


RESEARCH ARTICLE | MAY 19 2023

Intelligent performance inference: A graph neural network approach to modeling maximum achievable throughput in optical networks

Robin Matzner ; Ruijie Luo; Georgios Zervas; ... et. al

APL Machine Learning 1, 026112 (2023)

<https://doi.org/10.1063/5.0137426>View
OnlineExport
Citation

CrossMark

Articles You May Be Interested In

Deep learning for energetic material detonation performance

AIP Conference Proceedings (November 2020)

Cartesian message passing neural networks for directional properties: Fast and transferable atomic multipoles

J. Chem. Phys. (June 2021)

Solving ordinary differential equations using neural networks

AIP Conference Proceedings (June 2018)

Intelligent performance inference: A graph neural network approach to modeling maximum achievable throughput in optical networks

Cite as: APL Mach. Learn. 1, 026112 (2023); doi: 10.1063/5.0137426

Submitted: 2 December 2022 • Accepted: 3 April 2023 •

Published Online: 19 May 2023



View Online



Export Citation



CrossMark

Robin Matzner,^{a)}  Ruijie Luo,  Georgios Zervas  and Polina Bayvel 

AFFILIATIONS

Optical Networks Group, Department of Electronic and Electrical Engineering, University College London, Roberts Building, Torrington Place, London WC1E 7JE, United Kingdom

^{a)} Author to whom correspondence should be addressed: robin.matzner.19@ucl.ac.uk

ABSTRACT

One of the key performance metrics for optical networks is the maximum achievable throughput for a given network. Determining it, however, is a nondeterministic polynomial time (NP) hard optimization problem, often solved via computationally expensive integer linear programming (ILP) formulations. These are infeasible to implement as objectives, even on very small node scales of a few tens of nodes. Alternatively, heuristics are used although these, too, require considerable computation time for a large number of networks. There is, thus, a need for an ultra-fast and accurate performance evaluation of optical networks. For the first time, we propose the use of a geometric deep learning model, message passing neural networks (MPNNs), to learn the relationship between node and edge features, the network structure, and the maximum achievable network throughput. We demonstrate that MPNNs can accurately predict the maximum achievable throughput while reducing the computational time by up to five-orders of magnitude compared to the ILP for small networks (10–15 nodes) and compared to a heuristic for large networks (25–100 nodes)—proving their suitability for the design and optimization of optical networks on different time- and distance-scales.

© 2023 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/5.0137426>

I. INTRODUCTION

Multiwavelength optical networks underpin the global data communication network infrastructure and use the wavelength domain both for routing and to increase point-to-point data transmission rates. Optical networks are also being used for intra- and inter-data center communications and in high performance computing.¹ To enhance the throughput of these networks, optical fibers carry many wavelength channels on a single fiber, known as wavelength division multiplexing (WDM), which are also used for wavelength routing.² This has greatly improved the overall capacity of optical networks but requires the solution of the routing and wavelength assignment (RWA) problem. These have been shown to be nondeterministic polynomial time (NP) hard and, therefore, computationally difficult to solve optimally.^{2–4}

The overarching goal of the physical network design is to maximize network performance, measured by throughput, latency, and resilience, while minimizing the cost and/or resource use, making networks intelligent and adaptive.^{5,6} This is an evolution from the

previous goals of minimizing the number of wavelengths needed to optically route data within the network,⁷ which quantified the relationship between wavelength requirements and the physical topology.^{8,9} However, the understanding of linear and nonlinear physical layer impairments, in the presence of multiple wavelength channels, has shown that these play a key role in determining both routing and throughput, highlighting the importance of including optical fiber physical properties in network analysis and design.⁶

The maximum achievable network throughput is defined here as the throughput when the resources (wavelengths) are fully exploited, at zero blocking, for a given demand distribution (measured in bits per second). To quantify the maximum achievable throughput in a network requires an optimal solution to the RWA problem. Integer linear programming (ILP) formulations have been shown to solve the RWA problem optimally;^{10–12} however, these are infeasible for networks larger than ~30 nodes. Problem-agnostic optimization frameworks, which aim at efficiently exploring the solution space, i.e., meta-heuristics, have also been shown to give

good solutions; however, these have no guarantee of achieving a global optimum, while often still taking a long time to solve.^{13–16} Handcrafted algorithms, i.e., heuristics, are designed for specific purposes and are based on the automated rules-of-thumb. These are usually highly scalable but have shown limited success in terms of reaching optimum values, compared to ILP solutions.⁴ In addition, graph cut algorithms have been analyzed to estimate the maximum achievable throughput; however, these also succumb to computational complexity for larger graphs and have been shown to be non-exact for non-uniform traffic distributions.^{7,17} The efficient and accurate calculation of the maximum achievable throughput of optical networks, thus, remains a considerable challenge.

To make this task more computationally efficient, machine learning has been introduced to learn the relationship between the topology and performance parameters based on previously labeled datasets.

More specifically, supervised deep learning has been used, where large labeled datasets are used to train learnable functions. There are two broad approaches to deep learning: (i) artificial neural networks (ANNs) based on Euclidean data or (ii) geometric deep learning, using graph neural networks (GNNs), based on graph structured data to perform classification or regression tasks. In Ref. 18, an ANN was used to estimate the blocking probability of a network, and in Ref. 19, it was used to estimate the fast optical packet loss rate for bufferless optical packet-switched networks. The problem, however, with ANNs is that they operate on grid-style data, i.e., vector/matrix inputs. Therefore, they are not able to use relational data, i.e., the relation of one node to other nodes, nor fully represent structural features of graphs, such as node degree variance or algebraic connectivity.⁹ This makes it difficult to learn a general relationship between these features and the performance parameters as the structure and physical properties of graphs have a significant impact on the overall network performance. More recently, there have been many works that have applied novel combinatorial optimization techniques, using reinforcement learning to solve the RWA problem.^{20–24} However, these have yet to achieve performance comparable to the ILP solutions.

Geometric deep learning applies deep learning to graph structured data, incorporating the graph structure within the learning process. By including node features, edge features, and learnable functions that operate in a graph-structured manner, geometric deep learning learns relationships on graphs better than traditional deep learning. Graph neural networks (GNNs) are a collection of supervised learning frameworks for geometric deep learning, able to aggregate information in graphs. Message passing neural networks (MPNNs) are a specific formulation of GNN, shown to perform well for regression tasks, such as the evaluation of the maximum achievable throughput of an optical network, one of the fundamental goals of our work.²⁵ The benefit of this type of model is that it has been shown to generalize well to graphs of different sizes and structures and to be computationally efficient. The downside, however, is that it requires large datasets to learn and predict graph properties. More often than not, these are far and few between, making it difficult to apply these frameworks. Previously, this approach has been successfully applied in quantum chemistry and non-optical communication networks.^{25–27}

In this paper, we apply a MPNN architecture to model the performance of optical networks, for the first time, by learning the relationship between the topology and the maximum achievable throughput. We generated three large datasets (~80 000 graphs per dataset), with the number of nodes in the range 10–100, expanding our previous work.^{28,29} The datasets were used to train an accurate MPNN model to learn the relationship between the topology and the performance of optical networks. We demonstrate the significant computational time reduction with this model to estimate graph performance. This, in turn, reduces the complexity of optical network topology design problems by replacing computationally complex objectives with surrogate models, such as the one demonstrated here. This enables future topology optimization to maximize the achievable throughput, making topology design more intelligent and computationally efficient.

The rest of this paper is structured as follows. Section II describes the MPNN model and its operation. Section III details the methodology used to generate the three graph datasets. Section IV explains the training procedure and the different parameter settings for the MPNN model. Finally, using the generated datasets, Sec. V analyzes the model's capability to predict the maximum achievable throughput in terms of accuracy, computation time, and generalization capability.

