}\})}{r/2} f(\{\mathbf{x}_{i\cdot}\})\right) = \frac{1}{t} L_{t,n}^{(r)} f(\{\mathbf{x}_{i\cdot}\})$$

Thus, the Laplacian Eq. (4.18) can be seen as a discrete approximation of the continuous Laplacian for $r/2$ variables heat equation Eq. (4.23). We also see that Prop. 4.10 introduces the coefficient $r/2$ in Eq. (4.18).

Finally, we remark that all the approximation here is justified by Thm. 4.7.

4.F.3 Proof of Proposition 4.10

This section provides a proof of Prop. 4.10.

Before we proceed, we review the one variable case. Using single variable Gaussian kernel characteristics Eq. (4.48) and Eq. (4.49), we compute for multivariate version of Eq. (4.48) as

$$\begin{aligned}
&\lim_{t\to 0} \int_{\mathcal{M}'^{r/2}} H_t^{(r)}(\{\mathbf{x}_i\}_{i=1}^{r/2},\{\mathbf{y}_i\}_{i=1}^{r/2}) f(\mathbf{y}) d\mu(\mathbf{y}_*) \\
&= \lim_{t\to 0} \int_{\mathcal{M}'^{r/2}} \sum_{i=1}^{r/2}\sum_{j=1}^{r/2} G_t(\mathbf{x}_i,\mathbf{y}_i) \sum_{i=1}^{r/2} f'(\mathbf{y}_i) d\mu(\mathbf{y}_*) \qquad (4.53) \\
&= \mathrm{vol}(\mathcal{M})^{r/2-1} \left(\frac{m}{2}\right)^2 \sum_{i=1}^{r/2} f'(\mathbf{y}_i), \qquad (4.54)
\end{aligned}$$

and for multivariate version of Eq. (4.49) as

$$\lim_{t\to 0}\int_{\mathcal{M}'^{r/2}} H_t^{(r)}(\{\mathbf{x}_i\}_{i=1}^{r/2},\{\mathbf{y}_i\}_{i=1}^{r/2})d\mu(\mathbf{y}_*) = \lim_{t\to 0}\int_{\mathcal{M}'^{r/2}}\sum_{i=1}^{r/2}\sum_{j=1}^{r/2} G_t(\mathbf{x}_i,\mathbf{y}_i)d\mu(\mathbf{y}_*)$$
$$= \mathrm{vol}(\mathcal{M})^{r/2-1}\left(\frac{m}{2}\right)^2.$$

We now prepare to expand the left hand side of Eq. (4.47). The left hand side of Eq. (4.47) is expanded as

$$\lim_{t\to 0}\int_{\mathcal{M}'} d\mu(\mathbf{y}_*)H_t(\{\mathbf{x}_i\}_{i=1}^{r/2},\{\mathbf{y}_\cdot\})f(\{\mathbf{y}_\cdot\}) \tag{4.55}$$
$$= \lim_{t\to 0}\sum_{i=1}^{r/2}\left(\int_{\mathcal{M}'^{r/2}} H_t^{(r)}(\{\mathbf{x}_i\}_{i=1}^{r/2},\{\mathbf{y}_i\}_{i=1}^{r/2})f'(\mathbf{y}_i)d\mu(\mathbf{y}_*)\right) \tag{4.56}$$

We further compute the inside of the parenthesis in Eq. (4.56) as

$$\lim_{t\to 0}\int_{\mathcal{M}'^{r/2}} H_t^{(r)}(\{\mathbf{x}_i\}_{i=1}^{r/2},\{\mathbf{y}_i\}_{i=1}^{r/2})f'(\mathbf{y}_i)d\mu(\mathbf{y}_*) \tag{4.57}$$
$$= \lim_{t\to 0}\sum_{j=1}^{r/2}\left(\int_{\mathcal{M}'^{r/2}} G_t(\mathbf{x}_j,\mathbf{y}_i)f'(\mathbf{y}_i)d\mu(\mathbf{y}_*) + \sum_{l\neq i}\int_{\mathcal{M}'^{r/2}} G_t(\mathbf{x}_j,\mathbf{y}_l)f'(\mathbf{y}_i)d\mu(\mathbf{y}_*)\right) \tag{4.58}$$
$$= \sum_{j=1}^{r/2}\left(\mathrm{vol}(\mathcal{M})^{r/2-1}f'(\mathbf{x}_j) + \frac{r/2-1}{2}\mathrm{vol}(\mathcal{M})^{r/2-1}\int_{\mathcal{M}} d\mu(\mathbf{y}_i)f'(\mathbf{y}_i)\right) \tag{4.59}$$
$$= \mathrm{vol}(\mathcal{M})^{r/2-1}\sum_{j=1}^{r/2} f'(\mathbf{x}_j) + \frac{m}{2}\frac{r/2-1}{2}\mathrm{vol}(\mathcal{M})^{r/2-1}\int_{\mathcal{M}}\mu(\mathbf{y}_i)f'(\mathbf{y}_i) \tag{4.60}$$

Putting this into Eq (4.56), we obtain

$$\lim_{t\to 0}\sum_{i=1}^{r/2}\left(\int_{\mathcal{M}'^{r/2}} H_t^{(r)}(\{\mathbf{x}_i\}_{i=1}^{r/2},\{\mathbf{y}_i\}_{i=1}^{r/2})f'(\mathbf{y}_i)d\mu(\mathbf{y}_*)\right)$$
$$= \sum_{i=1}^{r/2}\left(\mathrm{vol}(\mathcal{M})^{r/2-1}\sum_{j=1}^{r/2} f'(\mathbf{x}_j) + \frac{m}{2}\frac{r/2-1}{2}\mathrm{vol}(\mathcal{M})^{r/2-1}\int_{\mathcal{M}} d\mu(\mathbf{y}_i)f'(\mathbf{y}_i)\right)$$
$$= \frac{m}{2}\mathrm{vol}(\mathcal{M})^{r/2-1}\sum_{i=1}^{r/2} f'(x_i) + \frac{m}{2}\frac{r/2-1}{2}\mathrm{vol}(\mathcal{M})^{r/2-1}\sum_{i=1}^{r/2}\int_{\mathcal{M}} d\mu(\mathbf{y}_i)f'(\mathbf{y}_i)$$

$$= \lim_{t\to 0} \int_{\mathcal{M}'} d\mu(\mathbf{y}_*) \frac{H_t^{(r)}(\{\mathbf{x}_i\}_{i=1}^{r/2}, \{\mathbf{y}_i\}_{i=1}^{r/2})}{r/2} f(\{\mathbf{x}_i\}_{i=1}^{r/2}). \tag{4.61}$$

The last equality follows from that due to Eq. (4.54) we can proceed the first term as this, and that the second term vanishes due to the constraint $\langle f, c\mathbf{1}\rangle = 0$, i.e.,

$$\begin{aligned}
0 &= \langle f, c\mathbf{1}\rangle \\
&= c \int_{\mathcal{M}^{r/2}} d\mu(\mathbf{y}_*) f(\{\mathbf{y}_\cdot\}) \\
&= c \int_{\mathcal{M}^{r/2}} \prod_{i=1}^{r/2} d\mu(\mathbf{y}_i) \left(\sum_{i=1}^{r/2} f'(\mathbf{y}_i) \right) \\
&= c \sum_{i=1}^{r/2} \mathrm{vol}(\mathcal{M})^{r/2-1} \int_{\mathcal{M}} d\mu(\mathbf{y}_i) f'(\mathbf{y}_i).
\end{aligned}$$

Eq. (4.61) concludes the proof.

4.G Proof of Proposition 4.8

By definition of the star adjacency matrix, the matrix can be computed

$$(A_s)_{ij} = \sum_{e\in E; i,j\in e} \frac{a_{ij}}{r}$$

Considering the order of edges, this would be

$$\begin{aligned}
(A_s)_{ij} &= \sum_{e\in E; i,j\in e} \frac{a_{ij}}{r} \\
&= \sum_{e\in E; i,j\in e} \left(\frac{r}{2}\right)^2 w(e = i, \cdots, j, \cdots)
\end{aligned} \tag{4.62}$$

Eq. (4.62) is shown to be equivalent to A_g/r. Therefore, the problems Eq. (4.27) and Eq. (2.97) are equivalent.

Chapter 5

Multi-Class Clustering via Approximated p-Resistance

This chapter develops an approximation to the (effective) p-resistance and applies it to multi-class clustering. Spectral methods based on the graph Laplacian and its generalization to the graph p-Laplacian have been a backbone of non-Euclidean clustering techniques. The advantage of the p-Laplacian is that the parameter p induces a controllable bias on cluster structure. The drawback of p-Laplacian eigenvector based methods is that the third and higher eigenvectors are difficult to compute. Thus, instead, we are motivated to use the p-resistance induced by the p-Laplacian for clustering. For p-resistance, small p biases towards clusters with high internal connectivity while large p biases towards clusters of small "extent," that is a preference for smaller shortest-path distances between vertices in the cluster. However, the p-resistance is expensive to compute. We overcome this by developing an approximation to the p-resistance. We prove upper and lower bounds on this approximation and observe that it is exact when the graph is a tree. We also provide theoretical justification for the use of p-resistance for clustering. Finally, we provide experiments comparing our approximated p-resistance clustering to other p-Laplacian based methods.

5.1 Introduction

As we see in Chapter 2, various graph methods have been considered, such as clustering and semi-supervised learning [von Luxburg, 2007, Zhu et al., 2003]. Common to these methods, graph 2-seminorm, 2-seminorm induced from the graph Laplacian, is actively used. Its generalization to the graph p-seminorm is known to exhibit performance improvement [Bühler

and Hein, 2009, Slepcev and Thorpe, 2019].

This chapter considers multi-class clustering over a graph using the graph p-seminorm. For this purpose, spectral clustering is the most popular. In the 2-seminorm based (i.e., standard) spectral clustering, we use the first k eigenvectors of the graph Laplacian for k-class clustering [von Luxburg, 2007]. This use of the first k eigenvectors is theoretically supported [Lee et al., 2014]. Using the p-seminorm, this graph Laplacian is extended to the graph p-Laplacian [Bühler and Hein, 2009]. Similar to the standard case, using the first k eigenvectors of this graph p-Laplacian for k-class clustering is also theoretically supported [Tudisco and Hein, 2018]. However, as discussed in Sec. 2.1.4, there is not yet known an exact identification for the third or higher eigenpairs of p-Laplacian [Lindqvist, 2008], and hence in practice, it is difficult to obtain them. Due to this limitation, the existing methods using p-Laplacian propose an ad-hoc resolution of this limitation for multi-class clustering [Bühler and Hein, 2009, Ding et al., 2019, Luo et al., 2010]. On the other hand, this limitation makes the p-Laplacian difficult to use in practice to leverage the full potential of graph p-seminorm for multi-class clustering purposes. Note that the same limitation applies to Chapter 3 even if we generalize to hypergraph p-Laplacian, where we conducted experiments only for two-class clustering.

Thus, in order to aim to exploit the graph p-seminorm more for multi-class clustering, we explore an alternative way to spectral clustering; in this chapter, we propose multi-class clustering via approximated p-resistance. The p-resistance is also induced by the graph p-seminorm. The use of p-resistance for clustering is motivated in the following way. Looking back to the 2-seminorm case discussed in Sec. 2.4, the 2-resistance is defined as an inverse of the constrained optimization problem using the graph 2-seminorm and is known to be a metric over a graph [Klein and Randić, 1993]. Moreover, 2-resistance is characterized by a semi-supervised learning problem of the graph 2-seminorm regularization [Alamgir and Luxburg, 2011]. Given these properties, the 2-resistance is used for the multi-class graph clustering [Yen et al., 2005, Alev et al., 2017]. However, for the large graph under certain conditions, the 2-resistance converges to a meaningless limit function [Nadler et al., 2009, von Luxburg et al., 2010]. Using the graph p-seminorm, the 2-resistance is generalized to the p-resistance [Herbster and Lever, 2009], which overcomes this problem [Slepcev and Thorpe, 2019]. The $1/(p-1)$-th power of the p-resistance is also shown to be a metric [Herbster, 2010,

Kalman and Krauthgamer, 2021]. Furthermore, since different p of p-resistance captures a different characteristic of a graph [Alamgir and Luxburg, 2011], we expect that the parameter p serves as a tuning parameter for the clustering result. Thus, the natural idea for the multi-class clustering is to use the $1/(p-1)$-th power of p-resistance.

While the discussion above motivates us to use the $1/(p-1)$-th power of the p-resistance to multi-class clustering, there remain two issues; i) computational cost of p-resistances for many pairs ii) lack of theoretical justification for using p-resistance for clustering other than the metric property. In this chapter, we address these in the following way. For i), it is computationally expensive to compute p-resistances for many pairs. The reason is that we need to solve the constrained optimization problem for many pairs. Looking back at the 2-resistance, we can compute the 2-resistance efficiently in the following way. Recall that we can compute 2-resistance as

$$r_{G,2}(i,j) = \|L^+\mathbf{e}_i - L^+\mathbf{e}_j\|_{G,2}^2, \tag{5.1}$$

where $r_{G,p}(i,j)$ is p-resistance for a graph G, i and j are vertices, L^+ is a pseudoinverse of the graph Laplacian L for G, $\mathbf{e}_i$ is the i-th coordinate vector of $\mathbb{R}^n$, and $\|\cdot\|_{G,2}$ is a 2-seminorm induced from the graph Laplacian L. By this representation, once we compute L^+, we can "reuse" L^+ to compute 2-resistance for different pairs. This reuse makes the computation of 2-resistances for many pairs faster than naively solving the optimization problem for each pair. However, we do not know such representation for p-resistance. Thus, to obtain p-resistance for many pairs, we need to solve many constrained optimization problems. The significant result of this work is that in Thm. 5.4, we give a theoretical guarantee for the approximation of p-resistance as

$$r_{G,p}(i,j) \approx \|L^+\mathbf{e}_i - L^+\mathbf{e}_j\|_{G,q}^p, \tag{5.2}$$

where q satisfies $1/p + 1/q = 1$, and $\|\cdot\|_{G,q}$ is a graph q-seminorm whose formal definition is given later. We also show that for a tree, the approximation of Eq. (5.2) becomes exact (Thm. 5.5). By this approximation, we can compute the approximated p-resistance efficiently, similar to the $p = 2$ case. For ii), we do not have a theoretical justification for using p-resistance for clustering other than the metric property. While the p-resistance has the metric

property, this property itself does not support the clustering quality. For spectral clustering and 2-resistance, we have theoretical justifications for clustering. For spectral clustering, using the first k eigenvectors of the graph p-Laplacian is theoretically justified [Lee et al., 2014, Tudisco and Hein, 2018]. The 2-resistance has a theoretical connection to a semi-supervised learning problem of graph 2-seminorm regularization [Alamgir and Luxburg, 2011]. For p-resistance, we show that p-resistance is characterized by the semi-supervised learning problem of p-seminorm regularization. This resolves the open problem stated in [Alamgir and Luxburg, 2011]. This gives a theoretical foundation for using p-resistance for clustering from a view of the semi-supervised learning problem. Addressing the two issues above, as a multi-class clustering algorithm, we propose to apply the k-medoids algorithm to the distance matrix obtained from the approximated p-resistance. With these two results, our algorithm can be said to be more theoretically supported than existing multi-class spectral clusterings via graph p-Laplacian. Our experiment demonstrates that our algorithm outperforms the existing multi-class clustering using graph p-Laplacian and 2-resistance-based methods.

Our contributions are as follows: i) We give a guarantee for the approximated representation of p-resistance using the q-seminorm. ii) We show that the p-resistance characterizes the solution of semi-supervised learning of p-seminorm regularization of a graph. iii) We provide graph p-seminorm-based multi-class clustering. iv) We numerically show that our method outperforms the existing and standard methods. *All proofs are in Appendix.*

5.2 Hölder's Inequality and Matrix Norm

We review Hölder's inequality and matrix norm, which we use in this chapter.

First, we recall the weighted p-norm. Given positive weights $\mathbf{r} \in \mathbb{R}^{n_1}$ where $r_i > 0$, for a vector $\mathbf{x} \in \mathbb{R}^{n_1}$ we define the weighted p-norm $\|\mathbf{x}\|_{\mathbf{r},p}$, and its inner product $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{r}}$ as

$$\|\mathbf{x}\|_{\mathbf{r},p} := \left(\sum_{i=1}^{n_1} r_i |x_i|^p \right)^{1/p}, \langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{r}} := \sum_{i=1}^{n_1} r_i x_i y_i. \tag{5.3}$$

For this weighted p-norm and inner product, we have Hölder's inequality as follows;

Lemma 5.1 (Hölder's inequality)**.** *For $p, q > 1$ s.t. $1/p + 1/q = 1$, $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{r}} \leq \|\mathbf{x}\|_{\mathbf{r},p} \|\mathbf{y}\|_{\mathbf{r},q}$.*

For a matrix $M \in \mathbb{R}^{n_1 \times n_2}$, we define an *image* of M as $\mathrm{Im}(M) := \{\mathbf{y} | \mathbf{y} = M\mathbf{x}, \mathbf{x} \in \mathbb{R}^{n_2}\} \subseteq \mathbb{R}^{n_1}$, that is a space spanned by the matrix M. Note that $MM^{+}\mathbf{y}$ is an orthogonal

projection of $\mathbf{y}$ onto $\mathrm{Im}(M)$, where M^+ is a pseudoinverse of M. We introduce a *matrix operator p-norm* $|||M|||_p$ for a matrix M as

$$|||M|||_p := \sup_{\mathbf{x}\in\mathbb{R}^n} \frac{\|M\mathbf{x}\|_p}{\|\mathbf{x}\|_p}. \tag{5.4}$$

This operator p-norm can be bounded as follows.

$$|||M|||_p \leq \max(|||M|||_1, |||M|||_\infty). \tag{5.5}$$

Note that if M is symmetric, then $|||M|||_1 = |||M|||_\infty$. We refer to [Horn and Johnson, 2012] for the details.

5.3 Graph p-seminorm and Approximating p-Resistance

This section defines a graph p-seminorm, which is a foundation of our discussion. We then discuss several properties of the graph p-seminorm. Using these properties, we provide the approximation of p-resistance.

5.3.1 Graph p-seminorm

In this section, we define a graph p-seminorm and discuss its characteristics. For a vector over vertices $\mathbf{x} \in \mathbb{R}^n$, we define a graph p-seminorm over a graph using a weighted p-norm for a graph weight vector $\mathbf{w} \in \mathbb{R}^m$. Recall that we defined a graph p-seminorm $\|\mathbf{x}\|_{G,p}$ for $\mathbf{x} \in \mathbb{R}^n$ as

$$\|\mathbf{x}\|_{G,p} = \|C\mathbf{x}\|_{\mathbf{w},p} = \left(\sum_{i\in E} w_i |(C\mathbf{x})_i|^p\right)^{1/p} = \left(\sum_{i,j\in V} a_{ij}|x_i - x_j|^p\right)^{1/p}. \tag{5.6}$$

From the definition of p-energy Eq. (2.31), $S_{G,p}(\mathbf{x}) = \|\mathbf{x}\|_{G,p}^p$. Also, we immediately know that this norm is induced by the inner product $\langle C\mathbf{x}, C\mathbf{y}\rangle_{\mathbf{w}}$ from the definition of the graph p-seminorm. We now see that this graph seminorm can also be induced from the inner product $\langle \mathbf{x}, \mathbf{y}\rangle_L$, because

$$\langle C\mathbf{x}, C\mathbf{y}\rangle_{\mathbf{w}} = \mathbf{x}^\top C^\top W C\mathbf{y} = \mathbf{x}^\top L\mathbf{y} = \langle \mathbf{x}, \mathbf{y}\rangle_L. \tag{5.7}$$

From this observation, we see that $\|\mathbf{x}\|_{G,2} = \|\mathbf{x}\|_L$. Also, we can restrict graph p-seminorm to a norm if we consider $\mathbf{x} \in \mathrm{Im}(L)$. Note that this graph p-seminorm is same as the graph p-seminorm defined in [Herbster and Lever, 2009]. For this graph p-seminorm, using Lemma 5.1, the Hölder's inequality holds;

$$\langle \mathbf{x}, \mathbf{y} \rangle_L \leq \|\mathbf{x}\|_{G,p} \|\mathbf{y}\|_{G,q}, 1/p + 1/q = 1. \tag{5.8}$$

When $p = 2$ Hölder's inequality plays a fundamental role to show the representation of 2-resistance by Eq. (2.71) in the following way. Using the equality condition of the Hölder's inequality Eq. (5.8) for $p = 2$, we have a lemma.

Lemma 5.2 (Classical, e.g., Herbster and Pontil [2006])**.** *For $\mathbf{y} \in \mathbb{R}^n$, we have*

$$\|\mathbf{y}\|_{G,2}^{-2} = \min_{\mathbf{x}} \{\|\mathbf{x}\|_{G,2}^2 \text{ s.t. } \langle \mathbf{x}, \mathbf{y} \rangle_L = 1\}. \tag{5.9}$$

This lemma is a classical result rewritten with our notation of graph p-seminorm. By substituting $\mathbf{y} := L^+\mathbf{e}_i - L^+\mathbf{e}_j$, the right hand side of Lemma 5.2 becomes the inverse of 2-resistance (see Appendix 5.B.5). Thus, we obtain $r_{G,2}(i,j) = \|L^+\mathbf{e}_i - L^+\mathbf{e}_j\|_L^2 = \|L^+\mathbf{e}_i - L^+\mathbf{e}_j\|_{G,2}^2$. For p-resistance, the question is how is the coordinate spanning set $\mathcal{V}(L^+)$ related to the p-resistance? Can we derive such relation using Hölder's inequality Eq. (5.8), similarly to the $p = 2$ case? Next section will show such connection between p-resistance and the coordinate spanning set using Eq. (5.8).

5.3.2 Approximating p-Resistance via Coordinate Spanning Set

This section discusses approximation of p-resistance via the coordinate spanning set $\mathcal{V}(L^+)$. Looking at the $p = 2$ case, we see that $L^+\mathbf{e}_i \in \mathcal{V}(L^+)$ can be regarded as coordinate, and $r_{G,2}(i,j) = \|L^+\mathbf{e}_i - L^+\mathbf{e}_j\|_{G,2}^2$. This expression aids us to compute all the pairs of 2-resistance much faster than naively obtaining 2-resistance. For p-resistance, a natural question to ask is that does there exist some norm $\|\cdot\|^\ddagger$ such that $r_{G,p}(i,j) = \|L^+\mathbf{e}_i - L^+\mathbf{e}_j\|^\ddagger$? If not, how can we approximate as $r_{G,p}(i,j) \approx \|L^+\mathbf{e}_i - L^+\mathbf{e}_j\|^\ddagger$? If we can write p-resistance by such expression, we expect to obtain all the pairs of approximated p-resistance much faster than naively computing all the pairs of p-resistance. This section addresses this problem.

As we see in Sec. 5.3.1, Lemma 5.2 is a key to show that $r_{G,2}(i,j) = \|L^+\mathbf{e}_i - L^+\mathbf{e}_j\|_{G,2}^2$.

In the following, we now extend Lemma 5.2 from the case of $p = 2$ to the general p.

Proposition 5.3. *For a graph G and $p, q > 1$ such that $1/p + 1/q = 1$, we have*

$$\|\mathbf{y}\|_{G,q}^{-p} \leq \min_{\mathbf{x}}\{\|\mathbf{x}\|_{G,p}^{p} \text{ s.t. } \langle \mathbf{y}, \mathbf{x}\rangle_L = 1\} \leq \|\mathbf{z}\|_{G,p}^{p} \tag{5.10}$$

where

$$\mathbf{z} := C^{+}\frac{f_{q/p}(C\mathbf{y})}{\|\mathbf{y}\|_{G,q}^{q}}, \quad (f_\theta(\mathbf{x}))_i := \mathrm{sgn}(x_i)|x_i|^\theta. \tag{5.11}$$

When $f_{q/p}(C\mathbf{y}) \in \mathrm{Im}(C)$, we have

$$\|\mathbf{y}\|_{G,q}^{-p} = \min_{\mathbf{x}}\{\|\mathbf{x}\|_{G,p}^{p} \text{ s.t. } \langle \mathbf{y}, \mathbf{x}\rangle_L = 1\} = \|\mathbf{z}\|_{G,p}^{p} \tag{5.12}$$

We first note that the minimization problem of Eq. (5.10) is the inverse of p-resistance Eq. (2.31). The left hand side of inequality Eq. (5.10) immediately follows from Hölder's inequality (Eq. (5.8)) with $\langle \mathbf{y}, \mathbf{x}\rangle_L = 1$. We now turn our attention to the right hand side. Recall that when $p = 2$ we always have $f_{q/p}(C\mathbf{x}) \in \mathrm{Im}(C)$ and $\|\mathbf{y}\|_{G,2}^{-1} = \|\mathbf{z}\|_{G,2}$, which matches Lemma 5.2. In the general p case, $f_{q/p}(C\mathbf{x}) \notin \mathrm{Im}(C)$ and $\|\mathbf{y}\|_{G,q}^{-1} \neq \|\mathbf{z}\|_{G,p}$. Thus, neither $\|\mathbf{y}\|_{G,q}^{-p}$ nor $\|\mathbf{z}\|_{G,p}^{p}$ gives the solution to the minimization problem. However, this theorem tells us that we can upper bound the solution to the minimization problem by $\|\mathbf{z}\|_{G,p}^{p}$.

Applying Prop. 5.3, we obtain the bound for p-resistance as follows;

Theorem 5.4. *For a graph G and $p, q > 1$ such that $1/p + 1/q = 1$, the p-resistance can be bounded as*

$$\frac{1}{\alpha_{G,p}^{p}}\|L^{+}\mathbf{e}_i - L^{+}\mathbf{e}_j\|_{G,q}^{p} \leq r_{G,p}(i,j) \leq \|L^{+}\mathbf{e}_i - L^{+}\mathbf{e}_j\|_{G,q}^{p},$$

where

$$\alpha_{G,p} := \|\!|W^{1/p}CC^{+}W^{-1/p}|\!\|_p. \tag{5.13}$$

Theorem 5.5. *For a tree G and $p, q > 1$ such that $1/p + 1/q = 1$, the p-resistance can be*

written as

$$r_{G,p}(i,j) = \|L^+\mathbf{e}_i - L^+\mathbf{e}_j\|_{G,q}^p. \tag{5.14}$$

Thm. 5.4 and Thm. 5.5 show the relationship between p-resistance and $\|L^+\mathbf{e}_i - L^+\mathbf{e}_j\|_{G,q}^p$. For general graphs, we do not obtain the exact representation of p-resistance. However, Thm. 5.4 guarantees the quality of approximation as

$$r_{G,p}(i,j) \approx \|L^+\mathbf{e}_i - L^+\mathbf{e}_j\|_{G,q}^p = \|\mathbf{v}_i - \mathbf{v}_j\|_{G,q}^p, \tag{5.15}$$

where $\mathbf{v}_i$,$\mathbf{v}_j \in \mathcal{V}(L)$. By this approximation, we obtain a similar representation of p-resistance to the $p = 2$ case Eq. (2.71). The term $\alpha_{G,p}$ is a p-norm of the orthogonal projector to $\mathrm{Im}(W^{1/p}C)$. Note that we always have $\alpha_{G,p} \geq 1$. For a tree graph, Thm. 5.5 shows that $\|L^+\mathbf{e}_i - L^+\mathbf{e}_j\|_{G,q}^p$ becomes the exact representation of p-resistance.

The next question is what is $\alpha_{G,p}$. We bound $\alpha_{G,p}$ as follows;

Proposition 5.6. *For a general graph G and $p > 1$, we have $\alpha_{G,p} \leq m^{|1/2-1/p|}$.*

This proposition gives the guarantee for the approximation in Thm. 5.4. Although Prop. 5.6 gives the quality guarantee, we expect this upper bound to be loose, i.e., we expect that the actual approximation value is closer to the exact value than this bound. The reason why we expect in this way is that to prove the bound we only use the general technique that holds for any matrix and we do not use any graph structural information. In fact, we have a bound for the specific graphs as follows.

Proposition 5.7. *If a graph is complete or cyclic, then $\|\|CC^+\|\|_p \leq 4$ and hence $\alpha_{G,p} \leq 4w_{\max}^{1/p}/w_{\min}^{1/p}$.*

For these specific graphs, we can bound the p-resistance (Thm. 5.4) by a constant. In the real dataset, we observe that the approximation of p-resistance and $\alpha_{G,p}$ is far better than this guarantee, see Sec. 5.6.

Finally, we discuss computational times of the p-resistance. To compute Eq. (5.15), it takes $O(m)$, given L^+. Also, in general it takes $O(n^3)$ to compute L^+. Note that we can reuse L^+ to compute p-resistance for different pairs. We now consider to obtain the p-resistance by naively solving the optimization problem. We can rewrite the constrained problem Eq. (2.74)

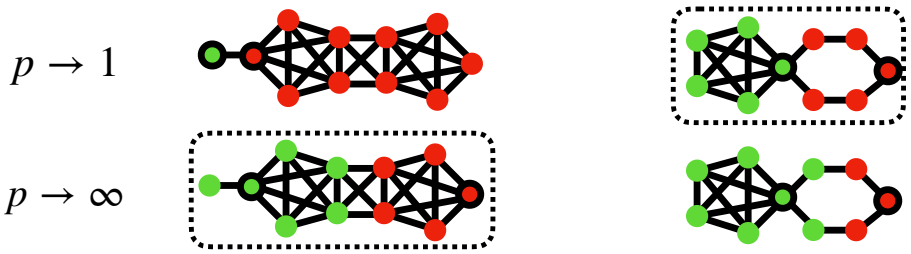


Figure 5.1: The illustrative examples where p changes the results of the clustering using p-resistance. These examples conduct clustering with k-center algorithm using p-resistance as a metric. The red and green colors show the clustering result. Also, the vertices with borders show the obtained centers. The dotted boxes exhibit natural clustering results. These examples show varying p tunes the clustering result; the left example gives a more natural clustering result when $p \to \infty$ whereas for right $p \to 1$ gives more natural result. Details are in Sec. 5.4.1 and Appendix 5.F.

as unconstrained problem, which is solvable by gradient descent. In each step of the gradient descent, we compute $\nabla_{\mathbf{x}} \|\mathbf{x}\|_{G,p}^p$, which takes almost same time as Eq. (5.15). Moreover, we cannot reuse the result of a single pair to compute for other pairs, while we can reuse L^+. Thus, to compute p-resistance for a single pair, our approximation is expected to be faster than naively solving the optimization problem. Moreover, if we compute for many pairs, our approximation is much faster by reusing L^+.

5.4 Clustering via p-Resistance

This section considers using the p-resistance for the clustering algorithm. Firstly, we propose a clustering algorithm using the approximated p-resistance. We next characterize our clustering algorithm from the semi-supervised problem point of view. From this characterization, we can see that our clustering algorithm inherits properties from semi-supervised learning.

5.4.1 Proposed Clustering Algorithm via p-Resistance

This section proposes an algorithm using p-resistance. The triangle inequality Eq. (2.75) gives a metric property to $r_{G,p}^{1/(p-1)}(i,j)$. We call this $1/(p-1)$-th power of p-resistance as *p-resistance metric*. This metric property motivates us to use p-resistance for clustering algorithms.

Furthermore, the parameter p serves as a tuning parameter of the clustering result. The general p of p-resistance captures the graph structure somewhere between the cut and shortest

path. Using this characteristic, we expect varying p tunes the clustering result somewhere suitable between cut-based and path-based. When p is small, the clustering result biases towards clusters with high internal connectivity, like a min-cut. When p is large, the clustering result focus more on path-based topology, that is a preference for smaller shortest-path distances between vertices in the cluster. We illustrate this with examples of the two-class clustering in Fig. 5.1. In these examples, we conduct clustering with k-center algorithm using p-resistance. The left example is intuitively "symmetric"; for this kind, $p \to \infty$, which looks at the path-based topology, gives more natural result. The more natural clustering of the right example is "cut"; for this kind, $p \to 1$, where we focus on the graph cut, gives the more natural result. More details are in Appendix 5.F.

While the discussion above motivates us to use the p-resistance metric for clustering, computing the p-resistance metric for all pairs is costly. Thus, we approximate this metric by Thm. 5.4, and we obtain

$$r_{G,p}^{1/p-1}(i,j) \approx \|L^{+}\mathbf{e}_i - L^{+}\mathbf{e}_j\|_{G,q}^{p/(p-1)} = \|L^{+}\mathbf{e}_i - L^{+}\mathbf{e}_j\|_{G,q}^{q} = \|\mathbf{v}_i - \mathbf{v}_j\|_{G,q}^{q}, \tag{5.16}$$

where $\mathbf{v}_i, \mathbf{v}_j \in \mathcal{V}(L)$. We then apply k-medoids to the distance matrix obtained by Eq. (5.16). The overall proposed algorithm is summarized in Alg. 5

We discuss the choice of k-medoids over the other distance based method, such as k-means [Bishop and Nasrabadi, 2006] and k-center [Gonzalez, 1985]. Although the main emphasis of our algorithm does not comes from the choice of k-medoids but from the approximation of p-resistance metric, k-medoids has some advantages. Since p-resistance metric cannot define a distance between other than the data points defined as $\mathcal{V}(L^{+})$, we cannot define distance for the some "mean", which is outside of the data points. Therefore, the mean-based method such as k-means is not appropriate for this setting. Instead, k-medoids is similar to the k-means [Kaufman and Rousseeuw, 1990] but more appropriate since k-medoids assigns the centers to the actual data points. The other potential choice is k-center algorithm. The k-center algorithm also assigns the center to the actual data point, and is known to be faster than k-medoids. Also, k-center algorithm is approximated by the fast greedy farthest first algorithm [Gonzalez, 1985, Herbster, 2010]. However, the k-medoids is more robust to the outliers than k-center. Thus, we propose to use k-medoids. The overall computational time for Alg. 5 is dominated by the computation of the all the pairs of the approximated

Algorithm 5 Clustering Algorithm via p-Resistance

Input: Graph $G = (V, E)$ and p

1: Compute pseudoinverse of the graph Laplacian L^+.
2: Compute all the pairs of the p-resistance metrics $r_{G,p}^{1/(p-1)}$ using Eq. (5.16) and obtain a distance matrix.
3: Apply k-medoids to the distance matrix.

