From Instance to Metric Calibration: A Unified Framework for Open-World Few-Shot Learning

Yuexuan An, Hui Xue, Xingyu Zhao and Jing Wang

Abstract—Robust few-shot learning (RFSL), which aims to address noisy labels in few-shot learning, has recently gained considerable attention. Existing RFSL methods are based on the assumption that the noise comes from known classes (in-domain), which is inconsistent with many real-world scenarios where the noise does not belong to any known classes (out-of-domain). We refer to this more complex scenario as open-world few-shot learning (OFSL), where in-domain and out-of-domain noise simultaneously exists in few-shot datasets. To address the challenging problem, we propose a unified framework to implement comprehensive calibration from instance to metric. Specifically, we design a dual-networks structure composed of a contrastive network and a meta network to respectively extract feature-related intra-class information and enlarged inter-class variations. For instance-wise calibration, we present a novel prototype modification strategy to aggregate prototypes with intra-class and inter-class instance reweighting. For metric-wise calibration, we present a novel metric to implicitly scale the per-class prediction by fusing two spatial metrics respectively constructed by the two networks. In this way, the impact of noise in OFSL can be effectively mitigated from both feature space and label space. Extensive experiments on various OFSL settings demonstrate the robustness and superiority of our method. Our source codes is available at https://github.com/anyuexuan/IDEAL.

Index Terms—Few-Shot Learning, Self-Supervised Learning, Open-World, Label Noise, Metric Learning.

1 Introduction

Few-shot learning (FSL), which can mimic humans to recognize new classes with very few examples in a task, has drawn increasing attention due to the high cost and effort of collecting large amounts of data [1], [2]. Most existing FSL methods are based on the assumption that the label information is fully clean and intact without considering the robustness of models faced with noisy labels. In fact, noisy labels are ubiquitous due to the existence of limited knowledge and unintentional impairments in real-world environments.

Take medical image analysis for example, the training samples are tremendously difficult to acquire due to the prohibitive cost of data collection [3], [4]. In the meantime, the manual annotations may not be completely clean since the training set might involve annotation bias which could come from ambiguous medical images that confuse clinical experts, images of unknown diseases that are beyond experts' knowledge or unknown data corruption during transmission [5], [6]. As a result, even carefully annotated and curated datasets could contain mislabeled samples [7], [8], [9]. Training on the biased dataset can adversely affect the learned representation and generalization ability of models since it is far from mirroring the ground truth distribution [10], [11].

Recently, robust few-shot learning (RFSL) is proposed to attempt to address the noisy label FSL problem [12], [13]. However, these methods simply concentrate on dealing with the in-domain (ID) noise FSL problem, where noise only comes from the known domain. The more common out-of-domain (OOD) noise that comes from wider unknown domains and more often exists in the real world has not been explored in FSL. Moreover, the existing few-shot settings with label noise are of great inconsistency and confusion. RNNP [13] simply concentrates on the in-task ID noise problem that samples are mislabeled by other classes in the same task and RapNets [12] only considers out-of-task ID noise that samples are still from the same domain but from other classes except for classes in the current task.

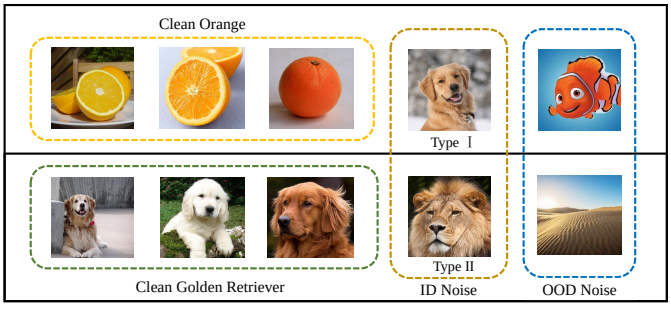
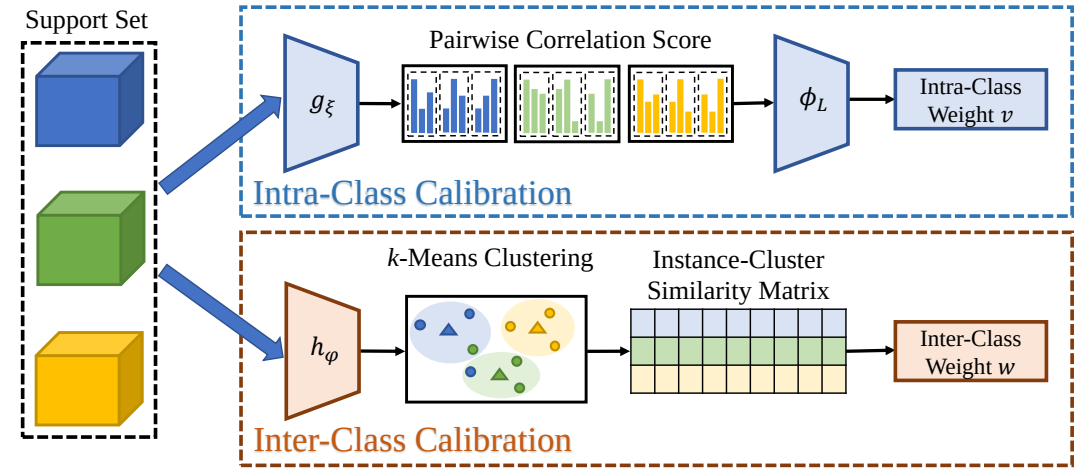


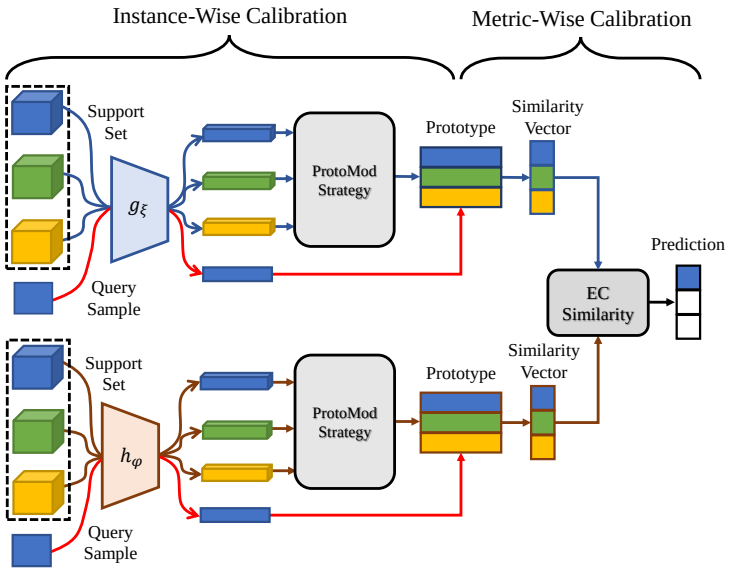
Fig. 1: An example of open-world few-shot learning.