II. MESSAGE PASSING NEURAL NETWORKS (MPNNs)

Optical networks are a set of nodes (for example, major cities or data center or servers within data centers) inter-connected, with arbitrary connectivity, by optical fibers. Optical fibers carry multiple discrete wavelength channels, each carrying individually modulated data. To transfer data between source and destination nodes, for each source–destination demand, a lightpath is set up, consisting of a route (a series of edges) and a wavelength designated for transmission. The process of selection of wavelength (color) and route is determined via the RWA algorithm. Optical networks are usually modeled as graphs and so ideally lend themselves as inputs to graph neural networks. Recently, there have been works that apply graph neural networks in the context of reinforcement learning (RL) to the RWA problem in optical networks.^{23,24} These works, however, are only compared to heuristics, where sometimes they perform better and sometimes worse. We apply MPNNs to learn from the solutions directly from integer linear programming formulations and heuristics to consistently predict performance properties of optical networks, such as the maximum achievable throughput.

We start with a digraph denoted as $G(N, E)$, where N and E are its set of nodes and edges, respectively. All nodes and edges have a pre-determined set of node and edge features, x_n and e_{nu} , respectively, where $n \in N$ and $(n, u) \in E$. In this work, the degree (δ_n) and the total traffic originating from a node are used to describe node features, and the worst case noise-to-signal ratio (NSR) of a fully populated link is used to describe the corresponding edge feature. These node and edge features are vectors with information related to either nodes: degree and traffic, or the edges: distance and signal-to-noise (SNR) ratio. MPNNs use abstract vectors, referred to as a node or edge hidden state. In this work, for simplicity, we confine the hidden states to nodes, represented as h_n^t , where t represents the message passing iteration. These hidden states are vectors that

hold embeddings for nodes, i.e., for a specific node, they capture structural information from the rest of the graph. We define the set of node features/edge features as X_N/X_E , respectively, and the set of hidden node states as H_N . GNNs can be used for either regression or classification tasks, and in this work, we focus on the regression of graph properties.

The MPNN framework centers around three functions: message function $M_t(h_n^t, h_u^t, e_{nu})$, update function $U_t(h_n^t, m_n^{t+1})$, and readout function $R(H_N, X_N)$, where n is a node in the node set N , t is the message passing iteration out of T iterations, and u is a node in the neighborhood of $n [u \in \mathcal{N}(n)]$. T generally is chosen to be in the order of either the average shortest path length or diameter of the graph. There have been several different formulations of these functions in the past,^{30–37} however, they all follow the same general algorithm.

The MPNN—outlined in Algorithm 1—is made up of three stages: (i) message passing (line 7), (ii) update (line 8), and (iii) readout (line 11). In (i), each node in the graph requests information (messages) from its neighborhood $[\mathcal{N}(n)]$. This information (messages) is given by feeding node and edge information into the message function $[M_t(h_n^t, h_u^t, e_{nu})]$. To form the nodal message (m_n^{t+1}) of node n , we have to aggregate the information (messages) given by the neighborhood of n . This can be done via different operations, i.e., averaging, sampling, or summing. We obtain the message (m_n^{t+1}) of node n by summing the messages from the neighborhood of n . This is then fed through an update function (line 8) defined as $U_t(h_n^t, m_n^{t+1})$, which updates the state (h_n^{t+1}) of each node. These two steps are illustrated in the inner block of the diagram shown in Fig. 2, where one can see that the process is repeated for each node $n \in N$. This procedure iteratively distributes the information of the graph to every node by collecting local messages and using these to update the new hidden vectors. Figure 1 demonstrates this process of message passing for the computation of the node state of node 1 on an example six-node topology. Here, the process is shown for two message passing iterations, $t = 1, 2$. Working backward from $t = 2$ and node 1, we can see that we use the neighbors of node 1, which feed their messages into the aggregation that are then updated. Before this, their neighbors do the same for their states. This demonstrates how the information about the graph is distributed across the graph during message passing.

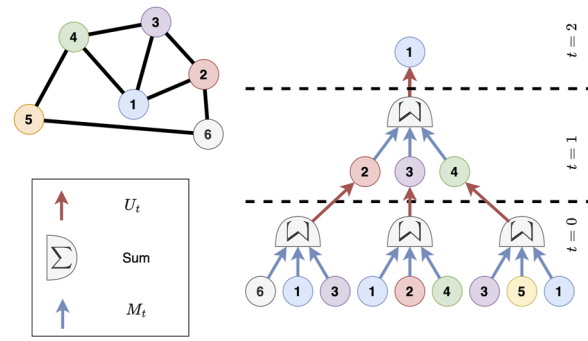


FIG. 1. An example of message passing on a six node topology.

After T message passing rounds and update layers, shown by the outer blocks in Fig. 2, the hidden states are aggregated and used to create a graph level prediction y , seen in line 11 of Algorithm 1. This process of aggregation and graph-level readout is summarized by the readout function $R(H_N, X_N)$, where H_N denotes the set of hidden states and X_N denotes the set of node features. The readout function outputs a scalar value used for prediction. This architecture has been shown to provide good learning capability for general graph structured data,²⁶ where one can incorporate individual node and edge features. Another advantage of such a model is that the architecture is size agnostic, meaning that the model can generalize to graphs of different sizes.

A. Message function

The message function is used to extract information from both hidden states of neighboring nodes and adjacent edge features, thus producing messages (line 7). The functions that are used to formulate these messages are generally learnable functions. Within optical networking, the fiber lengths of edges significantly impact the transmission performance over these edges; therefore, it is essential to include this information in the formulation of messages,

ALGORITHM 1. Message passing neural network algorithm.²⁶

```

Input:  $G, X_N, X_E$ 
Output:  $y$ 
begin
  for  $n$  to  $N$  do
     $h_n^1 \leftarrow [x_n, 0, 0, \dots, 0];$ 
  end
  for  $t = 1$  to  $T$  do
    for  $n \in N$  do
       $m_n^{t+1} \leftarrow \sum_{u \in \mathcal{N}(n)} M_t(h_n^t, h_u^t, e_{nu});$ 
       $h_n^{t+1} \leftarrow U_t(h_n^t, m_n^{t+1});$ 
    end
  end
   $y \leftarrow R(H_N, X_N);$ 
end

```

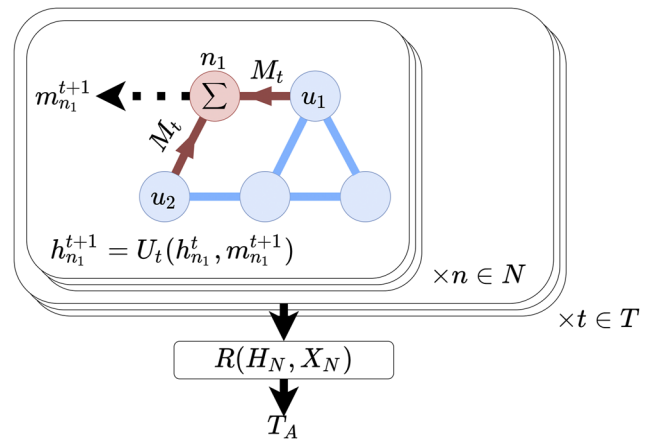


FIG. 2. Process of message passing and readout. T_A : maximum achievable throughput.

$$M_t(h_n^t, h_u^t, e_{nu}) = A(e_{nu}) \times h_u^t + b(e_{nu}). \quad (1)$$

As shown in Eq. (1), we use a matrix ANN (A) and vector valued ANN (b) to extract features from the edge feature vectors e_{nu} in constructed messages. These messages are then aggregated via a sum operation, as seen in line 7 of Algorithm 1 and the inner block of Fig. 2, to give the future message of node n , m_n^{t+1} . The sum operator is chosen due to it being permutation agnostic and for its simplicity.

B. Update function

The update function is used to take the information (messages) aggregated from the neighborhood of node n and learn a new hidden state vector h_n^{t+1} to incorporate this new information in the state (line 8). As this is a sequential process, we use a recurrent neural network (RNN) architecture, which can take previous states into account and learn how much of the previous states to use in the next state, while producing new abstract representations of the data. RNNs have been shown to struggle with vanishing gradients during training; therefore, we chose to use a gated recurrent unit (GRU). GRUs use reset and update gates to learn how much of the state and input to use or forget depending on the target. They exhibit improved performance over standard RNNs; however, they have been shown to have reasonable computational complexity,³⁸

$$U_t(h_n^t, m_n^{t+1}) = GRU(h_n^t, m_n^{t+1}). \quad (2)$$

As shown in Eq. (2), the update function solely consists of a GRU function. The current state, h_n^t , and the aggregated messages, m_n^{t+1} , are fed in to produce an updated representation, h_n^{t+1} .

