Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Clarity in chaos: Boosting few-shot classification through information suppression and sparsification

Xiaoxu Li [a], Luchen Ji [a], Rui Zhu [b,*], Zhanyu Ma [c], Jing-Hao Xue [d]

[a] School of Computer and Communication, Lanzhou University of Technology, Lanzhou, 730050, China
[b] Faculty of Actuarial Science and Insurance, Bayes Business School, City St George's, University of London, London, EC1Y 8TZ, UK
[c] Pattern Recognition and Intelligent System Laboratory, School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, 100876, China
[d] Department of Statistical Science, University College London, London, WC1E 6BT, UK

ARTICLE INFO

Keywords:
Few-shot classification
Irrelevant information suppression

ABSTRACT

The advance of deep learning has invigorated the research of few-shot classification. However, the interference of non-target information in feature representations hampers classification generalization. To tackle this issue, we propose an irrelevant information suppression (IIS) module, which is focused on *suppressing the weight of unimportant information and elevating the sparsity of feature representations*. An IIS network with three consecutive IIS modules is developed, to illustrate the progressive suppression of unimportant information and highlighting of key discriminative features of the target. Extensive experiments showcase the superior performance of our IIS network on five widely-used benchmark datasets. Furthermore, we show that the IIS module can be readily used as a plug-in module by state-of-the-art few-shot classifiers, and can clearly further improve their performance. Our code is available on GitHub at https://github.com/LC4188/IISNet.

1. Introduction

Deep learning-based classifiers traditionally require a large amount of annotated data for model training. In practice, however, it is often very expensive if not impossible to obtain a large number of labeled samples. The objective of few-shot image classification [1,2] is to accurately learn new visual concepts from only very few labeled samples. This task is defined as classifying query images into new classes, where the new classes were not involved in the model training, and only a few images are available to support new classes.

However, due to the interference of non-target information, the learned embeddings often suffer from overfitting to irrelevant information, hindering the generalization to unseen novel classes [3]. Therefore, it is necessary to calibrate feature embeddings by reducing the effect of the image content irrelevant to the target. This is particularly prominent and challenging in few-shot learning, as few samples are available for learning.

To achieve this goal, we propose an irrelevant information suppression (IIS) module, which is aimed at suppressing irrelevant information in the base feature representation that is unrelated to a class, thereby reducing its impact on classification. The IIS module consists of two components: the perceptive information (PI) module and the self-subtraction (SS) module.

The PI module is responsible for recognizing the key areas of the target. Firstly, it transforms the base feature representation into a self-similarity tensor [4] to obtain the structural information of an image by computing the product between each position in the feature map and its neighborhood, which has been utilized for object detection and few-shot classification [5]. Subsequently, a dual-branch convolutional operation with two different kernel sizes is developed to aggregate the local and global similarity information. The key areas of the target are determined as the positions with strong agreement between the local and global windows.

The SS module then utilizes the key area information provided by the PI module for IIS. To achieve this, we propose a novel inverse operation to subtract the irrelevant information or the inversion of the key target information, from the base features, ensuring that only regions deemed important by both the base representation and the PI module are retained and those with irrelevant information receive very low or zero activation values. Therefore, the output feature representation from the IIS module is sparse and can identify the most vital information to help recognize the target.

For illustrative purposes, we introduce an irrelevant information suppression network (IISNet), which is composed of three concatenated IIS modules. As depicted in Fig. 1, the input image undergoes feature

* Corresponding author.
E-mail address: rui.zhu@city.ac.uk (R. Zhu).

https://doi.org/10.1016/j.patcog.2025.111726
Received 12 November 2024; Received in revised form 31 March 2025; Accepted 16 April 2025
Available online 30 April 2025

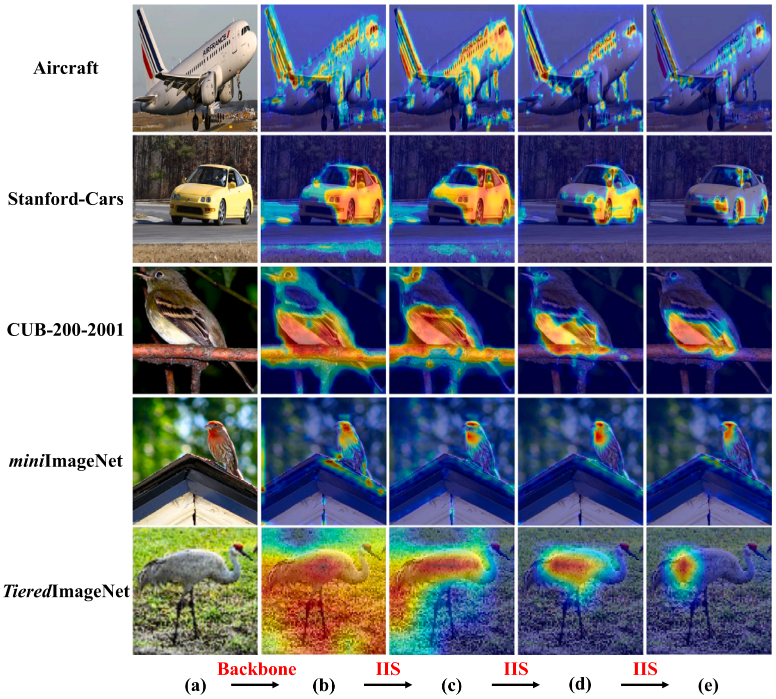


Fig. 1. The effect of the proposed irrelevant information suppression (IIS) module. (a) Five randomly selected images from five benchmark datasets. (b) Feature mapping by backbone (ResNet-12), with many irrelevant details mapped to the feature space. To progressively eliminate these irrelevant features while retaining discriminative foreground features, we introduce three concatenated IIS modules, with progressively improved results shown in (c), (d), and (e).

extraction through the backbone network, resulting in base feature representations (Fig. 1(a)→(b)). It is observed that many details irrelevant to the target are mapped to the feature space. By progressively applying IIS modules layer by layer, these irrelevant features can be effectively eliminated gradually while the discriminative features of the target is more and more highlighted (see Fig. 1(c)→(e)).

In short, our main novelties and contributions are as follows:

- We propose an irrelevant information suppression (IIS) module, which can be readily applied to progressively suppress non-target information in the base representations.
- We show that the IIS module can also be readily used as a plug-in module by state-of-the-art few-shot classifiers.
- We also show that, when incorporated into the state-of-the-art few-shot classification models, the IIS module can clearly improve their performance.

2. Related work

2.1. Few-shot classification

Recent methods in few-shot classification can be broadly categorized into three types. Optimization-based methods [6–8] focus on designing appropriate objective functions and choosing suitable optimization algorithms. Data augmentation-based methods [9,10] improve the generalization ability of classifiers by increasing the diversity and quantity of training data. Metric-based methods use or learn distance or similarity measures between samples [11–13]. Common metric functions are Euclidean distance [14], cosine similarity [15,16], Manhattan distance, etc. These functions project features into a metric space and measure the similarity between samples [5].

One challenging task in few-shot image classification is to classify fine-grained images, where images are labeled by subcategories with quite similar appearance details. To take the subtle inter-class variation into consideration, attention mechanisms [17–19] are used to generate features with strong discriminative power. In a recent work, [20] proposes to enhance the discriminative ability by tailoring multi-scale features.

In our work, after suppressing irrelevant information, we adopt a metric-based approach using the cosine similarity to measure the similarity between query images and class prototypes. Our work is also demonstrated to work well on both coarse-grained and fine-grained images.