In this paper, we unify the setting of noisy label FSL problems, which is composed of ID (both in-task ID and out-of-task ID) and OOD noise in FSL, and refer to the new and challenging scenario as open-world few-shot learning (OFSL). Fig. 1 provides an example of a 2-way 5-shot OFSL task with multiple types of ID noise and OOD noise. The known domain is from *mini*ImageNet. In the 2-way 5-shot task, we have an image dataset with two classes, orange (on the top) and golden retriever (at the bottom). The in-

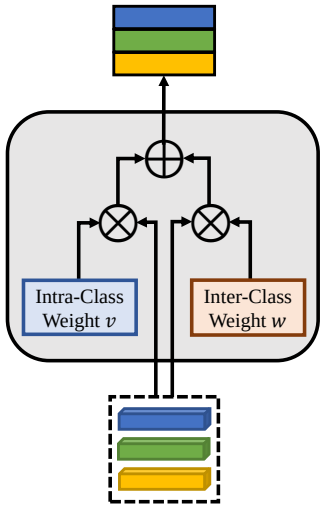
- *Yuexuan An, Hui Xue,Xingyu Zhao and Jing Wang are with the School of Computer Science and Engineering, Southeast University, Nanjing 210096, China, and the Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China. (Corresponding author: Hui Xue)*
  E-mail: {yx_an,hxue,xyzhao,wangjing91}@seu.edu.cn



(a) The generation of intra-class weight and inter-class weight



(b) Dual-networks structure for IDEAL



(c) ProtoMod strategy

Fig. 2: Overview of the proposed IDEAL (a) The generation of intra-class weight and inter-class weight. (b) A brief illustration of the dual-networks structure for IDEAL. (c) The architecture of ProtoMod strategy.

task (type I) ID noise represents that a golden retriever in the current task disturbs oranges and the out-of-task (type II) ID noise represents that a lion from other classes in *mini*ImageNet corrupts golden retriever. OOD noise represents that samples e.g., the cartoon character and the landscape from any unknown domains disturb the current distribution. Compared with RFSL, the OFSL problem contains more general FSL tasks in the real world, which is more challenging and meaningful.

The key challenges of open-world few-shot learning come from two aspects: 1) complexity: due to unknown sources of noise, distinctive characteristics are inherent in each kind of noise. Therefore, the model requires to adopt appropriate strategies to respond to each noisy situation. 2) uncertainty: due to the rarity of samples, each sample might contain important category information and potential noise is hard to detect. Therefore, the model should eliminate the uncertainty and disturbance from noisy samples.

To handle the above challenges, we propose a unified dual-networks framework for calibration from instance to metric perspectives. For the challenge of complexity, the instance-wise calibration with dual-networks distinctively tackles these different types of noise by leveraging intra-class information to discard dissimilar noise in each class, and mining inter-class information to assign in-task ID noise to its potential correct class and filter out the out-of-task ID and OOD noise. For the challenge of uncertainty, the instance-wise calibration leverages intra-class and inter-class information to respectively generate class centroids in two embedding spaces constructed by dual networks, and the metric-wise calibration fuses the two semantic spatial metrics to adaptively calibrates the metric scores. Specifically, as shown in Figure 2, our Instance-wise anD mEtric-wise cALibration (IDEAL) framework is composed of a contrastive network and a meta network. The contrastive network leverages the correlation of support samples from the same class (i.e., intra-class information) to set intra-class weights for support samples. The meta network leverages

the correlation between support samples and all classes (i.e., inter-class information) by dividing all support samples into different clusters and evaluating their contributions to each cluster to set inter-class weights of support samples. Then, each network respectively constructs modified prototypes with intra-class and inter-class weighted samples in their corresponding embedding spaces. After that, for a query sample, each network measures the similarity between the query sample and each prototype in corresponding embedding spaces. Moreover, IDEAL constructs a novel unified metric, i.e., Ensemble with Consistency (EC) similarity, by fusing multiple semantic spatial metrics to adaptively calibrate the label confidences of the query samples. EC similarity can implicitly scale the similarity score of each class with the consistency of similarity scores measured by both networks, so as to further mitigate the impact of noise.

Our main contributions can be summarized as follows:

- We identify open-world few-shot learning (OFSL) as a new challenging topic and unify the setting of noisy label FSL problems.
- We propose a unified framework with a dual-networks structure to tackle the OFSL problem, which collaboratively performs instance-wise and metric-wise calibration.
- For instance-wise calibration, we propose a prototype modification strategy by aggregating prototypes with intra-class and inter-class information extracted by the dual networks to calibrate feature space.
- For metric-wise calibration, we propose a novel similarity metric by fusing multiple semantic spatial metrics to calibrate label space. The generalization and superiority of the proposed metric are verified by detailed theoretical analyses.

The rest of the paper is organized as follows. We review related work in Section 2. The problem definition and related notation are presented in Section 3. The proposed IDEAL is detailed in Section 4. The experimental results are reported in Section 5 and the conclusion is given in Section 6.

2 Related Work

Few-shot Learning. As a challenging machine learning problem, FSL has been explored with different ideas to alleviate the overfitting problems [1], [14]. In general, the prevalent methods can be divided into four types: data augmentation method, model-based method, optimization-based method, and metric-based method. Data-augmentation methods aim to compensate for the insufficient number of available samples by generating some samples. Specifically, MetaGAN [15] synthesizes data by introducing generative adversarial networks (GANs) and distribution calibration [11] synthesizes data by estimating the distribution of novel classes with the statistics information from base classes. The optimization-based method is to find a set of model parameters that can be adapted with a few steps of gradient descent to individual tasks [16], [17], [18]. MAML [17] is a representative of the gradient-based model, which advocates learning a suitable initialization of model parameters from base classes, and transfers these parameters to novel classes in a few gradient steps. The idea of the metric-based method is to leverage similarity information between samples to classify query samples to corresponding classes [19], [20], [21], [22]. Prototypical networks [20] aim to learn the prototypes as class centroids and classify samples by comparing the distance to each prototype. Since the prototypical networks were proposed, many methods [23], [24] have been devoted to modifying the prototype to maintain the robustness of FSL. Recently, ReProto [24] attempts to restore prototypes in one-shot problems by learning a variance of the prototype of one example and the prototype of all examples of the class. ProtoComNet [23] extracts features from WordNet to complement the semantic information of prototypes. However, due to the existence of unknown label noise, these methods might easily learn a biased distribution, which inevitably causes severe degradation of performances.

Self-Supervised Learning. Self-supervised learning learns a feature representation by constructing pseudo-labels for annotation-free pretext tasks and learning to predict them [25], [26]. One of the most general pretext tasks is designed by contrastive loss, which aims to learn a feature representation by bringing the positive pairs closer, and spreading negative pairs apart. Many studies dedicate to contrastive learning and achieve promising performances [27], [28], [29]. MoCo [27] trains a representation encoder by matching an encoded query sample to an encoded key in a memory bank. SimCLR [28] uses the normalized temperature-scaled cross-entropy loss as contrast loss. SwAV [30] introduces a scalable online clustering loss without computing pairwise comparisons. BYOL [31] abandons the negative pairs and only relies on positive pairs to learn a feature representation. SimSiam [29] removes the momentum encoder and designs a simpler one based on BYOL.

However, most existing contrastive learning methods are dependent on the availability of large training samples and unsuitable for few-shot scenarios. Therefore, some papers leverage auxiliary information to improve the performance of self-supervised learning [32], [33], [34]. In this paper, inspired by our preliminary work CSS [32], we leverage prior knowledge learned by the supervised network without auxiliary data to reduce the representation bias of the contrastive model for few-shot settings.