C. Readout function

The final graph-level aggregation of the states and features is carried out via the readout function (line 11). The aim is to aggregate all the relevant inter-dependencies between the nodes via the hidden states, h_n^T , and represent it as a single vector, on which one can regress to the target outputs (throughput, in our case). The readout function is shown in the following equation:

$$R(H_N, X_N) = b \left(\sum_n \sigma[i(h_n^T, x_n)] \odot j(h_n^T) \right). \quad (3)$$

To learn which parts of the hidden vector, h_n^T , are important for the prediction of the target, an attention mechanism was used. The attention mechanism learns weights in the range of $[0, 1]$, which are used to weight a vector, with the goal of learning which parts of the vector are important for the learning task. This is achieved by feeding the concatenated hidden vector, h_n^T , and node features, x_n , through an ANN and passing the output through a sigmoid function σ , which normalizes the output between $[0, 1]$. Using an element-wise multiplication (Hadamard product), this vector acts as attention scores to the original hidden vector, h_n^T , which is fed through another ANN j .²⁶ Summing these operations over all nodes gives us the final vector used for the regression layer. The regression layer b consists of a single ANN layer, which reduces the output to a scalar value.

Having defined the MPNN model architecture, it now needs to be trained on graph labels, i.e., maximum achievable throughput, to predict this on all unseen graphs. Section III details the methodology for generating training and testing datasets.

III. DATASET GENERATION

To train the MPNN, three large sets of graphs, with different numbers of nodes, were generated. To calculate the maximum achievable throughput for nodes sizes below 25, an ILP was used to find optimal RWAs, although for the two larger sets of graphs ($N = 25-100$), a heuristic algorithm was used. Using these RWAs, the throughput was calculated and then stored as a graph label to be used for training and testing. This section details the methodology used for generating topologies, the ILP formulation, heuristic algorithms, the physical layer impairment (PLI) model, and the final calculation of the maximum achievable throughput.

A. Topology generation

A single generative graph model was chosen to create the graph structures for training and testing. These were created via the SNR-Barabasi-Albert (BA) model, which has been shown to reflect optical core network structures and physical properties.⁶ In this generative graph model, distances between nodes are incorporated in the process of edge selection, creating localized hubs within graphs. Nodes are chosen uniformly over a grid, representing the size of the north-American continent, resulting in unique node locations for each graph and therefore giving a greater range of graph structures and physical properties than in Ref. 6, where the same real network node locations were used in all generated graphs. The SNR-BA generative graph model then uses these unique node locations to generate the graphs. Inter-node distances were constrained to be at least 100 km,

$$D_{fibre} = \begin{cases} 1.5 \cdot D_{hav} & \text{if } D_{hav} < 1000 \text{ km,} \\ 1500 \text{ km} & \text{if } 1000 \text{ km} \leq D_{hav} \leq 1200 \text{ km,} \\ 1.25 \cdot D_{hav} & \text{if } D_{hav} > 1200 \text{ km.} \end{cases} \quad (4)$$

To model distance, D_{hav} , in the generated graphs, their geographical coordinates were used in conjunction with the haversine formula.³⁹ The haversine formula takes into account the curvature of the earth and calculates distances over a sphere rather than a plain. We also account for realistic fiber distances, as shown in Eq. (4). The fiber distances are estimated according to the European Telecommunications Standards Institute (ETSI) guidelines for estimating distance overheads in communication networks.⁴⁰

To demonstrate the model operation on different size topologies, three separate datasets were generated: (i) alpha set with $10 \leq |N| \leq 15$ (~75 000 graphs), beta set with $25 \leq |N| \leq 45$ (~95 000 graphs), and gamma set with $55 \leq |N| \leq 100$ (~75 000 graphs). The nodes increased by 1 and 5 for the alpha and beta and gamma datasets, respectively, giving 6, 5, and 10 node scales for the alpha, beta, and gamma datasets, respectively. To make sure that the model learns performance trends over a variety of edge numbers, edge numbers were chosen by adding an empirically determined percentage of the nodes for a given graph, multiple times, as seen in Eq. (5). Here, d_n was chosen to be 0.2, typical of the relatively sparse core

networks, and i varied between 1 and 10 so that the graph, empirically, has ~20% more edges than nodes; as the sparsest core networks have about 20% more edges than nodes, we have used this as the minimum value,²

$$|E| = |N| + d_n \cdot |N| \cdot i. \quad (5)$$

To generate the labels for the maximum achievable throughput values, an optimum RWA that maximizes the number of allocated lightpaths (simply referred to as the optimum RWA) is required for each network considered. To find these optimum RWAs, an ILP formulation was used for $|N| < 25$ and a heuristic was used for $|N| = 25-100$, as described in Secs. III B and III C.

B. Integer linear program

To maximize the maximum achievable throughput of the network, an ILP is developed that maximizes the number of lightpaths allocated, maximized using the objective function (6). The ILP considered here does not directly optimize the maximum achievable throughput; however, it assumes that maximizing the number of lightpaths allocated is highly correlated with the maximizing throughput within the network. Maximizing the number of lightpaths allows the problem to be specified in terms of an integer objective, making the problem less computationally complex. Maximizing throughput directly by solving the routing and wavelength allocation problem, a more complex problem to be solved because of nonlinear interactions between lightpaths. A decision variable $\delta_{w,k,z}$ is used to define a lightpath, where $w \in W$, $k \in K$, and $z \in Z$ are the set of wavelengths, k-shortest paths, and node-pairs, respectively. It is defined as in Eq. (7) and is constrained to assigning a lightpath, subject to the normalized traffic matrix (T_z^c) and the objective M , defined in Eq. (8). For this work, all the training data were generated with uniform traffic, meaning that uniform bandwidth was assumed to be routed between all node-pairs. Here, the objective is to maximize M , as summarized in Eq. (6). The wavelength continuity and edge-disjoint constraints of paths are defined in Eq. (9), where $I(j \in k)$ refers to whether edge (j) occurs on path (k),

$$\max(M), \quad (6)$$

$$\delta_{w,k,z} = \begin{cases} 1 & \text{if } (k, w) \text{ is the lightpath assignment} \\ & \text{for node pair } z, \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

$$\sum_{w \in W} \sum_{k \in K} \delta_{w,k,z} = \lceil M \cdot T_z^c \rceil \quad \forall z \in Z, \quad (8)$$

$$\sum_{z \in Z} \sum_{k \in K} \delta_{w,k,z} I(j \in k) \leq 1 \quad \forall j \in E, \quad \forall w \in W. \quad (9)$$

Using this ILP formulation, optimum RWAs were found for each graph in the alpha dataset using 156 wavelengths and 20 k-shortest paths. ILP formulations, however, do not scale for larger graphs, and heuristics in conjunction with sequential loading were used for the beta and gamma datasets, as described in Sec. III C.

C. Heuristics

Previously, it has been argued that heuristics, and specifically, first-fit k-shortest-paths (FF-kSPs) can achieve similar performance for estimating capacity in optical networks.⁴ These heuristics are used in combination with sequential loading to load the network until the maximum achievable throughput can be found, and the FF-kSP heuristic has been shown to give very good performance for this type of task, with low computational complexity.⁴ The heuristics here are used for two purposes: (i) to benchmark the performance of the MPNN ($N \leq 25$) and (ii) to generate training labels for larger graphs ($N \geq 25$), for which the ILP is infeasible.