Output: The clustering result.

p-resistance, $O(mn^2)$. If we use the farthest first algorithm instead of k-medoids the algorithm is dominated either by the computation of L^+, $O(n^3)$, or farthest first $O(kmn)$. Thus, farthest first is faster since in general $m \gg n$ but less robust than k-medoids.

5.4.2 Connection between Semi-supervised Learning and p-Resistance

This section explores connection from the p-resistance to the semi-supervised learning (SSL) via graph p-seminorm. As we saw in Sec. 5.2, Herbster [2010] shows the metric property of p-resistance. While the metric property itself can motivates us to use p-resistance for our clustering, we do not know how much p-resistance shows connectivity of a graph. This section shows that p-resistance can be seen from as an SSL perspective. This connection assures us to use p-resistance for the clustering problem. In the following we explain the connection by taking the following steps; i) SSL problem in the clustering context ii) the connection between the SSL and p-resistance.

We first consider an SSL problem for two known labels as

$$\min_{\mathbf{x}}\{S_{G,p}(\mathbf{x}) \text{ s.t. } x_i - x_j = 1\} = \min_{\mathbf{x}}\{S_{G,p}(\mathbf{x}) \text{ s.t. } x_i = 1, x_j = 0\}. \tag{5.17}$$

The equality holds since $S_{G,p}(\mathbf{x}) = S_{G,p}(\mathbf{x} + c\mathbf{1})$, $\forall c \in \mathbb{R}$. We first note that Eq. (5.17) is an inverse of the p-resistance and we use the optimal value of this problem to p-resistance. This learning problem for $p = 2$ case has been considered in many literature, such as [Zhu et al., 2003], and extended to the p-seminorm setting [Herbster and Lever, 2009, Alamgir and Luxburg, 2011, Slepcev and Thorpe, 2019].

We now put Eq. (5.17) into clustering context; the solution of Eq. (5.17) tells us the graph structural information on clustering. We recognize that Eq. (5.17) is two fixed-label problem. Let $\mathbf{x}^{*ij}$ be a solution of the problem Eq. (5.17). It is straightforward to interpret Eq. (5.17) if i and j is in different binary classes; we see which clusters the third point ℓ belongs to, the

cluster which i or j is in. More specifically, by comparing $x_\ell^{*ij} - x_j^{*ij}$ and $x_i^{*ij} - x_\ell^{*ij}$ we know which cluster the third point ℓ belongs to. If we take a pair of vertices (i, j) arbitrarily, the assumption that "i and j in different binary classes" is not always appropriate. In this case, rather than assuming i and j in different binary classes, it is more natural to interpret in the following way; the two-pole binary SSL problem tells us that which of i and j the third point ℓ is close to in a graph. From this observation, if we look at $\mathbf{x}^{*ij}$ for all pairs, we know "graph structural information" from the SSL point of view.

We next show the connection between p-resistance and the solution of Eq. (5.17), $\mathbf{x}^{*ij}$.

Theorem 5.8. *Let $\mathbf{x}^{*ij}$ be the solution of the problem Eq.* (5.17)*, and $\ell \in V$ be the third unlabeled point. Then we have*

$$x_\ell^{*ij} - x_j^{*ij} \geq x_i^{*ij} - x_\ell^{*ij} \iff r_{G,p}(j, \ell) \geq r_{G,p}(\ell, i).$$

First note that Alamgir and Luxburg [2011] proved Thm. 5.8 only for the $p = 2$ case in a different context than clustering (See Appendix 5.H.2), and posed the case of general p as an open problem. We resolve this open problem.

Thm. 5.8 means that the p-resistance has a good property inherited from the SSL problem Eq. (5.17) in a following sense. Thm. 5.8 tells us that the "graph structural information", which can be obtained by comparing $x_\ell^{*ij} - x_i^{*ij}$ and $x_j^{*ij} - x_\ell^{*ij}$, is equivalent to comparing p-resistances $r_{G,p}(i, \ell)$ and $r_{G,p}(\ell, j)$. Henceforth, Thm. 5.8 further translates the intuition about $\mathbf{x}^{*ij}$ into p-resistance.

Thus, combining the two observations above, looking at the distance matrix computed from p-resistances can be interpreted as follows. Each distance shows how close the pair is in terms of two-pole binary SSL problem. Doing clustering with this distance matrix assigns a cluster by looking at all the graph structural information of two-pole binary SSL problems, which tells us that "which the third point ℓ is close to, i or j?"

From the observations above, we see that Thm. 5.8 motivates us to use p-resistance metrics for multi-class problem. Without Thm. 5.8, our algorithm is somewhat naive; even though p-resistance has a metric property, we do not know how much p-resistance contains the structural information.

5.5 Related Work

This section reviews the related work to the clustering via graph p-seminorm. Since our work uses graph p-seminorm for the clustering purpose, spectral clustering using graph p-Laplacian is relevant. The graph p-Laplacian is induced from graph p-seminorm and used for the clustering purpose [Bühler and Hein, 2009]. Tudisco and Hein [2018] showed a theoretical guarantee for the use of the first k variational eigenvectors (i.e., eigenvectors obtained by variational theorem) of p-Laplacian for k-class clustering. While we know the exact identification for the second eigenvectors of p-Laplacian, we do not know how to obtain the third or higher eigenvectors [Lindqvist, 2008]. Thus, it is practically difficult to use spectra of p-Laplacian for multi-class clustering. To bypass this limitation, Bühler and Hein [2009] applied two-class clustering method to multi-class by recursively bisectioning a subgraph into two subgraphs, which does not exploit the full structure of the graph. The earlier works [Luo et al., 2010, Ding et al., 2019, Pasadakis et al., 2022] used approximated orthogonality between eigenvectors of p-Laplacian for multi-class clustering. However, we do not have theoretical supports that this approximated k eigenvectors are the approximation of *the first* k variational eigenvectors. Thus, we need to say that these methods rely on the "ad-hoc bypasses" and do not fully exploit the graph p-seminorm. For more details, see Sec. 2.1.4.5.

Another relevant approach is resistance-based clustering. In Yen et al. [2005], k-medoids algorithm is applied to the square of 2-resistance. For clustering purpose, similar distances to the 2-resistance is proposed [Fouss et al., 2007, Nguyen and Mamitsuka, 2016, Yen et al., 2008] The most relevant approach in this category is the k-center algorithm for the "distance" matrix obtained from the *exact* p-resistance in Herbster [2010]. Herbster [2010] did not numerically verify the algorithm. Our work uses p-resistance metric instead of p-resistance since without the $1/(p-1)$-th power operation p-resistance does not satisfy the metric property (Eq. (2.75)). However, if we use k-center algorithm to the *exact* p-resistance metric, we obtain the same result as Herbster [2010]. The reason is that the k-center algorithm only matters the order of the distance, and the $1/(p-1)$-th power operation does not change the order of the p-resistance. On the other hand, we emphasize that the most significant difference between our work and Herbster [2010] is that while we use the *approximated* p-resistance Herbster [2010] uses exact p-resistance. We also mention that the work [Nguyen and Mamitsuka, 2016]

Table 5.1: Dataset summary. Since Hopkins 155 contains 155 different videos, we report the sum of the data points and sum of the dimensions of videos. Also, Hopkins 155 dataset contains 120 2-class datasets and 35 3-class datasets.

	ionosphere	hop 155 2cls	iris	wine	hop 155 3cls
# of class	2	2	3	3	3
size	351	31981	150	178	13983
dimension	34	3542	4	13	999

proposed a distance from the p-seminorm flow point of view. However, this distance does not have characterization from the learning problem (Thm. 5.8).

Also, the graph p-seminorm is actively used in semi-supervised learning (SSL). The SSL problem using graph p-seminorm is relevant to p-resistance since the p-resistance can be seen as SSL for two known labels. Earlier, the SSL using graph 2-seminorm is considered [Zhou et al., 2003, Zhu et al., 2003, Calder et al., 2020]. The SSL via graph 2-seminorm and effective resistance is known to be "ill-posed" when the size of the unlabeled data points is asymptotically large [Nadler et al., 2009]. To overcome this problem, graph p-seminorm based SSL and p-resistance are considered [Alamgir and Luxburg, 2011, Bridle and Zhu, 2013, El Alaoui et al., 2016, Slepcev and Thorpe, 2019], where the p-resistance is shown to be meaningful when p is large. Finally, the graph p-seminorm is widely used in the machine learning community, such as online learning [Herbster and Lever, 2009, Pasteris et al., 2024] and the local graph clustering task, where we find a cluster which the given vertices belong to [Veldt et al., 2019, Fountoulakis et al., 2020, Liu and Gleich, 2020].

5.6 Experiments

This section numerically demonstrates the performance of our Alg. 5 using approximated p-resistance.

Objective of the Experiments. The purpose of the experiments is to evaluate if our algorithm on two-class and multi-class clustering problems improves the existing p-seminorm-based graph clustering algorithm. Thus, we compared it with existing resistance-based algorithms and spectral clustering algorithms using graph p-seminorm and its $p = 2$ setting.

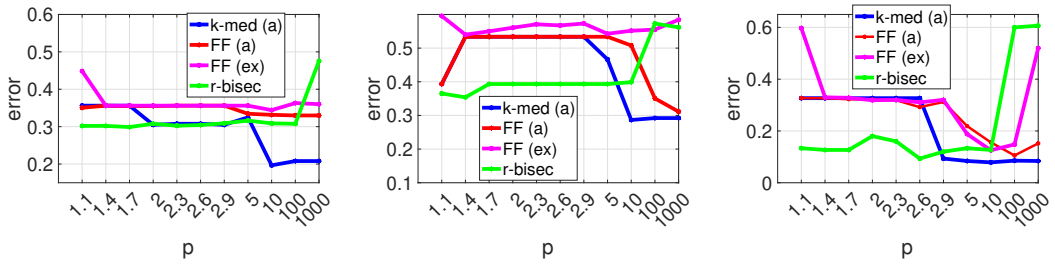
Datasets. Our experiments were conducted on the same classification datasets. We used the ionosphere, iris, and wine datasets from the UCI repository, as well as the Hopkins155 dataset [Tron and Vidal, 2007], which includes 120 two-class and 35 three-class motion

Table 5.2: Experimental results. The "type" shows the type of methods; (ER) for effective resistance based methods and (SC) for spectral clustering methods. The "Hop" stands for Hopkins 155 dataset. In method of ER, "(a)" shows that the method uses the approximation by (Eq. (5.16)) and "(ex)" computes the exact p-resistance by gradient descent. Also, "k-med" is k-medoids, and "FF" is the farthest first. Thus, the method "k-med (a) p" is our proposed algorithm, and "FF (ex) p" and "FF $p = 2$" is a method proposed by [Herbster, 2010]. The "p-Flow" is [Nguyen and Mamitsuka, 2016], "ECT" is [Yen et al., 2005], "Rec-bi p" is [Bühler and Hein, 2009], and "p-orth" is [Luo et al., 2010]. Since "Rec-bi p" is a deterministic method, we only report error. Also, since Hop contains multiple datasets, we only show the average. Due to the significant computational time, we were unable to finish some of the experiments, which are shown as "–".

		2 clsss		multi-class		
Type	Method	ionosphere	Hop 2 cls	iris	wine	Hop 3 cls
ER	k-med (a) p	**0.196** $\pm$ 0.000	**0.056**	0.078 $\pm$ 0.013	**0.287** $\pm$ 0.000	**0.144**
ER	k-med (ex) p	–	–	**0.075** $\pm$ 0.000	0.427 $\pm$ 0.000	–
ER	k-med $p = 2$	0.305 $\pm$ 0.000	0.236	0.331 $\pm$ 0.000	0.534 $\pm$ 0.000	0.306
ER	FF (a) p	0.330 $\pm$ 0.023	0.109	0.108 $\pm$ 0.045	0.339 $\pm$ 0.054	0.313
ER	FF (ex) p	0.344 $\pm$ 0.020	–	0.109 $\pm$ 0.019	0.524 $\pm$ 0.046	–
ER	FF $p = 2$	0.355 $\pm$ 0.035	0.274	0.320 $\pm$ 0.000	0.530 $\pm$ 0.000	0.357
ER	p-Flow	0.291 $\pm$ 0.000	0.231	0.247 $\pm$ 0.000	0.543 $\pm$ 0.043	0.243
ER	ECT	0.376 $\pm$ 0.000	0.155	0.247 $\pm$ 0.000	0.534 $\pm$ 0.000	0.310
SC	Rec-bi p	0.225	0.200	0.089	0.354	0.237
SC	SC p-orth	0.215 $\pm$ 0.123	0.237	0.087 $\pm$ 0.089	0.327 $\pm$ 0.116	0.221
SC	SC $p = 2$	0.308 $\pm$ 0.000	0.216	0.093 $\pm$ 0.000	0.438 $\pm$ 0.000	0.251

Table 5.3: Computational time for approximated vs exact p-resistance. (a) denotes approximation and (ex) denotes exact. In "r" we reuse L^+. In "et" we compute L^+ each time. All time is in second.

	ionosphere	iris	wine
(a) + r	0.08 $\pm$ 0.04	0.07 $\pm$ 0.03	0.01 $\pm$ 0.00
(a) + et	0.39 $\pm$ 0.00	0.32 $\pm$ 0.00	0.05 $\pm$ 0.00
(ex)	1.11 $\pm$ 0.00	1.03 $\pm$ 0.04	0.36 $\pm$ 0.00



(a) Ion (b) Wine (c) Iris

Figure 5.2: Plots of the error vs p for the methods. The k-med (a) stands for k-median using our approximated p-resistance. FF (a) stands for furthest first using approximated p-resistance. FF (e) stands for furthest first using exact p-resistance. The legend r-bisec stands for recursive bisectioning using p-Laplacian.

segmentation datasets. Thus, we conducted our experiments on 158 datasets. Dataset sizes are summarized in Table 4.2, part of which we share with Chpater 4. Since our algorithm as well as resistance based comparisons take $O(mn^2)$, we choose small to medium size datasets. Although the datasets are the same as in Chapter 4, the graphs used are different in the following way. Chapter 4 focused on constructing graphs from vector data using similarity or kernel functions for all pairs. However, due to higher computational costs in our experiments, we used k-NN graphs with $k = \mu n$ $(0 < \mu \leq 1)$ to achieve sparsity. While performance results may vary between the chapters, our primary aim is to compare with baselines for each experiment following the experimental objectives, not to achieve the best performance on the datasets. See more discussion in Sec 7.3.

Experimental Settings. For the resistance based method, we compared with the farthest first algorithm on our approximated p-resistance. Additionally, we compared with existing methods; the farthest first using *exact* p-resistance [Herbster, 2010], a p-seminorm flow based method [Nguyen and Mamitsuka, 2016], and 2-resistance based method [Yen et al., 2005]. We especially note that for the farthest first [Herbster, 2010], we computed the exact p-resistance by the gradient descent as discussed in Sec. 5.3.2. We also apply this exact p-resistance to k-medoids. Note that k-medoids is more costly than k-center as discussed in Sec. 5.4.1. For spectral clustering methods, we compared with a recursive bisection method [Bühler and Hein, 2009] and a method using the approximated orthogonality [Luo et al., 2010]. Since the p-resistance is related to the unnormalized graph Laplacian, we use

the unnormalized graph Laplacian for the spectral methods. We created a graph using the following procedure: We construct a k-NN graph using the Euclidean distance. Then, we computed the edge weight with a Gaussian kernel ($\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\sigma\|\mathbf{x}_i - \mathbf{x}_j\|^2)$) for two vectors $\mathbf{x}_i, \mathbf{x}_j$. We used free parameters $\mu \in \{0.04, 0.06, 0.08, 0.1\}$, $\sigma \in \{10^{-3}, \dots, 10^2\}$ and $p \in \{1.1, 1.4, \dots, 2.9, 5, 10, 100, 1000\}$. For comparisons, we followed the original parameters other than above μ, σ, and p. We evaluated the performance by error rate, similar to the previous study [Bühler and Hein, 2009]. Since the Hopkins155 dataset contains multiple two-class and three-class tasks, we take an average of error rates among a set of two-class tasks and three-class tasks and report both. For computational time, due to its significant computational time, we parallelized the distance computation for the exact method, while we did not use such a technique for the others. Thus, we first compare the approximation of the p-resistance and the exact p-resistance. Then, we compare the computational time among the methods except for the exact methods. Finally, we remark that our experiment was conducted on Mac Studio with M1 Max Processor and 32GiB RAM. Also, we use an Intel binary Matlab translated by Rosetta, which is standard in MacOS with Apple Silicon environment at the time when experiments are conducted.

Notes on the Comparison Procedures. We make a detailed note on the comparison methods. Firstly, for the comparison method Rec-bi [Bühler and Hein, 2009], the algorithm is originally defined for $p \leq 2$. Thus, we apply the same technique for $p \geq 2$. In [Bühler and Hein, 2009], in order to avoid the conversion to too close or too far local optimum, at the step t [Bühler and Hein, 2009] minimizes Eq. (2.37) via the gradient descent method using the initial condition as the obtained eigenvector of the previous step $p_t = 0.9p_{t-1}$. If we increase the p, we use the same technique; $p_t = p_{t-1}/0.9$ until p reaches 5. Beyond 5, we use $p_t = 2p_{t-1}$. When $p = 2$, we know that 2-resistance is further computed as

$$r_{G,2}(i,j) = \|L^+\mathbf{e}_i - L^+\mathbf{e}_j\|_{G,2}^2 = \|(L^+)^{1/2}\mathbf{e}_i - (L^+)^{1/2}\mathbf{e}_j\|^2. \tag{5.18}$$

The graph p-seminorm is the size m norm, while the latter is based on the size n norm. However, we use graph 2-seminorm even for the $p = 2$ case. The reason is that we needed to be consistent in the experiment since we observed numerical round-off errors in the other methods. On the other hand, ECT [Yen et al., 2005] uses the square of 2-resistance. For this method, we use $\|(L^+)^{1/2}\mathbf{e}_i - (L^+)^{1/2}\mathbf{e}_j\|^2$ since the original paper [Yen et al., 2005] uses

Table 5.4: Computational time for the main experiment (unit:sec). Here we use E notation, e.g., E-6= 10^{-6} or E1 =10^{1}. Since "Rec-bi p" is a deterministic method, we only report time. Also, since Hop contains multiple datasets, we only show the average.

		2 clsss		multi-class		
Type	Method	ionosphere	Hop 2 cls	iris	wine	Hop 3 cls
ER	k-med (a) p	1.37E1 $\pm$ 0.01E1	1.21E1	4.93E0 $\pm$ 0.01E0	1.30E-1 $\pm$ 0.13E-1	1.78E1
ER	k-med $p=2$	3.76E0 $\pm$ 0.01E0	1.01E1	1.36E0 $\pm$ 0.00E0	9.07E-2 $\pm$ 1.05E-2	1.34E1
ER	FF (a) p	9.71E-2 $\pm$ 0.32E-1	4.88E-1	5.45E-1 $\pm$ 0.33E-1	4.72E-2 $\pm$ 0.18E-2	8.01E0
ER	FF $p=2$	8.71E-2 $\pm$ 0.12E-1	3.56E-1	4.21E-1 $\pm$ 0.12E-1	3.82E-2 $\pm$ 0.06E-2	6.45E0
ER	p-Flow	1.46E1 $\pm$ 0.01E1	1.33E1	5.21 E0 $\pm$ 0.02E0	1.60E-1 $\pm$ 0.15E-1	1.81E1
ER	ECT	1.01E-1 $\pm$ 0.10E-1	8.12E-1	2.03E-2 $\pm$ 0.42E-2	2.05E-2 $\pm$ 0.49E-2	9.26E-1
SC	Rec-bi p	1.18E-1	8.11E-1	5.35E-1	9.07E-2	1.01E0
SC	SC p-orth	8.60E-2 $\pm$ 0.01E-2	6.31E-1	4.78E-1 $\pm$ 1.65E-1	3.02E-2 $\pm$ 0.57E-2	8.10E-1
SC	SC $p=2$	1.60E-2 $\pm$ 0.01E-2	3.43E-2	1.28E-1 $\pm$ 0.00E-1	8.78E-3 $\pm$ 0.00E-3	6.12E-2

this.

Overall Results. The results are summarized as follows. We see that ours outperforms the others except for iris. As we expected, seeing the deviation, k-medoids offers more robust performance than the farthest-first algorithms. Moreover, our approximation provides faster computation than the exact method since we could not finish some of the experiments using the exact p-resistance even for the farthest first. Seeing Fig. 5.2, in k-medoids large p offers better performance. Also, if we look at $p=2$, the k-medoids with 2-resistance is not always better than spectral clustering. However, for general p the k-medoids with p-resistance performs better. Thus, the k-medoids with p-resistance can be said to be more benefited by p than spectral clustering. These correspond to the existing theoretical indication; p-resistance with large p becomes meaningful function while 2-resistance is not [Alamgir and Luxburg, 2011]. Comparing exact and approximation p-resistance in Fig. 5.2, while we observe similar performance in the middle range of p, we observe the better performance for approximation at the very small p or very large ps. This might come from the numerical computation of the gradient $\nabla_{\mathbf{x}}\|\mathbf{x}\|_{G,p}^{p}$ as follows. For the exact solution of the very small p, the gradient of each step tends to be very small. For the very large p, there is a risk of amplifying round-off numerical error at each step of optimization by taking the power of large p. On the other hand, the approximation offers a robust computation, especially for important large ps, because instead we compute by taking the power of small q in Eq. (5.16), by which we can avoid the risk discussed.

Computational Time to Compute Resistances. Next, we compare times to compute the exact and approximated p-resistance for 100 randomly chosen pairs of vertices. We made a

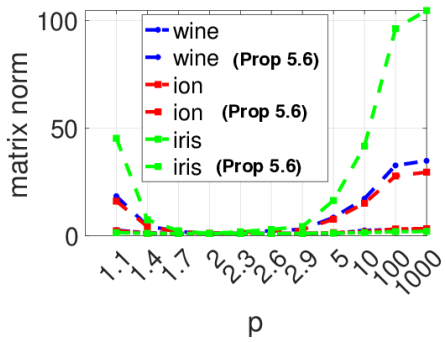
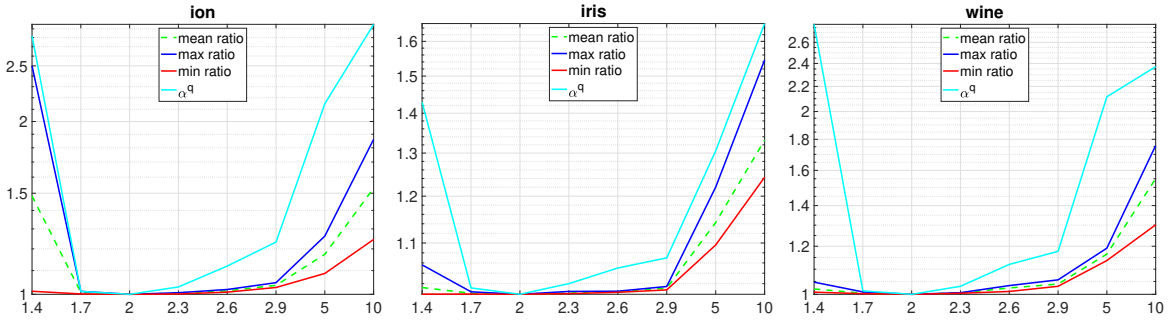


Figure 5.3: Plot of matrix norm $\|CC^+\|_p$ vs. the bound $m^{1/2-1/p}$ in Prop. 5.6.

graph for ionosphere, iris, and wine by choosing the best-performing parameters in Table. 5.2. We measure time for exact p-resistance by naively optimizing Eq. (2.74) by gradient descent. For approximation, since we can reuse L^+ for our approximation, we report the time in two ways: i) we compute L^+ each time and compute the approximation, and ii) we reuse L^+. The "each time" scenario reports the computational time of p-resistance for a single pair. In the reuse scenario, we measure time t to compute L^+ and p-resistance 100 pairs using L^+. Then, we report the time $t/100$. By this, the reuse scenario is much faster than the each time scenario. Table 5.3 summarizes the computing time. We observe that the approximation method for a single pair is faster than the exact method by comparing the "each time" and exact. As expected, the reuse scenario provides much faster computation than the exact p-resistance

Computational Time of the Experiments. In Table 5.3, we compare the approximation by Eq. (5.16) and the exact computation of p-resistance by naively optimizing Eq. (2.74) by the gradient descent. For ionosphere, iris, and wine, we made a graph for the best performing parameters in Table. 5.2. In Table 5.4, we compare the computing time for the best-performing parameters in Table. 5.2. We are unable to share the computing time of the exact resistances since due to their severe time complexity, we needed to parallelize the computations, which makes us difficult to track the running time. We can see that ours are slower than spectral clustering methods. This slowness is because ours takes $O(mn^2)$ while spectral clustering methods using p-Laplacian are $O(n^3)$-based convergence methods. Looking at the computational time for ECT, the time is similar to the spectral methods since ECT is also the $O(n^3)$ method.

Comparison of the Values of Approximated and Exact p-Resistance. Here we observe that the value of approximation from the real experiments is tighter than Prop. 5.6. Firstly,



(a) ion (b) iris (c) wine

Figure 5.4: The ratio of the approximated value of p-resistance to the exact p-resistance, i.e., $\|L^+\mathbf{e}_i - L^+\mathbf{e}_j\|_{G,q}^q / r_{G,p}^{1/(p-1)}(i,j)$. Also, the factor of the bound $\alpha_{G,p}^q$.

since $\alpha_{G,p} = \|\|W^{1/p}CC^+W^{-1/p}\|\|_p$ involves p in the matrix as well as norm, it is somewhat difficult to how $\alpha_{G,p}$ behaves by changing p. Thus, we focus on the unweighted graph; we numerically investigate the unweighted $\|CC^+\|_p$ that is a matrix norm evaluated in Eq. (5.21). We plot $\|CC^+\|_p$ for wine, ion, and iris with the μ when k-medoids performs the best in Fig. 5.3. We use the matrix p-norm estimation algorithm proposed by [Higham, 1992]. We plot $p \to 1$ and $p \to \infty$ as the exact value. We remark on the estimation of the matrix norm by [Higham, 1992]; let ξ be the output by the estimation of the matrix norm of CC^+, then $\|\|CC^+\|\|_p / m^{|1/2-p|} \leq \xi \leq \|\|CC^+\|\|_p$. Fig. 5.3 shows that $\|CC^+\|_p$ is far lower than the worst bound in Prop. 5.6. Moreover, Fig. 5.3 shows that the estimation algorithm proposed by Higham [1992] outputs is reliable results on this problem since this follows theory in terms of $\|\|CC^+\|\|_{p'} \leq \|\|CC^+\|\|_p$ if $2 < p' < p$ or $p < p' < 2$. Next, we evaluate the quality of the approximation comparing to Thm. 5.4. To do so, we would like to compute the ratio of approximation to the exact p-resistance as $\|L^+\mathbf{e}_i - L^+\mathbf{e}_j\|_{G,q}^q / r_{G,p}^{1/(p-1)}(i,j)$. Using Thm. 5.4, this ratio can be theoretically evaluated as

$$1 \leq \frac{\|L^+\mathbf{e}_i - L^+\mathbf{e}_j\|_{G,q}^q}{r_{G,p}^{1/(p-1)}(i,j)} \leq \alpha_{G,p}^q, \alpha_{G,p} := \|\|W^{1/p}CC^+W^{-1/p}\|\|_p. \tag{5.19}$$

This experiment also aims to evaluate this inequality. We numerically computed the approximated p-resistance, exact p-resistance, and $\alpha_{G,p}$. To compute $\alpha_{G,p}$, we use the same algorithm for the Table. 5.3. To compute the approximated and exact p-resistance, we conducted with the following procedure. To create a graph, we used $\mu = 0.1$, in order to make k-nn graph.

This means that we use $k = \lfloor 0.1n \rfloor$. To make the comparison simple, we use an unweighted graph. This is because if we incorporate weights, it is not trivial how $\alpha_{G,p}$ behaves, and thus, the results might not be a consistent of a comparison among different ps. The rest of the analysis was carried in the same procedure as Table 5.4. We perform this experiment for $p \in \{1.4, 1.7, 2, 2.3, 2.6, 2.9, 5, 10\}$. We cannot perform too small and too large p cases since we observed the dominance of the numerical errors to obtain the exact ones. In fact, this phenomenon is similar to what we observed in the main experiment. Thus, we omitted them from the table. The result is summarized in Fig. 5.4. We can see that the all the ratios are in the bound of Thm. 5.4. We also remark that using Prop. 5.6 we have the bound as

$$\alpha_{G,p}^q \leq m^{|q/2-1|}, \tag{5.20}$$

all of the plots of $\|L^+\mathbf{e}_i - L^+\mathbf{e}_j\|_{G,q}^q / r_{G,p}^{1/(p-1)}(i,j)$ in Fig. 5.4 is obviously far lower than this bound. For this result we observe the looser bound for larger p, since $\|\|CC^+\|\|_p \leq \|\|CC^+\|\|_\infty$ assuming an unweighted graph. By incorporating the weight, we might observe $\alpha_{G,p}$ differently, since we do not know which is larger $\|\|W^{1/p}CC^+W^{-1/p}\|\|_p$ and $\|\|W^{1/\infty}CC^+W^{-1/\infty}\|\|_\infty$. Further, we have

$$\alpha_{G,p}^q = \|\|W^{1/p}CC^+W^{-1/p}\|\|_p^q \leq \left(\frac{w_{\max}}{w_{\min}}\right)^{q/p=q-1} \|\|CC^+\|\|_p^q. \tag{5.21}$$

Seeing the current derived bound Eq. (5.21), the bound may be looser if we involve the weights and p is small and hence q is large. A tighter bound particularly for smaller p is a possible future direction, but this might be a lower priority due to the low performance at the smaller p. The reason is that, for small p, it is known that $r_{G,p}(i,j)$ converges to a meaningless function [Alamgir and Luxburg, 2011, Slepcev and Thorpe, 2019] under certain graph building conditions. Also, possibly due to this, our method performs better for larger p.

5.7 Summary

We have proposed the multi-class clustering algorithm using the approximated p-resistance. For this purpose, we have shown the guarantee for the approximation of p-resistance. This has enabled to compute the p-resistance much faster than the naive optimization methods. We also have shown that p-resistance characterizes the solution of the semi-supervised learning

problem. Our algorithm has outperformed the existing clustering methods using the graph p-seminorm.

The limitation of this work is that we cannot exploit the sparse structures of graphs. It is because we use L^+, which becomes dense even if the graph is sparse. For future work, there remains an ample opportunity to further speed up the procedures involving pseudoinverse of graph Laplacian, such as sparsification techniques [Spielman and Srivastava, 2011, Spielman and Teng, 2014]. Moreover, instead of our approximated representation of p-resistance approach, the exciting approach is to obtain exact representation of p-resistance. We leave some discussion on the difficulty of this approach in Appendix. 5.I. Finally, it would be also interesting if we apply this p-resistance framework to the recently growing space of hypergraph clustering using hypergraph p-Laplacians [Hein et al., 2013, Saito et al., 2018, Li and Milenkovic, 2018].

Appendices for Chapter 5

In the following sections, we provide the omitted proofs, additional discussions for Chapter 5.

5.A Additional Definitions for Proofs

First, we make an additional note for an intuition behind the analog between graph and electric circuit. In this analog, a vertex is a point at a circuit, and an edge is a resistor with resistance $1/a_{ij}$. A flow over a graph mapped to a current, and a distribution over V as $\mathbf{x}$ is seen as a potential at each vertex point. For the equations Eq. (2.70), the energy is defined as a sum of the inverse of resistance times square of the difference of the potential. The effective resistance between i and j is computed as follows; we inverse the energy that is minimized with the constraint that the difference of potential between i and j is unit. Given an electrical network the effective resistance between two vertices is the voltage difference needed to induce a unit "current" flow between the vertices i.e., it is resistance measured across the vertices.