2.2. Irrelevant information suppression

Irrelevant information suppression (IIS) is to highlight target regions by suppressing unimportant information. Previous work has delved deeply into the influence of image background on learning-based visual systems from various perspectives. The study by [21] provides preliminary evidence of false correlations between background and image categories, and further reveals the negative impact of background on visual model predictions. A simple approach to IIS is to set a threshold to filter out background areas that are not suppressed by backbone. [22] introduces a background suppression and foreground alignment (BSFA) method with the background activation suppression (BAS) module to generate foreground masks by suppressing background activation values. However, BAS simply calculates the mean of features in the base representation as a threshold to roughly filter out background information, with no guarantee about the importance of the features kept. [23] adopts a similar approach to obtain the background map and suppress its values. Contrastive learning is also utilized to distinguish foreground and background features. For instance, [24] presents a novel contrastive learning framework, COSOC, to extract foregrounds by identifying shared patterns among images. [25] proposes to extract feature representations in the frequency domain using discrete cosine transform to filter out high-frequency signals considered as noise. [26] proposes the foreground-background contrastive learning (FBCL) method to separate features of objects and background with supervised contrastive learning. However, contrastive learning requires significant computational resources and training time. [27] proposes the foreground object transformation (FOT) method, which utilizes a

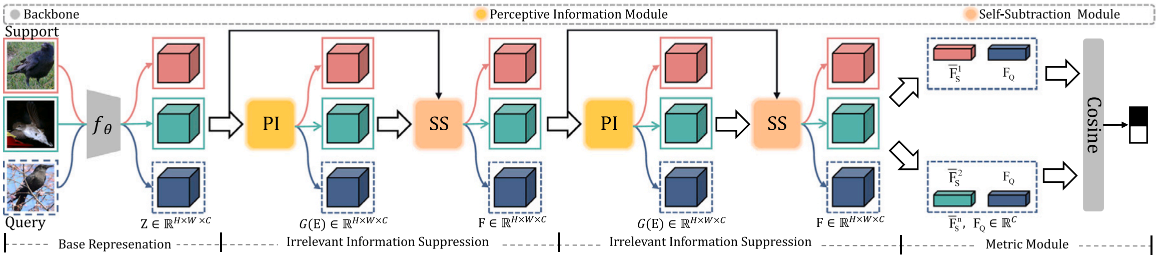

Backbone
Perceptive Information Module
Self-Subtraction Module
Support
Query
f_θ
PI
SS
PI
SS
$Z \in \mathbb{R}^{H\times W\times C}$
$G(E) \in \mathbb{R}^{H\times W\times C}$
$F \in \mathbb{R}^{H\times W\times C}$
$G(E) \in \mathbb{R}^{H\times W\times C}$
$F \in \mathbb{R}^{H\times W\times C}$
$\bar{F}_S^1$
F_Q
$\bar{F}_S^2$
F_Q
$\bar{F}_S^n$, $F_Q \in \mathbb{R}^C$
Cosine
Base Represenation
Irrelevant Information Suppression
Irrelevant Information Suppression
Metric Module

Fig. 2. The architecture of the IISNet with two IIS layers. The base representations $\mathbf{Z}$ are obtained through the feature extractor f_θ, which are then processed by d concatenated IIS modules to progressively suppress irrelevant information. In the PI module, the key structural information $G(\mathbf{E})$ is extracted. Subsequently, in the SS module, the irrelevant information is obtained as an inversion of $G(\mathbf{E})$, which is then subtracted from $\mathbf{Z}$, resulting in the irrelevant information-suppressed feature representation $\mathbf{F}$. The support and query features $\mathbf{F}$ are then averaged in the spatial dimensions, leading to C-dimensional representations, $\mathbf{F}_S$ and $\mathbf{F}_Q$, respectively. The cosine similarity between the query feature $\mathbf{F}_Q$ and the nth support class prototype, $\bar{\mathbf{F}}_S^n$, is calculated to determine the class label.

pre-trained BASNet [28] to identify the foreground and background regions in the image.

In our work, the proposed IIS module suppresses irrelevant information by jointly considering the base representation and the key areas of targets identified by local and global structural information. In addition, our approach can ensure competitive performance with relatively low computational resource consumption.

3. Our approach

3.1. Preliminaries

Few-shot classification aims to achieve superior generalization performance on new classes, by training on base classes with few images per class. The majority of few-shot classifiers employ the N-way K-shot episodic training strategy to enhance the generalization ability [1,2,14]. That is, the classifier is trained on a large number of randomly sampled tasks from the base dataset, and each task is composed of N classes with K images per class. The evaluation of the classifier is then performed on the novel dataset with classes not seen during training.

Specifically, given a dataset with image and label pairs, $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^T$, where $\mathbf{x}_i$ denotes the ith image, y_i is its label and T is the total number of observations, we randomly divide it to three subsets: a base set $\mathcal{D}_{\text{base}}$, a validation set $\mathcal{D}_{\text{valid}}$ and a novel set $\mathcal{D}_{\text{novel}}$. The label sets of the three subsets are mutually exclusive and their union is the label set of $\mathcal{D}$. During the training process, we repeat random sampling of tasks from $\mathcal{D}_{\text{base}}$. Each task is composed of a *support set* $\mathcal{S} = \{\mathbf{x}_j^S, y_j^S\}_{j=1}^{NK}$ with N classes and K images per class, and a *query set* $\mathcal{Q} = \{\mathbf{x}_l^Q, y_l^Q\}_{l=1}^{Nq}$ with the same N classes and q images per class. The model is updated by learning discriminative features from $\mathcal{S}$ to correctly classify images in $\mathcal{Q}$. The same task sampling strategy is applied to the validation set $\mathcal{D}_{\text{valid}}$, helping choose the hyper-parameters of the model. Lastly, the trained model is evaluated by classifying tasks randomly sampled from $\mathcal{D}_{\text{novel}}$ and the average accuracy is usually adopted as the evaluation metric.

3.2. Architecture overview

The structure of the IIS module and the overall architecture of the IISNet are illustrated in Fig. 2. Given the support and query images, we first extract their base feature representations $\mathbf{Z}$ from a feature extractor f_θ. Then the IIS module operates on $\mathbf{Z}$ to remove target-irrelevant feature. First, the PI module extract features $G(\mathbf{E})$ containing information about the key structural information of the target, through the self-similarity tensor $\mathbf{E}$ and the dual-branch convolutional operation with two different kernel sizes. Note that the Hadamard product of the two branches is adopted to determine the positions with strong agreement between the local and global perceptions. Afterwards, the SS module conducts an inverse operation on the key target information in $G(\mathbf{E})$ to extract the irrelevant information, which is then subtracted from the base feature $\mathbf{Z}$. The latter subtraction leads to a large proportion of zero activations and few large activations highlighting the target, resulting in more sparse and discriminative feature representations. To boost the suppression effect, we propose to concatenate d IIS modules in the network and obtain $\mathbf{F}$. The default value of d is set to 3. Next, average pooling is applied to the H and W dimensions of $\mathbf{F}$. Lastly, the cosine similarity is adopted to measure the similarity between the query image $\mathbf{F}_Q$ and the support class prototypes $\bar{\mathbf{F}}_S$.

3.3. Irrelevant Information Suppression (IIS)

The IIS module takes the base representation as input and sequentially processes it with two components: the PI module to capture the key target information and the SS module to suppress the non-target information based on the output from the PI module. The operations in the IIS module are the same for support and query features, so we do not distinguish them in this section.

3.3.1. Perceptive Information (PI) module

The structure of the PI module is depicted in Fig. 3. Given the base feature representation $\mathbf{Z} \in \mathbb{R}^{H\times W\times C}$, where H, W and C are the height, width and number of channels, respectively, we compute the self-similarities, extracting the structural patterns of the image. Specifically, for each spatial position in $\mathbf{Z}$, we take its C-dimensional channel vector and calculate its Hadamard product with the channel vectors of all spatial positions in the neighborhood window of size $U \times V$, resulting in the self-similarity tensor $\mathbf{E} \in \mathbb{R}^{H\times W\times U\times V\times C}$.

To extract important and target-related structural patterns from $\mathbf{E}$, we design a dual-branch convolutional operation with two kernel sizes, capturing the global and local perceptions about the key information. The local branch is denoted as C_3 while the global branch is denoted as C_5. C_3 consists of two three-dimensional convolution kernels of size $k_3 = (1, 3, 3)$, while C_5 comprises two three-dimensional convolution kernels of sizes $k_5 = (1, 5, 5)$ and $k_1 = (1, 1, 1)$, respectively. The convolution gradually aggregates the information in the $U \times V$ dimensions, squeezing the structural patterns at different kernel scales. To reduce the computational cost and speed up processing, we decrease the channel dimensions of both branches from C to C_0 by factor r ($C_0 = C/r$) and restore them to C after the second kernel calculation. Lastly, average pooling is applied to the $U \times V$ dimensions, and as a result, the outputs $C_3(\mathbf{E})$ and $C_5(\mathbf{E})$ have the same sizes as $\mathbf{Z}$, i.e., C_3, C_5: $\mathbb{R}^{H\times W\times U\times V\times C} \to \mathbb{R}^{H\times W\times C}$.