The sequential loading applied is described by Algorithm 2. Here, the same objective (M) is maximized as in the ILP, where T_r and T_c are $N \times N$ matrices representing the connections to be routed and the normalized traffic matrix, respectively. F_R is the routing function, which takes three inputs: a graph (G), a connections matrix (T_r), and the previous RWA. Once blocking is achieved, the iteration (m_s) is halved and the network is loaded until blocking is achieved. This iterates m_i times, after which the final RWA is returned, where m_i was chosen to be 6, as more would be adding negligible amounts to the objective. In this work, kSP-FF and FF-kSP⁴ were used, both of which are common heuristics for solving the RWA problem. FF-kSP was used to generate optimum RWA configurations for the graphs in the beta and gamma datasets due to its high linear correlation to the ILP-determined performance data, as seen later in Sec. V A.

Having optimized the RWA of a given network, we need to use these optimum RWA values to calculate the maximum achievable throughput. For this, the physical layer impairments (PLIs) over the fiber in which the data are routed need to be taken into account to calculate the SNR values of the individual lightpaths.

D. Physical layer impairment model

To calculate the resultant throughput for a given RWA, we calculated the accumulated SNR for each lightpath assignment. A lightpath $i = \{p_i, \lambda_i\} \in R$ has a path p_i and a wavelength λ_i associated with it and is part of the set of lightpaths for a routing R . To

ALGORITHM 2. Sequential loading algorithm.

Input: T_c, G
Output: RWA_t

```

1 begin
2    $i = 0;$ 
3    $M = 0;$ 
4    $m_s = 100;$ 
5   for  $i \leq m_s$  do
6     while  $RWA_t \neq blocked$  do
7        $M \leftarrow M + m_s;$ 
8        $T_r \leftarrow \lceil M \cdot T_c \rceil;$ 
9        $RWA_t \leftarrow F_R(G, T_r, RWA_{t-1});$ 
10    end
11     $M \leftarrow M - m_s;$ 
12     $m_s \leftarrow \lceil m_s / 2 \rceil;$ 
13  end
14 end

```

calculate the capacity for this lightpath, one first needs to take into account the edges along which the lightpath travels and their corresponding transmission SNR values. Using Eq. (10) and the state of the edge (which wavelengths are occupied), one can calculate the $SNR_{(i,e)}$ value on each of the links $e \in p_i$. Here, P_i is the launch power; η_n is the nonlinear coefficient, which can be calculated by Eq. (5) in Ref. 41; and P_{ASE} is the power of the amplified spontaneous emission. In this work, just by way of an example, we assumed multiples of 80 km standard single mode fiber spans, with $\alpha = 0.2 \frac{dB}{km}$, $D = 18 \frac{ps}{nm km}$, and $\gamma = 1.2 \frac{1}{W km}$, amplified with erbium doped fiber amplifiers (EDFAs) with a noise figure of 4 dB. Nyquist spaced 32Gbd channels were interfaced with colorless, directionless, and contentionless reconfigurable optical add-drop multiplexers (CDC-ROADMs) over a constrained C-band (1530–1570 nm) optical bandwidth. The total SNR of that path is then calculated by taking the inverse sum of the noise-to-signal ratio (NSR) values of each link traversed by the path p_i , shown in Eq. (11),

$$SNR(i, e) \approx \frac{P_i}{P_{ASE} + \eta_n P_i^3}, \tag{10}$$

$$SNR_i = \left(\sum_{e \in p_i} \frac{1}{SNR_{(i,e)}} \right)^{-1}. \tag{11}$$

This SNR is used to calculate the maximum achievable data rate over this lightpath using Eq. (12).⁴² It can be seen that the capacity of a lightpath mainly depends on the SNR of that path, which, in turn, depends on the length and congestion along the path. B_{CH} represents the channel bandwidth used, which was kept constant at 32 GHz for all channels,

$$T_i = 2B_{CH} \log_2(1 + SNR_i), \tag{12}$$

$$T_A = \sum_{i \in R} T_i. \tag{13}$$

The throughputs for all the lightpaths were calculated and summed, as in Eq. (13), to give the total achievable throughput for a particular RWA and network. For the routing of larger networks, multiple fibers needed to be incorporated to allow for all-to-all connectivity, where four fibers per edge for $25 \leq |N| \leq 45$ and 16 fibers per edge for $55 \leq |N| \leq 100$ were used. Multiple fibers generally increase the throughput of a network by at least ξ times, where ξ is the number of fibers per edge.⁴³ The hierarchy of graphs, however, does not change when adding more fibers, i.e., which network is better does not change, therefore not changing the overall distribution of the generated graphs.

Using this methodology, the three datasets, alpha, beta, and gamma, were generated via the SNR-BA model; their RWAs were

optimized via either ILP or FF-kSP (depending on their node scale); and finally their maximum achievable throughputs (T_A) were calculated. The whole process is illustrated in Fig. 3. Using T_A as a training label (target), the MPNN model, defined in Sec. II, was trained in a supervised manner, described in Sec. IV.

IV. TRAINING

Using the datasets created as described in Sec. III, the three MPNN models were trained to predict the maximum achievable throughput, T_A . The node features (x_n) chosen to be the degree of each node (δ_n) and its normalized traffic

$$x_n = \left[\delta_n, \sum_{u \in N} T_{(n,u)}^c \right]$$

were used to initialize the node hidden vectors (h_n^1). As throughput depends on the physical properties of the fiber links, the overall transmission quality metric is a critical feature. The number of lightpaths carried over an edge or set of edges, as well as the length of paths taken, influences the overall system performance via the achievable signal-to-noise ratio in the nonlinear optical regime. Throughout testing, it was seen that the inverse of the SNR (NSR) was a better feature to use as it is an additive quantity over a set of edges. In an ideal case, optimal launch powers would be found for each wavelength in the network; however, due to the computational complexity associated with carrying this out for many networks (240 000 topologies), the assumption of uniform launch powers, i.e., equal input power/lightpath for all lightpaths, was used, similarly to other work.⁴

A hidden vector size of 16, with eight message passing rounds, was used for all model training. Larger hidden vector sizes did not seem to provide more accuracy but added higher computational complexity. The message passing rounds were chosen to work well over the whole spectrum of node scales, where $T = 8$ gave good performance for all. As overfitting was initially a problem, a dropout rate of 0.65 with L2 regularization at a rate of 0.03 was used, where higher values started reducing the accuracy of the model. The learning rate was initialized with a value of 0.001 and decayed exponentially using 10 000 steps at a rate of 0.95. The Adam optimization algorithm was used to train all models, with the graphs in batches of 50, and a single fully connected regression layer consisting of 256 neurons was used for the final regression output, where larger values did not provide further accuracy without overfitting. To monitor whether the model was tending to overfit, a validation set of 500 graphs was used to evaluate the performance at each epoch. The training was conducted on a Nvidia V100 16 GB graphics processing unit (GPU), with training generally taking around 72 h, covering 2000 epochs.

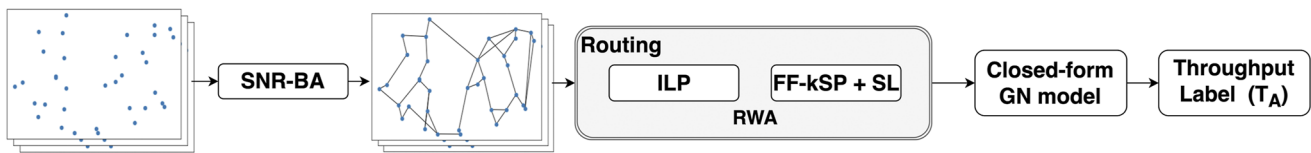


FIG. 3. Data generation process for the maximum achievable throughput labels. SL: sequential loading. T_A : maximum achievable throughput.

Having trained the three separate models on the different datasets, we used three separate test sets to gauge the performance of the trained inference models. These were generated identically to the methodology laid out in Sec. III, but were unseen (not used for training) by the model. Section V analyzes the accuracy, computation time, and generalization capabilities of the MPNN model using these test sets.

V. RESULTS

To understand how well the MPNN model performs in comparison to other methodologies for estimating the maximum achievable throughput of an optical network, there are three aspects we evaluate: (i) model accuracy, (ii) time complexity, and (iii) generalization capability to unseen graphs.