Next, on top of the image for a matrix $M \in \mathbb{R}^{n_1 \times n_2}$, $\mathrm{Im}(M)$, we also define a *kernel*[1] of M, which is a subclass of $\mathbb{R}^n$, as

$$\mathrm{Ker}(M) := \{\mathbf{x} | M\mathbf{x} = 0, \mathbf{x} \in \mathbb{R}^n\}. \tag{5.22}$$

From the elementary result in the linear algebra area, we note that

$$\mathrm{Im}(M)^\perp = \mathrm{Ker}(M^\top), \tag{5.23}$$

where $\mathrm{Im}(M)^\perp$ is an orthogonal space to $\mathrm{Im}(M)$.

The matrix norm is *submultiplicative*, i.e., $|||M_1 M_2|||_p \leq |||M_1|||_p |||M_2|||_p$ whenever a product of matrices $M_1 M_2$ can be defined. A matrix norm is shown to be bounded as follows;

Lemma 5.9 (Higham [1992])**.** *For a square matrix* $M \in \mathbb{R}^{n_1 \times n_1}$, $\|M\|_p \leq n_1^{|1/2-1/p|} \|M\|_2$.

Lemma 5.10 (Higham [1992])**.** *For a matrix* $M \in \mathbb{R}^{n_1 \times n_2}$, $\|M\|_p \leq \max(\|M\|_1, \|M\|_\infty)$.

[1]This kernel is a linear algebraic kernel, not a kernel function which often appears in the machine learning context.

We elaborate more on Lemma 5.10. For a symmetric matrix, since we have

$$|||M|||_1 = |||M|||_\infty = \max_j \sum_i |m_{ij}|. \tag{5.24}$$

From the Lemma 5.10 and Eq.(5.24), for a symmetric matrix M, we have

$$|||M|||_p \leq |||M|||_1 = |||M|||_\infty. \tag{5.25}$$

By this we can bound $|||M|||_p$ by 1 or infinity norm of the matrix M.

An operator weighted matrix norm is defined for any matrix $M \in \mathbb{R}^{n_1 \times n_2}$ and weights $\mathbf{r}$ as

$$|||M|||_{\mathbf{r},p} := \sup_{\mathbf{x} \in \mathbb{R}^{n_2}} \frac{\|M\mathbf{x}\|_{\mathbf{r},p}}{\|\mathbf{x}\|_{\mathbf{r},p}}. \tag{5.26}$$

Recall that

$$\|\mathbf{x}\|_{\mathbf{r},p} = \|R^{1/p}\mathbf{x}\|_p, \tag{5.27}$$

where R is a diagonal matrix whose diagonal element is a weight of the norm.

From this definition, we can rewrite $|||M|||_{\mathbf{r},p}$ as

$$\begin{aligned}
|||M|||_{\mathbf{r},p} &= \sup_{\mathbf{x} \in \mathbb{R}^{n_2}} \frac{\|M\mathbf{x}\|_{\mathbf{r},p}}{\|\mathbf{x}\|_{\mathbf{r},p}} \\
&= \sup_{\mathbf{x} \in \mathbb{R}^{n_2}} \frac{\|R^{1/p}M\mathbf{x}\|_p}{\|R^{1/p}\mathbf{x}\|_p} && (5.28) \\
&= \sup_{\mathbf{x}' := R^{1/p}\mathbf{x}, \mathbf{x} \in \mathbb{R}^{n_2}} \frac{\|R^{1/p}MR^{-1/p}\mathbf{x}'\|_p}{\|\mathbf{x}'\|_p} && (5.29) \\
&= \sup_{\mathbf{x}' \in \mathbb{R}^{n_2}} \frac{\|R^{1/p}MR^{-1/p}\mathbf{x}'\|_p}{\|\mathbf{x}'\|_p} && (5.30) \\
&= \sup_{\mathbf{x} \in \mathbb{R}^{n_2}} \frac{\|R^{1/p}MR^{-1/p}\mathbf{x}\|_p}{\|\mathbf{x}\|_p} && (5.31) \\
&= |||R^{1/p}MR^{-1/p}|||_p, && (5.32)
\end{aligned}$$

5.B Proof of Proposition 5.3

For the proof we divide the proof into lower bound, upper bound and equal condition.

5.B.1 Lower Bound

In the following we give a proof of this Proposition. From Hölder's inequality, we have

$$\langle \mathbf{x}, \mathbf{y} \rangle_L \leq \|\mathbf{x}\|_{G,p} \|\mathbf{y}\|_{G,q} \tag{5.33}$$

Assuming $\langle \mathbf{x}, \mathbf{y} \rangle_L = 1$, we have

$$1 \leq \|\mathbf{x}\|_{G,p} \|\mathbf{y}\|_{G,q}, \tag{5.34}$$

and hence

$$\|\mathbf{y}\|_{G,q}^{-p} \leq \|\mathbf{x}\|_{G,p}^{p}, \tag{5.35}$$

which proves the lower bound.

5.B.2 Upper Bound

This section proves the upper bound of Prop. 5.3.

Recall the variable of the minimization problem in Eq. (5.10) is $\mathbf{x}$. If we prove that when $\mathbf{x} = \mathbf{z}$, this $\mathbf{z}$ satisfies the condition in the minimization problem $\langle \mathbf{z}, \mathbf{y} \rangle_L = 1$, from the minimization problem nature we can prove the upper bound. For this strategy, we use the following lemma.

Lemma 5.11. *For $\boldsymbol{\alpha} \in \mathbb{R}^n$ and $\boldsymbol{\beta} \in \mathbb{R}^m$, we have*

$$\langle C\boldsymbol{\alpha}, \boldsymbol{\beta} \rangle_{\mathbf{w}} = \langle C\boldsymbol{\alpha}, \boldsymbol{\beta}' \rangle_{\mathbf{w}} \tag{5.36}$$

where

$$\boldsymbol{\beta} := \boldsymbol{\beta}' + \boldsymbol{\beta}'', \quad \boldsymbol{\beta}' := CC^{+}\boldsymbol{\beta}, \boldsymbol{\beta}'' := (I - CC^{+})\boldsymbol{\beta} \tag{5.37}$$

For readability, we move the proof of Lemma 5.11 to Sec. 5.B.4.2.

By using Lemma 5.11, we now prove the upper bound.

Note that

$$\frac{f_{q/p}(C\mathbf{y})}{\|C\mathbf{y}\|_{\mathbf{w},q}^{q}} = CC^{+}\frac{f_{q/p}(C\mathbf{y})}{\|C\mathbf{y}\|_{\mathbf{w},q}^{q}} + (I - CC^{+})\frac{f_{q/p}(C\mathbf{y})}{\|C\mathbf{y}\|_{\mathbf{w},q}^{q}}. \tag{5.38}$$

Recall that we define as

$$\mathbf{z} := C^{+}\frac{f_{q/p}(C\mathbf{y})}{\|\mathbf{y}\|_{G,q}^{q}} = C^{+}\frac{f_{q/p}(C\mathbf{y})}{\|C\mathbf{y}\|_{\mathbf{w},q}^{q}}, \quad (f_{\theta}(\mathbf{x}))_i := \operatorname{sgn}(x_i)|x_i|^{\theta}. \tag{5.39}$$

By using this relation and Lemma 5.11, we have

$$\begin{aligned}
\langle \mathbf{z}, \mathbf{y} \rangle_L &= \langle C\mathbf{z}, C\mathbf{y} \rangle_{\mathbf{w}} && (5.40)\\
&= \langle C\mathbf{y}, C\mathbf{z} \rangle_{\mathbf{w}} && (5.41)\\
&= \langle C\mathbf{y}, CC^{+}\frac{f_{q/p}(C\mathbf{y})}{\|C\mathbf{y}\|_{\mathbf{w},q}^{q}} \rangle_{\mathbf{w}} && (5.42)\\
&= \langle C\mathbf{y}, \frac{f_{q/p}(C\mathbf{y})}{\|C\mathbf{y}\|_{\mathbf{w},q}^{q}} \rangle_{\mathbf{w}} && (5.43)\\
&= 1. && (5.44)
\end{aligned}$$

From Eq. (5.42) to Eq. (5.43) we apply Lemma 5.11. The last equality comes from the same nature of Eq. (5.57) in Prop. 5.13, which we will discuss in Sec. 5.B.4.1.

From the discussion above, $\mathbf{z}$ satisfies the condition of the minimization problem of Eq. (5.10). Therefore, we obtain

$$\min_{\mathbf{x}} \{ \|\mathbf{x}\|_{G,p}^{p} \text{ s.t. } \langle \mathbf{y}, \mathbf{x} \rangle_L = 1 \} \leq \|\mathbf{z}\|_{G,p}^{p} \tag{5.45}$$

5.B.3 Proof of the Equal Condition

Finally, we turn into the equality condition. We obtain the following lemma.

Lemma 5.12. *For any p, q such that $1/p + 1/q = 1$, we have*

$$\min_{\mathbf{x}} \{ \|\mathbf{x}\|_{G,p}^{p} \text{s.t. } \langle \mathbf{x}, \mathbf{y} \rangle_L = 1 \} = \|\mathbf{z}\|_{G,q}^{p} = \|\mathbf{y}\|_{G,q}^{-p}, \tag{5.46}$$

where

$$\mathbf{z} := C^{+}\frac{f_{q/p}(C\mathbf{y})}{\|\mathbf{y}\|_{G,q}^{q}} = C^{+}\frac{f_{q/p}(C\mathbf{y})}{\|C\mathbf{y}\|_{\mathbf{w},q}^{q}}, \tag{5.47}$$

when $f_{q/p}(C\mathbf{y}) \in \mathrm{Im}(C)$

The proof of Lemma 5.12 is given in Sec. 5.B.4.3

5.B.4 Proof of Lemma 5.11 and Lemma 5.12

This section provides proofs for Lemma 5.11 and Lemma 5.12. These lemmas are critical components for the proof of Prop. 5.3. In order to enhance the readability, we gather proofs for these claims in this section. We first give auxiliary lemmas, that hold for the general setting. Then, using these auxiliary lemmas, we provide the proofs for Lemma 5.11 and Lemma 5.12.

5.B.4.1 Auxiliary Lemmas

This section provides auxiliary lemmas. We start with the following claim.

Proposition 5.13. *For any* $p, q > 1$ *such that* $1/p + 1/q = 1$*, we have*

$$\min_{\mathbf{x}}\{\|\mathbf{x}\|_{\mathbf{r},p}^{p} \text{ s.t. } \langle \mathbf{y}, \mathbf{x}\rangle_{\mathbf{r}} = 1\} = \|\mathbf{y}\|_{\mathbf{r},q}^{-p} \tag{5.48}$$

Proof. Using the Hölder's inequality, we get

$$\|\mathbf{y}\|_{\mathbf{r},q}\|\mathbf{x}\|_{\mathbf{r},p} \geq \langle \mathbf{x}, \mathbf{y}\rangle_{\mathbf{r}} \tag{5.49}$$

Assuming $\langle \mathbf{x}, \mathbf{y}\rangle_{\mathbf{r}} = 1$, we can rearrange as

$$\|\mathbf{x}\|_{\mathbf{r},p} \geq \|\mathbf{y}\|_{\mathbf{r},q}^{-1}. \tag{5.50}$$

Now we consider when the minimum of the right hand side of Eq. (5.50). The minimum with the assumption $\langle \mathbf{y}, \mathbf{x}\rangle_{\mathbf{r}} = 1$ is achieved when $\mathbf{x} = \boldsymbol{\zeta}$ such that

$$\boldsymbol{\zeta} := \frac{f_{q/p}(\mathbf{y})}{\|\mathbf{y}\|_{\mathbf{r},q}^{q}}, \quad (f_{\theta}(\mathbf{y}))_i := \mathrm{sgn}(y_i)|y_i|^{\theta} \tag{5.51}$$

which means

$$\zeta_i = \frac{\mathrm{sgn}(y_i)|y_i|^{q/p}}{\|\mathbf{y}\|_{\mathbf{r},q}^q}. \tag{5.52}$$

For this $\boldsymbol{\zeta}$, we compute

$$\begin{aligned}
\langle \mathbf{y}, \boldsymbol{\zeta} \rangle_{\mathbf{r}} &= \sum_{i=1}^n r_i y_i \zeta_i && (5.53)\\
&= \sum_{i=1}^n r_i y_i \frac{\mathrm{sgn}(y_i)|y_i|^{q/p}}{\|\mathbf{y}\|_{\mathbf{r},q}^q} && (5.54)\\
&= \frac{\sum_{i=1}^n r_i |y_i|^{q/p+1}}{\|\mathbf{y}\|_{\mathbf{r},q}^q} && (5.55)\\
&= \frac{\sum_{i=1}^n r_i |y_i|^{q}}{\|\mathbf{y}\|_{\mathbf{r},q}^q} && (5.56)\\
&= \frac{\|\mathbf{y}\|_{\mathbf{r},q}^q}{\|\mathbf{y}\|_{\mathbf{r},q}^q} = 1. && (5.57)
\end{aligned}$$

The transition from Eq. (5.55) to Eq. (5.56) comes from $q/p + 1 = q$. Also, we have

$$\begin{aligned}
\|\boldsymbol{\zeta}\|_{\mathbf{r},p} &= \left\| \frac{f_{q/p}(\mathbf{y})}{\|\mathbf{y}\|_{\mathbf{r},q}^q} \right\|_{\mathbf{r},p} && (5.58)\\
&= \left(\sum_{i=1}^n r_i \left| \frac{\mathrm{sgn}(y_i)|y_i|^{q/p}}{\|\mathbf{y}\|_{\mathbf{r},q}^q} \right|^p \right)^{1/p} && (5.59)\\
&= \frac{1}{\|\mathbf{y}\|_{\mathbf{r},q}^q} \left(\sum_{i=1}^n r_i \left| \mathrm{sgn}(y_i)|y_i|^{q/p} \right|^p \right)^{1/p} && (5.60)\\
&= \frac{1}{\|\mathbf{y}\|_{\mathbf{r},q}^q} \left(\sum_{i=1}^n r_i |y_i|^{q} \right)^{1/p} && (5.61)\\
&= \frac{1}{\|\mathbf{y}\|_{\mathbf{r},q}^q} \|\mathbf{y}\|_{\mathbf{r},q}^{q/p} && (5.62)\\
&= \|\mathbf{y}\|_{\mathbf{r},q}^{q/p-q} = \|\mathbf{y}\|_{\mathbf{r},q}^{-1}. && (5.63)
\end{aligned}$$

By substituting $\mathbf{x} = \boldsymbol{\zeta}$ in Eq. (5.50), the assumption $\langle \mathbf{y}, \mathbf{x} \rangle_{\mathbf{r}} = 1$ is satisfied and the equality holds. Thus, we obtain

$$\|\mathbf{x}\|_{\mathbf{r},p} \geq \|\mathbf{y}\|_{\mathbf{r},q}^{-1} \iff \|\mathbf{x}\|_{\mathbf{r},p}^p \geq \|\mathbf{y}\|_{\mathbf{r},q}^{-p}, \tag{5.64}$$

where the equality holds when $\mathbf{x} = \boldsymbol{\zeta}$. □

We bring another lemma about spaces spanned by matrices.

Lemma 5.14 ([Ben-Israel and Greville, 2003] Ex.9, §1.3, p.43 & §2.6, p.71)**.** *For a matrix $M \in \mathbb{R}^{n_1 \times n_2}$, we define a generalized inverse of matrix M denoted by $M^\dagger \in \mathbb{R}^{n_2 \times n_1}$, satisfying that*

$$MM^\dagger M = M. \tag{5.65}$$

Then,

$$\mathrm{Im}(M) = \mathrm{Im}(MM^\dagger). \tag{5.66}$$

Also,

$$S = \{\mathbf{y} : \mathbf{y} = (I - MM^\dagger)\mathbf{x}, \mathbf{x} \in \mathbb{R}^{n_1}\} \subseteq \mathrm{Im}(M)^\perp. \tag{5.67}$$

Note that the generalized inverse $M^\dagger$ is not unique. However, the pseudoinverse M^+ is unique, and also be one of generalized inverses $M^\dagger$. From this lemma, we can write as

$$\mathrm{Im}(M) = \{\mathbf{a} : \mathbf{a} = MM^\dagger\mathbf{b}, \mathbf{b} \in \mathbb{R}^{n_1}\}, \quad \mathrm{Im}(M)^\perp \supseteq \{\mathbf{a} : \mathbf{y} = (I - MM^\dagger)\mathbf{b}, \mathbf{b} \in \mathbb{R}^{n_1}\}. \tag{5.68}$$

5.B.4.2 Proof of Lemma 5.11

This section provides a proof of Lemma 5.11.

For the illustrative purpose, we start with the $\mathbf{w} = \mathbf{1}$ case. If $\mathbf{w} = \mathbf{1}$, then

$$\begin{aligned} \langle C\boldsymbol{\alpha}, \boldsymbol{\beta}\rangle_{\mathbf{w}} &= \langle C\boldsymbol{\alpha}, \boldsymbol{\beta}' + \boldsymbol{\beta}''\rangle_{\mathbf{w}} && (5.69)\\ &= \langle C\boldsymbol{\alpha}, \boldsymbol{\beta}'\rangle_{\mathbf{w}} + \langle C\boldsymbol{\alpha}, \boldsymbol{\beta}''\rangle_{\mathbf{w}} && (5.70)\\ &= \langle C\boldsymbol{\alpha}, \boldsymbol{\beta}'\rangle_{\mathbf{w}}. && (5.71) \end{aligned}$$

The last equality follows for the following reason. By composition, $C\boldsymbol{\alpha} \in \mathrm{Im}(C)$. Also, from Lemma 5.14, $\boldsymbol{\beta}'' \in \mathrm{Im}(C)^\perp$. Hence, $C\boldsymbol{\alpha}$ and $\boldsymbol{\beta}''$ are orthogonal to each other and we get

$\langle C\boldsymbol{\alpha}, \boldsymbol{\beta}'' \rangle_{\mathbf{w}} = 0$.

We now turn into the case where $\mathbf{w}$ is arbitrary. Things are less trivial when we introduce the weight. Thus, we further analyze the weighted inner product.

Since the matrix W is a full rank diagonal matrix, we obtain

$$W^{1/2}C(C^{+}W^{-1/2})W^{1/2}C = W^{1/2}CC^{+}C = W^{1/2}C, \tag{5.72}$$

and thus $C^{+}W^{-1/2}$ is a generalized inverse of $W^{1/2}C$, i.e.,

$$(W^{1/2}C)^{\dagger} = C^{+}W^{-1/2}. \tag{5.73}$$

Also, since W is a full rank diagonal matrix,

$$\{\mathbf{b} : \mathbf{b} = W^{-1/2}\mathbf{a}, \mathbf{a} \in \mathbb{R}^m\} = \{\mathbf{b}' : \mathbf{b}' \in \mathbb{R}^m\} = \mathbb{R}^m. \tag{5.74}$$

Using these relations and Lemma 5.14, we get

$$\begin{aligned}
\mathrm{Im}(W^{1/2}C) &= \{\mathbf{b} : \mathbf{b} = W^{1/2}C(W^{1/2}C)^{\dagger}\mathbf{a}, \mathbf{a} \in \mathbb{R}^m\} && (5.75)\\
&= \{\mathbf{b} : \mathbf{b} = W^{1/2}CC^{+}W^{-1/2}\mathbf{a}, \mathbf{a} \in \mathbb{R}^m\} && (5.76)\\
&= \{\mathbf{b} : \mathbf{b} = W^{1/2}CC^{+}\mathbf{a}', \mathbf{a}' \in \mathbb{R}^m\}, && (5.77)
\end{aligned}$$

where we use Eq. (5.73) for Eq. (5.76) and we use Eq. (5.74) for Eq. (5.77). Moreover, we have

$$\begin{aligned}
\mathrm{Im}(W^{1/2}C)^{\perp} &\supseteq \{\mathbf{b} : \mathbf{b} = (I - W^{1/2}C(W^{1/2}C)^{\dagger})\mathbf{a}, \mathbf{a} \in \mathbb{R}^m\} && (5.78)\\
&= \{\mathbf{b} : \mathbf{b} = (I - W^{1/2}CC^{+}W^{-1/2})\mathbf{a}, \mathbf{a} \in \mathbb{R}^m\} && (5.79)\\
&= \{\mathbf{b} : \mathbf{b} = W^{1/2}(I - CC^{+})W^{-1/2}\mathbf{a}, \mathbf{a} \in \mathbb{R}^m\} && (5.80)\\
&= \{\mathbf{b} : \mathbf{b} = W^{1/2}(I - CC^{+})\mathbf{a}', \mathbf{a}' \in \mathbb{R}^m\}, && (5.81)
\end{aligned}$$

where we use Eq. (5.73) for Eq. (5.79) and we use Eq. (5.74) for Eq. (5.81). Therefore,

$$\langle C\boldsymbol{\alpha}, \boldsymbol{\beta} \rangle_{\mathbf{w}} = \langle W^{1/2}C\boldsymbol{\alpha}, W^{1/2}\boldsymbol{\beta} \rangle \tag{5.82}$$

$$= \langle W^{1/2} C\boldsymbol{\alpha}, W^{1/2}\boldsymbol{\beta}' \rangle + \langle W^{1/2} C\boldsymbol{\alpha}, W^{1/2}\boldsymbol{\beta}'' \rangle \tag{5.83}$$

$$= \langle W^{1/2} C\boldsymbol{\alpha}, W^{1/2} CC^{+}\boldsymbol{\beta} \rangle + \langle W^{1/2} C\boldsymbol{\alpha}, W^{1/2}(I - CC^{+})\boldsymbol{\beta} \rangle \tag{5.84}$$

$$= \langle W^{1/2} C\boldsymbol{\alpha}, W^{1/2} CC^{+}\boldsymbol{\beta} \rangle. \tag{5.85}$$

$$= \langle W^{1/2} C\boldsymbol{\alpha}, W^{1/2}\boldsymbol{\beta}' \rangle. \tag{5.86}$$

$$= \langle C\boldsymbol{\alpha}, \boldsymbol{\beta}' \rangle_{\mathbf{w}} \tag{5.87}$$

The line Eq. (5.85) follows because from Eq. (5.81) the $W^{1/2}(I - CC^{+})\boldsymbol{\beta} \in \mathrm{Im}(W^{1/2}C)^{\perp}$ and therefore $W^{1/2}(I - CC^{+})\boldsymbol{\beta}$ is orthogonal to $W^{1/2}C\boldsymbol{\alpha}$, which induces $\langle W^{1/2}C\boldsymbol{\alpha}, W^{1/2}(I - CC^{+})\boldsymbol{\beta} \rangle = 0$.

Eq. (5.87) concludes the proof.

5.B.4.3 Proof of Lemma 5.12

This section proves Lemma 5.12.

If $f_{q/p}(C\mathbf{y}) \in \mathrm{Im}(C)$, then

$$C\mathbf{z} = CC^{+} \frac{f_{q/p}(C\mathbf{y})}{\|\mathbf{y}\|_{G,q}^{q}} \tag{5.88}$$

$$= \frac{f_{q/p}(C\mathbf{y})}{\|\mathbf{y}\|_{G,q}^{q}} \tag{5.89}$$

$$= \frac{f_{q/p}(C\mathbf{y})}{\|C\mathbf{y}\|_{\mathbf{w},q}^{q}} \tag{5.90}$$

From Eq. (5.88) to Eq. (5.89), we use the following relation; for a vector $\mathbf{a} \in \mathrm{Im}(C)$ we have $\mathbf{a} = CC^{+}\mathbf{a}$ since CC^{+} is an orthogonal projection onto the space $\mathrm{Im}(C)$. Eq. (5.90) is a form of Eq. (5.51), and thus from Prop. 5.13, Eq. (5.90) satisfies the equality condition of the Hölder's inequality as

$$\|C\mathbf{z}\|_{\mathbf{w},p}^{p} = \|C\mathbf{y}\|_{\mathbf{w},q}^{-p} \iff \|\mathbf{z}\|_{G,p}^{p} = \|\mathbf{y}\|_{G,q}^{-p}, \tag{5.91}$$

where we use the definition of the graph p-seminorm. Thus, we obtain the claim.

5.B.5 Remark on the Constraints

This section provides detailed explanation for the constraints of Lemma 5.2 and Prop. 5.3. We first note that the trick in this transformation is as same as done in [Herbster and Pontil,

2006, Klein and Randić, 1993]. The elaboration here follows these earlier works.

Using the reproducing property Eq. (2.67) as done in [Herbster and Pontil, 2006, Klein and Randić, 1993], the constraints of 2-resistance (and also for p-resistance Eq. (2.74)) can be rewritten as

$$1 = x_i - x_j = \langle \mathbf{x}, L^+\mathbf{e}_i - L^+\mathbf{e}_j \rangle_L. \tag{5.92}$$

Now, since $L\mathbf{1} = 0$, there exists $c \in \mathbb{R}$ such that $\mathbf{x} - c\mathbf{1} \in \mathcal{H}(L)$. We now define as $\mathbf{x}' := \mathbf{x} - c\mathbf{1}$. We then compute

$$\begin{aligned} \langle \mathbf{x}', L^+\mathbf{e}_i \rangle_L &= \mathbf{x}'^\top L L^+ \mathbf{e}_i && (5.93)\\ &= \mathbf{x}'^\top \mathbf{e}_i && (5.94)\\ &= x'_i. && (5.95) \end{aligned}$$

The second line follows since we have $LL^+\mathbf{x}' = \mathbf{x}'$ for $\mathbf{x}' \in \mathcal{H}(L)$. Note that this computation is same as the reproducing kernel property characteristics Eq. (2.67). Also, the definition of $\mathbf{x}'$ immediately leads to

$$L\mathbf{x}' = L(\mathbf{x} - c\mathbf{1}) = L\mathbf{x} - cL\mathbf{1} = L\mathbf{x}, \tag{5.96}$$

and thus for $\mathbf{u} \in \mathbb{R}^n$ we have

$$\begin{aligned} \langle \mathbf{x}', \mathbf{u} \rangle_L &= \mathbf{x}'^\top L \mathbf{u} && (5.97)\\ &= \mathbf{x}^\top L \mathbf{u} && (5.98)\\ &= \langle \mathbf{x}, \mathbf{u} \rangle_L && (5.99) \end{aligned}$$

From these discussions, we obtain

$$\begin{aligned} 1 &= x_i - x_j && (5.100)\\ &= (x_i - c) - (x_j - c) && (5.101)\\ &= x'_i - x'_j && (5.102)\\ &= \langle \mathbf{x}', L^+\mathbf{e}_i \rangle_L - \langle \mathbf{x}', L^+\mathbf{e}_j \rangle_L && (5.103) \end{aligned}$$

$$= \langle \mathbf{x}, L^+\mathbf{e}_i \rangle_L - \langle \mathbf{x}, L^+\mathbf{e}_j \rangle_L \tag{5.104}$$

$$= \langle \mathbf{x}, L^+\mathbf{e}_i - L^+\mathbf{e}_j \rangle_L. \tag{5.105}$$

The third line follows from the definition of $\mathbf{x}'$. The fourth line follows from Eq. (5.95) and the fifth line follows from Eq. (5.99). Thus, we obtain Eq. (5.92)

5.C Proof of Theorem 5.4

In this section we prove Thm. 5.4. The general strategy is applying Prop. 5.3.

By definition,

$$r_{G,p}(i,j) = \frac{1}{\min_{\mathbf{x}} \|\mathbf{x}\|_{G,p}^p \text{ s.t. } x_i - x_j = 1}. \tag{5.106}$$

First, we rewrite the condition of the minimization problem. Using Eq. (5.92), we observe that the denominator of Eq.(2.74) can be written as

$$\min_{\mathbf{x}} \{\|\mathbf{x}\|_{G,p}^p \text{ s.t. } x_i - x_j = 1\} = \min_{\mathbf{x}} \{\|\mathbf{x}\|_{G,p}^p \text{ s.t. } \langle L^+\mathbf{e}_i - L^+\mathbf{e}_j, \mathbf{x} \rangle_L = 1\} \tag{5.107}$$

From this rewrite, we see that Eq. (5.107) is exactly same as the minimization problem of Prop. 5.3 if we substitute $\mathbf{y} := L^+(\mathbf{e}_i - \mathbf{e}_j)$. Thus, we apply Prop. 5.3 to this problem in order to obtain lower and upper bounds of Eq. (5.107).

Lower Bound of Eq. (5.107)**.** Now, we come to the lower bound of this problem Eq. (5.107). By applying the lower bound of Prop. 5.3 with substituting $\mathbf{y} := L^+(\mathbf{e}_i - \mathbf{e}_j)$, we obtain

$$\|L^+(\mathbf{e}_i - \mathbf{e}_j)\|_{G,q}^{-p} \leq \min_{\mathbf{x}} \{\|\mathbf{x}\|_{G,p}^p \text{ s.t. } \langle L^+(\mathbf{e}_i - \mathbf{e}_j), \mathbf{x} \rangle_L = 1\}. \tag{5.108}$$

This conclude the lower bound.

Upper Bound of Eq. (5.107)**.** Next, we turn to the upper bound of this problem Eq. (5.107).

We first compute

$$\|\mathbf{z}\|_{G,p} = \|C\mathbf{z}\|_{\mathbf{w},p} = \left\| \frac{CC^+ f_{q/p}(C\mathbf{y})}{\|C\mathbf{y}\|_{\mathbf{w},q}^q} \right\|_{\mathbf{w},p} \tag{5.109}$$

$$= \frac{\left\|CC^+ f_{q/p}(C\mathbf{y})\right\|_{\mathbf{w},p}}{\|C\mathbf{y}\|_{\mathbf{w},q}^q} \tag{5.110}$$

$$\leq |||CC^{+}|||_{\mathbf{w},p} \frac{\|f_{q/p}(C\mathbf{y})\|_{\mathbf{w},p}}{\|C\mathbf{y}\|^{q}_{\mathbf{w},q}} \tag{5.111}$$

$$= |||CC^{+}|||_{\mathbf{w},p} \|C\mathbf{y}\|^{-1}_{\mathbf{w},q} \tag{5.112}$$

$$= |||W^{1/p}CC^{+}W^{-1/p}|||_{p} \|C\mathbf{y}\|^{-1}_{\mathbf{w},q} \tag{5.113}$$

$$= |||W^{1/p}CC^{+}W^{-1/p}|||_{p} \|\mathbf{y}\|^{-1}_{G,q} \tag{5.114}$$

$$= \alpha_{G,p} \|\mathbf{y}\|^{-1}_{G,q}, \tag{5.115}$$

where we recall that we defined as $\alpha_{G,p} := |||W^{1/p}CC^{+}W^{-1/p}|||_{p}$. The transformation from Eq. (5.110) to Eq. (5.111) follows from the submultiplicative characteristics of the matrix norm discussed in Sec. 5.2. The equality from Eq. (5.111) to Eq. (5.112) holds due to the same discussion as Eq. (5.63) in Prop. 5.13, which we discussed in Sec. 5.B.4.1. The transformation from Eq. (5.112) to Eq. (5.113) follows from a characteristics of the weighted matrix norm discussed in Eq. (5.30). Hence, by taking the p-th power of the inequality Eq. (5.115) and observing that we substitute $\mathbf{y} := L^{+}(\mathbf{e}_i - \mathbf{e}_j)$, we obtain

$$\|\mathbf{z}\|^{p}_{G,p} \leq \alpha^{p}_{G,p} \|L^{+}(\mathbf{e}_i - \mathbf{e}_j)\|^{-p}_{G,q} \tag{5.116}$$

Thus, from Prop. 5.3 and the inequality Eq. (5.115) we get

$$\min_{\mathbf{x}} \{\|\mathbf{x}\|^{p}_{G,p} \text{ s.t. } \langle L^{+}(\mathbf{e}_i - \mathbf{e}_j), \mathbf{x}\rangle_{L} = 1\} \leq \|\mathbf{z}\|^{p}_{G,p} \leq \alpha^{p}_{G,p} \|L^{+}(\mathbf{e}_i - \mathbf{e}_j)\|^{-p}_{G,q}. \tag{5.117}$$

Combining Lower and Upper Bounds of Eq. (5.107)**.** We now combine the lower bound Eq. (5.108) and the upper bound Eq. (5.117). By combining these two and using Eq. (5.107), we get

$$\begin{aligned} &\|L^{+}\mathbf{e}_i - L^{+}\mathbf{e}_j\|^{-p}_{G,q} \leq \min_{\mathbf{x}} \{\|\mathbf{x}\|^{p}_{G,p} \text{ s.t. } \langle L^{+}(\mathbf{e}_i - \mathbf{e}_j), \mathbf{x}\rangle_{L} = 1\} \leq \alpha^{p}_{G,p} \|L^{+}\mathbf{e}_i - L^{+}\mathbf{e}_j\|^{-p}_{G,q} \\ \iff &\|L^{+}\mathbf{e}_i - L^{+}\mathbf{e}_j\|^{-p}_{G,q} \leq \min_{\mathbf{x}} \{\|\mathbf{x}\|^{p}_{G,p} \text{ s.t. } x_i - x_j = 1\} \leq \alpha^{p}_{G,p} \|L^{+}\mathbf{e}_i - L^{+}\mathbf{e}_j\|^{-p}_{G,q} \end{aligned} \tag{5.118}$$

For the p-effective resistance, taking the inverse we obtain

$$\frac{1}{\alpha^{p}_{G,p}} \|L^{+}\mathbf{e}_i - L^{+}\mathbf{e}_j\|^{p}_{G,q} \leq r_{G,p}(i,j) \leq \|L^{+}\mathbf{e}_i - L^{+}\mathbf{e}_j\|^{p}_{G,q}. \tag{5.119}$$

5.D Proof of Theorem 5.5

This section proves Thm. 5.5. The proof here is a special case of Thm. 5.4.