Label Noise. Label noise is ubiquitous and can not be ignored in the real world [35]. To address this problem, different approaches have been proposed [36], [37]. One line of these approaches is instance reweighting, which usually selects a certain number of small-loss training samples as true-labeled samples [38], [39], [40]. MentorNet [38] pretrains a sample selecting network, and uses the network to select clean samples for model training. Co-teaching trains two networks and each network selects clean samples for the other [39]. Meta-Weight-Net adaptively learns an explicit weighting function from data [40]. The other line of approaches is label refurbishing, which obtains a refurbished label to reduce the adverse effect of false labels [10], [41], [42]. [43] fits a two-component beta mixture model to model clean and noisy samples. [10] and [42] update the target model with soft labels provided by a meta model with bilevel learning optimization. However, in FSL scenarios, the above methods are difficult to train with limited examples. Therefore, it is urgent to study a new method to deal with label noise in FSL.

Compared to FSL and label noise, FSL with label noise has been hardly explored. [13] generates a new feature of samples with features of other samples in the same class to eliminate the effect of In-task (type I) noise. [12] considers the attention module to reweight each sample from one class according to its contribution to its corresponding class for out-of-task (type II) noise. However, their settings for noisy label few-shot problems are inconsistent and less comprehensive. Moreover, their methods only consider intra-class information while ignoring the inter-class information which is more valuable for classifying the in-task ID noise to its potential correct class and filtering out the isolated noise from unknown classes in both the known domain (out-of-task ID) or unknown domains (OOD).

In contrast to the aforementioned literature, we propose a more general topic, that is, open-world few-shot learning. To the best of our knowledge, it is the first work to consider both ID noise and OOD noise in few-shot learning. To address the problem, we propose a unified IDEAL framework by combining the advantages of the above two lines to clean the open-world noise, i.e., instance-wise and metric-wise calibration.

3 Problem Setting

3.1 Few-Shot Learning

For N-way K-shot problems, we are given two datasets: a training set with a few labeled samples which can be called support set $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N\times K}$ and a test set consisting of unlabeled samples which can be called query set $Q = \{(\boldsymbol{x}_i, y_i)\}_{i=NK+1}^{NK+M}$. x_i denotes sample, $y_i \in C_{novel}$ is its corresponding label, N is the number of classes in S, K is the number of samples extracted for each class, and M is the number of samples in Q. Each FSL problem that estimates the class of samples in the query set with the support set can be regarded as a task. Meanwhile, an auxiliary dataset with abundant labeled samples $D_b = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{T}$ which is also divided into support sets S_b and query sets Q_b is used for meta-training, where $y_i \in C_{base}$ and $C_{base} \bigcap C_{novel}{=}\emptyset$. This strategy can be regarded as a rehearsal to train a model on the data from the base classes C_{base} so that the model can generalize well to the novel classes C_{novel}.

3.2 Open-World Few-Shot Learning

In this paper, we consider an open-world few-shot learning (OFSL) setting where labels of support sets are unreliable and some samples are randomly polluted by noisy samples of other classes especially the classes that are unseen during the whole learning process. The existing noise conditions can be divided into three types: type I ID noise, type II ID noise and OOD noise. Type I ID noise is from other classes in the current task [36], which is an extreme situation. Due to sparse data and simultaneous confusion of two classes, type I ID noise severely disturbs both distributions of the two classes and inevitably degrades the performance of learned models. Type II ID noise is from other classes in the known domain except for classes in the current task, which disturbs the distribution of the current corrupted class and causes semantic shift [44]. OOD noise is from wilder external domains, which is more common in real world. Due to the introduction of unknown domains, semantic shift and covariate shift [45] might both exist in OOD noise, which leads to unforeseen shift of feature and label distribution. Formally, we define the OFSL problem as follows.

Problem 1 (OFSL)**.** *Consider a known domain $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is a d-dimensional feature space while $\mathcal{Y}$ is a label set. For an OFSL problem with N-way K-shot tasks, given a dataset which contains a support set (training set) $S = \{(\boldsymbol{x}_i, y_i)|y_i \in \mathcal{Y}'\}_{i=1}^{N\times K}$ and a query set (test set) $Q = \{(\boldsymbol{x}_i, y_i)|y_i \in \mathcal{Y}'\}_{i=NK+1}^{NK+M}$, where $\mathcal{Y}'$ is a subset sampled from $\mathcal{Y}$ and contains N classes. For the OFSL dataset, S might not be from a clean distribution and there might be different rates of label noise in support samples. The undesirable label noise could be from ID noise and OOD noise. Each OFSL task estimates labels of query samples with the support set.*

In OFSL, we consider three types of label noise:

Definition 1 (Type I ID Noise)**.** *Given an OFSL task, for the specific class j, the type I ID noise is defined to be the noise generated from the label subset $\mathcal{Y}' - \{j\}$, i.e., the label set that contains other $N-1$ classes sampled in the current task.*

Definition 2 (Type II ID Noise)**.** *Given an OFSL task, the type II ID noise is defined to be the noise generated from the label subset $\mathcal{Y} - \mathcal{Y}'$, i.e., the label set that contains other classes in the known domain except for the N classes sampled in the current task.*

Definition 3 (OOD Noise)**.** *Given an OFSL task, the OOD noise is defined to be the noise generated from the label set $\mathcal{Y}^*$ which is from other domain $\mathcal{Z}^* = \mathcal{X}^* \times \mathcal{Y}^*$ and $\mathcal{Z}^* \bigcap \mathcal{Z}{=}\emptyset$.*

In order to mimic the training process with noisy labeled datasets, we also take the rehearsal strategy and deliberately add label noise to the support set of D_b to boost the robustness of our model.

3.3 Compared with Traditional Open World Recognition

Contrary to the closed world, the open world is posted to port the learning system from controlled lab environments to the real world. Therefore, the learning system should face unknown, uncontrolled environments, where unknown classes or even unknown domains might exist in the training set (support set) or test set (query set). The settings proposed in [46], [47] and [48] emphasize the discrimination ability of the model for query samples (test samples) of unknown classes, while our paper mainly focuses on the scenario that there are open-world samples, i.e. noisy samples from known classes, unknown classes in the known domain or unknown domains in the support set which could severely damage the robustness of models in the open world.

The detailed differences between the settings of open-world few-shot learning and traditional open-world recognition are as follows: 1) open-world few-shot learning tries to handle unknown classes in support samples, while traditional open-world recognition focuses on unknown classes in query samples; 2) open-world few-shot learning considers noise from not only the known domain but also unknown domains, while traditional open-world recognition only considers query samples from unknown classes regardless of domain variances; 3) open-world few-shot

learning aims to maintain the robustness of models in the open world while traditional open-world recognition tries to enhance the identification ability of models for unknown classes.

4 METHOD

4.1 An Overview of the Proposed Framework

To address the OFSL problem, we propose a dual-networks framework for instance-wise calibration and metric-wise calibration. Fig. 2(b) shows a brief illustration of the dual-networks framework which is composed of the contrastive network g_ξ and the meta network h_φ. First, the contrastive network extracts intra-class weights of support samples and the meta network extracts inter-class weights of support samples in Fig. 2(a). Then, each network respectively adopts a novel prototype modification (ProtoMod) strategy with intra-class and inter-class weighted samples to construct prototypes in Fig. 2(c). Finally, IDEAL uses the proposed EC similarity to adaptively calibrate the similarity score between a query and each prototype in the two semantic spaces for better prediction.