To evaluate the accuracy of the MPNN, the three test sets of 6000, 5000, and 10 000 graphs were used to test the alpha, beta, and gamma models, respectively. Here, 1000 graphs for each node scale were generated. Labels for the graphs in the alpha test set were generated via the ILP formulation, while for the beta and gamma test sets, FF-kSP was used. Again, uniform traffic was generated and used for all these results. To measure the accuracy of model predictions, the coefficient of determination R^2 was used, which measures how much of the variance in the data is explained by the model, by comparing the prediction variance to that of the original data.⁴⁴ Here, we define R^2 as in the following equation:

$$R^2 = 1 - \frac{\sum_n (y_n - p_n)^2}{\sum_n (y_n - \bar{y})^2}, \tag{14}$$

where y_n refers to the true label, p_n refers to the predicted label, and \bar{y} refers to the mean.

The Pearson’s correlation coefficient ρ was used as an alternative metric and determines how linearly labels and predictions are related, with a value of 1 signifying a perfect linear correlation. This coefficient is important in the context of surrogate models or cost functions, as a model might be inaccurate (low R^2), but might have a

high linear correlation (high ρ), which means that although inaccurate, it can predict the relative performance of a network well—vital for optimization. The predictive accuracy for all three models is analyzed and discussed in Sec. V A.

A. Maximum achievable throughput

The label that was evaluated is that of maximum achievable throughput (T_A). For each of the 6000 graphs in the alpha test set, the performance was evaluated via ILP (used for the labels), FF-kSP, kSP-FF, and MPNN and plotted in Fig. 4(a). It can be seen that the kSP-FF and FF-kSP heuristics underperform compared to the ILP, giving lower R^2 values in both, although kSP-FF is worse as it has even lower values for R^2 and ρ , as seen in Table I. This signifies poor predictive accuracy of the actual labels (T_A). The performance of kSP-FF is worse than that of FF-kSP as it prioritizes shorter paths over optimizing wavelength selection; this is an expected result seen in Ref. 4. FF-kSP generally spreads the usage more evenly across all network links compared to kSP-FF. This spreading of resources uses a slightly more spectrum, but achieves much less network congestion. However, the MPNN has learned on a variety of graphs of these sizes and can accurately predict the throughput trend here, as indicated by its high R^2 value of 0.951. Furthermore, one can see the difference in throughput distributions, where the CDF of the different methodologies is plotted in Fig. 4(b). Here, it is clear that the kSP-FF and FF-kSP heuristics give different distributions compared to the ILP. The MPNN, however, is able to replicate the original distribution of the ILP well, and therefore, it can predict the throughput of the networks better. In addition to the MPNN, linear and nonlinear ML regression methods were evaluated to compare other ML frameworks. Here, the degree variance, connectivity, algebraic connectivity, communicability distance, and communicability traffic index were used as input features for the graph. ElasticNet and the artificial neural network (ANN) were trained and tested with the same data, both scoring lower with R^2 values of 0.763 and 0.768, respectively.

Another metric evaluated was the Pearson’s correlation coefficient (ρ). As for the optimization, there is significant linear

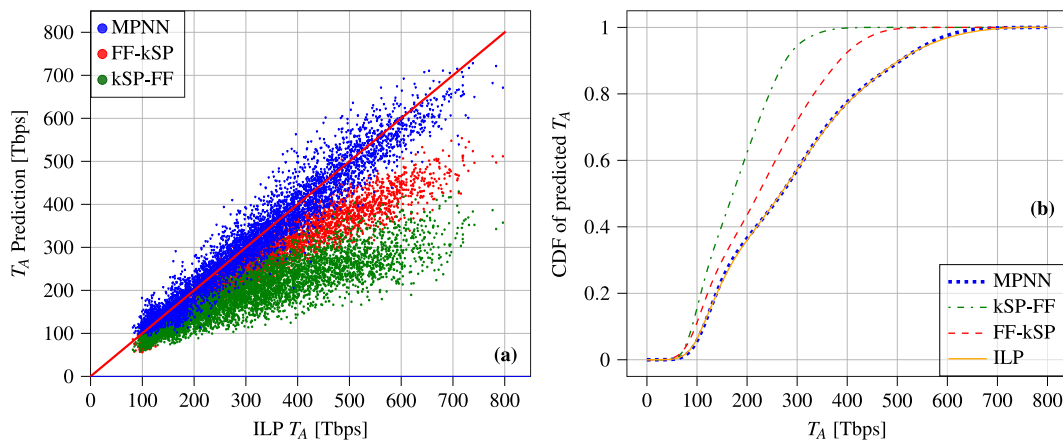


FIG. 4. (a) Throughput prediction for FF-kSP, kSP-FF, and the MPNN vs the ILP optimal value for $10 \leq N \leq 15$. (b) The cumulative distribution function (CDF) of the throughput distributions given by FF-kSP, kSP-FF, MPNN, and ILP for $10 \leq N \leq 15$.

TABLE I. Accuracy of the MPNN model and other capacity estimation methods, measured by the coefficient of determination (R^2) and the Pearson's correlation coefficient (ρ).

Node scale	Method	R^2	ρ
$10 \leq N \leq 15$	MPNN	0.951	0.975
$10 \leq N \leq 15$	FF-kSP	0.740	0.969
$10 \leq N \leq 15$	kSP-FF	0.102	0.901
$10 \leq N \leq 15$	ElasticNet	0.763	0.873
$10 \leq N \leq 15$	ANN	0.768	0.908
$25 \leq N \leq 45$	MPNN	0.973	0.986
$55 \leq N \leq 100$	MPNN	0.948	0.974

correlation between the real performance and predicted performance properties. It can be seen that the heuristics generally perform well, and the FF-kSP has a high linear correlation ($\rho = 0.969$) between the estimated throughput values and those calculated via the ILP, as seen in Table I. The MPNN has a similar correlation, with $\rho = 0.975$, meaning that it predicts the relative throughput performance of networks well. ElasticNet and the ANN both score ρ values lower than the MPNN and FF-kSP. The high linear correlation of FF-kSP compared to the ILP makes it a good candidate to evaluate the maximum achievable throughput for the larger graphs, even though the real maximum achievable throughput might be larger.

To further evaluate the performance of the model, the model was applied to the beta and gamma testing datasets that included larger graphs. Here, the labels were generated via the FF-kSP heuristic as the ILP is not able to scale to these larger topologies. The MPNN was used to infer the maximum achievable throughput of the graphs in the respective test sets and plotted in Figs. 5(a) and 5(b). It is clear that for the beta model, the MPNN performs better than the alpha and gamma models, with an R^2 value of 0.973. This is because the beta training set had the most training graphs per node scale. This was possible for the beta model as the heuristic runs

faster on the smaller beta graphs than the larger gamma set graphs and also faster than the ILP. This can be seen in Figs. 6(a) and 6(b). Although the gamma model still achieves a high R^2 of 0.948, one can see that the larger throughput values have more variance in them. To remedy this, more samples are needed in the training set for larger node scales. Both the beta and gamma models achieved high linear correlation values (ρ) of 0.986 and 0.974, respectively.

One can conclude that heuristics consistently underperform in estimating the throughput in optical networks compared to the ILP solutions. However, FF-kSP is able to predict the trend (linear correlation— ρ) well, making it suitable for optimization tasks and training data generation. The MPNN on the other hand is able to accurately predict throughput values of unseen data based on similar structure and sizes seen during training. It also has a high linear correlation between the labels and its own predictions, making it an excellent candidate for modeling the throughput of these optical networks. Therefore, the MPNN is able to predict maximum achievable throughput accurately although it does not provide a solution, for example, for the RWA, of how to reach it. The model could potentially be adjusted to learn the RWA; this is outside the scope of the current work and is left for future research. However, given the extensive training and data needs of this model, what is the real benefit of using it in place of heuristics? This question is addressed in Sec. V B.

B. Computational time comparison

The key advantage of using machine learning to model relationships in data is very low inference times compared to ILP and, even, heuristic methods.