For the incidence matrix of tree, $\mathrm{rank}(C) = n-1$ [Bapat, 2010]. Hence $\mathrm{Im}(C) = \mathbb{R}^{n-1}$. Thus, $f_{q/p}(C\mathbf{y}) \in \mathbb{R}^{n-1} = \mathrm{Im}(C)$. Using the Lemma 5.12 and substituting $\mathbf{y} = L^+\mathbf{e}_i - L^+\mathbf{e}_j$,

$$\min_{\mathbf{x}}\{\|\mathbf{x}\|_{G,p}^p \text{s.t. } \langle \mathbf{x}, L^+\mathbf{e}_i - L^+\mathbf{e}_j\rangle_L = 1\} = \|L^+\mathbf{e}_i - L^+\mathbf{e}_j\|_{G,q}^{-p}. \tag{5.120}$$

Recall that the minimization problem of Eq. (5.120) is the inverse of the p-resistance. Therefore, Eq. (5.120) leads to the claim.

5.E Proof of Proposition. 5.6

We recall that by definition of pseudoinverse, we have

$$|||CC^+|||_2 = 1, \tag{5.121}$$

since the eigenvalues of CC^+ is either 0 or 1. Also, for any matrix M and any invertible matrix P, PMP^{-1} and M share the same eigenvalues. By construction, W is also an invertible matrix. Thus, using Lemma 5.9, we obtain

$$\begin{aligned}
\alpha_{G,p} &= |||W^{1/p}CC^+W^{-1/p}|||_p && (5.122)\\
&\leq m^{|1/2-1/p|}|||W^{1/p}CC^+W^{-1/p}|||_2 && (5.123)\\
&= m^{|1/2-1/p|}. && (5.124)
\end{aligned}$$

5.F Proof of the Cut Results of Illustrative Examples Fig. 5.1

This section explains illustrative examples of clustering via p-resistance where p plays a role.

5.F.1 Preliminaries for Illustrative Examples

Before we discuss the details of the clustering, we setup preliminaries. We now setup the notions on the graph metrics. First, a *st-mincut* is defined as the minimum cut between the vertices s and t, i.e.,

$$\min_{V'} \mathrm{Cut}(s,t) := \min_{V'} \sum_{i\in V', j\in V'\setminus V | s\in V', t\in V'\setminus V} a_{ij}. \tag{5.125}$$

The act of the "cut" of the edges is defined to divide into two graphs so that the vertex s belongs to one and the vertex t belongs to the other. The minimum cut is that we want such a cut so that the sum of the weight of the edges to be cut is minimized.

Now, we also define the shortest path between vertices s and t is defined as

$$\min_{\mathbf{i}} \sum_{\ell \in E} w_\ell i_\ell \text{ s.t. } \mathbf{i} = (i_\ell)_{\ell \in E} \text{ unit flow from } i \text{ to } j, \tag{5.126}$$

where $\mathbf{i} \in \{0,1\}^m$. The shortest path problem is to finding the path with smallest sum of the weights of edges between s and t.

In the following, we show that p-resistance is connection with st-mincut and the shortest path. We recall the theorem in [Alamgir and Luxburg, 2011] as

Proposition 5.15 (Alamgir and Luxburg [2011])**.** *Consider a p-flow problem as*

$$F_{G,p}(i,j) := \min_{\mathbf{i}} \sum_{\ell \in E} w_\ell^{1-p} i_\ell^p \text{ s.t. } \mathbf{i} = (i_\ell)_{\ell \in E} \text{ unit flow from } i \text{ to } j, \tag{5.127}$$

where $\mathbf{i} \in \mathbb{R}^{+m}$ is a current at edges. Then, for $1/p + 1/q = 1$, we have

$$r_{G,p}^{1/(p-1)}(i,j) = F_{G,q}(i,j). \tag{5.128}$$

We first remark that $\mathbf{i}$ in q-flow problem is non-negative real value whereas $\mathbf{i}$ for the shortest path is either 0 or 1. We remark that when $p \to \infty$, q goes to 1 and q-flow problem is a simple shortest path flow problem.

This proposition means that the $1/(p-1)$-th power of p-resistance is equivalent to the q-flow. From this proposition, we now see the connection between p-resistance, and st-mincut and shortest path as follows.

- When $p \to 1$, p-resistance between s and t is 1/st-mincut.
- When $p \to \infty$, $1/(p-1)$-th power of the p-resistance is the discrete shortest path of the *unweighted* graph.

Thus, we intuitively characterize the p-resistance as

- When p is small, p-resistance more focus on a minimum cut.

- When p is large, p-resistance more focus on the "path", and also more focus on the "unweighted topology".

We next formulate the clustering problem as follows. We use the k-center algorithm using p-resistance as a metric as

$$C^*_{G,p} := \min_{v_1^*, v_2^* \in V} \max_{v \in V} \min_{i \in \{1,2\}} r^{1/(p-1)}_{G,p}(v, v_i^*), \tag{5.129}$$

where $\{v_1^*, v_2^*\}$ is a minimizer. Since when $p \to 1$ and $r^{1/(p-1)}_{G,p} > 0$, then $r^{1/(p-1)}_{G,p} \to \infty$ and therefore Eq. (5.129) cannot be used. In this case, we note that the following relation that is

$$x < y \iff x^{1/(p-1)} < y^{1/(p-1)} \tag{5.130}$$

we have

$$C^{*p-1}_{G,p} := \min_{v_1^*, v_2^* \in V} \max_{v \in V} \min_{i \in \{1,2\}} r_{G,p}(v, v_i^*). \tag{5.131}$$

Thus, we simply use the comparison of $r_{G,p}$ instead of $r^{(1/(p-1))}_{G,p}$ when $p \to 1$. We finally remark that Herbster [2010] showed that when $p \to 1$ the triangle inequality still holds, i.e.,

$$r_{G,p\to 1}(i,j) \le r_{G,p\to 1}(i,\ell) + r_{G,p\to 1}(\ell,j). \tag{5.132}$$

5.F.2 Illustrative Examples of Clustering via p-Resistance

Now, we discuss the examples in Fig. 5.1. We give notations as in Fig. 5.5. We denote by (V_{ij}, E_{ij}) the vertices and edges of the graph G_{ij}. We also give the example where the weight matters and its notation in Fig. 5.6.

5.F.2.1 The Case of G_1

For the case of $p \to 1$, since p-resistance is the 1 over min-cut, we have for $j > i$

$$r_{G,p}(i,j) = \begin{cases} 1 & i = 1 \text{ and } j \in V \backslash \{1\} \\ 1/5 & i,j \in V_{12} \text{ or } i,j \in V_{13} \\ 1/4 & i \in \{5,6\}, j \in \{7,8\} \end{cases} \tag{5.133}$$

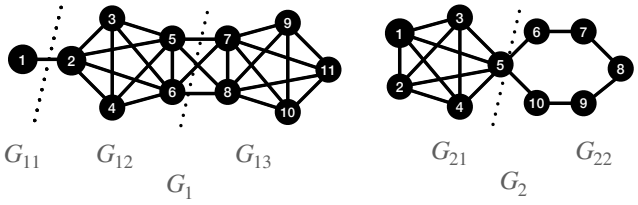


Figure 5.5: The notations of illustrative example graphs. In the graph G_2 the vertex 5 is in both G_{21} and G_{22}.

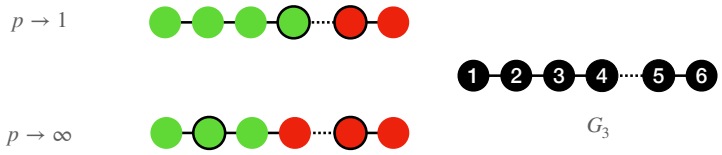


Figure 5.6: The illustrative example of a weighted graph and its notations. The weights of edge drawn in the line are 1, whereas weight of the dotted line is $\epsilon \ll 1$. The other drawing rule follows Fig. 5.1. In the example, we observe that we focus on the difference of the weight when $p \to 1$, while we ignore the weight when $p \to \infty$. For this example, "more natural result" depends on the perspective. If we look at the cut, the more natural result is obtained when $p \to 1$. If we look at the path-based topology, we obtain the natural result when $p \to \infty$. Details in Appendix 5.F.

Note that $r_{G,p}(i,j) = r_{G,p}(j,i)$. By using this p-resistance, the set satisfying Eq. (5.131) is $v_1^* = 1$ and $v_2^* \in V_{12} \cup V_{13}$. This is because if we do not take $v_1^* = 1$,

$$\min_{v_1^*, v_2^* \in V} \max_{v \in V} \min_{i \in \{1,2\}} r_{G,p}(v, v_i^*) = 1, \tag{5.134}$$

which is the maximum of the weight of edges of G.

For $p \to \infty$, since p-resistance is a shortest path, we have for $j > i$

$$r_{G,p}^{1/(p-1)}(i,j) = \begin{cases} 1 & i = 1, j \in V_2 \\ 1 & i, j \in V_{12} \text{ or } i, j \in V_{13} \\ 2 & i = 1, j \in V_{13} \\ 2 & i \in V_{12}, j \in V_{13}. \end{cases} \tag{5.135}$$

Then if we set $v_1^* = 2$ and $v_2^* \in V_{13}$, we have

$$\min_{v_1^*, v_2^* \in V} \max_{v \in V} \min_{i \in \{1,2\}} r_{G,p}^{1/(p-1)}(v, v_i^*) = 1. \tag{5.136}$$

Since this is the minimum of the weight of the edge, it is clear that this set is optimal.

Coloring the vertices in the same color if the vertices are closer to the same center than the others, we obtain Fig. 5.1.

5.F.2.2 The Case of G_2

For the case of $p \to 1$, we have for $j > i$

$$r_{G,p}(i,j) = \begin{cases} 1/5 & i,j \in V_{21} \\ 1/2 & i \in V_{21}, j \in V_{22} \\ 1/2 & i,j \in V_{22}. \end{cases} \tag{5.137}$$

Then if we set $v_1^* \in V_{21}$ and $v_2^* \in V_{22}$, we have

$$\min_{v_1^*, v_2^* \in V} \max_{v \in V} \min_{i \in \{1,2\}} r_{G,p}^{1/(p-1)}(v, v_i^*) = 1/2. \tag{5.138}$$

Since $\min_{i \in V_{22},} r_{G,p}(i,j) = 1/2$, this is the best possible minimum.

For the case of $p \to \infty$, we have for $j > i$

$$r_{G,p}(i,j)^{1/(p-1)} = \begin{cases} 1 & i,j \in V_{21} \\ 2 & i \in V_{21} \backslash \{5\}, j \in \{6, 10\} \\ 3 & i \in V_{21} \backslash \{5\}, j \in \{7, 9\} \\ 4 & i \in V_{21} \backslash \{5\}, j = 8 \\ \min\{j-i, 6-(j-i)\} & i,j \in V_{22} \end{cases} \tag{5.139}$$

Then if we set $v_1^* = 5$ and $v_2^* = 8$, we have

$$\min_{v_1^*, v_2^* \in V} \max_{v \in V} \min_{i \in \{1,2\}} r_{G,p}^{1/(p-1)}(v, v_i^*) = 1. \tag{5.140}$$

Since the minimum of p-resistance is 1, this is the best possible minimum.

Coloring the vertices in the same color if the vertices are closer to the same center than

the others, we obtain Fig. 5.1.

5.F.2.3 The Case of G_3

For the case of $p \to 1$, we have for $j > i$

$$r_{G,p}(i,j) = \begin{cases} 1 & i,j \in \{1,\dots,4\} \\ 1 & i,j \in \{5,6\} \\ 1/\epsilon & i \in \{1,\dots,4\}, j \in \{5,6\}. \end{cases} \tag{5.141}$$

Then if we set $v_1^* \in \{1,\dots,4\}$ and $v_2^* \in \{5,6\}$, we have

$$\min_{v_1^*,v_2^* \in V} \max_{v \in V} \min_{i \in \{1,2\}} r_{G,p}^{1/(p-1)}(v, v_i^*) = 1. \tag{5.142}$$

Since the minimum of p-resistance is 1, this is the best possible minimum.

For the case of $p \to \infty$, we have for $j > i$

$$r_{G,p}^{1/(p-1)}(i,j) = j - i \text{ if } j > i \tag{5.143}$$

Then if we set $v_1^* = 2$ and $v_2^* = 5$, we have

$$\min_{v_1^*,v_2^* \in V} \max_{v \in V} \min_{i \in \{1,2\}} r_{G,p}^{1/(p-1)}(v, v_i^*) = 1. \tag{5.144}$$

Since the minimum of p-resistance is 1, this is the best possible minimum.

Coloring the vertices in the same color if the vertices are closer to the same center than the others, we obtain Fig. 5.1.

5.G Proof of Proposition 5.7

This section provides the proof for Prop. 5.7. In practice, we want to know how close to the exact value and how far from this upper bound the value of $\|\|W^{1/p}CC^{+}W^{-1/p}\|\|_p$ is. In the following, we argue that in the general case $\alpha_{G,p}$ is far less than the bound given in Prop. 5.6.

Before we get into the detail, we give a brief overview of an interpretation of $\alpha_{G,p}$. From the definition of $\mathbf{z}$, $\mathbf{z}$ is a mapping of $f_{q/p}(C\mathbf{y})/\|\mathbf{y}\|_{G,q}^q$ from $\mathbb{R}^m \to \mathrm{Im}(C)$. Comparing the equality condition Eq. (5.90), we observe that if $f_{q/p}(C\mathbf{y}) \in \mathrm{Im}(C)$, we obtain the tightest bound since $\|\mathbf{z}\|_{G,p} = \|\mathbf{y}\|_{G,q}^{-1}$. By looking at this, we observe that the $\alpha_{G,p}$ is the worst

possible "overflow" of the mapping from $\mathrm{Im}(C)$ from $\mathbb{R}^m$, in a sense of the weighted p-norm.

5.G.1 Bound of $\alpha_{G,p}$ for Some Specific Graphs

In this section we give a constant bound of $\alpha_{G,p}$ for some specific graphs. We now divide the proof into the complete case and the cyclic case.

5.G.1.1 Complete Case

First, we obtain the pseudoinverse of C of a complete graph.

Lemma 5.16. *For an incidence matrix C' for a complete graph,*

$$C'^{+} = \frac{1}{n} C'^{\top} \tag{5.145}$$

Proof. For a graph Laplacian L of unweighted graph can be written as

$$L = nI - \mathbf{1}^{\top}\mathbf{1}, \tag{5.146}$$

and thus

$$L_{ij} = \begin{cases} n-1 & \text{if } i = j \\ -1 & \text{if } i \neq j \end{cases}. \tag{5.147}$$

Also, we know that

$$L = C'^{\top} C'. \tag{5.148}$$

Now we consider the the vector $\mathbf{x}_{ij} \in \mathbb{R}^n$ as

$$\mathbf{x}_{ij}^{\top} := \underbrace{(0, \ldots, 0, \overbrace{1}^{i\text{th element}}, 0, \ldots, 0, \overbrace{-1}^{j\text{th element}}, 0, \ldots, 0)}_{\text{size } n}. \tag{5.149}$$

Note that this $\mathbf{x}_{ij}$ is one row of the incidence matrix C'. Now we get

$$(L\mathbf{x}_{ij})_l = \begin{cases} (n-1)\times 1 + (-1)\times(-1) = n & \text{if } l = i \\ 1\times(-1) + (n-1)\times(-1) = -n & \text{if } l = j \\ (-1)\times 1 + (-1)\times(-1) = 0 & \text{otherwise} \end{cases} \tag{5.150}$$

$$= n(L\mathbf{x}_{ij})_l. \tag{5.151}$$

Since $\mathbf{x}_{ij}$ is one column of the transpose of the incidence matrix $\mathbf{C}'^{\top}$,

$$LC = C'^{\top}C'C'^{\top} = nC'^{\top} \iff \left(\frac{1}{n}C'^{\top}\right)C'\left(\frac{1}{n}C'^{\top}\right) = \frac{1}{n}C'^{\top} \tag{5.152}$$

Also,

$$(LC)^{\top} = C'C'^{\top}C' = nC' \iff C'\left(\frac{1}{n}C'^{\top}\right)C' = C' \tag{5.153}$$

From Eq. (5.152) and Eq. (5.153), the matrix $1/nC'$ satisfies the definition of C^{+}, which leads to the claim. □

Note that $|||CC^{+}|||_1 = |||CC^{+}|||_\infty$ due to the symmetricity of CC^{+}.

$$|||CC^{+}|||_p \leq |||CC^{+}|||_\infty \leq |||C|||_\infty |||C^{+}|||_\infty = 4\frac{n+1}{n} \leq 4. \tag{5.154}$$

5.G.1.2 Cyclic Case

In the cyclic graph, $m = n$, i.e., the number of vertices is equal to the number of edges. Thus, the incidence matrix C is square. However, in order to avoid confusion, in the following we use m and n. Now, we define the incidence matrix $C \in \mathbb{R}^{m\times n}$of the cyclic graph as

$$c_{i1} = \begin{cases} -1 & \text{when } i = 1 \\ 1 & \text{when } i = 2 \\ 0 & \text{otherwise} \end{cases} \tag{5.155}$$

$$c_{i2} = \begin{cases} -1 & \text{when } i = 1 \\ 1 & \text{when } i = n \\ 0 & \text{otherwise} \end{cases} \tag{5.156}$$

$$c_{ij} = \begin{cases} -1 & \text{when } i = j-1 \\ 1 & \text{when } i = j \\ 0 & \text{otherwise for} \end{cases} \quad \text{for } j \geq 3. \tag{5.157}$$

Before we explore C^+, we introduce *cyclic shift operator* of the vector. Given the vector $\mathbf{a}$, the shift operator (l) "cyclic shifts" the element, as

$$\mathbf{a}^{(l)} = (a_{n-l+1}, a_{n-l+2}, \ldots, a_n, a_1, \ldots, a_{n-l})^\top. \tag{5.158}$$

Thus, $\mathbf{a}^{(0)} = \mathbf{a}$. Also, we define the reverse operator rev for a vector $\mathbf{a}$ as

$$\mathrm{rev}(\mathbf{a}) = (a_n, a_{n-1}, \ldots, a_1). \tag{5.159}$$

We also define the vector $\boldsymbol{\xi} \in \mathbb{R}^n$ as

$$\boldsymbol{\xi} = (1/2 - 1/2n, 1/2 - 3/2n, \ldots, 1/2 - (2i-1)/2n, \ldots, -1/2 + 1/n). \tag{5.160}$$

Now, we define a matrix B as

$$B_{1\cdot} = \boldsymbol{\xi}^{(1)} \tag{5.161}$$

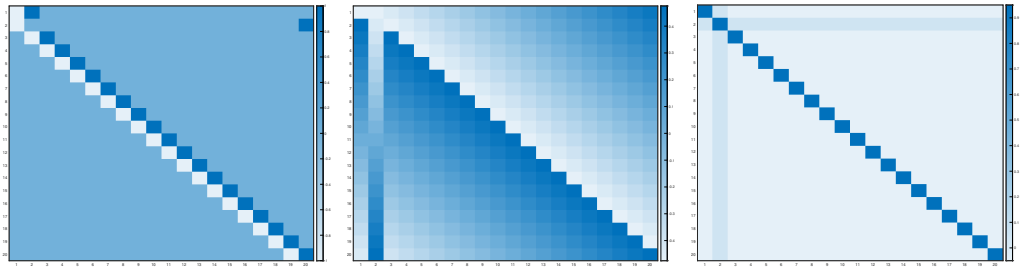
$$B_{2\cdot} = \mathrm{rev}(\boldsymbol{\xi}^{(0)}) = \mathrm{rev}(\boldsymbol{\xi}) \tag{5.162}$$

$$B_{j\cdot} = \boldsymbol{\xi}^{(j-1)} \text{for } j \geq 3, \tag{5.163}$$

where $B_{i\cdot}$ denotes i-th column of B. We plot a heatmap of C and B for the illustrative purpose.

Now we prove that $C^+ = B$. To claim that, it is enough to prove that $BC = I - \mathbf{1}^\top\mathbf{1}/n$ [Bapat, 2010]. From the construction,

$$(BC)_{ii} = \xi_1 - \xi_n = 1/2 - 1/2n - (-1/2 - 1/2n) = 1 - 1/n. \tag{5.164}$$



(a) C (b) $B = C^+$ (c) CC^+

Figure 5.7: Heatmap plot for the matrices C, $B = C^+$ and CC^+ of the cyclic graph for $n = 20$.

Also, when $i \neq j$,

$$(BC)_{ij} = \begin{cases} -\xi_{i-1} - \xi_{n-i+1} & \text{when when } j = 1 \\ \xi^{(j)}_{i+1} - \xi^{(j)}_{i} & \text{when when } 2 \leq j < n, \\ \xi_{n-i+1} + \xi_{i+1} & \text{when when } j = n, \end{cases} \tag{5.165}$$

$$= -1/n. \tag{5.166}$$

Thus, we can say that $B = C^+$. By doing a similar computation, we get

$$CC^+ = \begin{cases} 1 - 1/n & \text{when } i = j \\ 1/n & \text{when } i = 2 \text{ or } j = 2, i \neq j \\ -1/n & \text{otherwise} \end{cases} \tag{5.167}$$

We also plot a heatmap for CC^+ for the illustrative purpose in Fig. 5.7(c). Thus, applying Lemma 5.10, we get

$$\|\!|CC^+|\!\|_p \leq \|\!|CC^+|\!\|_1 = \max_i \sum_{j=1}^{n} |(CC^+)_{ij}| = 2 - 1/n \leq 4. \tag{5.168}$$

We leave a brief note for other concrete examples. Several attempts are made to obtain the concrete form of C^+ for the specific graph [Azimi and Bapat, 2018, Azimi et al., 2019]. However, due to their abstract ways to characterize the graph such as distance or cut, we think that it is hard to immediately obtain a non-trivial bound from these results. Also, C^+ for tree is studied [Bapat, 1997]. However, since we know the exact representation of p-resistance for

Table 5.5: The values of approximated 1-resistance for the graph Fig. 5.9 (a). The exact 1-resistance for this graph is $1/\delta$.

		ζ				
		5	10	20	40	80
δ	5	0.2	0.1	0.05	0.025	0.0125
	10	0.2	0.1	0.05	0.025	0.0125
	20	0.2	0.1	0.05	0.025	0.0125
	40	0.2	0.1	0.05	0.025	0.0125
	80	0.2	0.1	0.05	0.025	0.0125

Table 5.6: The values of approximated p-resistance for the graph Fig. 5.9 (b). The exact 1-resistance for this graph is $1/(\delta+1)$.

		ζ				
		5	10	20	40	80
δ	5	0.46	0.33	0.21	0.13	0.07
	10	0.61	0.48	0.33	0.21	0.12
	20	0.75	0.64	0.49	0.33	0.20
	40	0.85	0.77	0.65	0.49	0.33
	80	0.92	0.87	0.79	0.66	0.50

tree in Thm. 5.5, we do not have to discuss the tree case.

5.G.2 Condition Number Point of View

To prove the bound of Prop. 5.6, we only use $|||MM^+|||_2 = 1$ and Lemma 5.9, which holds for *any* matrix M. Hence, we can say that this is the "worst" bound and we expect a far lower value of $|||CC^+|||_p$ for a general incidence matrix of graph. To gain some qualitative observation on how close between the exact and approximation, we further decompose $\alpha_{G,p}$. By using the submultiplicity and Lemma 5.10,

$$\begin{aligned}
\alpha_{G,p} &= |||W^{1/p}CC^+W^{-1/p}|||_p && (5.169)\\
&\leq |||W^{1/p}|||_p|||CC^+|||_p|||W^{-1/p}|||_p && (5.170)\\
&\leq |||CC^+|||_p w_{\max}^{1/p}/w_{\min}^{1/p} && (5.171)\\
&\leq |||C|||_p|||C^+|||_p w_{\max}^{1/p}/w_{\min}^{1/p} && (5.172)
\end{aligned}$$

where $w_{\max} := \max_\ell w_\ell$ and $w_{\min} := \min_\ell w_\ell$. In numerical analysis, the term $|||C|||_p|||C^+|||_p$ is called as a *condition number* of the matrix C [Saad, 2003]. A condition number is related to the "difficulty" to numerically solve the linear equation $C\mathbf{x} = \mathbf{y}$. The larger the condition number gets, the more difficult to solve the linear equation. The linear equation is difficult to solve if we can make one or more pairs of column or row of C close to parallel by elementary operations. However, by construction of incidence matrix, no pairs of column or row of the incidence matrix are close to parallel. Thus, we expect that the condition number of C will not be large, and hence we expect a smaller value of $\alpha_{G,p}$ than Prop. 5.6 in general.

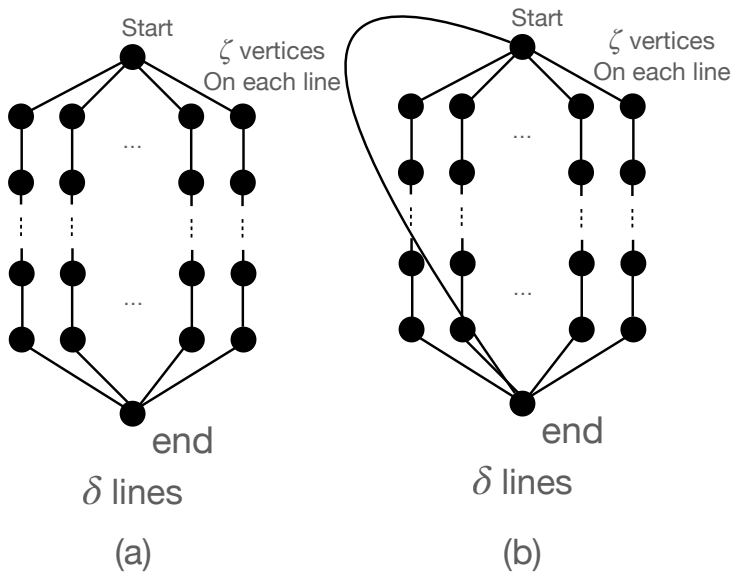


Figure 5.8: The example where the approximated value is far lower than the exact value. See 5.G.3 for details.

Table 5.7: The values of $|||CC^+|||_1/m^{1/2}$ for the graph Fig. 5.9 (a). If this value is close to 1, we have a looser bound.

		ζ 5	10	20	40	80
	5	0.51	0.33	0.23	0.16	0.11
	10	0.41	0.27	0.19	0.13	0.09
δ	20	0.31	0.2	0.14	0.1	0.07
	40	0.23	0.15	0.1	0.07	0.05
	80	0.16	0.11	0.07	0.05	0.04

Table 5.8: The values of $|||CC^+|||_1/m^{1/2}$ for the graph Fig. 5.9 (b). If this value is close to 1, we have a looser bound.

		ζ 5	10	20	40	80
	5	0.70	0.55	0.43	0.33	0.24
	10	0.68	0.59	0.51	0.41	0.32
δ	20	0.59	0.56	0.54	0.49	0.41
	40	0.47	0.48	0.51	0.52	0.48
	80	0.36	0.38	0.44	0.49	0.51

Table 5.9: The condition numbers $|||C|||_1|||C^+|||_1$ for the graph Fig. 5.9 (a).

		ζ 5	10	20	40	80
	5	20.4	45.2	95.1	195.1	395
	10	40.4	90.2	190.1	390.1	790
δ	20	80.4	180.2	380.1	780.1	1580
	40	160.4	360.2	760.1	1560.1	3160
	80	320.4	720.2	1520.1	3120.1	6320

Table 5.10: The condition numbers $|||C|||_1|||C^+|||_1$ for the graph Fig. 5.9 (b).

		ζ 5	10	20	40	80
	5	24.4	49.7	101.8	219.3	457.7
	10	51.0	136.7	346.2	812.8	1790
δ	20	120.2	362.3	1035.1	2702.6	6432.7
	40	266.2	871.6	2768.2	8119.5	21449
	80	563.8	1943.9	6670.3	21713	64438

5.G.3 Example where Approximation is Far Lower than the Exact Value

Lastly, we discuss an example where the approximation is far lower than the exact value and how this happens. We consider a graph depicted in Fig. 5.8. First, we see a graph in Fig. 5.8 (a). To build this graph, first consider the line graph, where the ζ vertices are in line. This graph is constructed with δ lines of diameter ζ each lines start vertex is glued to each other lines "start vertex" and similar to the "end vertices". For Fig. 5.8 (b), we add one edge to the graph in Fig. 5.8 between the start vertex and end vertex.

We now compare with approximation and the exact value of 1-resistance between the start vertex and end vertex. As we discussed in Sec. 5.F, we compute the exact 1-resistance between i and j as the minimum cut's inverse. Thus, for (a) $r_{G,1}(\text{start}, \text{end}) = 1/\delta$ and for (b) we have $r_{G,1}(\text{start}, \text{end}) = 1/(\delta + 1)$. We then compute the approximated values and $|||CC^+|||_1$ for Fig. 5.8. We give a result in Tables 5.5–5.8. From Tables 5.5 and 5.6, we observe that we have a far less accurate approximation for graph (b) than that for graph (a). In Tables 5.7 and 5.8, we observe that a far larger value of $|||CC^+|||_1$ for the graph (b) than that for the graph (a). We also observe that comparing with $|||CC^+|||_1$ of the graph constructed from the real dataset in Fig. 5.3, we see a far larger value of $|||CC^+|||_1$ for the graph (b). The larger value of $|||CC^+|||_1$ might be the reason why the approximation of the 1-resistance of graph (b) is far worse than the graph (a).

We now discuss why $|||CC^+|||_1$ for graph (b) is far larger than that for graph (a). We now revisit the condition number argument in Sec. 5.G.2. The condition number is the stableness of the linear equation of the matrix. The stable linear equation is even if we add small value ϵ to the linear equation, i.e., $C\mathbf{x} = \mathbf{y} + \epsilon$, the solution $\mathbf{x}$ is almost unchanged. If we add perturbation on each edge in graph (a), the graph can absorb the perturbation since each line graph is almost independent. However, on the graph (b), each line graph becomes dependent due to the additional edge. Moreover, the start and the end vertex are like "pivots" of the graph. The perturbations might be widely spread over the graph by connecting two pivots. By this spread, graph (b) becomes unstable, while graph (a), where we do not connect the pivots is more stable. In Tables 5.9 and 5.10 we see that graph (b) is far more unstable than graph (a).

Finally, we argue that we do not observe this phenomenon in the real setting. In the example of graph (b), the unstableness comes from the sparse connection over the graph and

connection of the "pivots" over such a spares graph. In a dense graph such as a complete graph, we saw far lower $\|\|\|CC^+\|\|\|_1$ as we observe in Sec 5.G.1. As we saw in the real dataset case, we can assume that there is a denser connection over the graph, even between the clusters.

5.H Proof of Theorem 5.8

This section discusses Thm. 5.8, including proof and some existing claim on Thm. 5.8.

5.H.1 Main Proof

We use the following characteristics of p-Laplacian, defined as Eq. (2.32).