To merge the results from the global and local branches, we propose to take the Hadamard product of the results from the two branches:

$$G(\mathbf{E}) = C_3(\mathbf{E}) \odot C_5(\mathbf{E}), \tag{1}$$

where $G(\mathbf{E}) \in \mathbb{R}^{H\times W\times C}$ is the output from the PI module and $\odot$ is the Hadamard product. This operation ensures that only positions with high values in both branches are identified as crucial to represent the target.

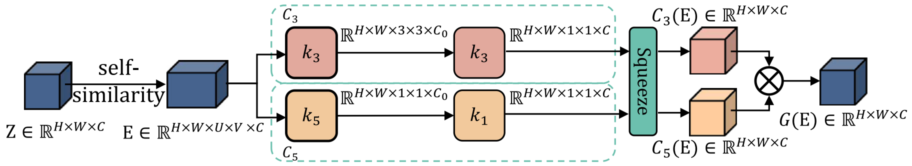

self-similarity
$Z \in \mathbb{R}^{H\times W\times C}$
$E \in \mathbb{R}^{H\times W\times U\times V\times C}$
C_3
k_3
$\mathbb{R}^{H\times W\times 3\times 3\times C_0}$
k_3
$\mathbb{R}^{H\times W\times 1\times 1\times C}$
k_5
$\mathbb{R}^{H\times W\times 1\times 1\times C_0}$
k_1
$\mathbb{R}^{H\times W\times 1\times 1\times C}$
C_5
Squeeze
$C_3(E) \in \mathbb{R}^{H\times W\times C}$
$C_5(E) \in \mathbb{R}^{H\times W\times C}$
$G(E) \in \mathbb{R}^{H\times W\times C}$

Fig. 3. The structure of the PI module. From the base representation **Z**, we calculate the self-similarity tensor **E**. Then, a dual-branch convolutional operation is applied to **E**, aggregating the local similarity information in the local branch C_3 and the global similarity information in the global branch C_5. Lastly, the Hadamard product of the two branches, $G(\mathbf{E})$, is calculated as the output of the PI module, identifying the key structural information related to the target.

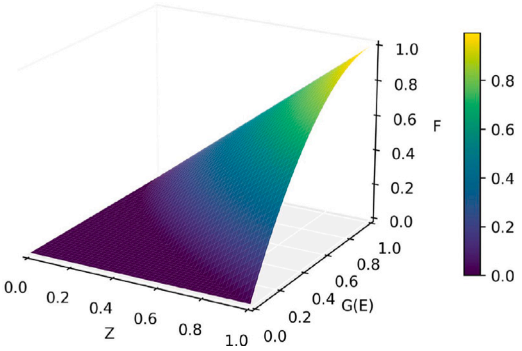


Fig. 4. The illustration of the operation of the SS module in Eq. (2). Given the base representation **Z** and the key structural information $G(\mathbf{E})$, if both **Z** and $G(\mathbf{E})$ are large, the information at the corresponding position of **F** will be preserved. However, if either **Z** or $G(\mathbf{E})$ or both of them are small, the information at that position will be suppressed, becoming insignificant for subsequent processing.

Table 1
The sparsity , i.e. the proportion of non-zero values, of **Z** and **F** after ReLU activation for five benchmark datasets. $\mathbf{F}_1$ to $\mathbf{F}_3$ denote the output of the three concatenated IIS modules, respectively.

Dataset	**Z**	$\mathbf{F}_1$	$\mathbf{F}_2$	$\mathbf{F}_3$
Aircraft	14.47%	55.22%	79.92%	90.21%
Cars	11.18%	52.39%	77.31%	86.93%
CUB-200–2001	14.39%	48.83%	74.00%	86.09%
*mini*ImageNet	14.34%	49.50%	75.29%	87.05%
*Tiered*ImageNet	12.47%	48.56%	77.63%	88.32%

3.3.2. Self-Subtraction (SS) module

Given the key areas of the target in $G(\mathbf{E})$, we calculate the irrelevant information via an inversion operation, which is then subtracted from the base representation **Z**:

$$\mathbf{F} = \mathbf{Z} - (1 - \min(G(\mathbf{E}), 1))^2, \tag{2}$$

where all operations are element-wise. By capping the maximum value of $G(\mathbf{E})$ at one, the second term with an inversion of $G(\mathbf{E})$ represents the irrelevant information. $G(\mathbf{E})$ is the output from the PI module, which should have high values for important target features. For those positions with values higher than 1, we simply keep the base representation **Z**, which we believe to be target-relevant. In **F**, only positions identified as important by both base representation and key target representation, i.e. the elements with high values in both **Z** and $G(\mathbf{E})$, are less affected and remains large. Otherwise, those only considered important by one type of representation or trivial by both representations are substantially reduced to negative and suppressed to zeros via ReLU activation. Such impact of **Z** and $G(\mathbf{E})$ on **F** is illustrated in Fig. 4.

Moreover, we calculate the sparsity of **Z**, $G(\mathbf{E})$ and **F** after ReLU activation in Table 1 for five benchmark datasets, Aircraft, Cars, CUB-200-2001, *mini*ImageNet and *Tiered*ImageNet, whose details will be revealed in Section "Experiments". It is clear that the sparsity of **F** gradually rises by passing through three concatenated IIS modules, and the sparsity of the output of the last IIS module, $\mathbf{F}_3$, is 6 to 8 times to that of the base representation **Z**.

Since the output **F** from the last IIS module is already sparse and highly discriminative, it can be directly used in a simple prototype-based metric for classification. As the metric functions usually take vectors as inputs, we average **F** in the $H \times W$ dimensions, resulting in C-dimensional feature representations.

3.4. Loss functions

As with [5,29,30], we train the IISNet end to end based on two classification losses: the anchor loss to supervise the correct classification of the query features, and the metric loss to guide the query features to be close to the corresponding support prototypes.

First, to obtain the anchor loss, we add a fully connected layer following $\mathbf{F}_Q$ and calculate

$$\mathcal{L}_\mathrm{a} = -\log \frac{\exp(\mathbf{u}_m^\top \mathbf{F}_Q + h_m)}{\sum_{m'=1}^{|\mathcal{Y}_\mathrm{train}|} \exp(\mathbf{u}_{m'}^\top \mathbf{F}_Q + h_{m'})}, \tag{3}$$

where $\mathcal{Y}_\mathrm{train}$ is the label set of the base dataset and $[\mathbf{u}_1^\top, \ldots, \mathbf{u}_{|\mathcal{Y}_\mathrm{train}|}^\top]$ and $[h_1, \ldots, h_{|\mathcal{Y}_\mathrm{train}|}]$ represent the weights and biases of the fully connected layer, respectively.

Second, the metric loss is calculated based on the cosine similarity between the query feature and the support class prototypes:

$$\mathcal{L}_\mathrm{m} = -\log \frac{\exp(\mathrm{sim}(\bar{\mathbf{F}}_S^n, \mathbf{F}_Q)/\sigma)}{\sum_{n'=1}^{N} \exp(\mathrm{sim}(\bar{\mathbf{F}}_S^{n'}, \mathbf{F}_Q)/\sigma)}, \tag{4}$$

where $\mathrm{sim}(\cdot)$ is the cosine similarity, $\bar{\mathbf{F}}_S^n$ is the prototype of the nth support class or the average of all support features in the nth class, and σ is the scale hyperparameter.

Finally, the total loss is calculated as the weighted sum of the two losses:

$$\mathcal{L}_\mathrm{total} = \mathcal{L}_\mathrm{a} + \gamma \mathcal{L}_\mathrm{m}, \tag{5}$$

where γ is the hyperparameter to control the relative contributions of the two losses.

In the test phase, only the metric loss $\mathcal{L}_\mathrm{m}$ is adopted to classify the query images.

4. Experiments

4.1. Datasets

To evaluate the effectiveness of the IIS network, we adopt three fine-grained few-shot datasets, CUB-200-2011 [31], Aircraft [32] and Stanford Cars [33], and two coarse-grained few-shot datasets, *mini*ImageNet [2] and *Tiered*ImageNet [34]. All these datasets are widely accepted benchmarks for few-shot classification [5,19].