4.2 Pre-Training

In FSL, the samples used for training are limited and thus it is necessary to find information irrelevant to the class from limited samples to overcome the impact of noise. Therefore, we first pre-train the contrastive network by contrastive self-supervised learning and then train the meta network and finetune the contrastive network. In our paper, inspired by our preliminary work CSS [32], we leverage supervised information as prior knowledge to condition the feature manifold of the contrastive network so as to be resilient to few-shot settings and reduce representation bias. Therefore, the pre-training of the contrastive network uses the conditional loss in CSS [32] and contrastive loss in SimSiam [29] as the final pre-training loss (more detail can be found in Appendix A). The training of meta network is described in detail in section 4.5.

4.3 Instance-Wise Calibration

Inspired by [23] and [49], the feature extractor trained by a self-supervised method forms loose but feature-related clusters and that trained by a supervised method can form sharp but isolated clusters. Therefore, we adopt two kinds of network, contrastive network and meta network, to respectively extract information for prototype modification. The contrastive network extracts the feature-related intra-class information to discard dissimilar noise in each class. Meanwhile, the meta network extracts sharp and isolated inter-class information to assign type I ID noise to its potential correct class and filter out type II ID and OOD noise. Therefore, the combination of the disentangled information might collaboratively benefit each other.

4.3.1 Intra-Class Calibration

We use the degree of correlation between samples and their class extracted by the contrastive network to determine the intra-class confidence weight for each sample, so as to better distinguish true-labeled samples from noise. For every two same labeled support samples $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$, we calculate their pairwise correlation score to compare their similarity

$$cor(\boldsymbol{x}_a, \boldsymbol{x}_b) = \cos\left(g_\xi\left(\boldsymbol{x}_a\right), g_\xi\left(\boldsymbol{x}_b\right)\right). \tag{1}$$

For a N-way K-shot task, for each sample, we construct an intra-class correlation vector composed of all correlation scores between the sample and the K same labeled samples, which represents the feature correlation between the sample and its corresponding class. Then, for each class, the correlation matrix $\boldsymbol{Z}_n \in R^{K\times K}$ of class n can be obtained, which reveals the whole correlation between samples labeled with class n and the class n. Our goal is to model the interconnection between congener correlation features and the potential correlations between samples. Each correlation feature $\boldsymbol{Z}_n$ contains the relevant information between $\boldsymbol{Z}_n^{(i)}$ and other $K-1$ congener support data, which simultaneously scatters across other $K-1$ correlation features. The properties of the interconnection make it more reasonable to take into consideration the full context of the congener correlation features to generate the intra-class weight. To model the relationship between samples, we adopt two types of encoders:

1) BiLSTM. Inspired by [19] and [12], we use BiLSTM to encode the relationship between samples. The key component which allowed for more expressive models is the introduction of content-based attention in BiLSTM. Similar to [19], we leverage the content-based attention capability of BiLSTM to encode support samples in the same class. Specifically, we regard each correlation matrix as a sequence and use the BiLSTM parameterized by ϕ_L to encode the correlation sequence. The forward pass can be given by

$$h_n^{(i)}, e_n^{(i)} = BiLSTM\left(\boldsymbol{Z}_n^{(i)}, h_n^{(i-1)}, e_n^{(i-1)} \,|\phi_L\right), \tag{2}$$

where h and e denote the hidden and cell state vectors inside BiLSTM respectively and $\boldsymbol{Z}_n^{(i)}$ is the i-th feature vector in $\boldsymbol{Z}_n$. Similarly, we can attain the backward pass $l_n^{(i)}$, $o_n^{(i)}$. We use a multi-layer neural network followed by a K-way softmax layer ρ to achieve the intra-class weight vector $\boldsymbol{V}_n$ for samples labeled with class n:

$$\boldsymbol{V}_n = \rho\left(\mathcal{C}\left(\mathcal{C}\left(h_n^{(1)}, l_n^{(1)}\right), \cdots, \mathcal{C}\left(h_n^{(K)}, l_n^{(K)}\right)\right)\right), \tag{3}$$

where $\mathcal{C}$ denotes the concatenation operation, $n = 1, \cdots, N$. We denote that $v_n^{(k)}$ is the k-element in $\boldsymbol{V}_n$, $k = 1, \cdots, K$.

It is worth noting that BiLSTM is used to extract the sequence information, which is not ordered agnostic. However, the key module in BiLSTM which mainly affects the final performance is the content-based attention. The content-based attention encodes the correlations of paired samples in the same class to extract the importance information of each sample to its category. Besides, we propose an order-agnostic encoder for intra-class calibration, the transformer layer to extract the similarity information of support samples in the same class.

2) Transformer. The self-attention mechanism of transformer is an effective way to leverage the similarities between support samples and naturally weigh them when aggregating them into prototypes. Specifically, we use the transformer layer proposed in [50]. Since shot and class

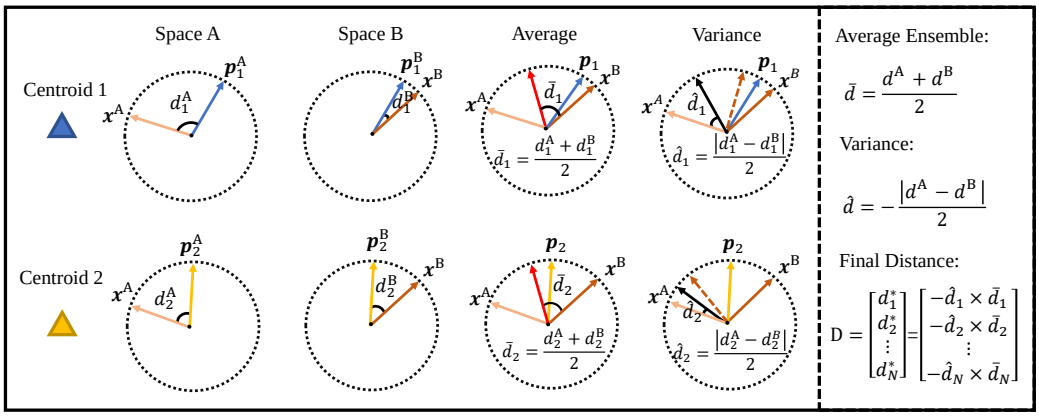


Fig. 3: Ideal illustration of Ensemble with Consistency principle.

order are both typically arbitrary in FSL, we direct input the correlation matrix $\boldsymbol{Z}_n \in R^{K\times K}$ into the transformer layer without positional encoding:

$$o_n = Transformer\left(\boldsymbol{Z}_n \,|\phi_T\right). \tag{4}$$

where ϕ_T is the parameters of the transformer layer. Then a multi-layer neural network followed by a K-way softmax layer is used to achieve the intra-class weight vector $\boldsymbol{V}_n$ for samples labeled with class n:

$$\boldsymbol{V}_n = \rho\left(o_n\right). \tag{5}$$

where ρ is the combination of the multi-layer neural networks and the softmax layer.

4.3.2 Inter-Class Calibration

The feature extractor trained by a meta network can form compact clusters that can attain enlarged inter-class variance to identify type I ID noise to their potential correct class and filter out noise from unknown classes (type II ID noise or OOD noise). Therefore, we use meta network as a feature extractor in the inter-class calibration.