Proposition 5.17 ([Bühler and Hein, 2009])**.**

$$S_{G,p}(\mathbf{x}) = \langle \mathbf{x}, \Delta_p \mathbf{x} \rangle_{\mathcal{H}(V)}, \tag{5.173}$$

$$\left(\frac{\partial S_p(\mathbf{x})}{\partial \mathbf{x}} \right)_i = p(\Delta_p \mathbf{x})_i. \tag{5.174}$$

Before we prove the main argument, we now explore a matrix expression of the p-Laplacian Δ_p. We define a matrix $A_{p,\mathbf{x}}$ as

$$A_{p,\mathbf{x}}(i,j) := a_{ij}|x_i - x_i|^{p-2}, \tag{5.175}$$

and its degree-like matrix $D_{p,\mathbf{x}}$ as

$$D_{p,\mathbf{x}}(i,j) = \begin{cases} \sum_{j=1}^n A_{p,\mathbf{x}}(i,j) & \text{if } l = i \\ 0 & \text{if } l \neq i \end{cases} \tag{5.176}$$

Define the matrix $L_{p,\mathbf{x}}$ as

$$L_{p,\mathbf{x}} := D_{p,\mathbf{x}} - A_{p,\mathbf{x}}. \tag{5.177}$$

Now,

$$(L_{p,\mathbf{x}}\mathbf{x})_i = D_{p,\mathbf{x}}(i,i)x_i - \sum_{j=1}^n A_{p,\mathbf{x}}(i,j)x_j \tag{5.178}$$

$$= \sum_{j=1}^{n} A_{p,\mathbf{x}}(i,j)x_i - \sum_{j=1}^{n} A_{p,\mathbf{x}}(i,j)x_j \tag{5.179}$$

$$= \sum_{j=1}^{n} A_{p,\mathbf{x}}(i,j)(x_i - x_j) \tag{5.180}$$

$$= \sum_{j=1}^{n} a_{ij}|x_i - x_i|^{p-1}\mathrm{sgn}(x_i - x_j) \tag{5.181}$$

$$= (\Delta_p \mathbf{x})_i. \tag{5.182}$$

Thus, we can say that $L_{p,\mathbf{x}}$ is a matrix expression of the p-Laplacian, satisfying

$$L_{p,\mathbf{x}}\mathbf{x} = \Delta_p \mathbf{x}. \tag{5.183}$$

Then, by Prop. 5.17, we can prove that

$$\mathbf{x}^\top L_{p,\mathbf{x}}\mathbf{x} = S_{G,p}(\mathbf{x}). \tag{5.184}$$

Now we turn to the optimization problem Eq. (5.17). By using the Lagrangian multiplier method, the optimal solution satisfies the following:

$$F(\mathbf{x},\lambda) := (S_{G,p}(\mathbf{x})) - \lambda(x_i - x_j - 1) \tag{5.185}$$

$$\frac{\partial F}{\partial \mathbf{x}} = pL_{p,\mathbf{x}}\mathbf{x} - \lambda(\mathbf{e}_i - \mathbf{e}_j) = 0 \tag{5.186}$$

$$\frac{\partial F}{\partial \lambda} = x_i - x_j - 1 = 0. \tag{5.187}$$

From Eq. (5.186), we have

$$\mathbf{x}^{*ij} = \frac{\lambda}{p} L^{+}_{p,\mathbf{x}^{*ij}}(\mathbf{e}_i - \mathbf{e}_j). \tag{5.188}$$

From Eq. (5.188) and Eq. (5.187), we have

$$\frac{\lambda}{p}\left((L^{+}_{p,\mathbf{x}^{*ij}}(i,i) - L^{+}_{p,\mathbf{x}^{*ij}}(i,j)) - (L^{+}_{p,\mathbf{x}^{*ij}}(j,i) - L^{+}_{p,\mathbf{x}^{*ij}}(j,j))\right) = 1. \tag{5.189}$$

Following Eq. (5.188), we substitute λ/p from Eq. (5.189) into Eq. (5.188), and we have

$$\mathbf{x}^{*ij} = \frac{L^+_{p,\mathbf{x}^*_{ij}}}{L^+_{p,\mathbf{x}^{*ij}}(i,i) + L^+_{p,\mathbf{x}^{*ij}}(j,j) - 2L^+_{p,\mathbf{x}^{*ij}}(i,j)}(\mathbf{e}_i - \mathbf{e}_j). \tag{5.190}$$

Since p-resistance is an inverse of the energy, we obtain

$$\begin{aligned} r_{G,p}(i,j) &= (\mathbf{x}^{*ij\top} L^+_{p,\mathbf{x}^{*ij}} \mathbf{x}^{*ij})^{-1} && (5.191)\\ &= L^+_{p,\mathbf{x}^{*ij}}(i,i) + L^+_{p,\mathbf{x}^{*ij}}(j,j) - 2L^+_{p,\mathbf{x}_{*ij}}(i,j) && (5.192)\\ &= (\mathbf{e}_i - \mathbf{e}_j)^\top L^+_{p,\mathbf{x}^{*ij}}(\mathbf{e}_i - \mathbf{e}_j) && (5.193) \end{aligned}$$

The rest of the proof is same as the original proof in Thm. 6 in [Alamgir and Luxburg, 2011]. The trick is that we do not have to the exact form of $L_{p,\mathbf{x}^{*ij}}$. Only this expression is enough to prove Theorem 5.8.

5.H.2 Original Context of Theorem 5.8

Originally in Sec. 5 [Alamgir and Luxburg, 2011], Thm. 5.8 when $p = 2$ has a different interpretation. Nadler et al. [2009] proves that the semi-supervised learning problem of $p = 2$ case is meaningless if the number of vertices are infinite. Thm. 5.8 for $p = 2$ supports this claim in [Nadler et al., 2009] for two-pole semi-supervised leaning problem for the following way. Since the equivalent 2-resistance is known to converge to a meaningless function, the solution of the semi-supervised problem is equivalently characterized by this meaningless function. Thus, the semi-supervised learning does not make sense, if the number of the vertices are large. If the conjecture for $p > 1$ case were proven, Thm. 5.8 can be interpreted that for some range of $p > 1$ two-pole semi-supervised learning problem is not meaningless, since the equivalent p-resistance is shown not to converge to a meaningless one. Later year, independent of p-resistance, the statement "for some range of $p > 1$ semi-supervised learning problem is not meaningles" is proven by [Slepcev and Thorpe, 2019]. Thm. 5.8 now supports [Slepcev and Thorpe, 2019] from a p-resistance view.

5.H.3 Remark on the Existing Claims on Theorem 5.8

Finally, we discuss several existing claims on this theorem. First, we need to mention a small *fixable* mistake in the original proof in [Alamgir and Luxburg, 2011] for the $p = 2$ case.

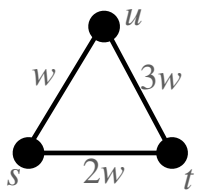


Figure 5.9: The graph example discussed in [Bridle and Zhu, 2013].

The original proof assumes that the solution to the semi-supervised learning Eq. (5.17) when $p = 2$ is that

$$\mathbf{x}^{*ij} = L^{+}(\mathbf{e}_i - \mathbf{e}_j). \tag{5.194}$$

However, this is not true since this does not satisfy the constraint

$$x_i^{*ij} - x_j^{*ij} = (\mathbf{e}_i - \mathbf{e}_j)^{\top} L^{+}(\mathbf{e}_i - \mathbf{e}_j) \neq 1. \tag{5.195}$$

Instead, the solution is given as

$$\mathbf{x}^{*ij} = \frac{L^{+}(\mathbf{e}_i - \mathbf{e}_j)}{(\mathbf{e}_i - \mathbf{e}_j)^{\top} L^{+}(\mathbf{e}_i - \mathbf{e}_j)}. \tag{5.196}$$

Note that this corresponds to Eq. (5.190). However, this does not affected the rest of the proof, since the proof exploits only $\mathbf{x}^{*ij} = \rho L^{+}(\mathbf{e}_i - \mathbf{e}_j)$ for $\rho \in \mathbb{R}$, and ρ does not matter. Thus, the validity of the original claim still remains.

Next, since Thm. 5.8 resolves the open problem in [Alamgir and Luxburg, 2011], there is an existing discussion on if this statement is true or not. The work [Bridle and Zhu, 2013] claims that there is a counterexample to Thm. 5.8 in the general p case. In the following, we argue that the discussion on the example in [Bridle and Zhu, 2013] does not work as a counterexample.

The "counterexample" given in [Bridle and Zhu, 2013] is based on the example shown as Fig. 5.9. However, unfortunately, we believe that there is invalidity in the discussion on this example. Firstly, we recall that

$$\min_{\mathbf{x}} \{S_{G,p}(\mathbf{x}) \text{ s.t. } x_s - x_t = 1\} \neq \min_{\mathbf{x}} S_{G,p}(\mathbf{x}), \tag{5.197}$$

since $\min_{\mathbf{x}} S_{G,p}(\mathbf{x}) = 0$ when $\mathbf{x} = c\mathbf{1}$, $\forall c \in \mathbb{R}$, while $c\mathbf{1}$ does not satisfy the constraint of the left hand side. However, the work [Bridle and Zhu, 2013] assumes the equality of Eq. (5.197), see the the first equality at the top of the left column in p.3 of [Bridle and Zhu, 2013]. Moreover, we note that

$$\frac{\partial S_{G,p}(\mathbf{x})}{\partial x_u} = \frac{\partial}{\partial x_u}\left(w|x_s - x_u|^p + 2w|x_u - x_v|^p + 3w|x_s - x_t|^p\right) \tag{5.198}$$

$$= p\left(w|x_s - x_u|^{p-1} + 2w|x_u - x_v|^{p-1}\right), \tag{5.199}$$

and therefore

$$\frac{\partial S_{G,p}(\mathbf{x})}{\partial x_u} \neq p\left(-w|x_s - x_u|^{p-1} + 2w|x_u - x_v|^{p-1}\right), \tag{5.200}$$

where the difference between Eq. (5.199) and Eq. (5.200) is the sign of the term $w|x_s - x_u|^{p-1}$. [2] However, the work [Bridle and Zhu, 2013] assumes the equality of Eq. (5.200), see the third equality at the top of the left column in p.3 of [Bridle and Zhu, 2013]. In [Bridle and Zhu, 2013], these invalid equality assumptions of Eq. (5.197) and Eq. (5.200) derive the fundamental relationship in order to bring a counterexample. The rest of the analysis in [Bridle and Zhu, 2013] is carried with this relationship. Due to this invalidity, we believe that there are serious flaws in the claim that the example Fig. 5.9 leads to a counterexample to Thm. 5.8. Hence, we claim that Thm. 5.8 holds with the proof in this section.

5.I On Difficulties of The Exact Solution

In this section, we briefly explain the difficulties to obtain the exact solution of the resistance. Again, we consider the minimization problem Eq. (5.17). The Lagrangian multiplier method gives Eq. (5.186) and Eq. (5.186). From Eq. (5.186), the optimal solution $\mathbf{x}$ satisfies

$$0 = p\Delta_p\mathbf{x} - \lambda(\mathbf{e}_i - \mathbf{e}_j) \tag{5.201}$$

[2] There is a slight difference between the definition of the p-resistance between ours and [Bridle and Zhu, 2013]. We follow the definition of [Herbster and Lever, 2009] and [Bridle and Zhu, 2013] follows the definition of [Alamgir and Luxburg, 2011]. However, these two have almost same properties. Moreover, while we write the equations in our form, this difference does not affect the discussion here. For more details of the difference, see §6.1 in [Alamgir and Luxburg, 2011] or Sec. 2.4.3.

To solve this problem, we want to consider Δ_p^+, which is "generalized inverse" function Δ_p, defined as

$$\Delta_p^+(\Delta_p(\Delta_p^+(\mathbf{x}))) = \Delta_p^+\mathbf{x} \tag{5.202}$$

Recall that we can write as

$$\Delta_p = C^\top W f_{p-1}(C\mathbf{x}). \tag{5.203}$$

For the convenience of notation, we write

$$f_{\mathbf{w},p} = W f_p(C\mathbf{x}). \tag{5.204}$$

If there exists $\boldsymbol{\alpha} \in \mathrm{Ker}(C)$ s.t.

$$f_{\mathbf{w},p-1}^{-1}(C^{+\top}\mathbf{x} - \boldsymbol{\alpha}) \in \mathrm{Im}(C), \tag{5.205}$$

the Δ_p^+ is given as

$$\Delta_p^+(\mathbf{x}) := C^+ f_{\mathbf{w},p-1}^{-1}(C^{+\top}\mathbf{x} - \boldsymbol{\alpha}), \tag{5.206}$$

The reason is as follows. We get

$$\begin{aligned}
\Delta_p(\Delta_p^+(\mathbf{x})) &= C^\top f_{\mathbf{w},p-1}(CC^+ f_{\mathbf{w},p-1}^{-1}(C^{+\top}\mathbf{x} - \boldsymbol{\alpha})) && (5.207)\\
&= C^\top f_{\mathbf{w},p-1}(f_{\mathbf{w},p-1}^{-1}(C^{+\top}\mathbf{x} - \boldsymbol{\alpha})) && (5.208)\\
&= C^\top (C^{+\top}\mathbf{x} - \boldsymbol{\alpha})) && (5.209)\\
&= C^\top C^{+\top}\mathbf{x}. && (5.210)
\end{aligned}$$

The second line follows from the assumption that $f_{\mathbf{w},p-1}^{-1}(C^{+\top}\mathbf{x} - \boldsymbol{\alpha}) \in \mathrm{Im}(C)$. Thus,

$$\begin{aligned}
\Delta_p^+(\Delta_p(\Delta_p^+(\mathbf{x}))) &= C^+ f_{\mathbf{w},p-1}^{-1}(C^{+\top}C^\top C^{+\top}\mathbf{x} - \boldsymbol{\alpha}) && (5.211)\\
&= C^+ f_{\mathbf{w},p-1}^{-1}(C^{+\top}\mathbf{x} - \boldsymbol{\alpha}) && (5.212)\\
&= \Delta_p^+\mathbf{x}. && (5.213)
\end{aligned}$$

From this property, if we substitute

$$\mathbf{x} = \Delta_p^+ \left(\frac{\lambda}{p} (\mathbf{e}_i - \mathbf{e}_j) \right) \tag{5.214}$$

the Eq. (5.201) satisfied. Therefore, the next question is what $\boldsymbol{\alpha}$ is. However, we do not know even if such $\boldsymbol{\alpha}$ satisfying Eq. (5.205) exists or not.

Chapter 6

ResTran: A GNN Alternative to Learn A Graph With Features

This chapter considers a vertex classification task where we are given a graph and associated vector features. The modern approach to this task is graph neural networks (GNNs). However, GNNs are biased to primarily learn homophilous information. To overcome this bias in GNN architectures, we take a simple alternative approach to GNNs. Our approach is to obtain a vector representation capturing both features and the graph topology. We then apply standard vector-based learning methods to this vector representation. For this approach, this chapter propose a simple transformation of features, which we call *Resistance Transformation* (abbreviated as *ResTran*). We provide theoretical justifications for ResTran from the effective resistance, k-means, and spectral clustering points of view. We empirically demonstrate that ResTran is more robust to the homophilous bias than established GNN methods.

6.1 Introduction

As discussed from Chapter 2 to Chapter 5, spectral clustering is used to cluster vertices in a given graph. While extensively studied, this approach typically considers a dataset where each vertex has both graph connections and associated features. The goal is to classify vertices by leveraging both the graph's topological structure and the vertex features. We may call this task as a vertex classification task in the "graph-with-features" setting. The modern approach to this task is graph neural networks (GNNs) [Gori et al., 2005, Kipf and Welling, 2016a, Veličković et al., 2018]. GNNs propagate features over the graph to build expressive latent embeddings; the embeddings are then consumed in downstream classification models.

However, due to the nature of these GNN architectures, GNNs are typically known to have a bias towards homophilous information and to not be effective in learning heterophilous information [Hoang and Maehara, 2019, Luan et al., 2022]. This bias worsens if we stack GNN layers (known as "over-smoothing" [Li et al., 2018, Oono and Suzuki, 2019]). Some recent GNN models mitigate this bias, such as [Azabou et al., 2023, Pei et al., 2020, Luan et al., 2021], while such models, including these examples, often involve complicated GNN architectures.

In this chapter, to overcome this homophilous bias in a simpler way, we propose an alternative approach to GNNs since this bias seems to be inherent in GNN architectures. Instead of mitigating biases by complicating the GNNs, our approach is to obtain a vector representation for the features and graph. Then, we apply standard vector-based learning methods to this vector representation, such as established neural network (NeuralNet) based models like variational autoencoder or even support vector machines (SVMs). For this approach, we propose a *Resistance Transformation* (abbreviated as *ResTran*), a simple transformation of feature vectors to incorporate graph structural information.

We theoretically justify ResTran from a connection between the k-means and spectral clustering. Our justification is inspired by Dhillon et al. [2004] and Chapter 4, which justifies using feature maps for spectral clustering applied to vector data. For this purpose, Dhillon et al. [2004] takes the following steps as i) setting up k-means objective for transformed vectors by a feature map and ii) showing the equivalence from this k-means objective to spectral clustering. For ResTran, we follow a similar strategy: i) modifying the k-means to incorporate the vector representation by ResTran and ii) showing the equivalence from this k-means to spectral clustering. We show that this modified k-means for the featureless setting (i.e., looking only at a graph by taking features as an identity matrix) is equivalent to spectral clustering. Moreover, for the graph-with-features setting, we show that this k-means can be seen as a natural extension of spectral clustering from the featureless to the graph-with-features setting. We also discuss why ResTran may preserve the homophilous and heterophilous information better than the established GNNs. Our experiments show that ResTran outperforms graph-only and feature-only representation in unsupervised tasks. We also numerically show that ResTran is more robust to the homophilous bias than established GNNs in the semi-supervised learning (SSL) tasks.

Note that, for ResTran, we use the same Laplacian coordinate, which is the same coordinate space as Chapter 5.

Contribution. In summary, our contributions in this chapter are as follows. i) We propose a simple *ResTran* for a graph-with-features problem. ii) We theoretically justify ResTran from an effective resistance, k-means, and spectral clustering perspective. iii) We numerically confirm that ResTran is more robust to homophilous bias than established GNNs for common datasets. *All proofs are in the Appendix.*

6.2 Basic Notions

This section introduces some basic notion related to this chapter.

6.2.1 Graph-with-features Problem vs. Featureless Problem.

This chapter considers a vertex classification task. This task is classifying vertices of the graph into k classes. For this task, we consider two settings. *i) Graph-With-Features Problem.* This problem assumes that the i-th vertex is associated with f dimensional *features* $\mathbf{x}_i \in \mathbb{R}^f$. We define a feature matrix as $X := (\mathbf{x}_1, \dots, \mathbf{x}_n)$. A popular technique for this is a GNN. *ii) Featureless Problem.* This problem only considers the topology of the graph. There are various methods specifically for this, such as spectral clustering. All the methods from Chapter 2 to Chapter 5 consider this featureless problem. We can also apply the graph-with-features methods to this featureless setting. A common technique to do so is by setting $X = I$, where I is an identity matrix [Kipf and Welling, 2016a].

6.2.2 Coordinate Spanning Set and Resistances Revisited

We introduced coordinate spanning set and resistance in Sec. 2.4.1. This section revisits and further develops coordinate spanning set and resistance related to this chapter.

Recall that the reproduced kernel associated with the PSD matrix M is M^+ since

$$\langle \mathbf{u}, \mathbf{v}_i \rangle_M = \mathbf{u}^\top M M^+ \mathbf{e}_i = u_i, \quad \forall \mathbf{v}_i \in \mathcal{V}(M)_{\langle \cdot, \cdot \rangle_M},\ \mathbf{u} \in \mathcal{H}(M)_{\langle \cdot, \cdot \rangle_M}. \tag{6.1}$$

For this inner product, we defined the coordinate spanning set

$$\mathcal{V}(M)_{\langle \cdot, \cdot \rangle_M} := \{v_i := M^+ \mathbf{e}_i : i = 1, \dots, n\}. \tag{6.2}$$

While this coordinate spanning set is same as $\mathcal{V}(M)$ in Eq. (2.67), we give a notation-wise addition to $\mathcal{V}(M)$; we subscript the inner product $\langle \cdot, \cdot \rangle_M$, which the coordinate spanning set is defined over.

We let $\mathcal{H}(M)_{\langle \cdot, \cdot \rangle_M} := \mathrm{span}(\mathcal{V}(M)_{\langle \cdot, \cdot \rangle_M})$. This $\mathcal{H}(M)_{\langle \cdot, \cdot \rangle_M}$ is a Hilbert space induced by inner product $\langle \cdot, \cdot \rangle_M$. As we see in Sec. 2.4.1, the set $\mathcal{V}$ acts as "coordinates" for $\mathcal{H}$, that is, if $\mathbf{w} \in \mathcal{H}$ we have $w_i = \mathbf{e}_i^\top M^+ M \mathbf{w} = \langle \mathbf{e}_i, M^+ \mathbf{e}_i \rangle_M$.

If we measure this space $\mathcal{H}(M)_{\langle \cdot, \cdot \rangle_M}$ over the plain dot product $\langle \cdot, \cdot \rangle_2$, the coordinate is instead

$$\mathcal{V}(M)_{\langle \cdot, \cdot \rangle_2} := \{ v_i := M^{+1/2} \mathbf{e}_i : i = 1, \dots, n \}, \tag{6.3}$$

since $\| M^+ \mathbf{e}_i \|_M = \| M^{+1/2} \mathbf{e}_i \|_2$. In the following, for brevity, we use

$$\mathcal{V}'(M) := \mathcal{V}(M)_{\langle \cdot, \cdot \rangle_2}, \quad \mathcal{H}'(M) := \mathcal{H}(M)_{\langle \cdot, \cdot \rangle_2} \tag{6.4}$$

Instead of $\mathcal{V}(L)$ we used in Chapter 5, we use $\mathcal{V}'(L)$ (Eq. (6.4)) for this chapter.

Recall that the resistance can be written as

$$r_{G,2}(i, j) = \| L^{+1/2} \mathbf{e}_i - L^{+1/2} \mathbf{e}_j \|_2^2. \tag{6.5}$$

Then, we can write the resistance using the coordinate spanning set $\mathcal{V}'(L)$ as

$$r_{G,2}(i, j) = \| \mathbf{v}_i - \mathbf{v}_j \|_2^2, \quad \mathbf{v}_i, \mathbf{v}_j \in \mathcal{V}'(L). \tag{6.6}$$

6.2.3 Homophily, Heterophily, and Eigenspace of Laplacian

A graph dataset may be classified into two notions. The *homophily* assumption is that adjacent vertices are more likely to be in the same group. The *heterophily* assumption is that vertices are collected in diverse groups, i.e., the contrary to homophily assumption. From the cut definition, spectral clustering assumes homophily. Recall that the spectral clustering looks at the eigenspace associated with smaller eigenvalues (i.e., low-frequencies) of L. Thus, we may see that this eigenspace contains homophilous information. Also, we may say that the eigenspace for larger eigenvalues (i.e., high-frequencies) of L captures heterophilous

information. In the following, we say "low-frequency" for homophily or "high-frequency" for heterophily. See Hoang and Maehara [2019] and Luan et al. [2022] for details.

6.3 Proposed Method: ResTran

This section presents our learning framework for the graph-with-features setting. A common method for this setting is a GNN, where we develop NeuralNets incorporating a graph. Instead, we propose a vector representation of the graph-with-features, which we call *ResTran*. We then apply vector-based machine learning methods to this vector, e.g., SVM and the standard NeuralNet methods. In Sec. 6.4, we will justify ResTran from the spectral connection and resistance view and also explore characteristics of ResTran.

For our framework, we use the *shifted graph Laplacian*, as done in [Herbster and Pontil, 2006], as

$$L_b^{-1} := L^+ + bJ_G, \quad \text{where} \quad b > 0, (J_G)_{ij} := \begin{cases} 1 & (i \text{ and } j \text{ are in the same component}) \\ 0 & (\text{otherwise}), \end{cases}. \tag{6.7}$$

Note that from the definition $J_G = \mathbf{1}\mathbf{1}^\top$ if the graph is connected, i.e., contains only one component. Note also that L_b is invertible since L_b is symmetric positive definite (PD) as we see later in Prop. 6.1.

Proposed Framework via *ResTran*. Below we propose our framework. The overall strategy is to i) have a vector representation of graph-with-features ii) apply a vector based machine learning method. For i), using the coordinate $\mathcal{V}'(L_b)$, we propose our *Resistance Transformation* (*ResTran* for abbreviation) X_G as

$$X_G := (\mathbf{x}_{G,1}, \ldots, \mathbf{x}_{G,n}), \quad \text{where } \mathbf{x}_{G,i} := X\mathbf{v}'_i,\ \mathbf{v}'_i \in \mathcal{V}'(L_b). \tag{6.8}$$

Recall that $\mathbf{v}'_i = L_b^{-1/2}\mathbf{e}_i$ by definition of $\mathcal{V}'(L_b)$ in Sec. 6.2.2. Note that $\mathbf{x}_i, \mathbf{x}_{G,i} \in \mathbb{R}^f$ and X, $X_G \in \mathbb{R}^{n\times f}$. For ii), we then use any vector based machine learning methods for X_G, such as SVM and NeuralNet-based methods.

Practical Implementation via Krylov Subspace Method. If we naively compute $L_b^{-1/2}$ and then multiply X to obtain ResTran Eq. (6.8), it costs prohibitive $O(n^3)$ complexity due

Algorithm 6 Proposed Practical Framework for SSL via ResTran and Krylov Subspace Method

Input: Graph $G = (V, E)$, Features X, Training and Test Indices Tr, Te, Krylov Subspace Dim r

Obtain the approximated ResTran $\tilde{X}_G$ (Eq. (6.8)) by applying Krylov subspace method, i.e.,

$$\tilde{X}_G = \textsc{KrylovSubspaceMethod}(L, X, r)$$

Obtain the model by applying any vector machine learning method to the training data whose indices are Tr as

$$\textsc{Model} = \textsc{AnyVectorMLMethod}(\{(\tilde{X}_G)_{\cdot i}, y_i\}_{i \in Tr})$$

Obtain the predicted label $\hat{\mathbf{y}}$ by applying Model to the test data whose indices are Te as

$$\hat{\mathbf{y}} = \textsc{Model}(\{(\tilde{X}_G)_{\cdot i}\}_{i \in Te})$$

Output: The predicted label $\hat{\mathbf{y}}$

to the computation of $L_b^{-1/2}$. Instead of this naive computing, we consider to approximate X_G. For this purpose, we apply the Krylov subspace method, by which we can approximate a solution of linear algebraic problems. The Krylov subspace method reduces the computational complexity from $O(n^3)$ to $O(rfm)$, where r is the dimension of the Krylov subspace. The dimension r is typically small, say $r < 100$. The Krylov subspace method approximates X_G by considering L and X *at the same time*. Thus, we expect a better approximation for Krylov than approximating $L_b^{-1/2}$ without using X; such methods include the polynomial approximation. Note that this polynomial approximation is common in the established convolutional GNNs, such as [Defferrard et al., 2016, Kipf and Welling, 2016a]. Refer to Appendix 6.A or [Higham, 2008] for details. The overall proposed framework is summarized in Alg. 6. Note that Alg. 6 can be interpreted as SSL even though we apply supervised methods such as SVM because we first observe X and G to obtain X_G. This is same as GNNs, where we observe X and G before we learn. Alg. 6 naturally generalizes to the unsupervised setting.

Coordinate Interpretation of ResTran. We first remark that $L_b^{-1/2} = (\mathbf{v}'_1, \ldots, \mathbf{v}'_n)$, $L_b^{-1/2}$ is symmetric, and $X_G = XL_b^{-1/2}$. The $X_G^\top$ can be seen as retaking basis of $X^\top$ by $\mathcal{V}'(L_b)$ if we see $L_b^{-1/2}$ in row-wise. Moreover, by comparing the original $X = (X\mathbf{e}_1, \ldots, X\mathbf{e}_n)$, the X_G can be seen as retaking $\mathbf{e}_i$ to $\mathbf{v}'_i$ to indicate i-th vertex if we see $L_b^{-1/2}$ in column-wise.

Comparison with GNNs. This approach is simpler than existing GNN approaches. The recent GNNs often involve complicated graph designs in layers of NeuralNet or pre/post-processing. However, our framework is simple since we transform X to X_G and then apply any vector-based methods.

6.4 Characteristics and Justification of ResTran

This section discusses the characteristics and justification of ResTran. We first discuss the characteristics of ResTran, by exploring theoretical properties of Laplacian coordinate $\mathcal{V}'(L_b)$ from a resistance view. Next, we justify using ResTran of X_G from a k-means perspective.

6.4.1 Characteristics of ResTran: An Effective Resistance View

This section discusses the characteristics of ResTran. We first explore theoretical properties of the Laplacian coordinate $\mathcal{V}'(L_b)$. We then interpret these results to explain characteristics of ResTran.

Theoretical Properties of $\mathcal{V}'(L_b)$. In the following, we assume that we have K connected components. We write $G_i := (V_i, E_i)$ for $i = 1, \ldots, K$, and $G = G_1 \cup \ldots \cup G_K$. We write as $n_i := |V_i|$. Whiteout loss of generality, we can assume that $n_1 \leq \ldots \leq n_K$. Denote $\mathbf{1}_{G_j}$ by all one vector for G_j, i.e., $(\mathbf{1}_{G_j})_i = 1$ if $j \in V_{G_j}$ otherwise 0. Note that $\sum_{j\in[K]} \mathbf{1}_{G_j} = \mathbf{1}$ and $(J_G)_{i\cdot} = \mathbf{1}_{G_s}$ if $i \in V_s$. Note also that $\mathbf{1}_{G_j}$ are eigenvectors of L. Using this notation, we have properties of $\mathcal{V}'(L_b)$ as follows.

Proposition 6.1. *Suppose that a graph G has K connected components. Let (λ_i, $\boldsymbol{\psi}_i$) be the i-th eigenpair of L. If $n_1 b > \lambda_{K+1}^{-1}$, the i-th eigenpair (λ'_i, $\boldsymbol{\psi}'_i$) of $L_b^{-1/2}$ is*

$$(\lambda'_i, \boldsymbol{\psi}'_i) = \begin{cases} \left(\lambda_{n+1-i}^{-1/2}, \boldsymbol{\psi}_{n+1-i}\right) & \text{for } i = 1, \ldots, n-K, \\ \left((n_{i-(n-K)}b)^{1/2}, \mathbf{1}_{G_{n_{i-(n-K)}}}\right) & \text{for } i = n-K+1, \ldots, n. \end{cases}$$

Corollary 6.2. $L_b^{-1/2}\mathbf{e}_i = (L^{+1/2} + \sqrt{b}J_G^{1/2})\mathbf{e}_i$, *where* $J_G^{1/2} = \sum_i^K (n_i^{-1/2}\mathbf{1}_{G_i}\mathbf{1}_{G_i}^\top)$

This proposition shows that L and $L_b^{-1/2}$ share eigenvectors and that $L_b^{-1/2}$ is PD since $\lambda'_i > 0$ for all i. Next, we explore the characteristics of the coordinates $\mathcal{V}'(L_b)$. We define an *extended resistance* as

$$r'_G(i,j) := \|\mathbf{v}'_i - \mathbf{v}'_j\|_2^2, \quad \mathbf{v}'_i, \mathbf{v}'_j \in \mathcal{V}'(L_b) \tag{6.9}$$

Recall that $\mathbf{v}'_i = L_b^{-1/2}\mathbf{e}_i$. The following can be claimed.

Proposition 6.3. *If two vertices i, j in the same component G_s, $r'_G(i,j) = r_{G_s}(i,j)$.*

Prop. 6.3 means that even if we use $\mathcal{V}'(L_b)$ instead of $\mathcal{V}_L$, the resistance, the distance between coordinates (Eq. (2.71)), is preserved within the connected component. For inter-component, the parameter b controls the connectivity among the components. If two vertices are in different components, it is natural to think that they are apart. However, in the graph-with-features setting, even if two vertices are in different components, the two vertices often belong to the same cluster; therefore, these are not apart so much. We parameterize this intuition by b; by taking larger b, we weigh more on the disconnected observation. Taking b large enough for two vertices i, ℓ in the different components, we can make $r'_G(i,\ell)$ greater than *any* resistances within the component as follows.

Proposition 6.4. *If $b > \sqrt{2}n_1/\lambda_{K+1}$, $r'_G(i,\ell) > r'_G(i,j)$ for $i, j \in V_s$ and $\ell \in V_t$ where $s \neq t$.*

Using these theoretical properties, we observe the following characteristics of the ResTran.

ResTran from a Resistance View. From Prop. 6.3 and Prop. 6.4, we observe that $\mathcal{V}'(L_b)$ serves as a coordinate offering an extended resistance. Our ResTran may be viewed as the basis transformation from $\mathbf{e}_i$ to $\mathbf{v}'_i$. This is why we call our transformation Eq. (6.8) as a "resistance" transformation.

ResTran Capturing a Mix of Homophilous and Heterophilous Information. Our ResTran can be seen as favoring the homophilous assumption but, at the same time, not ignoring the heterophilous assumption, while GNNs are biased toward homophily. Recall that the homophilous information is contained in the space spanned by ψ_i for the smaller eigenvalues λ_i while the heterophilous information is in the space spanned by ψ_j for larger eigenvalues λ_j, as seen in Sec. 6.2.3. GNNs are effective at homophilous data but not at heterophilous data [Luan et al., 2022]. Loosely speaking, this happens because each layer of GNNs multiplies the adjacency matrix A to the next layer, often several times. Stacking the layers enlarges the low-frequency components, which leads to a bias towards homophily. On the other hand, ResTran "balances" homophily and heterophily. Observe that we can see that $L_b^{-1/2}$ is "spectral reordering" of the graph Laplacian L (see Prop. 6.1); the largest eigenvalues of $L_b^{-1/2}$ are the smallest eigenvalues of L, and the order is reversed. Also, from Prop .6.1, the

eigenvalues of $L_b^{-1/2}$ associated with eigenvectors ψ_i is either $\sqrt{n_i b}$ or $\lambda_i^{-1/2}$, which is large since λ_i is small. Recall that ResTran multiplies $L_b^{-1/2}$ to X once. Thus, the space containing the homophilous information is amplified by large $\lambda_i^{-1/2}$. At the same time, we do not ignore the heterophilous space, but this is amplified by small $\lambda_j^{-1/2}$ since λ_j is large.

6.4.2 Justification of ResTran X_G from a k-means Perspective

This section justifies our ResTran X_G. Our justification is inspired by Dhillon et al. [2004]. As reviewed in Sec. 2.2, Dhillon et al. [2004] justifies using a feature map for spectral clustering applied to vector data. For this purpose, Dhillon et al. [2004] use the following steps: i) modify the k-mean objectives to incorporate a vector transformed by a feature map and ii) show a connection from this modified k-means objective to spectral clustering. Here, we aim to establish a similar connection for ResTran. For this purpose, following i), we use X_G in the k-means objective Eq. (2.49) as

$$\mathcal{J}_G(\{V_\ell\}_{\ell=1}^k) := \sum_{\ell\in[k]}\sum_{i\in V_\ell} \|\mathbf{x}_{G,i} - \mathbf{m}_{G,\ell}\|_2^2, \quad \mathbf{m}_{G,\ell} := \sum_{j\in V_\ell} \mathbf{x}_{G,j}/|V_\ell|. \tag{6.10}$$

This objective is a replacement of the standard k-means Eq. (2.49) from $\mathbf{x}_i$ to $\mathbf{x}_{G,i}$. Following ii), we establish connections from this k-means objective to spectral clustering as follows.