The ILP has a worst-case computational complexity, as described in the following equation, where D is the number of demands or connection requests ($D = \sum_{z \in Z} [T_z^C \cdot M]$), E is the number of edges, and W is the number of wavelengths used:⁴⁵

$$O(2^{D \cdot E \cdot W}). \tag{15}$$

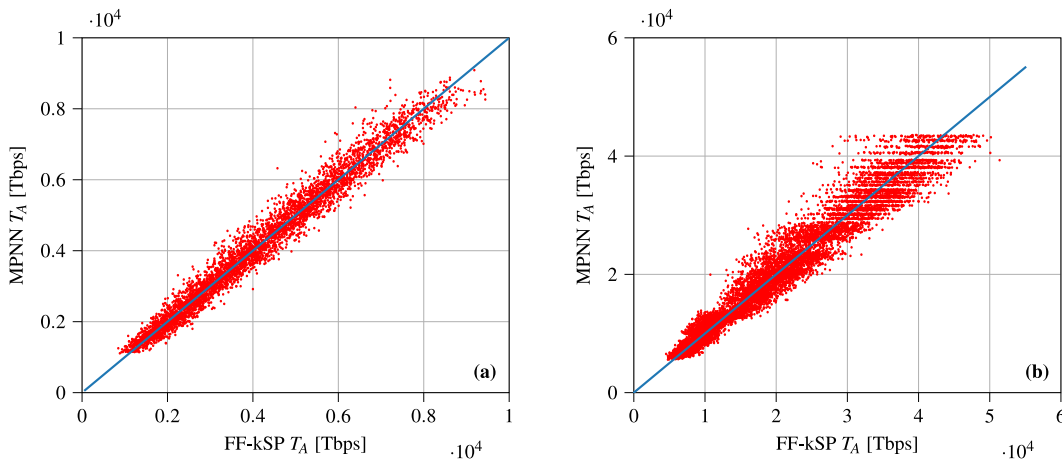


FIG. 5. (a) Throughput prediction of the MPNN vs the FF-kSP prediction for $25 \leq |N| \leq 45$. (b) Throughput prediction using MPNN vs the FF-kSP prediction for $55 \leq |N| \leq 100$.

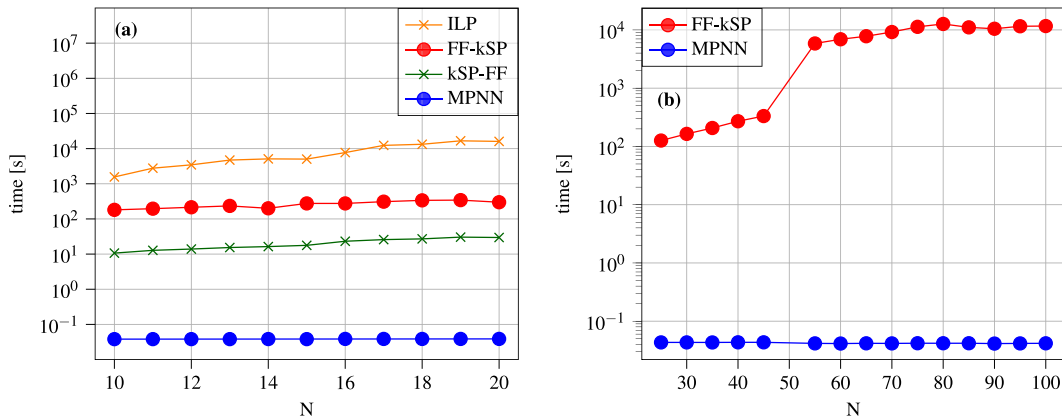


FIG. 6. (a) Comparison between computation times for the throughput of networks over different node scales using ILP, FF-kSP, kSP-FF, and MPNN. (b) Comparison between computation times for the throughput of networks using FF-kSP and MPNN for $25 \leq 100$.

For the heuristic algorithms used, the complexity generally scales as in Eq. (16),⁴⁶ but must be modified to include the term R —number of sequential loading iterations, as the heuristic needs to run many times before finding the optimum RWA,

$$O(RkN^3(E + N \log(N))). \quad (16)$$

The advantage of the MPNN, thus, is that it can directly evaluate the network properties, learned from previous data. For the inference of the MPNN, the complexity is defined in the following equation,⁴⁷ where T denotes the number of message passing rounds, defined previously, and d is the length of hidden dimensional vector used, 16 in our case:

$$O(T \cdot N^2 d^2). \quad (17)$$

Comparing Eqs. (15)–(17), it can be seen that the MPNN scales the best, computationally, with the number of nodes. To quantify the computational benefits of using MPNNs to model optical networks, graphs with nodes varying from 10 to 20 were used to evaluate the ILP, FF-kSP, kSP-FF, and MPNN time performance. Here, the node size has been expanded to 20 nodes for the test set to see the computational time scaling better with smaller graphs. This, however, was not possible in the case of the training set since the ILP computation is too complex. For each graph, the respective methodologies were used to calculate the maximum achievable throughput and their computation times measured and plotted in Fig. 6(a). The reduction in computation time through using the MPNN model can be clearly seen, where it takes ~10 s of ms, compared to 10, 100, and 1000 s of seconds for kSP-FF, FF-kSP, and ILP, respectively. The same trend, between FF-kSP and the MPNN, was analyzed for the larger networks using the beta and gamma test sets, with the results shown in Fig. 6(b). The MPNN again shows a minimum computation time increase for these node ranges, compared to the heuristic method.

Therefore, the MPNN model can be seen as an accurate and fast method for evaluating performance metrics of graphs, which are generally computationally difficult to evaluate. This would allow for

the fast evaluation of a large number of graphs within any topology design process. However, how well does this model generalize when varying the input structure and traffic distributions that affect the throughput of the network? This is the question explored in Sec. V C.

C. Generalization capability

The generalization capability of an machine learning (ML) model refers to the ability of the model to operate over distributions not seen during the training process. In the context of this work, this could encompass different graph structures, different traffic distributions, or distance scales. Here, we choose to evaluate the model against two different graph structures and a change in input traffic distribution.

To test different graph structures, two common generative models were used to vary the structures tested, the Erdos–Renyi (ER) and Barabasi–Albert (BA) models.^{48,49} Using these models, 5000 graphs with $25 \leq |N| \leq 45$ were generated per generative model. After calculating the throughputs of these graphs using the FF-kSP heuristic and the Gaussian Noise (GN) model, we feed the graphs through the MPNN model to predict the throughput values.

When comparing the accuracy of the model over these varied graph structures, one can see that the accuracy, in terms of coefficient of determination (R^2), drops significantly, as seen in Table II.

This means that the predictions vary largely from the actual labels of the graphs. However, this is to be expected as the graph structures resulting from the ER and BA models are largely different from those from SNR-BA graphs.⁶ This difference in the graph structure can be quantified by the weighted spectral density distance (WSD),⁵⁰ which measures the difference in structure between two sets of graphs. When the WSD (F_{WSD}) is smaller, the graph structures are more similar. The weighted spectral density distance is calculated between the original SNR-BA test graphs and the various test sets and are shown in Table II. Here, one can see that the WSD is close to zero for SNR-BA graphs as they are generated from the same distribution as the originally tested SNR-BA graphs. We

TABLE II. Accuracy for generalization capability, measured by the coefficient of determination R^2 and ρ . The variable γ determines how locally skewed the traffic is, and D is the absolute distance between the test throughput and the original training throughput (ks-2s test).