We infer that samples may be mislabeled by other classes in a task. Intuitively, a sample with low similarity to a class centroid should be given with a small weight for each class and a sample with low similarity to all class centroids might be noise. Therefore, the generation of reasonable prototypes should consider both the congeneric support samples and latent congeneric samples while mislabeled by other classes. In the inter-class calibration, we adopt k-Means clustering and perform clustering on all support samples. For a N-way K-shot problem, we set the number of clusters to N and use the class prototypes of support samples as initial cluster centers. Then, k-Means clustering is used to update cluster centers until they are converged. We calculate the similarity matrix $\boldsymbol{S} \in \mathbb{R}^{N\times NK}$ between final cluster centers and all support samples with cosine similarity. After that, a softmax operation with temperature scaling is applied to obtain the inter-class weight w_n^t of a support sample $\boldsymbol{x}_t$ for the class n. For a support sample $\boldsymbol{x}_t (t = 1, \cdots, NK)$, its inter-class weight for class n is:

$$w_n^{(t)} = \frac{e^{\boldsymbol{S}_n^{(t)}/\tau}}{\sum_{j=1}^{N\times K} e^{\boldsymbol{S}_n^{(j)}/\tau}}, \tag{6}$$

where $\boldsymbol{S}_n^{(t)}$ is a similarity score representing the contribution of sample $\boldsymbol{x}_t$ to class n, and τ is a temperature parameter.

4.3.3 Prototype Modification

In IDEAL, the contrastive network and the meta network respectively learn modified prototypes with intra-class and inter-class weighted support samples in their corresponding embedding spaces. The modified prototype of the class n in the space constructed by the contrastive network is given by:

$$\boldsymbol{p}_n = \alpha \sum\nolimits_{k=1}^{K} v_n^{(k)} g_\xi(\boldsymbol{x}_n^{(k)}) + \beta \sum\nolimits_{t=1}^{N\times K} w_n^{(t)} g_\xi\left(\boldsymbol{x}_t\right), \tag{7}$$

where $\alpha + \beta = 1$, $v_n^{(k)}$ is intra-class weight and $w_n^{(t)}$is inter-class weight for class n. Note that $\boldsymbol{x}_n^{(k)}$ is the k-th sample of the class n in a support set $(k = 1, \cdots, K)$ while $\boldsymbol{x}_t$ is the t-th sample in the support set $(t = 1, \cdots, NK)$. Similarly, the modified prototype of the class n formed by meta network is given by:

$$\boldsymbol{p}_n^{'} = \alpha \sum\nolimits_{k=1}^{K} v_n^{(k)} h_\varphi(\boldsymbol{x}_n^{(k)}) + \beta \sum\nolimits_{t=1}^{N\times K} w_n^{(t)} h_\varphi\left(\boldsymbol{x}_t\right). \tag{8}$$

4.4 Metric-Wise Calibration

Inspired by the communication theory that robustness is gained by adding variety in different levels of the transmission encoding [51], [52], dual networks jointly predict the similarity score between a query sample and each modified prototype. Moreover, to mitigate the impact of wrong predictions of two networks, we introduce a metric-wise calibration on top of the two networks and propose an Ensemble with Consistency (EC) principle to adaptively calibrate the prediction score. The basic idea is that, if similarity scores predicted by two networks are more consistent, the confidence of corresponding label prediction in the latent ensemble space should be magnified.

4.4.1 Motivation of Ensemble with Consistency Principle

To visualize this idea, one sample and two class centroids are mapped to two latent 2-dimensional embedding spaces (spaces A and B) as illustrated in Fig. 3. In this paper, a prototype is noted as a representative of a class centroid in an embedding space. For a given sample $\boldsymbol{x}$ and prototype $\boldsymbol{p}_n$

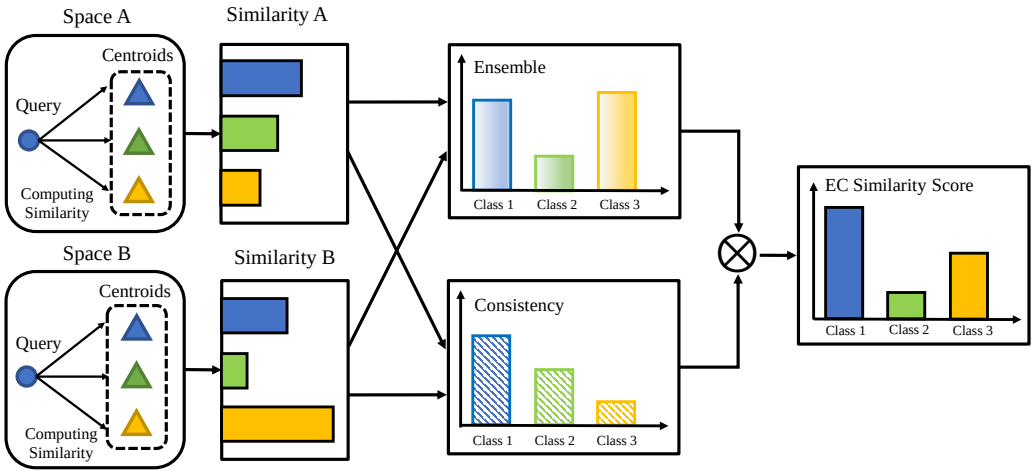


Fig. 4: Visualization of computation of Ensemble with Consistency similarity score.

of class n, $\boldsymbol{x}^i$ and $\boldsymbol{p}_n^i$ are their corresponding representations in space i, In the top line, d_1^{A} and d_1^{B} are distances between the sample $\boldsymbol{x}$ and the prototype $\boldsymbol{p}_1$ in two spaces. Similarly, d_2^{A} and d_2^{B} are distances between $\boldsymbol{x}$ and $\boldsymbol{p}_2$. From the third column pictures, the average distances in two representation spaces are the same $\bar{d}_1 = \bar{d}_2$. However, from the fourth column pictures, the variance (negative consistency) of distances between $\boldsymbol{x}$ and $\boldsymbol{p}_2$ in two spaces is smaller than $\boldsymbol{p}_1$, $\hat{d}_1 > \hat{d}_2$, indicating that its prediction of average distances might be more consistent and convincing. Therefore, the consistency of distances in two spaces can be regarded as confidence for the average ensemble distance to calibrate the label confidence prediction of two spaces.

4.4.2 Ensemble with Consistency Simiarity

The similarity score between a pair of instances can be set as the negative distance. We use the consistency of similarity scores in two spaces as a class scaling factor for the ensemble similarity results to achieve a more unbias metric as shown in Fig. 4. Our Ensemble with Consistency (EC) similarity score can be used to learn a joint similarity score from multiple metrics in different spaces considering both the diversity and consistency of different metrics. The definition of EC similarity score is given as follows:

Definition 4 (EC Similarity Score)**.** *Given a dataset containing N instances $\mathcal{D} = \{\boldsymbol{z}_i = (\boldsymbol{x}_i, y_i)\}_{i=1}^N$, where instance is sampled from a d-dimensional feature space $\mathcal{X}$ while label value y_i is generated from a scalar label space $\mathcal{Y}$, and $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. $\phi_1(\cdot)$ and $\phi_2(\cdot)$ are two different functions mapping from $\mathcal{X}$ to a high-dimensional space. For two instances $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, $\phi_1(\boldsymbol{x}_i)$ and $\phi_2(\boldsymbol{x}_i)$ are feature representations of $\boldsymbol{x}_i$ in two spaces, $\phi_1(\boldsymbol{x}_j)$ and $\phi_2(\boldsymbol{x}_j)$ are representations of $\boldsymbol{x}_j$ in two different spaces, the EC similarity can be defined according to Ensemble with Consistency of similarities in two spaces*

$$\mathcal{S}_{EC}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \hat{\mathcal{C}}(\boldsymbol{x}_i, \boldsymbol{x}_j) \times \hat{\mathcal{E}}(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{9}$$