- Sec. 6.4.2.1 shows that in the featureless setting where $X = I$, conducting k-means on $\mathbf{v}'_i = L_b^{-1/2}\mathbf{e}_i$ is equivalent to spectral clustering.
- Sec. 6.4.2.2 shows that conducting k-means on $\mathbf{x}_{G,i}$ can be seen as a natural generalization of the spectral clustering through the k-means discussion.

With these connections, we say that ResTran is justified in the same sense as the feature map for spectral clustering as done by [Dhillon et al., 2004] discussed in Sec. 2.2.

6.4.2.1 Justification for Featureless Setting: Revisiting the Spectral Connection

This section justifies Eq. (6.10) for the featureless setting, where we use $X = I$. Therefore, for featureless setting, $X_G = (\mathbf{v}'_1, \ldots, \mathbf{v}'_n)$ from the definition of X_G Eq. (6.8). Using this X_G, we can rewrite Eq. (6.10) and further expand using Frobenius norm $\|\cdot\|_{\mathrm{Fro}}$ and

indicator matrix Z_R (Eq. (1.11)) as

$$\mathcal{J}_R(\{V_\ell\}_{\ell=1}^k) := \sum_{\ell\in[k]}\sum_{i\in V_\ell} \|\mathbf{v}'_i - \mathbf{m}_\ell\|_2^2, \quad \mathbf{m}_\ell := \sum_{j\in V_\ell} \mathbf{v}'_j/|V_\ell|, \mathbf{v}'_j \in \mathcal{V}'(L_b) \tag{6.11}$$

$$= \|L_b^{-1/2} - Z_R Z_R^\top L_b^{-1/2}\|_{\mathrm{Fro}}^2. \quad (\because \mathbf{m}_\ell = (L_b^{-1/2} Z_R Z_R^\top)_{\cdot i} \text{ if } i \in C_\ell). \tag{6.12}$$

With Eq. (6.12), we may obtain the *relaxed* solution of k-means by relaxing Z_R into real values. We first claim that the objective Eq. (6.11) grounds on the extended resistance (Eq. (6.9)) as follows.

Proposition 6.5. *The objective function Eq.* (6.11) *can be rewritten as follows.*

$$\mathcal{J}_R(\{V_\ell\}_{\ell=1}^k) = \frac{1}{2}\sum_{\ell\in[k]}\sum_{i,j\in V_\ell} \frac{r'_G(i,j)}{|V_\ell|} \tag{6.13}$$

This proposition means that the k-means objective using $\mathbf{v}'_i$ (Eq. (6.11)) can be seen as the sum of the extended resistances. Since Eq. (6.13) itself seems a natural objective for graph clustering, our k-means Eq. (6.11) also may be seen as a natural objective. We also show that minimizing $\mathcal{J}_R(\{V_j\}_{j=1}^k)$ (Eq. (6.11) and its equivalence Eq. (6.13)) has a theoretical connection to spectral clustering as follows;

Theorem 6.6. *If we relax Z_R into real values and $n_1 b > \lambda_{K+1}^{-1}$, we have*

$$\arg\min_{Z_R}\{\mathrm{RCut}(\{V_\ell\}_{\ell=1}^k) \text{ s.t. } Z_R^\top Z_R = I\} = \arg\min_{Z_R}\{\mathcal{J}_R(\{V_\ell\}_{\ell=1}^k) \text{ s.t. } Z_R^\top Z_R = I\} \tag{6.14}$$

This theorem means that that ratio cut and k-means using $\mathbf{v}'_i$ are theoretically equivalent if we relax Z_R. By this theorem, Eq. (6.11), featureless version of Eq. (6.10) using the common featureless technique $X = I$, are theoretically justified in a sense of k-means.

Remark that Thm. 6.6 revisits the spectral connection between k-means and spectral clustering as seen in Sec. 2.2. However, the previous connections only hold for the vector data and a feature map, not for the discrete graph data like Thm. 6.6. Moreover, from Prop. 6.5 and Thm. 6.6, the clustering using resistance and spectral clustering are equivalent in a relaxed sense, which the previous connections have not shown. Finally, while the previous connections only hold for the normalized cut, Thm. 6.6 is the first to show the spectral connection for the

ratio cut. Note that Thm. 6.6 naturally generalizes to normalized cut. For more details on how the previous connection and Thm. 6.6 differ, see Sec. 6.4.3.

6.4.2.2 Justification for the Graph-With-Features Setting: A k-means View

This section justifies the k-means objective for the graph-with-features setting Eq. (6.10). In Sec. 6.4.2.1, we saw that Eq. (6.11), which is a featureless setting of Eq.(6.10), is equivalent to the spectral clustering. This section shows that Eq. (6.10) is a "natural extension" of spectral clustering through Eq.(6.11).

We first recall that the common technique (see, e.g., [Kipf and Welling, 2016a,b]) to apply a graph-with-features method to featureless setting is substituting $X = I$. Thus, it is natural to think in a "reverse way"; in order to generalize the featureless methods to graph with the features method, we replace I to the feature vector X. Since Eq. (6.12) is for a featureless setting, we now explicitly write I as

$$\mathcal{J}_R(\{V_\ell\}_{\ell=1}^k) = \|L_b^{-1/2} I - Z_R Z_R^\top L_b^{-1/2} I\|_{\mathrm{Fro}}^2. \tag{6.15}$$

Looking at Eq. (6.15), this can be thought as a featureless setting of the following objective function;

$$\mathcal{J}'_G(\{V_\ell\}_{\ell=1}^k) := \|L_b^{-1/2} X^\top - Z_R Z_R^\top L_b^{-1/2} X^\top\|_{\mathrm{Fro}}^2. \tag{6.16}$$

Using $\mathbf{m}_{G,j}$ in Eq. (6.10), we further rewrite Eq. (6.16) as

$$\mathcal{J}'_G(\{V_\ell\}_{\ell=1}^k) = \sum_{\ell\in[k]} \sum_{i\in V_\ell} \|\mathbf{x}_{G,i} - \mathbf{m}_{G,\ell}\|_2^2 = \mathcal{J}_G(\{V_\ell\}_{\ell=1}^k), \tag{6.17}$$

by which we show that $\mathcal{J}_G(\{V_\ell\}_{\ell=1}^k)$ Eq. (6.10) and $\mathcal{J}'_G(\{V_\ell\}_{\ell=1}^k)$ Eq.(6.16) are equal.

What does the equivalence between $\mathcal{J}_G(\{V_\ell\}_{\ell=1}^k)$ and $\mathcal{J}'_G(\{V_\ell\}_{\ell=1}^k)$ mean? We begin with $\mathcal{J}'_G(\{V_\ell\}_{\ell=1}^k)$. The objective $\mathcal{J}'_G(\{V_\ell\}_{\ell=1}^k)$ can be seen as a generalization of $\mathcal{J}_R(\{V_\ell\}_{\ell=1}^k)$ (Eq.(6.11)) from featureless to graph-with-features setting. Recall that from Thm. 6.6, the featureless $\mathcal{J}_R(\{V_\ell\}_{\ell=1}^k)$ is equivalent to the standard spectral clustering. Thus, by stretching this idea from the featureless to the graph-with-features, $\mathcal{J}'_G(\{V_\ell\}_{\ell=1}^k)$ can be seen as a natural

extension of spectral clustering to graph-with-features setting through a k-means perspective. Hence, since $\mathcal{J}'_G(\{V_\ell\}_{\ell=1}^k) = \mathcal{J}_G(\{V_\ell\}_{\ell=1}^k)$, we may say that the k-means $\mathcal{J}_G(\{V_\ell\}_{\ell=1}^k)$ we initially discuss in Eq. (6.10) can be seen as a natural "extended" spectral clustering for graph-with-features, seen through a k-means lens. Thus, we now establish a connection from k-means to the "extended" spectral connection using the common technique from the featureless to graph-with-features. In this sense, we may justify using X_G, similarly to Dhillon et al. [2004].

Finally, Thm. 6.6 also offers insights into the graph-with-features setting. From Thm. 6.6, we see that the basis $\mathbf{v}'_i$ has a graph structural information through spectral clustering. Thus, we can say that the ResTran $x_{G,i}$ captures more graph structure than $\mathbf{x}_i$ since ResTran replaces the basis from $\mathbf{e}_i$ to $\mathbf{v}'_i$.

6.4.3 Comparison with Theorem 6.6 and Weighted Kernel k-means

This section expands the explanation on the comparison between Thm. 6.6 and the previous weighted kernel k-means. We recall that Thm. 6.6 revisits the spectral connection between k-means and spectral clustering, extensively studied as we saw in Sec. 2.2. However, the previous connections is different than Thm. 6.6 in a number of sense.

Vector vs. Discrete. Most of the previous spectral connections are applied to vectors but not discrete graph data. Seeing Eq. (2.50), the weighted kernel k-means only applies to the vector data $X = (\mathbf{x}_1, \dots, \mathbf{x}_n)$. We construct a graph G whose adjacency matrix is a gram matrix, i.e., construct a graph whose weight is

$$a_{ij} = k_{ij} = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j), \tag{6.18}$$

where K is a gram matrix as defined in Sec. 2.2. The weighted kernel k-means is equivalent to the normalized cut on this graph. Thus, this previous connection assumes for the vector data. On the other hand, our connection can be for a "given" graph data $G = (V, E)$, and thus we do not have to assume any vector data. Note that Dhillon et al. [2007] connects discrete data to k-means in a different manner than us, which we discuss later.

Laplacian Coordinate Insights. Ours offers the Laplacian coordinate insights; seeing the Eq. (6.11), if we use $\mathbf{v}'_i$ to represent i-th vertex and put this vector into the standard k-means objective function, this is equivalent to the spectral clustering. On the other hand, the

weighted kernel k-means cannot be applied to this setting; the previous connection does not incorporate our connection Thm. 6.6. Two potential scenarios to reach Laplacian coordinate insights can be considered. One is a kernel mapping scenario. A naive application of the weighted k-means to the previous framework is to use L^+ as a kernel and $\langle \cdot, \cdot \rangle_L$ as an inner product. However, this Eq. (2.50) is not equivalent to the discrete spectral clustering. The other scenario is incorporating the weight to the standard setting. Recall that our insights come from the standard k-means. Thus, if we aim the standard k-means from the weighted kernel k-means, we compute

$$\mathcal{J}_\phi(\{C_\ell\}_{\ell=1}^k) = \sum_{j=1}^{k} \sum_{i \in C_j} w(\mathbf{x}_i) \|\phi(\mathbf{x}_i) - \mathbf{m}_{\phi,j}\|^2, \ \mathbf{m}_{\phi,j} := \sum_{\ell \in C_j} w(\mathbf{x}_\ell)\phi(\mathbf{x}_\ell) / \sum_{\ell \in C_j} w(\mathbf{x}_\ell) \tag{6.19}$$

$$= \sum_{j=1}^{k} \sum_{i \in C_j} \|w^{1/2}(\mathbf{x}_i)\phi(\mathbf{x}_i) - w^{1/2}(\mathbf{x}_i)\mathbf{m}_{\phi,j}\|^2. \tag{6.20}$$

However, this transformation does not go anywhere close to the standard k-means. To conclude, the previous connection does not incorporate Thm. 6.6, and thus does not offer the Laplacian coordinate insights.

Connection to the k-means using Resistance. Finally, we would like to point out that the previous connection does not connect to k-means. The previous connections, such as Dhillon et al. [2007], connect discrete data and ratio cut and normalized cut. We can connect to ratio cut if using $K = \sigma I - L$ and normalized cut if using $K = \sigma D^{-1} - L_N$ for $\sigma \geq 0$. However, this connection does not conclude Prop. 6.5, which provides the connection between k-means using resistance and spectral clustering. Furthermore, Thm. 6.6 naturally generalizes to normalized cut. Let $\mathbf{v}_i'' := \sqrt{d_i}\mathbf{v}_i'$. Then, we define the objective function and expand similarly in Sec. 6.G as

$$\mathcal{J}_N(\{V_\ell\}_{\ell=1}^k) := \sum_{\ell \in [k]} \sum_{i \in V_\ell} \|\mathbf{v}_i'' - \mathbf{m}_\ell\|_2^2, \ \mathbf{m}_\ell := \sum_{j \in V_\ell} \mathbf{v}_j''/|V_\ell|, \mathbf{v}_j' \in \mathcal{V}'(L_b) \tag{6.21}$$

$$= \mathrm{trace} D^{1/2} L_b^{-1} D^{1/2} - \mathrm{trace} Z_R D^{1/2} L_b^{-1} D^{1/2} Z_R. \tag{6.22}$$

Therefore, minimizing Eq. (6.21) subject to $Z_R^\top Z_R = I$ is equivalent to top k eigenvector problem of $D^{1/2} L_b^{-1} D^{1/2}$. This is equivalent to the smallest k eigenvectors of $D^{-1/2} L D^{-1/2}$,

by which we show that Thm. 6.6 naturally generalizes the ratio cut to the normalized cut. This normalized cut connects to the normalized measure discussed in Liben-Nowell and Kleinberg [2003].

6.5 Related Work

This section provides a review of the related work to our ResTran.

Spectral Connection. Our justification relies on the connection between spectral clustering, effective resistance and k-means. The spectral clustering using ratio and normalized cut has been extensively studied [Fiedler, 1975b, Shi and Malik, 2000]. The Laplacian coordinate and effective resistance are used for the various learning problem such as clustering [Fouss et al., 2007, Saito and Herbster, 2023b, Yen et al., 2008, 2005] and online learning [Herbster and Pontil, 2006, Herbster et al., 2005]. The connection between normalized cut and weighted kernel k-means has been developed, such as [Bach and Jordan, 2003, Dhillon et al., 2004, Saito, 2022]. The connection between ratio cut, effective resistance, and k-means are loosely studied [Saerens et al., 2004, Zha et al., 2001]. However, these studies do not give the "exact" connection between ratio cut and spectral clustering like Thm. 6.6. Also, the previous studies do not give the Resistance Transformation interpretation. We remark that there exist other connections between weighted kernel k-means and other matrix decomposition methods by using different constraints. Ding and He [2004] shows a connection between k-means with addition of constraints and principal component analysis [Pearson, 1901, Lakhina et al., 2004, Saito et al., 2015b, Wold et al., 1987], and Ding et al. [2005] provide a connection between k-means with other constraints and non-negative matrix factorization [Lee and Seung, 1999, 2000, Saito et al., 2015a, Wang and Zhang, 2012].

GNNs. Since our ResTran aims to address the graph-with-features problem, one popular approach to this problem is GNN. The GNN is firstly proposed as a neural network applied to the graph structural data [Gori et al., 2005, Scarselli et al., 2008]. The GCN [Kipf and Welling, 2016a] and GAT [Veličković et al., 2018] are established methods. The recent advancements include [Gasteiger et al., 2020, Hamilton et al., 2017, Pei et al., 2020, Xie et al., 2016] to name a few; see [Wu et al., 2020] for more comprehensive survey. The closest approach in the sense of formulation to our ResTran is SGC [Wu et al., 2019]. The SGC aims to simplify

ℓ layers of GCN. The SGC is formulated as

$$\begin{aligned}\hat{\mathbf{y}} =& \mathrm{softmax}(\tilde{A}^{\ell} X^{\top} \Omega), \\ & \text{where } \tilde{A} := (D+I)^{-1/2}(A+I)(D+I)^{-1/2},\ \Omega := \Omega^{(1)} \dots \Omega^{(\ell)},\end{aligned} \tag{6.23}$$

where $\Omega^{(i)}$ is a i-th layer of a fully-connected layer. This approach is close to ours for the following reason. If we apply ℓ layers of fully connected to ours, and then this can be written as $\hat{\mathbf{y}} = \mathrm{softmax}(X_G^{\top}\Omega) = \mathrm{softmax}(L_b^{-1/2} X^{\top}\Omega)$. The SGC is close since, in this setting, the difference is $\tilde{A}$ and $L_b^{-1/2}$. However, our approach is not limited to this formulation, but we can apply any building blocks, especially, activate functions such as ReLU. There have been some follow-ups on this simple approach [Chen et al., 2020, Salha et al., 2019, 2021, Zhu and Koniusz, 2021]. Another relevant approach is PinvGCN [Alfke and Stoll, 2021]. For a dense graph aiming for faster GCN, PinvGCN reconstructs three graphs by heuristic approximation of L^{+}, runs GCN for each graph, and then combines the results. While these studies heuristically simplify the GCN in some similar manner, we provide a theoretical justification on Resistance Transformation in Sec. 6.4. Also, again our ResTran is not limited to simplfied GCN models. In addition to various models of GNNs, transformers using the eigenvectors of Laplacian as positional encoding are considered [Dwivedi et al., 2023, Wang et al., 2022]. Also, Convolutional GNNs also exploit spectral properties such as [Bruna et al., 2014, Henaff et al., 2015]. The polynomial approximation strategy is a standard practice to obtain the spectra of graph Laplacian, such as [Defferrard et al., 2016, Kipf and Welling, 2016a]. Moreover, Krylov subspace method is used for the better approximation for the convolutional GNNs [Luan et al., 2019]. However, these studies are on specific GNNs while ours can be applied to any vector based model. Some common problems to GNN are reported: limited expressive power [Xu et al., 2019] and over-squashing [Di Giovanni et al., 2023, Topping et al., 2021, Black et al., 2023]. The most relevant problem to this study is the "low-frequency bias" of GNNs, where GNNs tend to learn only homophilous information [Chang et al., 2021, Du et al., 2022, Hoang and Maehara, 2019, Hoang et al., 2020, Zheng et al., 2022, Zhu et al., 2003, Luan et al., 2022, Platonov et al., 2023, Bonchi et al., 2023]. This phenomenon gets worse if we stack the GNN layers, which is known as "over-smoothing" [Li et al., 2018, Oono and Suzuki, 2019]. By construction, our Resistance Transformation are expected to represent not only homophilous information but also heterophilous information.

Table 6.1: Homophilous dataset summary.

	Cora	Citeseer	Pubmed	Photo	Computer
$\|V\|$	2708	3327	19717	7650	13752
$\|E\|$	5429	4732	44338	119081	245861
Classes	7	6	3	8	10
Features	1433	3703	500	745	767

SSLs. Since this work is related to semi-supervised learning problems, this section reviews the SSL studies in detail. The SSL over graph is extensively studied [Blum et al., 2004, Zhou et al., 2003, Zhu et al., 2003]. Unlike GNNs, these only use the graph topology. The Planetoid [Yang et al., 2016] is an SSL method that incorporates features and the topology at the same time, while most of the GNN models are known to outperform Planetoid. The SSL models for the vector dataset are also discussed. The early models include SVM-based one [Joachims, 1999], and early NeuralNet models [Ranzato and Szummer, 2008, Weston et al., 2008]. Also, we apply a kernel function to the vector to form a graph and apply the graph-based SSL models. One of the early established deep neural network-based SSL methods is variational autoencoder (VAE) [Kingma et al., 2014], which is simplified by the follow-up study called Auxiliary VAE (AVAE) [Maaløe et al., 2016]. Since then, there have been various improvements including [Laine and Aila, 2017, Miyato et al., 2018, Yang et al., 2022]. However, none of these aim to incorporate the graph and features. Instead, we can apply these methods to our X_G, unless the models are not designed to some specific tasks, e.g., images [Berthelot et al., 2019, Kurakin et al., 2020, Sohn et al., 2020, Zhang et al., 2021].

6.6 Experiments

This section numerically demonstrates the performance of ResTran.

Objective of the Experiments. The purpose of our experiments is to evaluate if our ResTran X_G improves i) the graph-only or feature-only representation and ii) the existing GNN methods. Recall that we propose to use ResTran X_G and to apply a vector-based machine learning method. Thus, various sophistications can be involved in both ResTran and the comparison methods. However, to focus on evaluating our ResTran, we want to exclude the effects of sophistication as much as possible. To do so, our experiments only used simple and established methods for both ResTran and the comparison. We used Alg. 6 for ResTran.

Datasets. For the homophilous dataset, we used the standard citation network benchmark;

Table 6.2: Heterophilous dataset summary.

	Texas	Cornell	Wisconsin	chameleon	squirrel	actor
$\|V\|$	183	183	251	2277	5201	7600
$\|E\|$	295	309	499	31421	198493	26752
Classes	5	5	5	5	5	5
Features	1703	1703	1703	2325	2089	932

Cora [McCallum et al., 2000], Citeceer [Sen et al., 2008], and Pubmed [Namata et al., 2012]. We also used the two Amazon co-purchase graphs, photo, and computer [McAuley et al., 2015]. The homophilous dataset statistics are summarized in Table 6.1 For the heterophilous dataset, we used web data, Wisconsin, Cornell, and Texas, all of which are a part of WebKB [Craven et al., 1998]. We also used the Wikipedia dataset chameleon and squirrel [Rozemberczki et al., 2021], as well as actor [Pei et al., 2020]. The heterophilous dataset statistics are summarized in Table 6.1. Note that the difference between homophilous datasets and heterophilous datasets has been discussed in a variety of the literature, such as [Luan et al., 2022, Platonov et al., 2023]. Note that we are aware of large pools of the benchmarks for this purpose, such as OGB. However, like the experimental purpose where this chapter focuses on the comparison with simple and established models and settings, we focus on the established and long-used benchmarks.

Experimental Settings for Unsupervised and Supervised tasks. We evaluated ResTran and existing methods by accuracy, same as the previous studies such as [Kipf and Welling, 2016a, Veličković et al., 2018]. Note that, throughout the experiments, we used $b = 1/(n\lambda_{K+1})$ for ResTran, that is the condition of Thm. 6.6. Also, we used the Krylov subspace dimension $r = 20$, since our preliminary experiements show that performances do not change when $r > 20$. Our experiments were conducted on Google Colab Pro+, Matlab, and Mac Studio with M1 Max Processor and 32GiB RAM. The experimental code is at `https://github.com/ShotaSAITO/ResTran`.

6.6.1 Comparing ResTran with Graph-Only and Feature-Only

This experiment compares ResTran with Graph-Only and Feature-Only.

Objective of the Experiments. This experiment briefly evaluates if our ResTran for representing the graph-with-features datasets improves the feature-only X and graph-only A. If we observe that the latent space is more separable for ResTran X_G than for graph-only and

Table 6.3: Experimental results for unsupervised learning. All measures are accuracy (%). "Graph-Only" uses only graph Laplacian. "Feature-only" uses a Gram matrix constructed only by features. "Graph + Feature" uses a Gram matrix constructed by our proposal X_G.

	Cora	Citeseer	Pubmed	Texas	Cornell	Wisconsin
Graph-Only	29.3 ± 0.5	23.7 ± 0.0	39.6 ± 0.0	49.6 ± 1.1	49.0 ± 5.6	45.6 ± 4.2
Feature-Only	32.6 ± 0.6	45.5 ± 0.9	45.4 ± 0.0	55.2 ± 0.5	55.2 ± 0.0	47.8 ± 0.0
Graph + Feature (Ours)	$\mathbf{58.9 \pm 4.5}$	$\mathbf{48.2 \pm 0.8}$	$\mathbf{71.6 \pm 0.6}$	$\mathbf{55.5 \pm 0.5}$	$\mathbf{55.7 \pm 0.0}$	$\mathbf{48.2 \pm 0.3}$

feature-only settings, we can say that ours improves the representation. For this purpose, we compare these using the simple setting of spectral clustering.

Experimental Settings. For the feature-only and ResTran, we used the Gaussian kernel to form a graph and applied spectral clustering. For graph-only, we used the graph Laplacian for the spectral clustering. We conducted a simple k-means on the first k eigenvectors of the graph Laplacian, and we reported the average. More specifically, for the feature only and ours, we computed the edge weight with a Gaussian kernel ($\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\sigma\|\mathbf{x}_i - \mathbf{x}_j\|^2)$) for two vectors $\mathbf{x}_i, \mathbf{x}_j$. We used free parameter $\sigma \in \{10^{-2}, \ldots, 10^3\}$. To gain the sparsity, we further constructed a 100-NN graph from these gram matrices, which is a common technique. We compute the smallest k eigenvectors of unnormalized Laplacian for all three graphs. Then, we apply the standard k-means to the smallest k eigenvectors in order to obtain the clustering results. Since the k-means algorithm depends on the initial condition, we repeated it 10 times and reported the average and standard errors. We conduct this experiment on the smaller sets of the datasets, both from homophilous and heterophilous datasets, since from the preliminary experiments, we observe similar patterns for the other datasets.

Overall Results. We see that ResTran offers better separation than graph-only and feature-only. The results of the unsupervised task are summarized in Table 6.3. In all datasets, we see that ResTran improves both graph-only and feature-only. These results further confirm that ResTran X_G better represents the dataset than the feature only X or the graph only A.

6.6.2 Comparing ResTran with GNN Methods.

This section compares ResTran with GNN methods.

Objective of the Experiments. Here, we evaluate whether ResTran improves the existing GNN methods. We evaluate this for the SSL tasks both on homophilous and heterophilous datasets.

Experimental Settings. We first introduce the comparison methods. Recall that our experiments only used simple and established methods for both our proposal and the comparison since we want to exclude the effects of sophistication as much as possible. For comparison, we used three established simple GNN models, GCN [Kipf and Welling, 2016a], GAT [Veličković et al., 2018], as well as SGC [Wu et al., 2019], which is a simplified GCN. For ResTran, we apply both non-NeuralNet vector-based models and NeuralNet-based models. We discuss some details of the methods we used for ResTran. For non-NeuralNet models, LP [Zhu et al., 2003] is one of the established models in SSL, as we saw in Appendix 6.5. The SVM [Cortes and Vapnik, 1995] is also an established model, while SVM is a supervised learning model in general. However, in this context, we can interpret the SVM as an SSL method, since, even though we only use the indices corresponding the training set, i.e., $\{(X_G)_{\cdot i}\}_{i\in Tr}$, in ResTran Eq. (6.8), the transformation uses the whole L and X but not $\{y_i\}_{i\in Te}$. Remark that we only use the training set $\{\mathbf{x}_{G,i}\}_{i\in Tr}$ to form a gram matrix and therefore the gram matrix is the size $|Tr|\times|Tr|$ matrix. For NeuralNet models, as we discussed in Appendix 6.5, AVAE [Kingma et al., 2014] is a simpler version of the SSL via VAE, which is the one of the earliest NeuralNet based SSL models. Also, VAT [Miyato et al., 2018] is the one early established NeuralNet based SSL model using generative adversarial network behind the scene. We then discuss the hyperparameters. To conduct a fair comparison, we endeavored to use the same settings for ours and compare as much as possible. We used non-normalized features for both methods. For non-NeuralNet based models, we again used a Gaussian Kernel and used free parameter $\sigma\in\{10^{-2},\dots,10^{3}\}$, as done in the unsupervised learning setting. For non-NeuralNet models, we apply label propagation (LP) [Zhou et al., 2003] and SVM [Cortes and Vapnik, 1995] with the Gaussian kernel for X_G. For NeuralNet models, we use two early and simple models, VAT [Miyato et al., 2018] and AVAE [Maaløe et al., 2016] to X_G. We only use fully connected layers and ReLU as an activation function for our NeuralNet models, which are simple and established NeuralNet components. For AVAE, the first fully connected layer contains 256 hidden units, and the second fully connected layer contains 128 hidden units. For VAT, the first fully connected layer contains 1028 hidden units, and the second fully connected layer contains 512 hidden units. Also, each layer was activated by ReLU. Finally, we passed to the output layer. For AVAE, we used the embedding dimension as 30 and the dimension of the auxiliary variable as 30. We used batch size 128. We applied the learning

Table 6.4: Experimental results for homophilous data using semi-supervised learning with some known labels. We use 5% labels. All measures are accuracy (%).

	Type	cora	citeseer	pubmed	photo	computer
GCN	GNN	**79.9 ± 0.9**	67.4 ± 1.1	83.8 ± 0.4	83.1 ± 1.2	80.4 ± 0.4
GAT	GNN	74.9 ± 4.2	67.6 ± 0.1	82.8 ± 0.2	**87.7 ± 1.3**	80.3 ± 1.2
SGC	GNN	79.3 ± 1.7	70.2 ± 0.8	67.9 ± 1.8	80.1 ± 2.9	81.4 ± 2.0
ResTran + LP	Non-NeuralNet	30.6 ± 0.6	20.6 ± 4.6	39.5 ± 1.4	25.3 ± 0.2	37.5 ± 2.2
ResTran + SVM	Non-NeuralNet	49.1 ± 5.7	45.5 ± 6.7	76.5 ± 2.2	24.3 ± 2.7	43.8 ± 3.4
ResTran + VAT	NeuralNet	77.6 ± 2.5	68.7 ± 1.1	82.8 ± 0.7	86.3 ± 0.8	78.1 ± 2.4
ResTran + AVAE	NeuralNet	78.2 ± 1.8	**71.7 ± 1.0**	**83.9 ± 0.7**	86.8 ± 1.5	**81.6 ± 0.9**

Table 6.5: Experimental results for heterophilous data using semi-supervised learning with some known labels. We use 5% labels. All measures are accuracy (%).

	Type	Texas	Cornell	Wisconsin	chameleon	squirrel	actor
GCN	GNN	50.9 ± 4.2	37.4 ± 9.3	46.3 ± 4.9	32.7 ± 2.0	23.5 ± 1.1	25.9 ± 0.9
GAT	GNN	50.3 ± 3.3	44.9 ± 4.9	44.0 ± 4.8	32.8 ± 1.8	23.4 ± 1.3	26.4 ± 0.9
SGC	GNN	44.6 ± 5.0	42.3 ± 5.3	44.6 ± 5.0	31.8 ± 1.8	23.5 ± 0.8	26.0 ± 0.8
ResTran + LP	Non-NeuralNet	46.3 ± 17.3	42.2 ± 20.6	37.3 ± 12.6	20.3 ± 0.8	20.0 ± 0.3	22.3 ± 2.8
ResTran + SVM	Non-NeuralNet	48.8 ± 14.1	45.7 ± 16.8	47.8 ± 9.6	33.6 ± 5.8	31.9 ± 0.9	29.4 ± 0.9
ResTran + VAT	NeuralNet	**55.9 ± 5.1**	**49.0 ± 3.8**	**51.2 ± 5.0**	34.0 ± 1.4	27.7 ± 3.5	27.8 ± 1.2
ResTran + AVAE	NeuralNet	51.4 ± 3.7	48.2 ± 3.7	50.0 ± 2.1	**40.7 ± 1.4**	**32.4 ± 0.8**	**29.5 ± 1.3**

rate of 0.01 to Adam for AVAE. We used two hidden layers for both of the NeuralNet methods for Restran and our comparisons. For all of the NeuralNet-based settings, we used a dropout rate of 0.2. We train all models for 100 epochs using the Adam optimizer. We conducted our experiments with the split where we know 5% labels, we use 25% for validation, and the rest for the test. We conducted our experiments on 10 random splittings and reported the average. For the comparison, apart from the setting above, we used the implementation and hyperparameters as implemented in the examples of PYTORCH-GEOMETRIC[1]. Finally, remark that for citation network benchmarks, although various studies use the public splittings in [Yang et al., 2016], we avoided using these since overfitting to this specific splitting is reported [Shchur et al., 2018].

Overall Results. The results are summarized in Table 6.4 and 6.5. On homophilous datasets, we observe comparable performances among GNNs and ResTran + NeuralNet models. On heterophilous datasets, we observe the performance improvement from GNNs to ResTran, sometimes even with SVM. This means that our ResTran is more robust to homophily bias. This robustness is expected from the construction of ResTran since, unlike

[1] `https://github.com/pyg-team/pytorch_geometric/tree/master/examples`

GNNs, X_G preserves not only homophilous information but also heterophilous information as seen in Sec. 6.4.1.

Computational Time. In the experiments, we have opted not to report computational time since it does not provide meaningful insights for comparison. This is primarily due to the fundamental differences between our proposed approach and existing GNN algorithms. Our approach involves the proposed transformation step (ResTran), followed by downstream SSL algorithms, which can vary. In contrast, the comparison methods are end-to-end. Our focus is on comparing the performance of our transformation (ResTran) with existing end-to-end GNN algorithms, which requires to plug ResTran into some SSL algorithms. Consequently, the computational time for our method depends on the specific choice of SSL algorithms and architectures, factors that are beyond the scope of this chapter and would only complicate comparison further. Moreover, the transformation step in our approach benefits from the pre-computation of the Krylov subspace method, which cannot be applied to the comparison methods. This difference makes a time comparison inherently unfair. For instance, increasing the number of repetitions in our experiments would disproportionately favor our method due to this pre-computation, but this does not imply that the comparison methods are inherently slower. Moreover, although we repeated the experiments 10 times, it is unclear if this number of repeats (or any number) is a fair basis for determining which method is faster. Therefore, reporting and comparing computational time would not lead to meaningful conclusions and could mislead the reader. However, both ResTran and GNNs exhibit similar computational complexity: ResTran requires $O(rfm)$ and GNNs require $O(tfm)$, where r and t are constants.