CUB-200-2011 (CUB): 11,788 bird images in total, distributed across 200 species. We randomly partition it into 100, 50 and 50 bird species for training, validation and test, respectively. The version of pre-cropped images with human bounding boxes is used in the experiments.

Aircraft: 10,000 model images of 100 aircraft classes. We randomly split the 100 classes into 50 for training, 25 for validation and 25 for test.

Table 2
The 5-way few-shot classification accuracy on the CUB, Aircraft and Cars datasets for the ResNet-12 backbone. **Bold** represents the best performance while underscore indicates the second best one; * represents the result of our reproduction while † indicates the result from the original paper. Discrepancies from the original papers are due to a reduced number of ways for training; detailed explanations are provided in main text.

Method	CUB		Aircraft		Cars	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MatchingNet [2]*	73.02 ± 0.88	85.17 ± 0.60	65.20 ± 0.80	78.99 ± 0.55	73.32 ± 0.93	87.61 ± 0.55
ProtoNet [14]*	78.37 ± 0.21	90.18 ± 0.12	86.37 ± 0.18	93.74 ± 0.09	86.58 ± 0.17	94.91 ± 0.08
Baseline++ [39]†	64.62 ± 0.98	81.15 ± 0.61	63.51 ± 0.90	78.06 ± 0.44	67.92 ± 1.01	84.17 ± 0.58
DeepEMD [35]†	71.11 ± 0.31	86.30 ± 0.19	61.86 ± 0.30	76.17 ± 0.28	73.30 ± 0.29	88.37 ± 0.17
RENet [5]*	83.04 ± 0.42	92.68 ± 0.22	87.32 ± 0.38	93.68 ± 0.17	85.36 ± 0.31	94.31 ± 0.37
FRN [40]*	83.07 ± 0.19	92.46 ± 0.10	87.68 ± 0.17	93.95 ± 0.09	88.20 ± 0.17	95.43 ± 0.08
MixtFSL [41]†	67.87 ± 0.94	82.18 ± 0.66	60.55 ± 0.86	77.57 ± 0.69	58.15 ± 0.87	80.54 ± 0.63
AGPF [42]†	78.73 ± 0.84	89.77 ± 0.49	82.65 ± 0.89	89.25 ± 0.45	85.35 ± 0.38	95.22 ± 0.20
HelixFormer [43]*	81.66 ± 0.30	91.93 ± 0.17	75.79 ± 0.23	83.03 ± 0.16	79.40 ± 0.43	92.26 ± 0.15
FicNet [44]*	80.97 ± 0.57	93.17 ± 0.32	69.11+0.62	83.71+0.39	86.81 ± 0.47	95.36 ± 0.22
TDM [19]†	82.64 ± 0.19	92.27 ± 0.10	70.35 ± 0.17	83.36 ± 0.13	87.21 ± 0.17	95.11 ± 0.07
LCCRN [45]*	81.93 ± 0.19	92.72 ± 0.10	88.48 ± 0.17	94.61 ± 0.08	87.27 ± 0.17	96.19 ± 0.06
IDEAL [46]*	77.56 ± 0.86	88.87 ± 0.51	81.37 ± 0.92	82.51 ± 0.55	74.02 ± 0.89	89.98 ± 0.50
MCL-katz [47]*	85.97 ± 0.18	93.09 ± 0.15	87.69 ± 0.17	93.28 ± 0.08	85.04 ± 0.19	93.92 ± 0.09
Bi-FRN [48]*	85.44 ± 0.19	**94.73+0.09**	87.05 ± 0.18	93.78 ± 0.09	87.90 ± 0.16	96.34 ± 0.07
C2-Net [49]*	83.34 ± 0.42	92.20 ± 0.23	87.98 ± 0.39	93.69 ± 0.20	84.81 ± 0.42	92.61 ± 0.23
BSFA [22]*	**86.00 ± 0.41**	92.53 ± 0.23	87.85 ± 0.35	94.93 ± 0.14	88.93 ± 0.38	95.20 ± 0.20
COSOC [24]*	76.95 ± 0.23	88.02 ± 0.14	83.86 ± 0.32	92.73 ± 0.21	86.62 ± 0.37	95.67 ± 0.23
IISNet	85.53 ± 0.41	93.71 ± 0.21	90.95 ± 0.35	95.38 ± 0.14	89.76 ± 0.33	96.16 ± 0.14
IISNet$^\sharp$	85.29 ± 0.41	93.64 ± 0.21	**91.30 ± 0.33**	**95.64 ± 0.13**	**90.12 ± 0.32**	**96.41 ± 0.14**

Stanford Cars (Cars): 16,185 images of 196 car classes. We randomly divide it into 130 classes for training, 17 classes for validation and 49 classes for test.

*Mini*ImageNet: a subset of ImageNet, consisting of 60,000 images evenly distributed over 100 classes. The training, validation and test sets are composed of 64, 16, and 20 object classes, respectively.

*Tiered*ImageNet:a subset of ImageNet, made up of images distributed over 608 classes. The training, validation, and test sets are composed of 351, 97, and 160 object classes respectively.

4.2. Implementation details

We adopt ResNet-12 [35] as the backbone network, which is widely used in recent works on few-shot classification [30,36–38]. ResNet-12 takes input images of size 84 × 84 and produces the base representation with size 640 × 5 × 5. In the training phase, we resize the images to 224 × 224 and randomly crop them to obtain 84 × 84 patches. In the test phase, we resize the images to 92 × 92 and obtain 84 × 84 patches through center cropping.

In the training phase, we conduct 5-way K-shot classification with $K = 1$ or 5. In the test phase, we randomly sample 15 query samples per class in a task and evaluate the average classification accuracy with a 95% confidence interval based on 2000 randomly sampled test tasks.

In the IIS module, to reduce the computational complexity, we decrease the number of channels by a factor of r. In the experiments, we set $r = 4$ and reduce the number of channels of $C = 640$ to $C_0 = C/r = 160$. For *mini*ImageNet, CUB, Stanford Cars, and Aircraft we set the hyperparameter $\gamma = 0.53, 1.5, 1.5, 1.5$, respectively. The temperature parameter σ for $\mathcal{L}_{\mathrm{m}}$ is set to 0.2.

In this work, we adopt the stochastic gradient descent (SGD) optimizer. The IISNet is trained for 60 and 80 epochs for the 5-shot and 1-shot settings, respectively.

4.3. Comparison with state-of-the-art methods

We report the classification accuracies and their 95% confidence intervals in Table 2 for fine-grained datasets and Table 3 for the coarse-grained dataset. Twenty-two state-of-the-art (SOTA) few-shot classifiers are selected in the comparison, including two background suppression-based classifiers as our main competitors, BSFA [22] and COSOC [24]. Since these two background suppression-based classifiers both involve a module to align the target object across support and query images, we also report the results of an enhanced IISNet by adding the existing cross-correlational attention (CCA) module [5] after the three IIS layers, to match their settings. The hyperparameter of the CCA module is set to 8 for all datasets. In Tables 2 and 3, the original IIS network is denoted as IISNet while the enhanced version is denoted as IISNet$^\sharp$. Moreover, some SOTA methods are originally trained using a large number of ways, e.g. C2-Net uses 30 ways and Bi-FRN uses 15 ways, while our methods are trained using only 5 ways. To make the comparisons fairer, we reproduce their methods by using 10 ways, leading to discrepancies on accuracies from their original papers.

Obviously, for the fine-grained datasets in Table 2, IISNet and IISNet$^\sharp$ rank the first two places in most settings. In the 1-shot CUB scenario, although IISNet$^\sharp$ and IISNet are not highlighted, they are still competitive with less than 0.5% difference from the best SOTA method. For both IISNet and IISNet$^\sharp$, noticeable improvement of over or around 2.5% can be observed in the 1-shot setting for the Aircraft dataset, which suggests that this dataset is severely affected by the irrelevant information while other methods fail to eliminate it effectively. In addition, the involvement of the CCA module in IISNet$^\sharp$ makes it better than the original IISNet in most cases, but the improvement is only minor with below or around 0.3% increase in mean accuracy. More importantly, IISNet and IISNet$^\sharp$ can beat the two background suppression-based classifiers in all settings on the three fine-grained datasets, except for the 1-shot setting for CUB. We also note that the results of the contrastive learning-based method, COSOC, are well below the SOTA performances on the CUB and Aircraft datasets.