Here,

$$\hat{\mathcal{C}}\left(x_i, x_j\right) = C\left(d(\phi_1(\boldsymbol{x}_i), \phi_1(\boldsymbol{x}_j)), d(\phi_2(\boldsymbol{x}_i), \phi_2(\boldsymbol{x}_j))\right) \tag{10}$$

is used to evaluate the confidence of similarity predictions for two instances and scale the similarity prediction score, $d(\cdot,\cdot)$ is a similarity metric in the same space, $C(\cdot,\cdot)$ indicates the consistency of similarities measured in two spaces.

$$\hat{\mathcal{E}}(x_i, x_j) = E\left(d(\phi_1(\boldsymbol{x}_i), \phi_1(\boldsymbol{x}_j)), d(\phi_2(\boldsymbol{x}_i), \phi_2(\boldsymbol{x}_j))\right) \tag{11}$$

is used to totally evaluate the similarity prediction for two instances, $E(\cdot,\cdot)$ indicates a unified ensemble evaluation between two representations in two spaces.

Theorem 1. *The average cosine similarity can be induced as the EC similarity if existing two spaces.*

Proof. For a paired sample $(\boldsymbol{x}_i, \boldsymbol{x}_j)$, the cosine similarity scores between the pair in two spaces are

$$\cos\alpha_1 = \cos\left(\phi_1\left(\boldsymbol{x}_i\right), \phi_1\left(\boldsymbol{x}_j\right)\right) = \frac{\phi_1\left(\boldsymbol{x}_i\right) \cdot \phi_1\left(\boldsymbol{x}_j\right)}{\|\phi_1\left(\boldsymbol{x}_i\right)\|_2 \|\phi_1\left(\boldsymbol{x}_j\right)\|_2}. \tag{12}$$

$$\cos\alpha_2 = \cos\left(\phi_2\left(\boldsymbol{x}_i\right), \phi_2\left(\boldsymbol{x}_j\right)\right) = \frac{\phi_1\left(\boldsymbol{x}_i\right) \cdot \phi_1\left(\boldsymbol{x}_j\right)}{\|\phi_2\left(\boldsymbol{x}_i\right)\|_2 \|\phi_2\left(\boldsymbol{x}_j\right)\|_2}. \tag{13}$$

We define the similarity score $d(\cdot,\cdot) = \arccos(\cdot,\cdot)$. Then, $d\left(\phi_1\left(\boldsymbol{x}_i\right), \phi_1\left(\boldsymbol{x}_j\right)\right) = \arccos\left(\phi_1\left(\boldsymbol{x}_i\right), \phi_1\left(\boldsymbol{x}_i\right)\right) = \alpha_1$, $d(\phi_2\left(\boldsymbol{x}_i\right), \phi_2\left(\boldsymbol{x}_i\right)) = \arccos(\phi_2\left(\boldsymbol{x}_i\right), \phi_2\left(\boldsymbol{x}_j\right)) = \alpha_2$ and $\alpha_1, \alpha_2 \in (0, \pi)$. According to the definition of EC similarity score, $C : \mathcal{R} \times \mathcal{R} \to \mathcal{R}$ and $E : \mathcal{R} \times \mathcal{R} \to \mathcal{R}$. $C(d_1, d_2)$ is denoted as $\cos(\frac{1}{2}(d_1 - d_2))$, $E(d_1, d_2)$ is set as $\cos(\frac{1}{2}(d_1 + d_2))$ where d_1 and d_2 are distances between two instances.

$$\begin{aligned}
& \cos\alpha_1 + \cos\alpha_2 \\
= \;& 2\cos\left(\frac{1}{2}\left(\alpha_1 - \alpha_2\right)\right) \times \cos\left(\frac{1}{2}\left(\alpha_1 + \alpha_2\right)\right) \\
= \;& 2\cos\left(\frac{1}{2}\left(d\left(\phi_1\left(\boldsymbol{x}_i\right), \phi_1\left(\boldsymbol{x}_j\right)\right) - d\left(\phi_2\left(\boldsymbol{x}_i\right), \phi_2\left(\boldsymbol{x}_j\right)\right)\right)\right) \\
& \times \cos\left(\frac{1}{2}\left(d\left(\phi_1\left(\boldsymbol{x}_i\right), \phi_1\left(\boldsymbol{x}_j\right)\right) + d\left(\phi_2\left(\boldsymbol{x}_i\right), \phi_2\left(\boldsymbol{x}_j\right)\right)\right)\right) \\
\propto \;& C\left(d\left(\phi_1\left(\boldsymbol{x}_i\right), \phi_1\left(\boldsymbol{x}_j\right)\right), d\left(\phi_2\left(\boldsymbol{x}_i\right), \phi_2\left(\boldsymbol{x}_j\right)\right)\right) \\
& \times E\left(d\left(\phi_1\left(\boldsymbol{x}_i\right), \phi_1\left(\boldsymbol{x}_j\right)\right), d\left(\phi_2\left(\boldsymbol{x}_i\right), \phi_2\left(\boldsymbol{x}_j\right)\right)\right) \\
= \;& C\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) E\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \\
= \;& \mathcal{S}_{EC}(\boldsymbol{x}_i, \boldsymbol{x}_j)
\end{aligned} \tag{14}$$

where the first equation can be proved according to Sum-to-product Identities [53]. □

In our paper, by defining the instance $\boldsymbol{x}$, we attain the corresponding representations $g_\xi(\boldsymbol{x})$ and $h_\varphi(\boldsymbol{x})$ of $\boldsymbol{x}$ in embedding spaces constructed by contrastive network and meta network respectively. Assume existing prototypes $\boldsymbol{p}_n$, $\boldsymbol{p}_n^{'}$ are corresponding class centriod $\boldsymbol{c}_n$ of class n in spaces A and B. Then, according to **Theorem 1**, the average cosine similarity can be induced as the EC similarity

$$\mathcal{S}_{EC}(\boldsymbol{x}, \boldsymbol{c}_n) = \cos\left(g_\xi\left(\boldsymbol{x}\right), \boldsymbol{p}_n\right) + \cos(h_\varphi\left(\boldsymbol{x}\right), \boldsymbol{p}_n^{'}). \quad (15)$$

4.4.3 Analyses of EC Similarity

EC Similarity for IDEAL. Based on the EC Similarity, IDEAL fuses multiple semantic spatial metrics to adaptively calibrate the label confidences of the query samples. EC similarity can implicitly scale the similarity score of each class with the consistency of similarity scores measured by both networks, so as to further mitigate the impact of noise. Specifically, in IDEAL, EC similarity integrates the consistency of the latent sample information in different decision to enhance the robustness of the model. For the type I ID noise, EC similarity leverages the decision information of different spaces to further confirm the corresponding relationship between the samples that may be mislabeled and the potential correct classes. For the type II ID noise and OOD noise, EC similarity can leverage the consistency of different spaces to relieve the influence brought by possible semantic shift and covariate shift. The experimental validity of metric-wise calibration is illustrated in Section Table 5.

Theoretical Analysis. We also give a theoretical analysis of the EC similarity under the metric learning framework [54] below.