6.7 Summary

We considered a vertex classification task on the graph-with-features setting, where we have a graph with associated features. While the modern approach to this task has been GNNs, we took an alternative approach to overcome the homophilous biases in GNNs. Our approach was to transform the feature vectors to incorporate the graph topology and apply standard learning methods to the transformed vectors. For this approach, we proposed a simple transformation of features, which we call *ResTran*. We established theoretical justifications for ResTran from resistance, k-means, and spectral clustering viewpoints. We also discusses why ResTran is robust to homophilous biases. We empirically demonstrated that ResTran is more robust on

the homophilous bias than existing GNN methods. Limitation and future work are that we are unsure how much ResTran has an expressive power, as done in [Xu et al., 2019]. We conjecture that the expressive power of ResTran is less than the 2-WL test. Thus, we speculate that we need a different setup for triangle counting problems.

6.A Note on Krylov Subspace Method

This section breifly explains the Krylov subspace method and its advantages over some natural ideas.

6.A.1 Krylov Subspace Method

In this section, the Krylov subspace method is an established way to approximate the solution of the linear algebraic solutions. In this case, we consider to approximate $f(A)\mathbf{b}$ for the matrix $A \in \mathbb{R}^{n\times n}$ and for a vector $\mathbf{b} \in \mathbb{R}^n$.

The r-th Krylov subsupace $\mathcal{K}_r$ for the matrix $A \in \mathbb{R}^{n\times n}$ and for a vector $\mathbf{b} \in \mathbb{R}^n$ is defined as

$$\mathcal{K}_r(A, \mathbf{b}) := \mathrm{span}\{\mathbf{b}, A\mathbf{b}, A^2\mathbf{b}, \ldots, A^{r-1}\mathbf{b}\}. \tag{6.24}$$

The Krylov subspace method approximates $f(A)\mathbf{b}$ into this Krylov subspace $\mathcal{K}_r(A, \mathbf{b})$. To obtain this approximation, the common way is Arnoldi process. The Arnoldi process at i-th iteration obtains $Q_i \in \mathbb{R}^{n\times i}$ and $H_i \in \mathbb{R}^{i\times i}$ as

$$AQ_i = Q_iH_i + h_{i+1,i}\mathbf{q}_{i+1}\mathbf{e}_i^\top, \text{where } Q_i := [\mathbf{q}_1 \ldots, \mathbf{q}_i], \mathbf{q}_1 := \mathbf{b}/\|\mathbf{b}\|_2^2. \tag{6.25}$$

Note that Q_i has orthonormal columns and H_i is upper Hessenberg matrix. Then, Krylov subspace based method approximates

$$f(A)\mathbf{b} \approx Q_r f(H_r)Q_r^\top \mathbf{b} = \|\mathbf{b}\|_2 Q_r f(H_r)\mathbf{e}_1. \tag{6.26}$$

This process overall takes $O(rm)$ time complexity. Typically, r is chosen small, say $r < 100$. See [Higham, 2008] for more details.

6.A.2 Advantages of Krylov Subspace Method

This section discusses the advantages of the Krylov subspace method over some natural ideas.

One natural idea to approximate $L_b^{-1/2}X$ is to approximate $L_b^{-1/2}$ using polynomial function. This technique is commonly used, even in the GNN research area, such as [Kipf

and Welling, 2016a]. For example, we first expand $L_b^{-1/2}$ as

$$L_b^{-1/2} = a_0 I + a_1 L_b + a_2 L_b^2 + \dots, \tag{6.27}$$

and then approximate in some order, say,

$$L_b^{-1/2} \approx a_0 I + a_1 L_b. \tag{6.28}$$

While this is straightforwardly understandable, the Krylov subspace method approximates $L_b^{-1/2}X$ better as follows. While this polynomial approximation only uses L when approximation, the Krylov subspace method approximates LX using both L and X as seen in Appendix 6.A.1. Hence, the Krylov subspace approximates $L_b^{-1/2}$ using more information than a polynomial approximation.

The other natural idea is to reduce the dimension, such as principal component analysis (PCA). We consider to eigendecompose the graph Laplacian as

$$L = \Psi \Lambda \Psi^\top, \tag{6.29}$$

where $\Psi := (\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_n)$ and $\Lambda := \mathrm{diag}(\lambda_1, \dots, \lambda_n)$, where $\boldsymbol{\psi}_i$ is the i-th eigenvector and λ_i is the i-the eigenvalue. Then, we compose $\Lambda_{r'} := \mathrm{diag}(\lambda_1, \dots, \lambda_{r'}, 0, \dots, 0)$. The value r' is again typically small compared to n. Then, we approximate $L^{+1/2}$ as

$$L^{+1/2} \approx \Psi \Lambda_r^{+1/2} \Psi^\top. \tag{6.30}$$

This approximation can be conducted much faster than obtaining naively $L^{+1/2}$.

While dimensional reduction is the standard way to make pseudoinverse faster, the Krylov subspace method provides a better approximation in the following sense. Firstly, as the polynomial approximation, the Krylov subspace approximates $L_b^{-1/2}X$ with more information. Secondly, as discussed in 6.4.1 and as seen in the experimental result as 6.6, ResTran also works for heterophilous datasets. However, from the construction of the eigendecomposition, the reduction cut down the high-frequency information corresponding to the heterophilous information. Therefore, the dimensional reduction throws away the information that ResTran is good at dealing with.

6.B Additional Definitions for Proofs

This section set ups additional preliminary definitions and facts for proofs.

Without loss of generality, we can reorder G as $G = G_1 \cup \ldots \cup G_K$ and $|G_1| \leq \ldots \leq |G_K|$. For the visual aid of J_G, we can write J_G as

$$J_G = \begin{array}{c} \\ |G_1| \\ \\ \vdots \\ \\ |G_K| \end{array} \begin{array}{c} \begin{array}{ccc} |G_1| & \ldots & |G_K| \end{array} \\ \left(\begin{array}{c|c|c} \begin{matrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{matrix} & & \\ \hline & \ddots & \\ \hline & & \begin{matrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{matrix} \end{array}\right) \end{array}, \tag{6.31}$$

Let $\mathbf{1}_{G_j}$ is all one vector for G_j, i.e., $(\mathbf{1}_{G_j})_i = 1$ if $j \in V_{G_j}$ otherwise 0. Then we have

$$(\mathbf{1}_{G_1} \cdots \mathbf{1}_{G_K})(\mathbf{1}_{G_1} \cdots \mathbf{1}_{G_K})^\top = J_G. \tag{6.32}$$

We also introduce the bound of resistance by the eigenvalue as follows.

Lemma 6.7 (Chandra et al. [1996])**.** *For any $i, j \in V$, we have*

$$r_G(i, j) \leq \frac{2}{\lambda_2} \tag{6.33}$$

Lemma 6.8 (Herbster and Pontil [2006])**.**

$$\max_i \|L^{+1/2}\mathbf{e}_i\|_2^2 \leq \max_{i,j} r_G(i, j) \tag{6.34}$$

By combining these two lemmas, we obtain

$$\max_i \|L^{+1/2}\mathbf{e}_i\|_2^2 \leq \frac{2}{\lambda_2} \tag{6.35}$$

6.C Proofs of Proposition 6.1 and Corollary 6.2

We conduct eigendecomposition on L, and obtain eigenpairs as (λ_k, ψ_k). We define a matrix U and diagonal matrix Λ as

$$\Psi := (\psi_1, \psi_2, \ldots, \psi_n), \Lambda_{kk} := \lambda_k. \tag{6.36}$$

We remark that the psuedoinverse of Λ can be written as

$$\Lambda_{ii} = 0, \quad \text{for } i = 1, \ldots, K \tag{6.37}$$

$$\Lambda_{ii}^{+1/2} = 1/\lambda_i^{1/2}, \quad \text{for } i \geq K+1. \tag{6.38}$$

Now we define an $n \times n$ matrix Λ_b which has only one element, as

$$(\Lambda_b)_{ii} = 1/n_{G_j} b \quad \text{for } i \in V_{G_j} \tag{6.39}$$

We can then write as

$$\begin{aligned} L_b^{-1/2} &= \Psi \Lambda^{+1/2} \Psi^\top + \sqrt{b} J_G \\ &= \Psi \Lambda^{+} \Psi^\top + \Psi \Lambda_b^{+1/2} \Psi^\top \\ &= \Psi (\Lambda^{+1/2} + \Lambda_b^{+1/2}) \Psi^\top. \end{aligned} \tag{6.40}$$

Thus, for $\ell > K$, the eigenvector associated with $\lambda_\ell^{-1/2}$ is ψ_i. From Eq. (6.39), for $\ell \leq K$ the eigenvalue associated with ψ_ℓ is $\sqrt{n_{G_\ell} b}$, where $|G_1| \leq \ldots \leq |G_\ell| \leq \ldots \leq |G_K|$. If $n_{G_1} b > \lambda_2^{-1}$, $n_{G_i} b$ is the largest K eigenvalues. This concludes the proof for Prop. 6.1.

Eq. (6.40) yields the Cor. 6.2.

Finally, by generalizing the fact that the square root of the all one matrix can be written as $(\mathbf{1}\mathbf{1}^\top)^{1/2} = \mathbf{1}\mathbf{1}^\top/\sqrt{n}$, we have

$$J_G^{1/2} = \begin{array}{c} \\ |G_1| \\ \\ \\ |G_K| \end{array} \begin{array}{c} \begin{array}{cccc} |G_1| & & \ldots & |G_K| \end{array} \\ \left(\begin{array}{c|c|c} \begin{matrix} 1/\sqrt{n_1} & \cdots & 1/\sqrt{n_1} \\ \vdots & \ddots & \vdots \\ 1/\sqrt{n_1} & \cdots & 1/\sqrt{n_1} \end{matrix} & & \\ \hline & \ddots & \\ \hline & & \begin{matrix} 1/\sqrt{n_K} & \cdots & 1/\sqrt{n_K} \\ \vdots & \ddots & \vdots \\ 1/\sqrt{n_K} & \cdots & 1/\sqrt{n_K} \end{matrix} \end{array}\right) \end{array} \tag{6.41}$$

From the proof of Prop. 6.1, we immediately have the following corollary.

Corollary 6.9. *Let $(\lambda_\omega, \psi_\omega)$ be the ω-th eigenpair of L. Suppose that a graph G is connected. If $nb > \lambda_2^{-1}$, i-th eigenpair (λ_i^+, ψ_i^+) of L_b^{-1} are*

$$(\lambda_i^+, \psi_i^+) = \left(\lambda_{n+1-i}^{-1}, \psi_{n+1-i}\right) \text{ for } i = 1, \ldots, n-1, \quad \left(nb, \mathbf{1}/\sqrt{n}\right) \text{ for } i = n$$

6.D Proof of Proposition 6.3

Using Cor. 6.2, we obtain

$$\|\mathbf{v}_i' - \mathbf{v}_j'\|_2^2 = \|(\mathbf{v}_i + b\mathbf{1}^\top\mathbf{1}\mathbf{e}_i) - (\mathbf{v}_j - b\mathbf{1}^\top\mathbf{1}\mathbf{e}_i)\|_2^2 = \|\mathbf{v}_i - \mathbf{v}_j\|_2^2 \tag{6.42}$$

Using the fact of Eq. (2.71), we conclude the proof.

6.E Proof of Proposition 6.4

Without loss of generality, we write as

$$r_G(i,j) = \left\| \begin{pmatrix} L_{G_s}^{+1/2} + \sqrt{bn_{G_s}}\mathbf{1}_{G_s} \\ 0 \\ 0 \end{pmatrix} \mathbf{e}_i - \begin{pmatrix} 0 \\ L_{G_t}^{+1/2} + b\sqrt{bn_{G_t}}\mathbf{1}_{G_t} \\ 0 \end{pmatrix} \mathbf{e}_j \right\|_2^2 \tag{6.43}$$

$$= \left\| \begin{pmatrix} \sqrt{bn_{G_s}}\mathbf{1}_{G_s} \\ \sqrt{bn_{G_t}}\mathbf{1}_{G_t} \\ 0 \end{pmatrix} (\mathbf{e}_i - \mathbf{e}_j) - \begin{pmatrix} L_{G_s}^{+1/2} \\ L_{G_t}^{+1/2} \\ 0 \end{pmatrix} (\mathbf{e}_j - \mathbf{e}_i) \right\|_2^2 \tag{6.44}$$

$$\geq \left(\left\| \begin{pmatrix} \sqrt{bn_{G_s}}\mathbf{1}_{G_s} \\ \sqrt{bn_{G_t}}\mathbf{1}_{G_t} \\ 0 \end{pmatrix} (\mathbf{e}_i - \mathbf{e}_j) \right\|_2 - \left\| \begin{pmatrix} L_{G_s}^{+1/2} \\ L_{G_t}^{+1/2} \\ 0 \end{pmatrix} (\mathbf{e}_j - \mathbf{e}_i) \right\|_2 \right)^2 \tag{6.45}$$

$$= \left((bn_{G_s} + bn_{G_t})^{1/2} - \left\| \begin{pmatrix} L_{G_s}^{+1/2}(\mathbf{e}_{G_s})_i \\ L_{G_t}^{+1/2}(\mathbf{e}_{G_t})_j \end{pmatrix} \right\| \right)^2 \tag{6.46}$$

$$\tag{6.47}$$

The second to third line follows from triangle inequality. We now show that the first term is strictly larger than the second term. The first term is bounded as

$$bn_{G_s} + bn_{G_t} \geq 2bn_{G_1}, \tag{6.48}$$

and

$$\left\| \begin{pmatrix} L_{G_s}^{+1/2}(\mathbf{e}_{G_s})_i \\ L_{G_t}^{+1/2}(\mathbf{e}_{G_t})_j \end{pmatrix} \right\| = (\|L_{G_s}^{+1/2}(\mathbf{e}_{G_s})_i\|_2^2 + \|L_{G_t}^{+1/2}(\mathbf{e}_{G_t})_j\|_2^2)^{1/2} \tag{6.49}$$

$$\leq (\max_{i,j} r_{G_s}(i,j) + \max_{i,j} r_{G_t}(i,j))^{1/2} \tag{6.50}$$

$$\leq (2/\lambda_{K+1} + 2/\lambda_{K+1})^{1/2} \tag{6.51}$$

$$= 2\lambda_{K+1}^{1/2} \tag{6.52}$$

Therefore, due to the assumption that $b > (1+\sqrt{2})^2/n_{G_1}\lambda_{K+1}$, we have

$$(bn_{G_s} + bn_{G_t})^{1/2} \geq \left\| \begin{pmatrix} L_{G_s}^{+1/2}(\mathbf{e}_{G_s})_i \\ L_{G_t}^{+1/2}(\mathbf{e}_{G_t})_j \end{pmatrix} \right\| \tag{6.53}$$

We also have if $\min x \geq \max y \geq 0$, then

$$(x-y)^2 > (\min x - \max y)^2 \tag{6.54}$$

since $x - y > \min x - \max y > 0$. By using these relations, we obtain

$$r_G(i,j) \geq \left((bn_{G_s} + bn_{G_t})^{1/2} - \left\| \begin{pmatrix} L_{G_s}^{+1/2}(\mathbf{e}_{G_s})_i \\ L_{G_t}^{+1/2}(\mathbf{e}_{G_t})_j \end{pmatrix} \right\| \right)^2 \tag{6.55}$$

$$\tag{6.56}$$

$$\geq \left((2bn_{G_1})^{1/2} - \frac{2}{\lambda_{K+1}^{1/2}} \right)^2 \tag{6.57}$$

$$\geq \frac{2}{\lambda_{K+1}} \geq r_G(i,j) \tag{6.58}$$

6.F Proof of Proposition 6.5

We now start with the standard k-means objective function using the general norm $\|\cdot\|$ is defined as

$$\mathcal{J}(\{C_\ell\}_{\ell=1}^k) := \sum_{\ell \in [k]} \sum_{i \in C_\ell} \|\mathbf{x}_i - \mathbf{m}_\ell\|^2, \quad \mathbf{m}_\ell := \sum_{j \in C_\ell} \mathbf{x}_j / |C_\ell|. \tag{6.59}$$

For each cluster C_ℓ of Eq. (6.59), we further rewrite the objective function of k-means as

$$\sum_{i \in C_\ell} \|\mathbf{x}_i - \mathbf{m}_\ell\|^2, \quad \mathbf{m}_\ell := \sum_{j \in C_\ell} \mathbf{x}_j / |C_\ell| \tag{6.60}$$

$$= \sum_{i \in C_\ell} \left(\|\mathbf{x}_i\|^2 - 2\langle \mathbf{x}_i, \mathbf{m}_\ell \rangle + \|\mathbf{m}_\ell\|^2 \right) \tag{6.61}$$

$$= \sum_{i \in C_\ell} \|\mathbf{x}_i\|^2 - |C_\ell| \, \|\mathbf{m}_\ell\|^2 \tag{6.62}$$

$$= \frac{1}{2|C_\ell|} \left(\sum_{i,j \in C_\ell} 2\|\mathbf{x}_i\|^2 - 2|C_\ell|^2 \, \|\mathbf{m}_\ell\|^2 \right) \tag{6.63}$$

$$= \frac{1}{2|C_\ell|} \left(\sum_{i,j \in C_\ell} (\|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2) - \sum_{i,j \in C_\ell} \langle \mathbf{x}_i, \mathbf{x}_j \rangle \right) \tag{6.64}$$

$$= \frac{1}{2} \sum_{i,j \in C_\ell} \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{|C_\ell|} \tag{6.65}$$

Summing up over the all cluster, we can rewrite Eq. (6.59) as

$$\mathcal{J}(\{C_\ell\}_{\ell=1}^k) = \frac{1}{2}\sum_{\ell\in[k]}\sum_{i,j\in C_\ell}\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{|C_\ell|}. \tag{6.66}$$

By replacing $\mathbf{x}_i$ and $\mathbf{x}_j$ to $\mathbf{v}'_i$ and $\mathbf{v}'_j$ and $\|\cdot\|$ to $\|\cdot\|_2$, we conclude the proof.

6.G Proof of Theorem 6.6

We now rewrite Eq. (6.11) as

$$\begin{aligned}
&J(\{V_\ell\}_{\ell=1}^k) = \sum_{\ell\in[k]}\sum_{i\in V_\ell}(\|\mathbf{v}'_i\|_2^2 - 2\langle\mathbf{v}'_i,\mathbf{m}_j\rangle_2 + \|\mathbf{m}_j\|_2^2)\\
&= \sum_{\ell\in[k]}\sum_{\mathbf{v}'_i\in V_\ell}\left(\langle\mathbf{v}'_i,\mathbf{v}'_i\rangle_2 - 2\left\langle\mathbf{v}'_i,\sum_{\mathbf{v}'_j\in V_\ell}\frac{1}{|V_\ell|}\mathbf{v}'_j\right\rangle_2 + \left\langle\sum_{\mathbf{v}'_j\in V_\ell}\frac{1}{|V_\ell|}\mathbf{v}'_j,\sum_{\mathbf{v}'_r\in V_\ell}\frac{1}{|V_\ell|}\mathbf{v}'_r\right\rangle_2\right)\\
&= \sum_{k\in[\ell]}\sum_{i\in V_\ell}\Big((L_b^{-1})_{ii} - 2\sum_{l\in V_j}\frac{1}{|V_j|}(L_b^{-1})_{il} + \sum_{r,t\in V_\ell}\frac{1}{|V_\ell|^2}(L_b^{-1})_{rt}\Big) \qquad (6.67)\\
&= \sum_{\ell\in[k]}\sum_{i\in V_\ell}(L_b^{-1})_{ii} - \sum_{\ell\in[k]}\sum_{r,t\in V_\ell}\frac{1}{|V_\ell|}(L_b^{-1})_{rt} \qquad (6.68)\\
&= \mathrm{trace}L_b^{-1} - \mathrm{trace}Z_R L_b^{-1} Z_R, \qquad (6.69)
\end{aligned}$$

where Z_R is an $n\times k$ matrix which serves as an indicator matrix, defined in Sec. 2.2. Thus, if we minimize Eq. (6.69) with respect to Z_R, we maximize the second term. Assuming Z_R is discrete, $Z_R^\top Z_R = I$. If we relax Z_R with this constraint, $\mathrm{trace}Z_R L_b^{-1} Z_R$ becomes a problem to obtain top k eigenvectors. From Prop. 6.1 and Cor. 6.9, the top k eigenvectors of L_b^{-1} are equivalent to the smallest k eigenvectors of L. Similarly to Sec. 2.2 case, using Cor. 6.1, optimal solutions of k-means on $\mathcal{H}_{L_b}$ and spectral clustering is given as the same set of vectors, which is the k smallest eigenvectors of L. This completes the proof.

Chapter 7

Conclusions and Future Directions

This chapter provides conclusions and future directions of this thesis.

7.1 Conclusions

As we see in Chapter 1, despite recent advancements, there remain untapped opportunities for further generalizing the graph spectral clustering framework. To address these gaps, this thesis has contributed to the development of hypergraph spectral clustering and graph-based learning algorithms.

In Chapter 3, we have considered hypergraph spectral clustering via hypergraph p-Laplacians. In the past many different hypergraph Laplacians were proposed, since generalizations can take different form. However, while these prior Laplacians have similar properties, they derive a patchwork of key features, such as nodal domain theorems, Cheeger inequalities, and partitioning algorithms for some particular cases of hypergraph p-Laplacians. To address this, we have proposed an abstract class of hypergraph p-Laplacians. We also provided theoretical results for our p-Laplacian and a hypergraph partitioning algorithm based on our abstract class of hypergraph p-Laplacians. Our experiments demonstrated that this algorithm outperforms existing hypergraph spectral clustering methods.

In Chapter 4, we have considered a hypergraph modeling from vector data. In the standard graph, vector data is commonly modeled by constructing vertices from data points and edges based on pairwise similarities, with theoretical justifications connecting this to the normalized cut. However, no comparable framework has existed for hypergraph cut problems. To address this, we have proposed a novel hypergraph modeling method with theoretical foundations. Furthermore, we have developed a spectral clustering algorithm

connected to hypergraph cut problems. Our experiments have showed that this method improves performance over standard graph-based modeling approaches.

In Chapter 5, we have focused on multi-class clustering exploiting the graph p-seminorm. While spectral clustering via p-Laplacian is effective to the bisectioning problem, it is known to be limited to apply to the multi-class settings, due to the long open problems. Thus, We have taken a different approach; we proposed the multi-class clustering algorithm using the p-resistance. However, p-resistance is expensive to compute. Thus, for this purpose, we have shown a guarantee for the approximation of p-resistance. This approximation has led to compute an approximation of p-resistance much faster than the naive optimization methods. We empirically confirmed that our algorithm has outperformed the existing clustering methods using the graph p-seminorm.

In Chapter 6, we have considered vertex classification in graph-with-features settings, where both the graph structure and node features are available. While graph neural networks (GNNs) are commonly used for this task, they tend to exhibit bias towards homophilous information. To mitigate this bias, we have proposed ResTran, a simple alternative to GNNs. ResTran transforms feature vectors by incorporating graph topology and then applies standard learning methods to these transformed vectors. We have provided theoretical justifications for ResTran from the perspectives of resistance, k-means clustering, and spectral clustering. Our experiments have demonstrated that ResTran is more robust against certain biases compared to existing GNN approaches.

7.2 Why Generalizations Mattered: A View from Mystical Power of Twoness

This thesis focused on generalizations of spectral clustering. As seen in Sec. 1.3, standard spectral clustering can be understood as solving a 2-seminorm problem on 2-uniform hypergraphs. We extended this to the p-seminorm and generalized the framework from graphs to hypergraphs.

In Sec. 1.3, we argue that generalizing the graph Laplacian provides a better understanding of the standard graph Laplacian. Using examples there, we illustrate that we identify what is essential in the graph Laplacian, specifically which properties hold in both the $p = 2$ case and the general p cases. Additionally, from generalizations we observe that properties

exclusive to $p = 2$ depend on "twoness," a feature unique to the $p = 2$ scenario that does not extend to other p values. These unique properties make problems easier to solve when $p = 2$. Specifically in the example in Sec. 1.3, the variational theorem holds for both cases, but orthogonality does not apply to the general p case, which makes the computation difficult for the general p case.

In the following, we revisit key observations through the lens of this "mystical power of twoness." By examining these observations, we also highlight the limitations of this thesis. Many of the computational conveniences are provided by the unique properties of "twoness." Thus, what does not generalize well to the p-case presents limitations.

Nodal Domain Theorem and Cheeger Inequality. In Chapter 3, we generalized the graph Laplacian to the hypergraph p-Laplacian in an abstract way. Despite this broad generalization, we preserved differential geometric structures, particularly the nodal domain theorem and the Cheeger inequality. These properties hold for both the graph Laplacian and the hypergraph p-Laplacian. However, the hypergraph p-Laplacian shares the same limitation as the graph p-Laplacian regarding higher eigenvalues. Neither can capture the third or higher eigenvalues, meaning that obtaining these values still depends on the "twoness" structure, as explained in Sec. 1.3.

Kernel Property. In Chapter 4, we generalized the kernel from graphs to r-uniform hypergraphs where r is even, focusing on the 2-seminorm problem. For r-uniform hypergraphs, the semi-definiteness of the kernel is preserved, which is also an important property for the standard graph. However, this property does not extend well to odd-uniform or general hypergraphs. Therefore, to accommodate such models, we may need to explore how to generalize semi-definiteness to broader settings.

Laplacian Coordinate. In Chapter 5, we considered generalization from resistance to p-resistance in graph settings. We showed that the Laplacian coordinates $\mathcal{V}(L) = \{v_i := L^{+}\mathbf{e}_i : i = 1, \ldots, n\}$ plays a critical role. Recall that the standard resistance can be written as

$$r_{G,2}(i,j) = \|\mathbf{v}_i - \mathbf{v}_j\|_{G,2}, \quad \text{where } \mathbf{v}_i, \mathbf{v}_j \in \mathcal{V}(L). \tag{7.1}$$

We approximated the p-resistance for general p as

$$r_{G,p}(i,j) \approx \|\mathbf{v}_i - \mathbf{v}_j\|_{G,p}, \quad \text{where } \mathbf{v}_i, \mathbf{v}_j \in \mathcal{V}(L). \tag{7.2}$$

This approximation becomes exact for tree graphs. Thus, the Laplacian coordinate is useful for both $p = 2$ and general p cases. When $p = 2$, the m-dimensional graph seminorm $\|\mathbf{x}\|_{G,2}$ can be rewritten as the n-dimensional seminorm $\|\mathbf{x}\|_L$. Reducing from m to n makes the computation faster for the $p = 2$ case. However, this simplification does not apply easily for general p-seminorms. The efficiency we gain in the $p = 2$ case, where we compute in $O(n)$ instead of $O(m)$, is another result of the "twoness" property as discussed in Appendix 5.I. Finally, we mention that we have not had effective resistance for hypergraphs because we do not have an immediate circuit analogy for hypergraphs. The circuit analogy is also due to "twoness."

7.3 Future Directions

This section outlines promising future directions based on this thesis.

More Hypergraph Modeling. In Chapter 4, we proposed a model for even-order uniform hypergraphs from vector data, which connects to certain hypergraph cut problems. Chapter 3 introduced an abstract class of hypergraph Laplacians. A key question arises: can we develop a hypergraph model aligned with the hypergraph Laplacian proposed in Chapter 3? Furthermore, can we establish theoretical foundations similar to those in Chapter 4 for this new modeling approach?

Hypergraph Multi-class Clustering via p-Seminorm. In the experiments from Chapter 4 and Chapter 5, we observed that the Iris dataset performed better in Chapter 4, despite the more sophistication in Chapter 5. This suggests that for some datasets, generalizing from graph to hypergraph is more effective than extending from the 2-seminorm to the p-seminorm. An interesting question is: under what conditions do datasets benefit from graph generalization, and when do they benefit from seminorm generalization? While this is an interesting question, it would be more convenient if we can combine both; a hypergraph multi-class clustering via p-seminorm. However, as noted in Chapter 3, the hypergraph p-Laplacian shares the same limitations as the graph p-Laplacian in multi-class clustering. Since there is no hypergraph counterpart for effective resistance, it would be fruitful to explore how the insights from Chapter 5 could be applied to hypergraphs.

p-Laplacian vs. p-Resistance. Another valuable direction is to further explore the relationship between the p-Laplacian and p-resistance. The first question is: how are these concepts connected? In Chapter 6, we showed that spectral clustering is connected to k-

means using resistance. Can we establish a similar connection between the p-Laplacian and p-resistance? Another question is whether the parameter p behaves the same way for both the graph p-Laplacian and p-resistance. In the experiments in Chapter 3, smaller p generally performed better, while in Chapter 5, larger p was more effective. This aligns with theoretical results: for the p-Laplacian, Cheeger's inequality gets progressively looser as p increases, while for p-resistance, smaller p loses global graph information for certain graph classes [Alamgir and Luxburg, 2011]. Investigating the causes of these differences would be a valuable next step.

More Scalability with Theoretical Guarantees. Much of the discussion in this thesis focuses on generalizing spectral clustering, which typically has a complexity of $O(n^3)$. Most of the generalized methods proposed in this thesis tend to be slower, with most having a complexity of *at least* $O(n^3)$, making practical computation for large graphs challenging. As a result, the datasets used for the experiments in this thesis are relatively small compared to current standards. From the observations of "twoness" (discussed in Sec. 1.3 and Sec. 7.2), this "twoness" can make algorithms faster, although general approaches may not benefit from such advantages. Given these fundamental challenges, an important future direction is how to achieve faster approximations with theoretical guarantees. For instance, can techniques like sparsification [Spielman and Teng, 2014], which approximates the Laplacian with theoretical guarantees, be extended to the generalized case?

More Applications of Laplacian Coordinates. As discussed in Sec. 7.2, the Laplacian coordinate is fundamental for both $p = 2$ and general p cases. In Chapter 6, we applied this insight to improve graph-with-features representation. Could Laplacian coordinates be useful in other areas? Possible applications include overlapping community detection [Xie et al., 2013] and temporal networks [Holme and Saramäki, 2012].

Bibliography

E. Abbe. Community detection and stochastic block models: recent developments. *J. Mach. Learn. Res.*, 18(177):1–86, 2018.

E. Abbe, A. S. Bandeira, and G. Hall. Exact recovery in the stochastic block model. *IEEE Trans. Inf. Theory*, 62(1):471–487, 2015.

S. Agarwal, K. Branson, and S. Belongie. Higher order learning with graphs. In *Proc. ICML*, pages 17–24, 2006.

C. C. Aggarwal and H. Wang. A survey of clustering algorithms for graph data. *Managing and mining graph data*, pages 275–301, 2010.

M. Alamgir and U. Luxburg. Phase transition in the family of p-resistances. *Proc. NIPS*, 24:379–387, 2011.

V. L. Alev, N. Anari, L. C. Lau, and S. O. Gharan. Graph clustering using effective resistance. *arXiv preprint arXiv:1711.06530*, 2017.

D. Alfke and M. Stoll. Pseudoinverse graph convolutional networks: Fast filters tailored for large eigengaps of dense graphs and hypergraphs. *Data Min. Knowl. Discov.*, 35:1318–1341, 2021.

N. Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986.

N. Alon and V. D. Milman. $\lambda 1$, isoperimetric inequalities for graphs, and superconcentrators. *J. Comb. Theory B*, 38(1):73–88, 1985.

S.-I. Amari. Natural gradient works efficiently in learning. *Neural Comput.*, 10(2):251–276, Feb. 1998. ISSN 0899-7667.

S. Amghibech. Eigenvalues of the discrete p-laplacian for graphs. *Ars Comb.*, 2003.

N. Aronszajn. Theory of reproducing kernels. *Trans. Am. Math. Soc*, 68(3):337–404, 1950.

G. Astarita and G. Marrucci. *Principles of non-Newtonian fluid mechanics*. McGraw Hill, 1974.

M. Azabou, V. Ganesh, S. Thakoor, C.-H. Lin, L. Sathidevi, R. Liu, M. Valko, P. Veličković, and E. L. Dyer. Half-hop: A graph upsampling approach for slowing down message passing. In *Proc. ICML*, pages 1341–1360, 2023.

A. Azimi and R. B. Bapat. Moore–penrose inverse of the incidence matrix of a distance regular graph. *Linear Algebra Appl.*, 551:92–103, 2018.

A. Azimi, R. B. Bapat, and E. Estaji. Moore–penrose inverse of incidence matrix of graphs with complete and cyclic blocks. *Discrete Math.*, 342(1):10–17, 2019.

F. Bach and M. Jordan. Learning spectral clustering. In *Proc. NIPS*, 2003.

N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Mach. Learn.*, 56:89–113, 2004.

R. B. Bapat. Moore-penrose inverse of the incidence matrix of a tree. *Linear Multilinear Algebra*, 42 (2):159–167, 1997.

R. B. Bapat. *Graphs and matrices*, volume 27. Springer, 2010.

S. T. Barnard and H. D. Simon. Fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems. *Concurrency: Practice and experience*, 6(2):101–117, 1994.