On the contrary, for the coarse-grained datasets, COSOC dominates all methods in Table 3 while IISNet$^\sharp$ is the second best. However, we further note that the superior performance of COSOC is mostly achieved by its SOC module to align the target objects across images. When the SOC module is removed and only background suppression is activated, COSOC provides substantially lower mean accuracies compared with IISNet. Thus, to fairly compare the irrelevant information suppression performance, we also present the results of COSOC and BSFA with the object alignment modules deactivated, which are listed with a superscript of $\flat$ in Table 3. Clearly, when only assessing the performance of irrelevant information suppression, IISNet is remarkably over COSOC$^\flat$ and BSFA$^\flat$. This pattern suggests that object alignment is important to correctly classify *mini*ImageNet, potentially caused by a large amount

Table 3
The 5-way few-shot classification accuracy on the *mini*ImageNet and *tiered*ImageNet dataset. **Bold** represents the best performance, while underscore indicates the second best one; * represents the result of our reproduction, while † indicates the result from the original paper.

Method	Backbone	*mini*ImageNet		*tiered*ImageNet	
		1-shot	5-shot	1-shot	5-shot
MatchingNet [2]†	ResNet-12	48.14 ± 0.72	63.49 ± 0.43	68.50 ± 0.92	80.60 ± 0.71
ProtoNet [14]†	ResNet-12	49.42 ± 0.78	68.20 ± 0.66	68.23 ± 0.23	84.03 ± 0.16
Baseline++ [39]†	ResNet-12	48.24 ± 0.75	66.43 ± 0.63	–	–
DeepEMD [35]†	ResNet-12	66.50 ± 0.80	82.41 ± 0.56	71.16 ± 0.87	85.28 ± 0.58
RENet [5]*	ResNet-12	67.60 ± 0.44	82.58 ± 0.30	71.61 ± 0.51	85.28 ± 0.35
FRN [40]*	ResNet-12	66.45 ± 0.19	82.83 ± 0.13	72.06 ± 0.22	86.69 ± 0.14
MixtFSL [41]†	ResNet-12	63.98 ± 0.79	82.04 ± 0.49	70.97 ± 1.03	86.16 ± 0.67
ATT [50]†	ResNet-12	67.64 ± 0.81	82.31 ± 0.49	69.34 ± 0.95	83.82 ± 0.63
SEMAN-G [50]†	ResNet-12	68.24 ± 0.82	83.48 ± 0.48	71.06 ± 0.92	86.02 ± 0.58
QSFormer [51]†	ResNet-12	65.24 ± 0.28	79.96 ± 0.20	72.47 ± 0.31	85.43 ± 0.22
DeepBDC [52]†	ResNet-12	60.76 ± 0.28	78.25 ± 0.20	63.03 ± 0.31	81.57 ± 0.22
Diff-ResNet [51]†	ResNet-18	68.47	80.02	–	–
UniSiam [53]†	ResNet-34	65.55 ± 0.36	83.40 ± 0.24	67.57 ± 0.39	84.12 ± 0.28
BSFA$^{\flat}$ [22]*	ResNet-12	66.54 ± 0.50	80.62 ± 0.34	70.12 ± 0.57	85.47 ± 0.32
BSFA [22]*	ResNet-12	66.67 ± 0.50	80.79 ± 0.34	70.46 ± 0.57	85.79 ± 0.32
COSOC$^{\flat}$ [24]*	ResNet-12	65.05 ± 0.06	81.16 ± 0.17	69.87 ± 0.48	85.43 ± 0.16
COSOC [24]†	ResNet-12	**69.28 ± 0.49**	**85.46 ± 0.12**	**73.57 ± 0.43**	**87.57 ± 0.10**
IISNet	ResNet-12	68.08 ± 0.44	83.54 ± 0.30	72.12 ± 0.47	86.53 ± 0.32
IISNet$^{\sharp}$	ResNet-12	68.63 ± 0.44	83.86 ± 0.30	72.73 ± 0.47	86.98 ± 0.32

Table 4
The accuracy of using one layer of the IIS module and the BAS module in BSFA as plug-ins after the ResNet-12 backbone for three few-shot classifiers. The values in the brackets are the differences between the means of the classifier with and without the plug-in module. Increases are indicated by ↑ while decreases are marked by ↓.

Method	CUB		Aircraft		Cars	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
ProtoNet [14]	78.37ś0.21	90.18ś0.12	86.37ś0.18	93.74ś0.09	86.58ś0.17	94.91ś0.08
+ **BAS**	77.09ś0.21(1.28 ↓)	89.67ś0.12(0.51 ↓)	85.16ś0.18(1.21 ↓)	93.17ś0.09(0.57 ↓)	84.25ś0.19(2.33 ↓)	94.03ś0.09(0.88 ↓)
+ **IIS**	80.53ś0.20(2.16 ↑)	90.92ś0.11(0.74 ↑)	88.10ś0.17(1.73 ↑)	94.40ś0.08(0.66 ↑)	87.43ś0.17(0.85 ↑)	95.10ś0.08(0.19 ↑)
RENet [5]	83.04ś0.42	92.38ś0.22	87.33ś0.38	93.68ś0.17	85.36ś0.31	94.31ś0.37
+ **BAS**	81.35ś0.44(1.69 ↓)	91.51ś0.12(0.87 ↓)	83.71ś0.41(3.62 ↓)	92.64ś0.19(1.04 ↓)	83.77ś0.39(1.59 ↓)	93.83ś0.18(0.48 ↓)
+ **IIS**	83.42ś0.43(0.38 ↑)	92.84ś0.22(0.46 ↑)	88.23ś0.36(0.90 ↑)	94.18ś0.16(0.50 ↑)	88.32ś0.35(2.70 ↑)	95.74ś0.15(1.43 ↑)
FRN [48]	83.07ś0.19	92.46ś0.10	87.68ś0.17	93.95ś0.09	88.20ś0.17	95.43ś0.08
+ **BAS**	81.20ś0.19(1.87 ↓)	91.51ś0.10(0.95 ↓)	87.59ś0.18(0.09 ↓)	91.03ś0.09(2.92 ↓)	84.85ś0.17(3.35 ↓)	92.03ś0.10(3.40 ↓)
+ **IIS**	84.20ś0.19(1.13 ↑)	93.25ś0.10(0.79 ↑)	88.10ś0.17(0.42 ↑)	94.40ś0.08(0.45 ↑)	88.55ś0.16(0.35 ↑)	96.42ś0.06(0.99 ↑)

of multi-object images. This is also evidenced by the increase of 0.55% of IISNet$^{\sharp}$ over IISNet.

Nonetheless, this paper aims at developing the IIS network to remove irrelevant information and generate sparse and discriminative features, whose superior performance is demonstrated on both fine-grained and coarse-grained datasets.

4.4. Evaluation of plug-in performance

The IIS module can be readily used as a plug-in module to improve the performance of few-shot classifiers. In this section, we embed one layer of IIS module after the ResNet-12 backbone to three state-of-the-art methods, ProtoNet [14], RENet [5] and FRN [48], and report the impact of the IIS module in Table 4. Additionally, we compare our performance with the background suppression module, BAS, in BSFA. The related module in COSOC is not included in the comparison here, as its performance on background suppression is the worst, especially on the fine-grained datasets.

Clearly, including one layer of IIS module produces positive improvement on all three methods under all settings. Remarkable increases in mean accuracies can be observed for the 1-shot setting of Cars and CUB. In contrast, BAS has negative impact on all methods and datasets. This indicates that BAS may only work within the BSFA method and cannot be easily adapted as a plug-in module, suggesting that the simple threshold-based strategy in BAS does not work well alone.