Given a dataset containing N instances $\mathcal{D} = \{\boldsymbol{z}_i = (\boldsymbol{x}_i, y_i)\}_{i=1}^N$, where instance is sampled from a d-dimensional feature space $\mathcal{X}$ while label value y_i is generated from a scalar label space $\mathcal{Y}$, and $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. Denote M_1 and M_2 as the similarity score matrices in different spaces according to the mapping functions ϕ_ξ and ϕ_φ. Then the empirical objective of the multiple metrics is

$$\begin{aligned}&\min_{\xi,\varphi} \frac{1}{N\left(N-1\right)} \sum_{i=1}^{N} \sum_{j \neq i} \ell\left(q_{ij}\left(\mathcal{S}_{\xi,\varphi}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)\right)\right) \\ &+ \lambda\left(\|M_1\|_F^2 + \|M_2\|_F^2\right) \\ =&\min_{\xi,\varphi} \hat{\mathcal{R}}_N\left(\mathcal{D}\right) + \lambda\left(\|M_1\|_F^2 + \|M_2\|_F^2\right),\end{aligned} \quad (16)$$

where $\ell\left(\cdot\right)$ is a non-increasing continuous convex loss function, $q_{ij} = \Pi\left[y_i = y_j\right] \in \{-1, 1\}$ denotes whether two instances are similar or not and

$$\mathcal{S}_{\xi,\varphi}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \cos\left(\phi_\xi\left(\boldsymbol{x}_i\right), \phi_\xi\left(\boldsymbol{x}_j\right)\right) + \cos\left(\phi_\varphi\left(\boldsymbol{x}_i\right), \phi_\varphi\left(\boldsymbol{x}_j\right)\right), \quad (17)$$

where $\mathcal{S}_{\xi,\varphi}$ are EC similarity score parametered by ξ and φ and $\mathcal{S}_{\xi,\varphi}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) > 0$ when two instances are similar, otherwise not.

The empirical objective function above constructs an unbiased estimation of the following expected risk

$$\begin{aligned}&\min_{\xi,\varphi} E_{\boldsymbol{x}_1,\boldsymbol{x}_2}\left[\ell\left(q_{12}\left(\mathcal{S}_{\xi,\varphi}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)\right)\right)\right] + \lambda\left(\|M_1\|_F^2 + \|M_2\|_F^2\right) \\ =&\min_{\xi,\varphi} \mathcal{R}\left(\mathcal{Z}\right) + \lambda\left(\|M_1\|_F^2 + \|M_2\|_F^2\right).\end{aligned} \quad (18)$$

The empirical and expected risk $\hat{\mathcal{R}}_N\left(\mathcal{D}\right)$ and $\mathcal{R}\left(\mathcal{Z}\right)$ are based on empirical dataset $\mathcal{D}$ and true distribution $\mathcal{Z}$, respectively. Denote $\|A_{ij}\| = \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)^T$ as the outer product of the difference between instances $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Assume ℓ is bounded by ℓ_u, we have $\hat{\mathcal{R}}_N\left(\mathcal{D}\right) + \lambda\left(\|M_1\|_F^2 + \|M_2\|_F^2\right) \leq \hat{\mathcal{R}}_N\left(0\right) \leq \ell_u$. Thus, $\|M_1\|_F^2 + \|M_2\|_F^2 \leq \frac{\ell_u}{\lambda}$.

Then we have the following theorem.

Theorem 2. *Given $\|A_{ij}\|_F \leq \alpha$ for all possible i and j, and loss $\ell\left(\cdot\right)$ is L-Lipschitz, then with probability at least $1-\delta$, we have*

$$\mathcal{R}\left(\mathcal{Z}\right) \leq \hat{\mathcal{R}}_N\left(\mathcal{D}\right) + \frac{4L\ell_u\alpha}{\lambda\sqrt{N}}\left(1 + \sqrt{\frac{\log\frac{1}{\delta}}{2}}\right). \quad (19)$$

Proof. define the supreme of excess loss function $\mathbb{U}\left(\mathcal{D}\right) = \sup_{\xi,\varphi}\mathcal{R}\left(\mathcal{Z}\right) - \hat{\mathcal{R}}_N\left(\mathcal{D}\right)$. Replacing one example $\boldsymbol{z}_o$ in $\mathcal{D} = \{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_N\}$ with $\boldsymbol{z}_o^{'}$, then the upper bound of the difference $\sup_{\mathcal{D},\boldsymbol{z}_o^{'}}\left|\mathbb{U}\left(\mathcal{D}\right) - \mathbb{U}\left(\mathcal{D}'\right)\right|$ is

$$\begin{aligned}&\sup_{\mathcal{D},\boldsymbol{z}_o^{'}}\left|\mathbb{U}\left(\mathcal{D}\right) - \mathbb{U}\left(\mathcal{D}'\right)\right| \\ \leq&\sup_{\mathcal{D},\boldsymbol{z}_o^{'},\xi,\varphi}\left|\hat{R}_N\left(\mathcal{D}\right) - \hat{R}_N\left(\mathcal{D}'\right)\right| \\ =&\sup_{\mathcal{D},\boldsymbol{z}_o^{'}}\left|\frac{2}{N\left(N-1\right)}\sum_{i=1,i\neq o}^{N}\ell\left(q_{oi}\left(S_{\xi,\varphi}\left(\boldsymbol{x}_i, \boldsymbol{x}_o\right)\right)\right)\right. \\ &\left.- \ell\left(q_{oi}\left(S_{\xi,\varphi}\left(\boldsymbol{x}_i, \boldsymbol{x}_{o'}\right)\right)\right)\right| \\ \leq&\sup_{\mathcal{D}}\frac{4L}{N\left(N-1\right)}\sum_{i=1,i\neq o}^{N}\left|q_{oi}\left(S_{\xi,\varphi}\left(\boldsymbol{x}_i, \boldsymbol{x}_o\right)\right)\right| \\ \leq&\sup_{\mathcal{D}}\frac{4L}{N\left(N-1\right)}\sum_{i=1,i\neq o}^{N}\left|\langle A_{oi}, M_1\rangle + \langle A_{oi}, M_2\rangle\right| \\ \leq&\frac{8L\left(N-1\right)}{N\left(N-1\right)}\left(\alpha(\|M_1\|_F + \|M_2\|_F)\right) \\ \leq&\frac{4L\ell_u\alpha}{\lambda N}.\end{aligned} \quad (20)$$

The first inequality comes from the basic property of $\sup(\cdot)$ operator. The second inequality comes from the Lipschitz property of the loss function. The third inequality comes from [54]. Since α give upper bound for $\|A\|_F$, we have the forth inequality. The fifth inequality comes from $\|M_1\|_F + \|M_2\|_F \leq \|M_1\|_F^2 + \|M_2\|_F^2 \leq \frac{\ell_u}{\lambda}$. Given the bounded difference condition, together with McDiarmid Inequality [55], then with probability $1-\delta$

$$\mathbb{U}\left(\mathcal{D}\right) \leq \mathbb{E}_{\mathcal{D}}\left[\mathbb{U}\left(\mathcal{D}\right)\right] + \frac{4L\ell_u\alpha}{\lambda}\sqrt{\frac{\log\frac{1}{\delta}}{2N}}. \quad (21)$$

According to [54], $\mathbb{E}_{\mathcal{D}}\left[\mathbb{U}\left(\mathcal{D}\right)\right]$ is bounded by:

$$\mathbb{E}_{\mathcal{D}}\left[\mathbb{U}\left(\mathcal{D}\right)\right] \leq \frac{4L\ell_u\alpha}{\lambda\sqrt{N}}. \quad (22)$$

Plug (22) into (21), then Theorem 2 can be obtained immediately. □

Theorem 2 illustrates that the generalization error bound of the EC similarity has a convergence rate of $\mathcal{O}(\frac{1}{\sqrt{N}})$, which is the same order as other metrics [54], [56], [57]. Although

Algorithm 1 Meta training of IDEAL

Require: The series of support sets S_b and query sets Q_b from base classes.