E. R. Barnes. An algorithm for partitioning the nodes of a graph. *SIAM J. Discrete Math.*, 3(4): 541–550, 1982.

E. R. Barnes and A. J. Hoffman. Partitioning, spectra and linear programming. In *Progress in Combinatorial Optimization*, pages 13–25. Elsevier, 1984.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, 2003.

M. Belkin and P. Niyogi. Towards a theoretical foundation for laplacian-based manifold methods. In *Proc. COLT*, pages 486–500. Springer, 2005.

M. Belkin and P. Niyogi. Convergence of laplacian eigenmaps. *Advances in Neural Information Processing Systems*, 19:129, 2007.

M. Belkin and P. Niyogi. Towards a theoretical foundation for laplacian-based manifold methods. *Journal of Computer and System Sciences*, 74(8):1289–1308, 2008.

M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res*, 7(Nov):2399–2434, 2006.

A. Ben-Israel and T. N. Greville. *Generalized inverses: theory and applications*, volume 15. Springer, 2003.

Y. Bengio, O. Delalleau, N. L. Roux, J.-F. Paiement, P. Vincent, and M. Ouimet. Learning eigenfunctions links spectral embedding and kernel pca. *Neural Comput.*, 16(10):2197–2219, 2004.

A. R. Benson, J. Kleinberg, and N. Veldt. Augmented sparsifiers for generalized hypergraph cuts. *arXiv preprint arXiv:2007.08075*, 2020.

C. Berge. *Hypergraphs: combinatorics of finite sets*, volume 45. Elsevier, 1984.

D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Proc. NeurIPS*, 2019.

P. A. Binding and B. P. Rynne. Variational and non-variational eigenvalues of the p-Laplacian. *Differ. Equ.*, 244(1):24–39, 2008.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*. Springer, 2006.

M. Black, Z. Wan, A. Nayyeri, and Y. Wang. Understanding oversquashing in GNNs through the lens of effective resistance. In *Proc. ICML*, pages 2528–2547, 2023.

V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *J. Stat. Mech.: Theory Exp*, 2008(10):P10008, 2008.

A. Blum, J. Lafferty, M. R. Rwebangira, and R. Reddy. Semi-supervised learning using randomized mincuts. In *Proc. ICML*, 2004.

M. Bolla. Spectra, euclidean representations and clusterings of hypergraphs. *Discrete Math.*, 117(1-3): 19–39, 1993.

F. Bonchi, C. Gentile, A. Panisson, and F. Vitale. Fast and effective gnn training with linearized random spanning trees. *arXiv preprint arXiv:2306.04828*, 2023.

S. Bougleux, A. Elmoataz, and M. Melkemi. Discrete regularization on weighted graphs for image and mesh filtering. In *Proc. SSVM*, pages 128–139, 2007.

S. Bougleux, A. Elmoataz, and M. Melkemi. Local and nonlocal discrete regularization on weighted graphs for image and mesh processing. *Int. J. Comput. Vision*, 84(2):220–236, 2009.

N. Bridle and X. Zhu. p-voltages: Laplacian regularization for semi-supervised learning on high-dimensional data. In *Proc. MLG*, 2013.

J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. In *Proc. ICLR*, 2014.

T. Bühler and M. Hein. Spectral clustering based on the graph p-Laplacian. In *Proc. ICML*, pages 81–88, 2009.

J. Calder. The game theoretic p-Laplacian and semi-supervised learning with few labels. *Nonlinearity*, 32(1):301, 2018.

J. Calder, B. Cook, M. Thorpe, and D. Slepcev. Poisson learning: Graph based semi-supervised learning at very low label rates. In *Proc. ICML*, pages 1306–1316, 2020.

P. K. Chan, M. D. Schlag, and J. Y. Zien. Spectral k-way ratio-cut partitioning and clustering. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 13(9):1088–1096, 1994.

T.-H. H. Chan, A. Louis, Z. G. Tang, and C. Zhang. Spectral properties of hypergraph laplacian and approximation algorithms. *J. ACM*, 65(3):1–48, 2018.

A. K. Chandra, P. Raghavan, W. L. Ruzzo, R. Smolensky, and P. Tiwari. The electrical resistance of a graph captures its commute and cover times. *Comput. Complex.*, 6:312–340, 1996.

H. Chang, Y. Rong, T. Xu, Y. Bian, S. Zhou, X. Wang, J. Huang, and W. Zhu. Not all low-pass filters are robust in graph convolutional networks. In *Proc. NuerIPS*, pages 25058–25071, 2021.

J. Chang, Y. Chen, L. Qi, and H. Yan. Hypergraph clustering using a new laplacian tensor with applications in image processing. *SIAM J. Imaging Sci.*, 13(3):1157–1178, 2020.

K. C. Chang. Spectrum of the 1-Laplacian and cheeger's constant on graphs. *J. Graph Theory*, 81(2): 167–207, 2016.

M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li. Simple and deep graph convolutional networks. In *Proc. ICML*, pages 1725–1735, 2020.

Y. Chen, L. Qi, and X. Zhang. The fiedler vector of a laplacian tensor for hypergraph partitioning. *SIAM J. Sci. Comput.*, 39(6):A2508–A2537, 2017.

P. S. Chodrow, N. Veldt, and A. R. Benson. Generative hypergraph clustering: From blockmodels to modularity. *Sci. Adv.*, 7(28):eabh1303, 2021.

F. Chung. Four proofs for the cheeger inequality and graph partition algorithms. In *Proc. of ICCM*, volume 2, pages 751–772, 2007.

F. R. Chung. Laplacians of graphs and cheeger's inequalities. *Combinatorics, Paul Erdös is eighty*, 2 (157-172):13–2, 1996.

A. Condon and R. M. Karp. Algorithms for graph partitioning on the planted partition model. *Random Struct. Algorithms*, 18(2):116–140, 2001.

J. Cooper and A. Dutle. Spectra of uniform hypergraphs. *Linear Algebra Appl.*, 436(9):3268–3292, 2012.

C. Cortes and V. Vapnik. Support-vector networks. *Mach. learn.*, 20:273–297, 1995.

R. Courant and D. Hilbert. *Methods of Mathematical Physics*. Methods of Mathematical Physics. Interscience Publishers, 1962.

M. Craven, A. McCallum, D. PiPasquo, T. Mitchell, and D. Freitag. Learning to extract symbolic knowledge from the world wide web. In *Proc. AAAI*, 1998.

L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4):1253–1278, 2000.

M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proc. NIPS*, volume 29, 2016.

P. Deidda, M. Putti, and F. Tudisco. Nodal domain count for the generalized graph p-laplacian. *arXiv preprint arXiv:2201.01248*, 2022.

I. S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means: spectral clustering and normalized cuts. In *Proc. KDD*, pages 551–556, 2004.

I. S. Dhillon, Y. Guan, and B. Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(11):1944–1957, 2007.

F. Di Giovanni, L. Giusti, F. Barbero, G. Luise, P. Lio, and M. Bronstein. On over-squashing in message passing neural networks: The impact of width, depth, and topology. In *Proc. ICML*, pages 7865–7885, 2023.

C. Ding and X. He. K-means clustering via principal component analysis. In *Proc. ICML*, page 29, 2004.

C. Ding, X. He, and H. D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proc. SDM*, pages 606–610. SIAM, 2005.

S. Ding, L. Cong, Q. Hu, H. Jia, and Z. Shi. A multiway p-spectral clustering algorithm. *Knowl. Based Syst.*, 164:371–377, 2019.

W. E. Donath and A. J. Hoffman. Algorithms for partitioning of graphs and computer logic based on eigenvectors of connection matrices. *IBM Tech. Dis. Bull.*, 15(3):938–944, 1972.

W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM J. Res. Dev.*, 17 (5):420–425, 1973.

P. G. Doyle and J. L. Snell. *Random walks and electric networks*, volume 22. American Mathematical Society, 1984.

L. Du, X. Shi, Q. Fu, X. Ma, H. Liu, S. Han, and D. Zhang. GBK-GNN: Gated bi-kernel graph neural networks for modeling both homophily and heterophily. In *Proc. WWW*, pages 1550–1558, 2022.

D. Dua and C. Graff. UCI machine learning repository, 2019. URL `http://archive.ics.uci.edu/ml`.

V. P. Dwivedi, C. K. Joshi, A. T. Luu, T. Laurent, Y. Bengio, and X. Bresson. Benchmarking graph neural networks. *J. Mach. Learn. Res.*, 24(43):1–48, 2023.

A. El Alaoui, X. Cheng, A. Ramdas, M. J. Wainwright, and M. I. Jordan. Asymptotic behavior of ℓ_p-based Laplacian regularization in semi-supervised learning. In *Proc. COLT*, pages 879–906, 2016.

A. Elmoataz, O. Lezoray, and S. Bougleux. Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing. *IEEE Trans. Image Process.*, 17(7):1047–1060, 2008.

M. Fiedler. Algebraic connectivity of graphs. *Czechoslov. Math. J.*, 23(2):298–305, 1973.

M. Fiedler. Eigenvectors of acyclic matrices. *Czechoslov. Math. J.*, 25(4):607–618, 1975a.

M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslov. Math. J.*, 25(4):619–633, 1975b.

D. E. Fishkind, D. L. Sussman, M. Tang, J. T. Vogelstein, and C. E. Priebe. Consistent adjacency-spectral partitioning for the stochastic block model when the model parameters are unknown. *SIAM J. Matrix Anal. Appl.*, 34(1):23–39, 2013.

K. Fountoulakis, D. Wang, and S. Yang. p-norm flow diffusion for local graph clustering. In *Proc. ICML*, pages 3222–3232, 2020.

F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Trans. Knowl. Data Eng.*, 19(3):355–369, 2007.

J. Gasteiger, A. Bojchevski, and S. Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *Proc. ICLR*, 2020.

D. Ghoshdastidar and A. Dukkipati. Consistency of spectral partitioning of uniform hypergraphs under planted partition model. In *Proc. NIPS*, pages 397–405, 2014.

D. Ghoshdastidar and A. Dukkipati. A provable generalized tensor spectral method for uniform hypergraph partitioning. In *Proc. ICML*, pages 400–409, 2015.

D. Ghoshdastidar and A. Dukkipati. Consistency of spectral hypergraph partitioning under planted partition model. *Ann. Stat.*, 45(1):289–315, 2017a.

D. Ghoshdastidar and A. Dukkipati. Uniform hypergraph partitioning: Provable tensor methods and sampling techniques. *J. Mach. Learn. Res.*, 18(1):1638–1678, 2017b.

D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamical systems. *VLDB J.*, 8(3-4):222–236, 2000.

E. Giné and V. Koltchinskii. Empirical graph laplacian approximation of laplace-beltrami operators: large sample results. *Lecture Notes-Monograph Series*, 51:238–259, 2006.

M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proc. Natl. Acad. Sci.*, 99(12):7821–7826, 2002.

O. Goldschmidt and D. S. Hochbaum. A polynomial algorithm for the k-cut problem for fixed k. *Math. Oper. Res.*, 19(1):24–37, 1994.

G. H. Golub and C. F. Van Loan. *Matrix computations*. JHU press, 2013.

T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.*, 38: 293–306, 1985.

M. Gori, G. Monfardini, and F. Scarselli. A new model for learning in graph domains. In *Proc. IJCNN*, pages 729–734, 2005.

V. M. Govindu. A tensor decomposition for geometric grouping and segmentation. In *Proc CVPR*, volume 1, pages 1150–1157, 2005.

P. Goyal and E. Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowl. Based Syst.*, 151:78–94, 2018.

L. Grady. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(11): 1768–1783, 2006.

L. Hagen and A. B. Kahng. New spectral methods for ratio cut partitioning and clustering. *EEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 11(9):1074–1085, 1992.

W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Proc. NIPS*, 2017.

M. Hein. Uniform convergence of adaptive graph-based regularization. In *Proc. COLT*, pages 50–64. Springer, 2006.

M. Hein, J.-Y. Audibert, and U. von Luxburg. Graph laplacians and their convergence on random neighborhood graphs. *J. Mach. Learn. Res.*, 8(6), 2007.

M. Hein, S. Setzer, L. Jost, and S. S. Rangapuram. The total variation on hypergraphs - learning on hypergraphs revisited. In *Proc. NIPS*, pages 2427–2435, 2013.

M. Henaff, J. Bruna, and Y. LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

C. Hennig and B. Hausdorf. Design of dissimilarity measures: A new dissimilarity between species distribution areas. In *Data science and classification*, pages 29–37. Springer, 2006.

M. Herbster. A triangle inequality for p-resistance. 2010.

M. Herbster and G. Lever. Predicting the labelling of a graph via minimum p-seminorm interpolation. In *Proc. COLT*, 2009.

M. Herbster and M. Pontil. Prediction on a graph with a perceptron. In *Proc. NIPS*, pages 577–584, 2006.

M. Herbster, M. Pontil, and L. Wainer. Online learning over graphs. In *Proc. ICML*, pages 305–312, 2005.

N. J. Higham. Estimating the matrix p-norm. *Numer. Math.*, 62(1):539–555, 1992.

N. J. Higham. *Functions of matrices: theory and computation*. SIAM, 2008.

C. J. Hillar and L.-H. Lim. Most tensor problems are np-hard. *J. ACM*, 60(6):1–39, 2013.

N. Hoang and T. Maehara. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*, 2019.

N. Hoang, T. Maehara, and T. Murata. Stacked graph filter. *arXiv preprint arXiv:2011.10988*, 2020.

P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Soc. Netw.*, 5(2): 109–137, 1983.

P. Holme and J. Saramäki. Temporal networks. *Phys. Rep.*, 519(3):97–125, 2012.

R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge university press, 2012.

S. Hu and L. Qi. Algebraic connectivity of an even uniform hypergraph. *J. Comb. Optim.*, 24(4): 564–579, 2012.

S. Hu and L. Qi. The Laplacian of a uniform hypergraph. *J. Comb. Optim.*, 29(2):331–366, 2015.

Y. Huang, Q. Liu, and D. Metaxas. Video object segmentation by hypergraph cut. In *Proc. CVPR*, pages 1738–1745, 2009.

M. Ikeda, A. Miyauchi, Y. Takai, and Y. Yoshida. Finding cheeger cuts in hypergraphs via heat equation. *arXiv preprint arXiv:1809.04396*, 2018.

T. Joachims. Transductive inference for text classification using support vector machines. In *Proc. ICML*, pages 200–209, 1999.

T. Joachims. Transductive learning via spectral graph partitioning. In *Proc. ICML*, pages 290–297, 2003.

A. Joseph and B. Yu. Impact of regularization on spectral clustering. *Ann. Stats.*, 44(4):1765–1791, 2016.

L. Kalman and R. Krauthgamer. Flow metrics on graphs. *arXiv preprint arXiv:2112.06916*, 2021.

R. M. Karp. *Reducibility among combinatorial problems*. Springer, 2010.

L. Kaufman and P. J. Rousseeuw. Partitioning around medoids (program pam). *Finding groups in data: an introduction to cluster analysis*, 344:68–125, 1990.

D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Proc. NIPS*, 2014.

T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *Proc. ICLR*, 2016a.

T. N. Kipf and M. Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016b.

G. Kirchhoff. Ueber die auflösung der gleichungen, auf welche man bei der untersuchung der linearen vertheilung galvanischer ströme geführt wird. *Ann. Phys. (Berl.)*, 148(12):497–508, 1847.

S. Klamt, U.-U. Haus, and F. Theis. Hypergraphs and cellular networks. *PLoS Comput. Biol.*, 5(5): e1000385+, 2009.

D. J. Klein and M. Randić. Resistance distance. *J. Math. Chem.*, 12(1):81–95, 1993.

A. Kurakin, C. Raffel, D. Berthelot, E. D. Cubuk, H. Zhang, K. Sohn, and N. Carlini. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *Proc. ICLR*, 2020.

S. S. Lafon. *Diffusion maps and geometric harmonics*. Yale University, 2004.

S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. In *Proc. ICLR*, 2017.

A. Lakhina, M. Crovella, and C. Diot. Diagnosing network-wide traffic anomalies. In *Proc. SIGCOMM*, pages 219–230. ACM New York, NY, USA, 2004.

C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators1. *J. Res. Natl. Inst. Stand. Technol.*, 45(4), 1950.

A. Lê. Eigenvalue problems for the p-laplacian. *Nonlinear Anal. Theory Methods Appl.*, 64(5): 1057–1099, 2006.

D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Proc. NIPS*, volume 13, 2000.

D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

J. R. Lee, S. O. Gharan, and L. Trevisan. Multiway spectral partitioning and higher-order cheeger inequalities. *J. ACM*, 61(6):37, 2014.

J. Lei and A. Rinaldo. Consistency of spectral clustering in stochastic block models. *Ann. Stat.*, pages 215–237, 2015.

J. K. Lenstra. The mystical power of twoness: in memoriam eugene l. lawler. *J. Sched.*, 1(1):3–14, 1998.

P. Li and O. Milenkovic. Inhomogoenous hypergraph clustering with applications. In *Proc. NIPS.*, pages 2305–2315, 2017.

P. Li and O. Milenkovic. Submodular hypergraphs: p-Laplacians, cheeger inequalities and spectral clustering. In *Proc. ICML*, pages 3020–3029, 2018.

Q. Li, Z. Han, and X.-M. Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proc. AAAI*, pages 3538–3545, 2018.

W.-C. W. Li and P. Solé. Spectra of regular graphs and hypergraphs and orthogonal polynomials. *Europ. J. Combinatorics*, 17(5):461 – 477, 1996. ISSN 0195-6698.

D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proc. CIKM*, pages 556–559, 2003.

L.-H. Lim. Singular values and eigenvalues of tensors: a variational approach. In *Proc. CAMSAP*, pages 129–132, 2005.

P. Lindqvist. A nonlinear eigenvalue problem. In *Topics in Mathematical Analysis*, pages 175–203. 2008.

M. Liu and D. F. Gleich. Strongly local p-norm-cut algorithms for semi-supervised learning and local graph clustering. In *Proc. NeurIPS*, pages 5023–5035, 2020.

M. Liu, N. Veldt, H. Song, P. Li, and D. F. Gleich. Strongly local hypergraph diffusions for clustering and semi-supervised learning. In *Proc. TheWebConf*, pages 2092–2103, 2021.

A. Louis. Hypergraph markov operators, eigenvalues and approximation algorithms. In *Proc. STOC*, pages 713–722, 2015.

S. Luan, M. Zhao, X.-W. Chang, and D. Precup. Break the ceiling: Stronger multi-scale deep graph convolutional networks. In *Proc. NeurIPS*, volume 32, 2019.

S. Luan, C. Hua, Q. Lu, J. Zhu, M. Zhao, S. Zhang, X.-W. Chang, and D. Precup. Is heterophily a real nightmare for graph neural networks to do node classification? *arXiv preprint arXiv:2109.05641*, 2021.

S. Luan, C. Hua, Q. Lu, J. Zhu, M. Zhao, S. Zhang, X.-W. Chang, and D. Precup. Revisiting heterophily for graph neural networks. In *Proc. NeurIPS*, pages 1362–1375, 2022.

D. Luo, H. Huang, C. Ding, and F. Nie. On the eigenvectors of p-Laplacian. *Mach. Learn.*, 81(1): 37–51, 2010.

L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther. Auxiliary deep generative models. In *Proc. ICML*, pages 1445–1453, 2016.

M. Mahajan, P. Nimbhorkar, and K. Varadarajan. The planar k-means problem is np-hard. *Theor. Comput. Sci.*, 442:13–21, 2012.

J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proc. SIGIR*, pages 43–52, 2015.

A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Inf. Retr.*, 3:127–163, 2000.

F. McSherry. Spectral partitioning of random graphs. In *Proc. FOCS*, pages 529–537. IEEE, 2001.

M. Meilă and J. Shi. A random walks view of spectral segmentation. In *Proc. AISTATS*, pages 203–208. PMLR, 2001.

C. A. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *J. Mach. Learn. Res.*, 7(12), 2006.

T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell*, 41(8): 1979–1993, 2018.

B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on web usage mining. *Commun. ACM*, 43(8):142–151, 2000.

K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

B. Nadler, N. Srebro, and X. Zhou. Semi-supervised learning with the graph Laplacian: The limit of infinite unlabelled data. In *Proc. NIPS*, pages 1330–1338, 2009.

G. Namata, B. London, L. Getoor, B. Huang, and U. Edu. Query-driven active surveying for collective classification. In *Proc. MLG*, 2012.

M. E. Newman. Modularity and community structure in networks. *Proc. Natl. Acad. Sci.*, 103(23): 8577–8582, 2006.

M. E. Newman. Spectral methods for community detection and graph partitioning. *Phys. Rev. E*, 88 (4):042822, 2013.

M. E. Newman. Equivalence between modularity optimization and maximum likelihood methods for community detection. *Phys. Rev. E*, 94(5):052315, 2016.

A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Proc. NIPS*, 14, 2001.

C. H. Nguyen and H. Mamitsuka. New resistance distances with global information on large graphs. In *Proc. AISTATS*, pages 639–647, 2016.

K. Oono and T. Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *Proc. ICLR*, 2019.

D. Pasadakis, C. L. Alappat, O. Schenk, and G. Wellein. Multiway p-spectral graph cuts on grassmann manifolds. *Mach. Learn.*, 111(2):791–829, 2022.

S. Pasteris, A. Rumi, M. Thiessen, S. Saito, A. Miyauchi, F. Vitale, and M. Herbster. Bandits with abstention under expert advice. *arXiv preprint arXiv:2402.14585*, 2024.

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philos. Mag*, 2(11): 559–572, 1901.

H. Pei, B. Wei, K. C.-C. Chang, Y. Lei, and B. Yang. Geom-gcn: Geometric graph convolutional networks. In *Proc. ICLR*, 2020.

E. F. Petricoin III et al. Use of proteomic patterns in serum to identify ovarian cancer. *Lancet*, 359 (9306):572–577, 2002.

O. Platonov, D. Kuznedelev, M. Diskin, A. Babenko, and L. Prokhorenkova. A critical look at the evaluation of GNNs under heterophily: are we really making progress? In *Proc. ICLR*, 2023.

A. Pothen, H. D. Simon, and K.-P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM journal on matrix analysis and applications*, 11(3):430–452, 1990.

L. Qi. Eigenvalues of a real supersymmetric tensor. *J. Symb. Comput.*, 40(6):1302–1324, 2005.

L. Qi. h^+-eigenvalues of Laplacian and signless Laplacian tensors. *arXiv preprint arXiv:1303.2186*, 2013.

T. Qin and K. Rohe. Regularized spectral clustering under the degree-corrected stochastic blockmodel. In *Proc. NIPS*, 2013.

M. Ranzato and M. Szummer. Semi-supervised learning of compact document representations with deep networks. In *Proc. ICML*, pages 792–799, 2008.

J. A. Rodriguez. On the Laplacian eigenvalues and metric parameters of hypergraphs. *Linear Multilinear Algebra*, 50(1):1–14, 2002.

K. Rohe, S. Chatterjee, and B. Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *Ann. Stat.*, 39(4):1878–1915, 2011.

S. Rosenberg and R. Steven. *The Laplacian on a Riemannian manifold: an introduction to analysis on manifolds*. Cambridge University Press, 1997.

B. Rozemberczki, C. Allen, and R. Sarkar. Multi-scale attributed node embedding. *J. Complex Netw.*, 9(2):cnab014, 2021.

Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

M. Saerens, F. Fouss, L. Yen, and P. Dupont. The principal components analysis of a graph, and its relationships to spectral clustering. In *Proc. ECML*, pages 371–383, 2004.

S. Saito. Hypergraph modeling via spectral embedding connection: Hypergraph cut, weighted kernel k-means, and heat kernel. In *Proc. AAAI*, pages 8141–8149, 2022.

S. Saito and M. Herbster. Generalizing p-Laplacian: spectral hypergraph theory and a partitioning algorithm. *Mach. Learn.*, 112(1):241–280, 2023a.

S. Saito and M. Herbster. Multi-class graph clustering via approximated effective p-resistance. In *Proc. ICML*, pages 29697–29733, 2023b.

S. Saito, Y. Hirata, K. Sasahara, and H. Suzuki. Tracking time evolution of collective attention clusters in twitter: time evolving nonnegative matrix factorisation. *Plos one*, 10(9):e0139085, 2015a.

S. Saito, R. Tomioka, and K. Yamanishi. Early detection of persistent topics in social networks. *Soc. Netw. Anal. Min.*, 5:1–15, 2015b.

S. Saito, D. P. Mandic, and H. Suzuki. Hypergraph p-Laplacian: A differential geometry view. In *Proc. AAAI*, pages 3984–3991, 2018.

S. Saito, T. Maehara, and M. Herbster. Restran: A gnn alternative to learn graph with features. In *MLGenX, Workshop of ICLR*, pages 8141–8149, 2024.

G. Salha, R. Hennequin, and M. Vazirgiannis. Keep it simple: Graph autoencoders without graph convolutional networks. *arXiv preprint arXiv:1910.00942*, 2019.

G. Salha, R. Hennequin, and M. Vazirgiannis. Simple and effective graph autoencoders with one-hop linear models. In *Proc. ECML PKDD*, pages 319–334, 2021.

P. Sarkar and P. J. Bickel. Role of normalization in spectral clustering for stochastic blockmodels. *Ann. Stat.*, 43(3):962–990, 2015.

F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Trans. Neural Netw.*, 20(1):61–80, 2008.

S. E. Schaeffer. Graph clustering. *Comput. Sci. Rev.*, 1(1):27–64, 2007.

G. L. Scott and H. C. Longuet-Higgins. Feature grouping by'relocalisation'of eigenvectors of the proximity matrix. In *Proc. BMVC*, pages 1–6, 1990.

P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad. Collective classification in network data. *AI Mag.*, 29(3):93–93, 2008.

J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell*, 22:888–905, 2000.

D. Singaraju, L. Grady, and R. Vidal. P-brush: Continuous valued mrfs with normed pairwise distributions for image segmentation. In *Proc. CVPR*, pages 1303–1310. IEEE, 2009.

D. Slepcev and M. Thorpe. Analysis of p-Laplacian regularization in semisupervised learning. *SIAM J. Math. Anal.*, 51(3):2085–2120, 2019.

K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Proc. NeurIPS*, pages 596–608, 2020.

D. A. Spielman and N. Srivastava. Graph sparsification by effective resistances. *SIAM J. Comput*, 40 (6):1913–1926, 2011.

D. A. Spielman and S.-H. Teng. Spectral partitioning works: Planar graphs and finite element meshes. In *Proc. FOCS*, pages 96–105, 1996.

D. A. Spielman and S.-H. Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM J. Matrix Anal. Appl.*, 35(3):835–885, 2014.

M. Stoer and F. Wagner. A simple min-cut algorithm. *J. ACM*, 44(4):585–591, 1997.

M. Struwe. *Variational Methods: Applications to Nonlinear Partial Differential Equations and Hamiltonian Systems, Third Edition*. Springer, 2000.

L. Su, W. Wang, and Y. Zhang. Strong consistency of spectral clustering for stochastic block models. *IEEE Trans. Inf. Theory*, 66(1):324–338, 2019.

Y. Sun, S. Wang, Q. Liu, R. Hang, and G. Liu. Hypergraph embedding for spatial-spectral joint feature extraction in hyperspectral images. *Remote Sens.*, 9(5):506, 2017.

J. Topping, F. Di Giovanni, B. P. Chamberlain, X. Dong, and M. M. Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *Proc. ICLR*, 2021.

N. G. Trillos and D. Slepčev. A variational approach to the consistency of spectral clustering. *Appl. Comput. Harmon. Anal.*, 45(2):239–281, 2018.

N. Trinajstic. *Chemical graph theory*. Routledge, 2018.

R. Tron and R. Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *Proc. CVPR*, pages 1–8, 2007.

F. Tudisco and M. Hein. A nodal domain theorem and a higher-order cheeger inequality for the graph p-Laplacian. *J. Spectr. Theory*, 8(3):883–908, 2018.

N. Veldt, C. Klymko, and D. F. Gleich. Flow-based local graph clustering with better seed set inclusion. In *Proc. SDM*, pages 378–386. SIAM, 2019.

N. Veldt, A. R. Benson, and J. Kleinberg. Hypergraph cuts with general splitting functions. *arXiv preprint arXiv:2001.02817*, 2020.

P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. In *Proc. ICLR*, 2018.

D. Verma and M. Meila. A comparison of spectral clustering algorithms. Technical Report UWCSE030501, University of Washington, 2003.

U. von Luxburg. A tutorial on spectral clustering. *Stat. Comput.*, 17(4):395–416, 2007.

U. von Luxburg, O. Bousquet, and M. Belkin. Limits of spectral clustering. In *Proc. NIPS*, volume 17, 2004.

U. von Luxburg, M. Belkin, and O. Bousquet. Consistency of spectral clustering. *Ann. Stat.*, pages 555–586, 2008.

U. von Luxburg, A. Radl, and M. Hein. Getting lost in space: Large sample analysis of the resistance distance. In *Proc. NIPS*, pages 2622–2630, 2010.

D. Wagner and F. Wagner. Between min cut and graph bisection. In *Proc. MFCS*, pages 744–750. Springer, 1993.

H. Wang, H. Yin, M. Zhang, and P. Li. Equivariant and stable positional encoding for more powerful graph neural networks. In *Proc. ICLR*, 2022.

Y.-X. Wang and Y.-J. Zhang. Nonnegative matrix factorization: A comprehensive review. *IEEE Trans. Knowl. Data Eng.*, 25(6):1336–1353, 2012.

Y. Weiss. Segmentation using eigenvectors: a unifying view. In *Proc. ICCV*, volume 2, pages 975–982. IEEE, 1999.

J. Weston, F. Ratle, and R. Collobert. Deep learning via semi-supervised embedding. In *Proc. ICML*, pages 1168–1175, 2008.

J. J. Whang, R. Du, S. Jung, G. Lee, B. Drake, Q. Liu, S. Kang, and H. Park. Mega: Multi-view semi-supervised clustering of hypergraphs. In *Proc. VLDB*, volume 13 (5), pages 698–711, 2020.

E. T. Whittaker. *A treatise on the analytical dynamics of particles and rigid bodies*. Cambridge University Press, 1964.

W. Williams, M. Dale, and P. Macnaughton-Smith. An objective method of weighting in similarity analysis. *Nature*, 201(4917):426–426, 1964.

S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemom. Intell. Lab. Syst.*, 2(1-3): 37–52, 1987.

F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger. Simplifying graph convolutional networks. In *Proc. ICML*, pages 6861–6871, 2019.

Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.*, 32(1):4–24, 2020.

J. Xie, S. Kelley, and B. K. Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Comput. Surv.*, 45(4):1–35, 2013.

J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *Proc. ICML*, pages 478–487, 2016.

K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In *Proc. ICLR*, 2019.

X. Yang, Z. Song, I. King, and Z. Xu. A survey on deep semi-supervised learning. *IEEE Trans. Knowl. Data Eng.*, 2022.

Z. Yang, W. Cohen, and R. Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *Proc. ICML*, pages 40–48, 2016.

L. Yen, D. Vanvyve, F. Wouters, F. Fouss, M. Verleysen, M. Saerens, et al. Clustering using a random walk based distance measure. In *Proc. ESANN*, pages 317–324, 2005.

L. Yen, M. Saerens, A. Mantrach, and M. Shimbo. A family of dissimilarity measures between nodes generalizing both the shortest-path and the commute-time distances. In *Proc. KDD*, pages 785–793, 2008.

Y. Yoshida. Cheeger inequalities for submodular transformations. In *Proc. SODA*, pages 2582–2601, 2019.

C.-A. Yu, C.-L. Tai, T.-S. Chan, and Y.-H. Yang. Modeling multi-way relations with hypergraph embedding. In *Proc. CIKM*, pages 1707–1710, 2018.

S. X. Yu and J. Shi. Multiclass spectral clustering. In *Proc. ICCV*, pages 11–17, 2003. ISBN 0-7695-1950-4.

H. Zha, X. He, C. Ding, M. Gu, and H. Simon. Spectral relaxation for k-means clustering. In *Proc. NIPS.*, 2001.

B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *Proc. NeurIPS*, pages 18408–18419, 2021.

D. Zhang. Homological eigenvalues of graph p-laplacians. *arXiv preprint arXiv:2110.06054*, 2021.

X. Zheng, Y. Liu, S. Pan, M. Zhang, D. Jin, and P. S. Yu. Graph neural networks for graphs with heterophily: A survey. *arXiv preprint arXiv:2202.07082*, 2022.

D. Zhou and B. Schölkopf. Regularization on discrete spaces. In *Pattern Recognition*, pages 361–368. Springer, 2005.

D. Zhou and B. Schölkopf. Discrete regularization. In *Semi-supervised Learning*. MIT Press, 2006.

D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Proc. NIPS*, pages 321–328, 2003.

D. Zhou, J. Huang, and B. Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Proc. NIPS*, pages 1601–1608, 2006.

H. Zhu and P. Koniusz. Simple spectral graph convolution. In *Proc. ICLR*, 2021.

X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. ICML*, pages 912–919, 2003.

J. Y. Zien, M. D. F. Schlag, and P. K. Chan. Multilevel spectral hypergraph partitioning with arbitrary vertex sizes. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 18(9):1389–1399, 1999.