Graph type	γ	R^2	ρ	D	F_{WSD}
ER	0.0	0.776	0.928	0.096	0.000 286
BA	0.0	0.830	0.938	0.125	0.000 201
SNR-BA	0.0	0.973	0.986	0.000	0.000 000
SNR-BA	0.2	0.901	0.983	0.079	2.13×10^{-6}
SNR-BA	0.4	0.907	0.983	0.075	2.13×10^{-6}
SNR-BA	0.6	0.906	0.983	0.060	2.13×10^{-6}
SNR-BA	0.8	0.911	0.984	0.058	2.13×10^{-6}
SNR-BA	1.0	0.907	0.982	0.062	2.13×10^{-6}

can see that the ER and BA graph's WSD is larger, where the ER structures are the most different from the original test graphs. In addition, due to the difference in graph structures, there are large differences in throughput distributions. This is measured by the Kolmogorov–Smirnov two sample test, which returns a distance D , which signifies the largest absolute difference between the CDF of two distributions. In Table II, the large D values for ER and BA test sets signify the difference in throughput distributions to those seen during training. These D values are larger than those of the SNR-BA test distributions, meaning that their throughput distributions are further from those arising from SNR-BA distributions. The Pearson's correlation coefficient (ρ) however remains high, which shows that it still indicates the throughput trends of the networks well. This is important for optimization as the cost function might not need to accurately describe throughput exactly; however, it needs to describe one graph being better than another.

To investigate how a change in the traffic demand matrix affects the accuracy of the model, we generated localized skewed traffic matrices for 5000 graphs. By defining the traffic as in the following equation, we generated five different traffic skews shown in Table II:

$$T_{i,j}^c = \frac{1}{\left(\frac{D(i,j)}{\sum_{k \in N} D(i,k)}\right)^\gamma} \quad (18)$$

For each of these skewed traffic matrices, we tested the MPNN accuracy in terms of R^2 and ρ . The R^2 values drop by about 7% and remain constant for the different skew values ($\gamma > 0$). The Pearson's correlation coefficient (ρ) value remains high at around 0.98, unchanged from the uniform traffic distribution results, meaning that it still predicts the throughput trend well.

The large variation in performance, in terms of R^2 for ER and BA graphs, indicates that the trained MPNN model does not generalize well to largely different graph structures. This highlights the importance of using a variety of different structures within the training dataset and that an expansion of the training data is necessary here to represent different graph structures. Once the training set is more representative, by generating a variety of graph structures, it accurately evaluates the vast solution space of the topology design problem.

VI. CONCLUSION

In this paper, the well-known NP-hard problem of estimating optimal performance of an optical network was investigated, focusing, in particular, on the maximum achievable network throughput. Understanding and being able to predict it are vital for optimizing optical networks for the future to maximize the longevity of investments made in infrastructure for communications and improve existing topologies. We have proposed the use of a geometric deep learning framework to learn this network property from a previously generated training dataset of graphs.

By applying this methodology, we showed that the model can accurately model the maximum achievable throughput of previously unseen graphs, proven by high coefficient of determination accuracy (R^2) and Pearson's correlation coefficient (ρ) values. This accuracy was achieved with significant time savings of multiple orders of magnitude, enabling rapid prediction (~ 30 ms for 100 node graphs) of maximal performance of a large number of graphs.

Furthermore, both the importance and weaknesses of the training datasets were highlighted by applying the model to other graph structures, such as ER and BA graphs, where the model struggled to give accurate predictions, showing that the training set of graphs needed to be expanded. The generalization of different traffic input distributions generally performed well and showed suitability for future optimization purposes.

This type of modeling is invaluable to further optimize the physical topologies of optical networks as it acts as a surrogate model in future optimization. It directly enables the use of the maximum achievable throughput in the cost function of any number of optimization algorithms and gives huge time savings. In addition, networks are ever-evolving and may not operate with a fixed grid of channels. They might have a flexible grid, regeneration devices, or intermediate grooming at routers or switches. These changes in network configuration could be included in future training sets by using them as node or edge features and generating more training data using a range of these configurations. Further investigations will include work on analyzing the resilience of networks, accurate traffic modeling, and generalization to multiple fiber systems using MPNNs.

ACKNOWLEDGMENTS

Financial support from UK EPSRC Doctoral Training Programme and the Programme Grant TRANSNET (Grant No. EP/R035342/1) is gratefully acknowledged. Microsoft is acknowledged for the support under the "Optics for the Cloud" program.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Robin Matzner: Conceptualization (lead); Data curation (lead); Formal analysis (lead); Investigation (lead); Methodology (lead); Visualization (lead); Writing – original draft (lead). **Ruijie Luo:**

Investigation (supporting); Writing – review & editing (supporting). **Georgios Zervas**: Writing – review & editing (supporting). **Polina Bayvel**: Supervision (lead); Writing – original draft (supporting); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available via UCL research data repository,⁵¹ and the code to train, test, and read the data is available from our GitHub repository.⁵²