Table 5
The impact of the number of layers d of the IIS modules on the classification accuracy.

d	Aircraft		Cars	
	1-shot	5-shot	1-shot	5-shot
0	88.01 ± 0.37	93.81 ± 0.16	88.46 ± 0.34	93.89 ± 0.17
1	89.28 ± 0.36	94.87 ± 0.14	88.75 ± 0.34	94.94 ± 0.17
2	90.04 ± 0.35	95.48 ± 0.13	89.08 ± 0.34	96.04 ± 0.15
3	**91.30 ± 0.33**	**95.64 ± 0.13**	90.12 ± 0.32	**96.41 ± 0.12**
4	90.53 ± 0.35	95.39 ± 0.14	**90.51 ± 0.32**	95.11 ± 0.16

4.5. Ablation experiments

To gain a deeper understanding of how hyperparameters of the IIS network affect the classification, we conduct ablation studies on the number of layers of the IIS modules. Since the SS module is responsible for background suppression and its calculation requires the output from the PI module, we do not perform ablation studies on the two modules in the IIS module. In this section, the classification accuracies of the Aircraft and Cars datasets under the 5-way 5-shot setting are recorded.

4.5.1. The number of layers d of the IIS modules

In IISNet, we concatenate d layers of IIS modules to gradually remove the irrelevant information. As shown in Table 5, when d is set to 3, IISNet can achieve the best classification accuracies in most scenarios. As d increases from 0 to 3, the classification performance progressively improves. Note that when d is 0, the base

Table 6
The impact of the channel compression factor r on the classification accuracy.

r	Aircraft		Cars	
	1-shot	5-shot	1-shot	5-shot
1	91.08 ± 0.34	95.20 ± 0.14	89.77 ± 0.33	95.24 ± 0.16
2	90.71 ± 0.33	95.58 ± 0.13	89.81 ± 0.35	95.59 ± 0.16
4	**91.30 ± 0.33**	**95.64 ± 0.13**	**90.12 ± 0.32**	**96.41 ± 0.14**
5	90.42 ± 0.34	95.27 ± 0.14	89.84 ± 0.32	96.13 ± 0.15
8	90.75 ± 0.34	95.28 ± 0.14	89.55 ± 0.33	96.26 ± 0.15
10	90.39 ± 0.34	95.18 ± 0.13	89.47 ± 0.33	96.21 ± 0.15

Table 7
The impact of the global similarity aggregation C_5 and the local similarity aggregation C_3 in the PI module on the classification accuracy.

	Aircraft		Cars	
	1-shot	5-shot	1-shot	5-shot
C_3	90.23 ± 0.34	95.44 ± 0.13	89.28 ± 0.34	96.18 ± 0.14
C_5	88.71 ± 0.36	95.28 ± 0.14	88.84 ± 0.34	96.27 ± 0.14
$C_3 + C_3$	90.42 ± 0.35	95.11 ± 0.14	89.89 ± 0.33	96.30 ± 0.14
$C_5 + C_5$	90.73 ± 0.34	95.23 ± 0.14	89.71 ± 0.33	96.29 ± 0.15
$C_3 + C_5$	**91.30 ± 0.33**	**95.64 ± 0.13**	**90.12 ± 0.32**	**96.41 ± 0.14**

features are directly used in the final classification, resulting in poor performance, especially on the Aircraft dataset. With the concatenated IIS modules, the irrelevant information are suppressed and the sparse but discriminative target-related features are emphasized, leading to better performance. However, when d exceeds 3, the excessive removal of information tends to weaken useful target-related information and results in decreases in classification accuracy. From these results, we suggest to set d to 3, to suppress the irrelevant information as much as possible while retaining the key information to classify the target.

4.5.2. The channel compression factor r in the PI module

In the PI module, to reduce the computational cost, we reduce the number of channels by dividing the original number of channels 640 with a factor r. A small r increases the model's complexity and computational cost, while a large r leads to a reduction in the model's representation capacity. As show in Table 6, setting r to 4 can balance the two sides of the trade-off and provides the best classification accuracies.

4.5.3. The dual-branch operation in the PI module

In the PI module, we design a dual-branch operation with two different kernel sizes to aggregate the global and local similarity information, where the global similarity aggregation with a kernel size of 5 is denoted as C_5 while the local similarity aggregation with a kernel size of 3 is denoted as C_3. Here we test the effectiveness of such dual-branch operation. In Table 7, we first test the performance of using only one type of information with one branch; that is, the dual-branch operation is changed to a single-branch operation with either C_3 or C_5. We then present the results of using the dual-operation but with the same kernel size: $C_3 + C_3$ only uses the local similarity while $C_5 + C_5$ only uses the global similarity. Lastly, the results of using the dual-branch with two different kernel sizes are shown as $C_3 + C_5$.

The results of $C_3 + C_5$ in Table 7 are the best, suggesting that letting the global and local information work together to find the key areas of the target is beneficial. Using the same kernel size in both branches only improves marginally and sometimes is harmful compared with the single-branch operation.

4.5.4. The impact of backbone

Lastly, we evaluate the impact of the backbone structure. In Table 8, IIS is used as a plug-in module for two Vision Transformer (ViT)-based few-shot classifiers, FewTURE [54] and CPEA [54]. Since ViT methods typically output a sequence of features without a spatial structure, the convolutional kernels in IIS have to be adapted for these models. In this experiment, we replace the convolutional kernels with linear layers, using two MLPs to substitute the local compression operations C_3 and C_5, respectively. The classification accuracies in Table 8 demonstrate the superiority of the IIS module with ViT backbones.

4.6. Visualization of $\mathbf{Z}$, $G(\mathbf{E})$ and $\mathbf{F}$

To qualitatively verify the effectiveness of the proposed method, Grad-CAM [55] visualization is performed on the base feature $\mathbf{Z}$, the key structural information $G(\mathbf{E})$ and the irrelevant information-suppressed feature $\mathbf{F}$ for four examples from the Aircraft dataset in Fig. 5. It can be observed that $\mathbf{Z}$ tends to identify the whole body of the aircraft as important and also includes class-irrelevant background information, such as the grass field in the third image, while $G(\mathbf{E})$ can focus on more delicate areas due to the global and local similarity fusion. The final output from the IIS module, $\mathbf{F}$, is the most sparse one, recognizing the most discriminative areas to distinguish aircraft classes. In addition, the areas highlighted by both $\mathbf{Z}$ and $G(\mathbf{E})$ are also emphasized by $\mathbf{F}$, for example, the yellow tail of the first aircraft and the front windows in the second one.

4.7. Computational complexity

Finally, we compare the computational complexities of IISNet, IISNet$^\sharp$, BSFA and COSOC in Table 9. IISNet and IISNet$^\sharp$ are the most efficient ones in terms of FLOPs and average training time. They require relatively low computational resources to achieve state-of-the-art performance. As expected, COSOC is the most expensive one. Although BSFA has the smallest number of parameters, its FLOPs and average training time are higher than IISNet and IISNet$^\sharp$, because in their published code, the mask matrix to identify the target-related positions is transferred from GPU to CPU.

4.8. Limitations

In our IIS module, fixed convolutional kernels are used to obtain local features and global features, resulting in a fixed ratio between them. This makes IISNet emphasize on features of specific scales.

Fig. 6 shows three misclassification examples of IISNet from the 1-shot CUB dataset. The values represent the posterior probabilities of assigning a query image to a support class, with red indicating the true label and green the misclassified label. When the highlighted feature is the small beak, query images of savannah and vesper are misclassified as belonging to the brewer support class, which also emphasizes the small beak. Thus, objects with similarly sized features tend to be classified into the same category, regardless of their true labels.

5. Conclusion

This work introduces the irrelevant information suppression network (IISNet) tailored for few-shot classification, aiming to enhance the accuracy of extracting class-related information from the base feature representation. Through three concatenated IIS modules, IISNet can gradually suppress the irrelevant information and generate highly sparse and discriminative features to assist classification. The experiments on both fine-grained and coarse-grained datasets showcase the exceptional generalization capability of IISNet to new classes. Moreover, we also demonstrate the remarkable improvement of using one layer of the IIS module as a plug-in module to enhance the state-of-the-art few-shot classifiers.

IISNet uses fixed global and local sizes, which may result in an emphasis on features of specific scales. In future work, we aim to incorporate multiscale features to address this limitation. Other possible future work includes designing a target alignment module to boost the performance of IISNet on coarse-grained datasets.

Table 8
The classification accuracies on *tiered*ImageNet and *mini*Imagenet for two ViT-based few-shot classifiers.

Method	Backbone	*mini*ImageNet		*tiered*ImageNet	
		1-shot	5-shot	1-shot	5-shot
FewTURE [54]	ViT-S/16	68.02 ± 0.88	84.51 ± 0.53	72.96 ± 0.92	86.43 ± 0.67
+**IIS**	ViT-S/16	68.88 ± 0.86 (0.76 ↑)	85.07 ± 0.52 (0.56 ↑)	73.34 ± 0.89 (0.38 ↑)	86.78 ± 0.65 (0.35 ↑)
CPEA [54]	ViT-S/16	73.36 ± 0.65	88.30 ± 0.36	73.94 ± 0.71	88.45 ± 0.44
+**IIS**	ViT-S/16	73.95 ± 0.62 (0.59 ↑)	88.73 ± 0.36 (0.43 ↑)	74.07 ± 0.71 (0.13 ↑)	88.77 ± 0.43 (0.32 ↑)

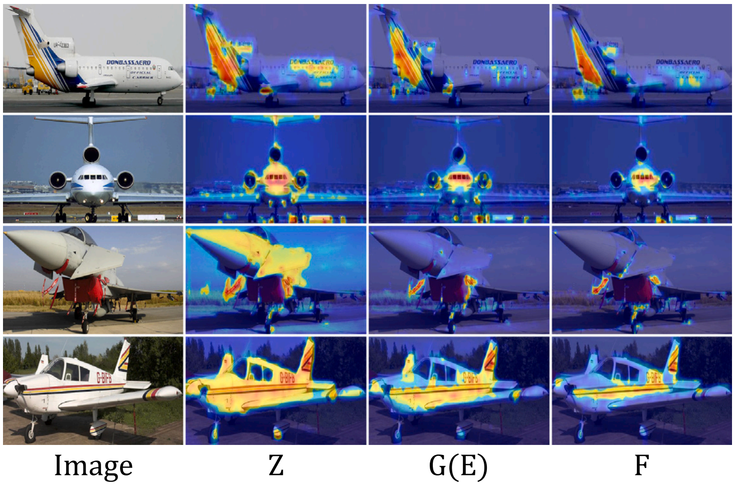
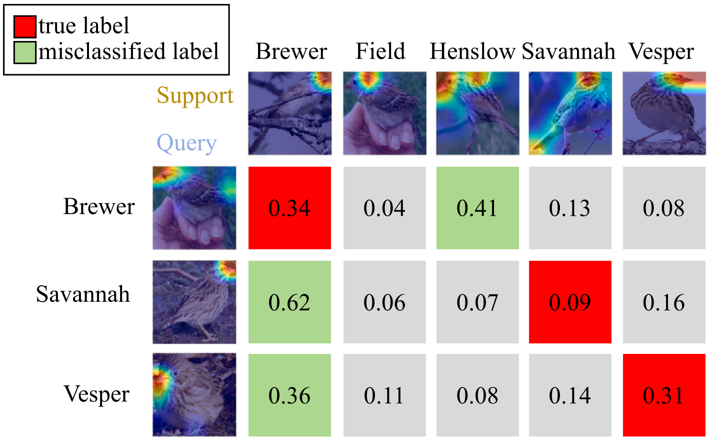


Fig. 5. Visualization of the base representation **Z**, the key structural information $G(\mathbf{E})$ and the irrelevant information-suppressed feature **F**. **Z** identifies a large part of the aircraft as important and also includes irrelevant background, while $G(\mathbf{E})$ focuses on more delicate areas of the aircraft. **F** is the most sparse feature, recognizing the most discriminative areas to distinguish aircraft classes.



	Brewer	Field	Henslow	Savannah	Vesper
Brewer	0.34	0.04	0.41	0.13	0.08
Savannah	0.62	0.06	0.07	0.09	0.16
Vesper	0.36	0.11	0.08	0.14	0.31

Fig. 6. Three misclassification examples of IISNet on the 1-shot CUB data.

Table 9
Computational complexity comparison of IISNet, IISNet$^\sharp$, BSFA and COSOC. Params is the number of parameters of the network, FLOPs is calculated for each task, and Time is the average training time of the network on three datasets with Nvidia GeForce RTX 4090.

	Params (M)	FLOPs (G)	Time (h)
BSFA	8.04	50.64	5.08
COSOC	49.9	8.78×10^5	98.4
IISNet	25.98	35.73	1.14
IISNet$^\sharp$	26.51	38.09	1.35

CRediT authorship contribution statement

Xiaoxu Li: Writing – review & editing, Supervision. **Luchen Ji:** Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization. **Rui Zhu:** Writing – review & editing, Writing – original draft, Supervision. **Zhanyu Ma:** Writing – review & editing, Supervision. **Jing-Hao Xue:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partly supported by the National Nature Science Foundation of China (Grants 62176110, 62225601, U23B2052), Key Talent Program of Gansu Province, China under Grant 2025RCXM002, Science and Technology Program of Hebei, China under Grant

SZX2020034, Hong-Liu Distinguished Young Talents Foundation of Lanzhou University of Technology, Beijing Natural Science Foundation Project No. L242025, and the Royal Society under International Exchanges Award IEC\NSFC\201071.

Data availability

Data used in this paper are publicly available.