REFERENCES

- Z. Zhao, B. Guo, Y. Shang, and S. Huang, “Hierarchical and reconfigurable optical/electrical interconnection network for high-performance computing,” *J. Opt. Commun. Networking* **12**, 50–61 (2020).
- J. M. Simmons, *Optical Network Design and Planning*, Optical Networks (Springer International Publishing, Cham, 2014).
- I. Chlamtac, A. Ganz, and G. Karmi, “Purely optical networks for terabit communication,” in *IEEE INFOCOM ’89, Proceedings of the Eighth Annual Joint Conference of the IEEE Computer and Communications Societies* (IEEE, 1989), Vol. 3, pp. 887–896.
- R. J. Vincent, D. J. Ives, and S. J. Savory, “Scalable capacity estimation for nonlinear elastic all-optical core networks,” *J. Lightwave Technol.* **37**, 5380–5391 (2019).
- P. Bayvel, R. Luo, R. Matzner, D. Semrau, and G. Zervas, “Intelligent design of optical networks: Which topology features help maximise throughput in the nonlinear regime?,” in *2020 European Conference on Optical Communications (ECOC)*, Brussels, Belgium, 6–10 December 2020 (IEEE, 2020), pp. 887–896.
- R. Matzner, D. Semrau, R. Luo, G. Zervas, and P. Bayvel, “Making intelligent topology design choices: Understanding structural and physical property performance implications in optical networks [invited],” *J. Opt. Commun. Networking* **13**, D53–D67 (2021).
- S. Baroni and P. Bayvel, “Wavelength requirements in arbitrarily connected wavelength-routed optical networks,” *J. Lightwave Technol.* **15**, 242–251 (1997).
- C. Fenger, E. Limal, U. Gliese, and C. J. Mahon, “Statistical study of the correlation between topology and wavelength usage in optical networks with and without conversion,” in *Networking 2000. Broadband Communications, High Performance Networking, and Performance of Communication Networks*, Lecture Notes in Computer Science, edited by G. Pujolle, H. Perros, S. Fdida, U. Körner, and I. Stavrakakis (Springer, Berlin, Heidelberg, 2000), pp. 168–175.
- B. Châtelain, M. P. Bélanger, C. Tremblay, F. Gagnon, and D. V. Plant, “Topological wavelength usage estimation in transparent wide area networks,” *J. Opt. Commun. Networking* **1**, 196–203 (2009).
- D. J. Ives, P. Bayvel, and S. J. Savory, “Routing, modulation, spectrum and launch power assignment to maximize the traffic throughput of a nonlinear optical mesh network,” *Photonic Network Commun.* **29**, 244–256 (2015).
- R. Ramaswami and K. N. Sivarajan, “Routing and wavelength assignment in all-optical networks,” *IEEE/ACM Trans. Networking* **3**, 489–500 (1995).
- S. Subramaniam and R. Barry, “Wavelength assignment in fixed routing WDM networks,” in *Proceedings of ICC’97–International Conference on Communications* (IEEE, 1997), Vol. 1, pp. 406–410.
- R. Rodriguez-Dagnino, E. Lopez-Caudana, H. Martinez-Alfaro, and J. Gonzalez-Velarde, “Simulated annealing and stochastic ruler algorithms for wavelength assignment planning in WDM optical networks,” in *IEEE SMC’99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 99CH37028)* (IEEE, 1999), Vol. 6, pp. 1015–1020.
- N. Banerjee, V. Mehta, and S. Pandey, “A genetic algorithm approach for solving the routing and wavelength assignment problem in WDM networks,” in *3rd IEEE/IEE international conference on networking, ICN* (2004), pp. 70–78.
- P. Wright, M. C. Parker, and A. Lord, “Minimum- and maximum-entropy routing and spectrum assignment for flexgrid elastic optical networking [invited],” *J. Opt. Commun. Networking* **7**, A66–A72 (2015).
- Á. Rubio-Largo, M. A. Vega-Rodríguez, J. A. Gomez-Pulido, and J. M. Sanchez-Pérez, “A comparative study on multiobjective swarm intelligence for the routing and wavelength assignment problem,” *IEEE Trans. Syst., Man, Cybern., Part C (Appl. Rev.)* **42**, 1644–1655 (2012).
- D. J. Ives, P. Bayvel, and S. J. Savory, “Physical layer transmitter and routing optimization to maximize the traffic throughput of a nonlinear optical mesh network,” *IEEE in 2014 International Conference on Optical Network Design and Modeling* (2014), pp. 168–173.
- D. R. B. de Araújo, C. J. A. Bastos-Filho, and J. F. Martins-Filho, “An evolutionary approach with surrogate models and network science concepts to design optical networks,” *Eng. Appl. Artif. Intell.* **43**, 67–80 (2015).
- M. Asghari, J. Bagherzadeh, and S. Yousefi, “Loss estimation and control mechanism in bufferless optical packet-switched networks based on multilayer perceptron,” *Photonic Network Commun.* **35**, 274–286 (2018).
- N. Di Cicco, E. F. Mercau, O. Karandin, O. Ayoub, S. Troia, F. Musumeci, and M. Tornatore, “On deep reinforcement learning for static routing and wavelength assignment,” *IEEE J. Sel. Top. Quantum Electron.* **28**, 3600112 (2022).
- J. W. Nevin, S. Nallaperuma, N. A. Shevchenko, Z. Shabka, G. Zervas, and S. J. Savory, “Techniques for applying reinforcement learning to routing and wavelength assignment problems in optical fiber communication networks,” *J. Opt. Commun. Networking* **14**, 733 (2022).
- H. T. Quang, O. Houidi, J. Errea-Moreno, D. Verchere, and D. Zeghlache, “MAGC-RSA: Multi-agent graph convolutional reinforcement learning for distributed routing and spectrum assignment in elastic optical networks,” *IEEE in 2022 European Conference on Optical Communication (ECOC)* (2022), pp. 1–4.
- P. Almasan, J. Suárez-Varela, K. Rusek, P. Barlet-Ros, and A. Cabellos-Aparicio, “Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case,” *Comput. Commun.* **196**, 184–194 (2022).
- L. Xu, Y.-C. Huang, Y. Xue, and X. Hu, “Deep reinforcement learning-based routing and spectrum assignment of EONs by exploiting GCN and RNN for feature extraction,” *J. Lightwave Technol.* **40**, 4945–4955 (2022).
- J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in *Proceedings of the 34th International Conference on Machine Learning* (PMLR, 2017), pp. 1263–1272.
- K. Rusek and P. Cholda, “Message-passing neural networks learn Little’s law,” *IEEE Commun. Lett.* **23**(2), 274–277 (2018).
- K. Rusek, J. Suárez-Varela, P. Almasan, P. Barlet-Ros, and A. Cabellos-Aparicio, “RouteNet: Leveraging graph neural networks for network modeling and optimization in SDN,” *IEEE J. Sel. Areas Commun.* **38**, 2260–2270 (2020).
- R. Matzner, R. Luo, G. Zervas, and P. Bayvel, “Ultra-fast optical network throughput prediction using graph neural networks,” *IEEE in 26th International Conference on Optical Network Design and Modelling* (2022), p. 3.
- R. Matzner, R. Luo, G. Zervas, and P. Bayvel, “Expanding Graph Neural Networks for Ultra-Fast Optical Core Network Throughput Prediction to Large Node Scales,” in *2022 European Conference on Optical Communication (ECOC)* (IEEE, 2022), (pp. 1–4).
- D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, “Convolutional networks on graphs for learning molecular fingerprints,” in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2015), Vol. 28.
- Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, “Gated graph sequence neural networks,” *arXiv:1511.05493* [cs, stat] (2017).
- P. W. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, and K. Kavukcuoglu, “Interaction networks for learning about objects, relations and physics,” in *NIPS*, 2016.
- S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley, “Molecular graph convolutions: Moving beyond fingerprints,” *J. Comput.-Aided Mol. Des.* **30**, 595–608 (2016).
- K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, “Quantum-chemical insights from deep tensor neural networks,” *Nat. Commun.* **8**, 13890 (2017).
- J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” *arXiv:1312.6203* [cs] (2014).
- M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2016), Vol. 29.

- ³⁷T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” [arXiv:1609.02907](https://arxiv.org/abs/1609.02907) [cs, stat] (2017).
- ³⁸K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder–decoder approaches,” in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation* (Association for Computational Linguistics, Doha, Qatar, 2014), pp. 103–111.
- ³⁹S. De Maesschalck, “Pan-European optical transport networks: An availability-based comparison,” *Photonic Network Commun.* **5**, 203–225 (2002).
- ⁴⁰S. De Maesschalck, “Network aspects (NA): Availability performance of path elements of international digital paths,” Technical Report No. REN/NA-042140 (3uo0ioo.PDF), European Telecommunications Standards Institute, France, 1998.
- ⁴¹D. Semrau, R. I. Killey, and P. Bayvel, “A closed-form approximation of the Gaussian noise model in the presence of inter-channel stimulated Raman scattering,” *J. Lightwave Technol.* **37**, 1924–1936 (2019).
- ⁴²C. E. Shannon, “A mathematical theory of communication,” *Bell Syst. Tech. J.* **27**, 379–423 (1948).
- ⁴³A. Ferrari, E. Virgillito, and V. Curri, “Band-division vs. space-division multiplexing: A network performance statistical assessment,” *J. Lightwave Technol.* **38**, 1041–1049 (2020).
- ⁴⁴A. Di Bucchianico, “Coefficient of determination (R^2),” in *Encyclopedia of Statistics in Quality and Reliability* (American Cancer Society, 2008).
- ⁴⁵R. Luo, Y.-Z. Xu, R. Matzner, G. Zervas, D. Saad, and P. Bayvel, “Message passing: Towards low-complexity, global optimal routing and wavelength assignment solutions for optical networks,” IEEE in *2022 Optical Fiber Communications Conference and Exhibition (OFC)* (2022), pp. 1–3.
- ⁴⁶E. Bouillet, G. Ellinas, J.-F. Labourdette, and R. Ramamurthy, *Path Routing in Mesh Optical Networks* (John Wiley & Sons, 2007).
- ⁴⁷Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A comprehensive survey on graph neural networks,” *IEEE Trans. Neural Networks Learn. Syst.* **32**, 4–24 (2021).
- ⁴⁸P. Erdos and A. Renyi, *On the Evolution of Random Graphs* (Publication of the Mathematical Institute of the Hungarian Academy of Sciences, 1960), pp. 17–61.
- ⁴⁹A.-L. Barabasi and R. Albert, “Emergence of scaling in random networks,” *Science* **286**, 509–512 (1999); [arXiv:cond-mat/9910332](https://arxiv.org/abs/cond-mat/9910332).
- ⁵⁰D. Fay, H. Haddadi, A. Thomason, A. W. Moore, R. Mortier, A. Jamakovic, S. Uhlig, and M. Rio, “Weighted spectral distribution for internet topology analysis: theory and applications,” *IEEE/ACM Trans. Networking* **18**, 164–176 (2009).
- ⁵¹R. Matzner (2022). “Intelligent performance inference: A graph neural network approach to modelling maximum achievable throughput in optical networks,” UCL Research. https://rdr.ucl.ac.uk/projects/Intelligent_Performance_Inference_a_Graph_Neural_Network_Approach_to_Modelling_Maximum_Achievable_Throughput_in_Optical_Networks/154985
- ⁵²R. Matzner (2022). “Intelligent performance inference: A graph neural network approach to modelling maximum achievable throughput in optical networks,” GitHub. <https://github.com/uceedat/Intelligent-Performance-Inference-a-Graph-Neural-Network-Approach-to-Modelling-Maximum-Achievable-T>