References

[1] S. Ravi, H. Larochelle, Optimization as a model for few-shot learning, in: International Conference on Learning Representations, 2016.
[2] O. Vinyals, C. Blundell, T.P. Lillicrap, K. Kavukcuoglu, D. Wierstra, Matching networks for one shot learning, in: Neural Information Processing Systems, 2016.
[3] C. Doersch, A. Gupta, A. Zisserman, CrossTransformers: spatially-aware few-shot transfer, 2020, arXiv, arXiv:2007.11498.
[4] E. Shechtman, M. Irani, Matching local self-similarities across images and videos, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[5] D. Kang, H. Kwon, J. Min, M. Cho, Relational embedding for few-shot classification, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 8802–8813.
[6] X. Li, Z. Sun, J.-H. Xue, Z. Ma, A concise review of recent few-shot meta-learning methods, Neurocomputing 456 (2021) 463–468.
[7] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: International Conference on Machine Learning, 2017.
[8] K. Lee, S. Maji, A. Ravichandran, S. Soatto, Meta-learning with differentiable convex optimization, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 10649–10657.
[9] Y. Meng, M. Michalski, J. Huang, Y. Zhang, T. Abdelzaher, J. Han, Tuning language models as training data generators for augmentation-enhanced few-shot learning, in: International Conference on Machine Learning, PMLR, 2023, pp. 24457–24477.
[10] M. Sreenivas, S. Biswas, Similar class style augmentation for efficient cross-domain few-shot learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 4589–4597.
[11] X. Li, X. Yang, Z. Ma, J.-H. Xue, Deep metric learning for few-shot image classification: A review of recent developments, Pattern Recognit. 138 (2023) 109381.
[12] M. Dong, F. Li, Z. Li, X. Liu, PRSN: Prototype resynthesis network with cross-image semantic alignment for few-shot image classification, Pattern Recognit. 159 (2025) 111122.
[13] J. Wu, S. Wang, J. Sun, AMMD: Attentive maximum mean discrepancy for few-shot image classification, Pattern Recognit. 155 (2024) 110680.
[14] J. Snell, K. Swersky, R.S. Zemel, Prototypical networks for few-shot learning, in: Neural Information Processing Systems, 2017.
[15] S. Gidaris, N. Komodakis, Dynamic few-shot visual learning without forgetting, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4367–4375.
[16] Y. Chen, X. Wang, Z. Liu, H. Xu, T. Darrell, A new meta-baseline for few-shot learning, 2020, arXiv, arXiv:2003.04390.
[17] S.-L. Xu, F. Zhang, X.-S. Wei, J. Wang, Dual attention networks for few-shot fine-grained recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, (3) 2022, pp. 2911–2919.
[18] H. Tang, C. Yuan, Z. Li, J. Tang, Learning attention-guided pyramidal features for few-shot fine-grained recognition, Pattern Recognit. 130 (2022) 108792.
[19] S. Lee, W. Moon, J.-P. Heo, Task discrepancy maximization for fine-grained few-shot classification, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 5321–5330.
[20] W. Zhang, Y. Zhao, Y. Gao, C. Sun, Re-abstraction and perturbing support pair network for few-shot fine-grained image classification, Pattern Recognit. 148 (2024) 110158.
[21] K.Y. Xiao, L. Engstrom, A. Ilyas, A. Madry, Noise or signal: The role of image backgrounds in object recognition, in: International Conference on Learning Representations, 2021.
[22] Z. Zha, H. Tang, Y. Sun, J. Tang, Boosting few-shot fine-grained recognition with background suppression and foreground alignment, IEEE Trans. Circuits Syst. Video Technol. 33 (2022) 3947–3961.
[23] P.-Y. Chou, Y.-Y. Kao, C.-H. Lin, Fine-grained visual classification with high-temperature refinement and background suppression, 2023, arXiv preprint arXiv:2303.06442.
[24] X. Luo, L. Wei, L. Wen, J. Yang, L. Xie, Z. Xu, Q. Tian, Rectifying the shortcut learning of background for few-shot learning, Adv. Neural Inf. Process. Syst. 34 (2021) 13073–13085.
[25] Z. Chen, T. Ji, S. Zhang, F. Zhong, Noise suppression for improved few-shot learning, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2022, pp. 1900–1904.
[26] J. Geng, B. Xue, W. Jiang, Foreground–background contrastive learning for few-shot remote sensing image scene classification, IEEE Trans. Geosci. Remote Sens. 61 (2023) 1–12.
[27] C. Wang, S. Song, Q. Yang, X. Li, G. Huang, Fine-grained few shot learning with foreground object transformation, 2021, arXiv e-prints.
[28] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, M. Jagersand, Basnet: Boundary-aware salient object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7479–7489.
[29] R. Hou, H. Chang, B. Ma, S. Shan, X. Chen, Cross attention network for few-shot classification, in: Neural Information Processing Systems, 2019.
[30] B.N. Oreshkin, P.R. López, A. Lacoste, TADAM: Task dependent adaptive metric for improved few-shot learning, in: Neural Information Processing Systems, 2018.
[31] C. Wah, S. Branson, P. Welinder, P. Perona, S.J. Belongie, The caltech-UCSD birds-200-2011 dataset, 2011.
[32] S. Maji, E. Rahtu, J. Kannala, M.B. Blaschko, A. Vedaldi, Fine-grained visual classification of aircraft, 2013, arXiv, arXiv:1306.5151.
[33] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3D object representations for fine-grained categorization, in: 2013 IEEE International Conference on Computer Vision Workshops, 2013, pp. 554–561.
[34] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J.B. Tenenbaum, H. Larochelle, R.S. Zemel, Meta-learning for semi-supervised few-shot classification, 2018, arXiv preprint arXiv:1803.00676.
[35] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015, pp. 770–778.
[36] A. Ravichandran, R. Bhotika, S. Soatto, Few-shot learning with embedded class models and shot-free meta training, in: 2019 IEEE/CVF International Conference on Computer Vision, ICCV, 2019, pp. 331–339.
[37] H.-J. Ye, H. Hu, D. chuan Zhan, F. Sha, Few-shot learning via embedding adaptation with set-to-set functions, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2018, pp. 8805–8814.
[38] C. Zhang, Y. Cai, G. Lin, C. Shen, DeepEMD: Few-shot image classification with differentiable earth mover's distance and structured classifiers, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 12200–12210.
[39] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y. Wang, J.-B. Huang, A closer look at few-shot classification, 2019, arXiv, arXiv:1904.04232.
[40] D. Wertheimer, L. Tang, B. Hariharan, Few-shot classification with feature map reconstruction networks, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 8008–8017.
[41] A. Afrasiyabi, J.-F. Lalonde, C. Gagn'e, Mixture-based feature space learning for few-shot image classification, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, 2020, pp. 9021–9031.
[42] H. Tang, C. Yuan, Z. Li, J. Tang, Learning attention-guided pyramidal features for few-shot fine-grained recognition, Pattern Recognit. 130 (2022) 108792.
[43] B. Zhang, J. Yuan, B. Li, T. Chen, J. Fan, B. Shi, Learning cross-image object semantic relation in transformer for few-shot fine-grained image classification, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 2135–2144.
[44] H. Zhu, Z. Gao, J. Wang, Y. Zhou, C. Li, Few-shot fine-grained image classification via multi-frequency neighborhood and double-cross modulation, 2022, arXiv, arXiv:2207.08547.
[45] X. Li, Q. Song, J. Wu, R. Zhu, Z. Ma, J.-H. Xue, Locally-enriched cross-reconstruction for few-shot fine-grained image classification, IEEE Trans. Circuits Syst. Video Technol. (2023).
[46] Y. An, H. Xue, X. Zhao, J. Wang, From instance to metric calibration: A unified framework for open-world few-shot learning, IEEE Trans. Pattern Anal. Mach. Intell. (2023).
[47] Y. Liu, W. Zhang, C. Xiang, T. Zheng, D. Cai, Learning to affiliate: Mutual centralized learning for few-shot classification, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 14391–14400.
[48] J. Wu, D. Chang, A. Sain, X. Li, Z. Ma, J. Cao, J. Guo, Y.-Z. Song, Bi-directional feature reconstruction network for fine-grained few-shot image classification, 2022, arXiv, arXiv:2211.17161.
[49] Z.-X. Ma, Z.-D. Chen, L.-J. Zhao, Z.-C. Zhang, X. Luo, X.-S. Xu, Cross-layer and cross-sample feature optimization network for few-shot fine-grained image classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, (5) 2024, pp. 4136–4144.
[50] S. Huang, J. Ma, G. Han, S.-F. Chang, Task-adaptive negative envision for few-shot open-set recognition, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 7161–7170.
[51] X. Wang, X. Wang, B. Jiang, B. Luo, Few-shot learning meets transformer: Unified query-support transformers for few-shot classification, 2022, arXiv, arXiv:2208.12398.
[52] J. Xie, F. Long, J. Lv, Q. Wang, P. Li, Joint distribution matters: Deep Brownian distance covariance for few-shot classification, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 7962–7971.
[53] Y. Lu, L. Wen, J. Liu, Y. Liu, X. Tian, Self-supervision can be a good few-shot learner, in: European Conference on Computer Vision, 2022.

[54] F. Hao, F. He, L. Liu, F. Wu, D. Tao, J. Cheng, Class-aware patch embedding adaptation for few-shot image classification, in: 2023 IEEE/CVF International Conference on Computer Vision, ICCV, 2023, pp. 18859–18869, http://dx.doi.org/10.1109/ICCV51070.2023.01733.

[55] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.

Xiaoxu Li received the Ph.D. degree from Beijing University of Posts and Telecommunications in 2012. She is currently a Professor with the School of Computer and Communication, Lanzhou University of Technology. Her research interests include machine learning fundamentals with a focus on applications in image and video understanding. She is also a member of the China Computer Federation.

Luchen Ji is currently working toward the M.E. degree with Lanzhou University of Technology. His research interests include computer vision and few-shot learning.

Rui Zhu received the Ph.D. degree in statistics from University College London in 2017. She is a Senior Lecturer in Statistics in the Faculty of Actuarial Science and Insurance, City St George's, University of London. Her research interests include machine learning, computer vision and interdisciplinary applications in actuarial science. She serves as the Associate Editor for IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Circuits and Systems for Video Technology and Neurocomputing.

Zhanyu Ma is currently a Professor at Beijing University of Posts and Telecommunications, Beijing, China, since 2019. He received the Ph.D. degree in electrical engineering from KTH Royal Institute of Technology, Sweden, in 2011. From 2012 to 2013, he was a Postdoctoral Research Fellow with the School of Electrical Engineering, KTH. He has been an Associate Professor with the Beijing University of Posts and Telecommunications, Beijing, China, from 2014 to 2019. His research interests include pattern recognition and machine learning fundamentals with a focus on applications in computer vision, multimedia signal processing. He is a Senior Member of IEEE.

Jing-Hao Xue received the Dr.Eng. degree in signal and information processing from Tsinghua University in 1998 and the Ph.D. degree in statistics from the University of Glasgow in 2008. He is a Professor of Statistical Pattern Recognition in the Department of Statistical Science, University College London. His research interests include statistical pattern recognition, machine learning, and computer vision. He received the Best Associate Editor Award of 2021 from the IEEE Transactions on Circuits and Systems for Video Technology, and the Outstanding Associate Editor Award of 2022 from the IEEE Transactions on Neural Networks and Learning Systems.