Require: The pre-trained contrastive network g_ξ.

1: Initialize the meta network h_φ and BiLSTM ϕ_L.
2: **while** not done **do**
3: Sample a support set $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N\times K}$ and a query set $Q = \{(\tilde{\boldsymbol{x}}_i, \tilde{y}_i)\}_{i=1}^{M}$ from S_b and Q_b.
4: Compute the modified prototypes according to Eq.(7) and Eq.(8).
5: Compute the EC similarity scores between different query samples and the modified prototypes according to Eq.(15).
6: Update h_φ and ϕ_L by forward computation and back-propagation according to Eq.(29).
7: **end while**

Output: The meta network h_φ and BiLSTM ϕ_L.

the proposed method is learning in different spaces, the generalization error bound can be bound uniformly. It is also noteworthy that our method can promote the representation ability of the model by learning in different spaces without the raise of the generalization error.

4.5 Model Training and Inference

In this section, we detail the meta-training and inference. In the meta-training stage, we train the $h_\varphi(\cdot)$ in meta network and finetune the feature extractor $g_\xi(\cdot)$ in contrastive network. According to section 4.1, disentangled information learned by the two networks can remedy each other. Therefore, contrastive network and meta network respectively learns modified prototypes according to intra-class and intra-class weights of support samples in their corresponding embedding spaces. Finally, the EC similarity score is used to measure the similarity to each class according to prototypes information from different networks.

4.5.1 Meta-Training

In the meta-training, we train meta network $h_\varphi(\cdot)$ and finetune the contrastive network $g_\xi(\cdot)$ with three loss functions.

Meta Loss. Given a query sample $\tilde{\boldsymbol{x}}_q \in Q_b$ in a task, the classifier outputs the normalized EC similarity score between the query sample $\tilde{\boldsymbol{x}}_q$ and class centroid $\boldsymbol{c}_n$ of class n

$$P_\varphi\left(y=n\left|\tilde{\boldsymbol{x}}_q\right.\right)=\frac{\exp\left(sim_{EC}\left(\tilde{\boldsymbol{x}}_q,\boldsymbol{c}_n\right)/T\right)}{\sum_{n'}\exp\left(sim_{EC}\left(\tilde{\boldsymbol{x}}_q,\boldsymbol{c}_{n'}\right)/T\right)}, \tag{23}$$

where

$$sim_{EC}(\tilde{\boldsymbol{x}}_q,\boldsymbol{c}_n)=\cos\left(g_\xi\left(\tilde{\boldsymbol{x}}_q\right),\boldsymbol{p}_n\right)+\cos(h_\varphi\left(\tilde{\boldsymbol{x}}_q\right),\boldsymbol{p}_n^{'}), \tag{24}$$

according to the theory analysis in section 4.4 and T is a temperature scaling parameter. Then, the meta loss is given by

$$\mathcal{L}_{me}=\frac{1}{M}\sum\nolimits_{q=1}^{M}\left[-\log P_\varphi\left(y=y_q\left|\tilde{\boldsymbol{x}}_q\right.\right)\right], \tag{25}$$

where $M=|Q_b|$ is the number of query samples in a task.

Intra-Class Noise Loss. Similiar to [12], we consider the auxiliary set $D_b=\{(\boldsymbol{x}_i,y_i)\}_{i=1}^{T}$ for meta-training to be an offline data set containing sufficient clean supervised data. Even if D_b might contain noise-polluted instances, we have enough time to sift the data in D_b, or utilize the suitable data cleansing methods [35], [58] to filter D_b [12]. Therefore, D_b is regarded as a clean dataset. Since the mislabeled data in is artificially introduced, we know which one is noise during the training process. The intra-class noise loss is acted upon the intra-class weight of support samples in a task:

$$\mathcal{L}_{ra}=\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K}\left(-\Pi\left(\boldsymbol{x}_n^{(k)}\right)\log\left(1-\boldsymbol{v}_n^{(k)}\right)\right), \tag{26}$$

where the indicator function $\Pi\left(\boldsymbol{x}_n^{(k)}\right)$ is 1 when $\boldsymbol{x}_n^{(k)}$ is an artificially introduced mislabeled sample and 0 otherwise.

Inter-Class Noise Loss. As we all know, a good feature extractor can benefit clustering and evaluation of the class. We introduce inter-class noise loss to measure the inter-class noise information. For all support samples in each task, we construct similarity matrix $\boldsymbol{M}\in\mathbb{R}^{NK\times NK}$ to measure the similarity of each pair of samples. Then the loss of inter-class noise is given by

$$\mathcal{L}_{er}=-\frac{1}{(NK)^2}\sum\nolimits_{i=1}^{NK}\sum\nolimits_{j=1}^{NK}\boldsymbol{M}_{i,j}\boldsymbol{A}_{i,j}, \tag{27}$$

where

$$\boldsymbol{A}_{i,j}=\begin{cases}1, & \text{i}\neq j \text{ and } l(\boldsymbol{x}^{(i)})=l(\boldsymbol{x}^{(j)})\\ -1, & \text{i}\neq j \text{ and } l(\boldsymbol{x}^{(i)})\neq l(\boldsymbol{x}^{(j)})\\ 0, & i=j\end{cases} \tag{28}$$

and $l\left(\boldsymbol{x}^{(i)}\right)$ denotes the groud-truth label of $\boldsymbol{x}^{(i)}$.

Final Loss. Finally, the following objective function is used to optimize the meta network

$$\mathcal{L}_{total}=\mathcal{L}_{me}+\eta\mathcal{L}_{ra}+\gamma\mathcal{L}_{er}, \tag{29}$$

where η and γ are positive constants trading off the importances of different losses. Algorithm 1 summarizes the algorithmic description of the training process the proposed IDEAL.

4.5.2 Inference

In the inference stage, we use the EC similarity to measure the score for each class. For a query sample $\tilde{\boldsymbol{x}}_q$, the class with the highest score is obtained as the final label prediction:

$$\begin{aligned}\hat{y}_q&=\underset{n=1,\ldots,N}{\arg\max}\ sim_{EC}(\tilde{\boldsymbol{x}}_q,\boldsymbol{c}_n)\\&=\underset{n=1,\ldots,N}{\arg\max}\ \left(\cos\left(g_\xi\left(\tilde{\boldsymbol{x}}_q\right),\boldsymbol{p}_n\right)+\cos(h_\varphi\left(\tilde{\boldsymbol{x}}_q\right),\boldsymbol{p}_n^{'}).\end{aligned} \tag{30}$$

5 Experiments

In this section, we perform typical few-shot learning problem (noise-free) and open-world few-shot learning problem to verify the efficacy of our method in. Then, we perform ablation experiments to analyze the efficacy of different modules and parameters. Moreover, we compare methods on the union of base classes and novel classes to verify the plasticity and stability in the open world.

TABLE 1: Comparisons on FSL tasks (5-way 5-shot, Acc.%).

Methods	Backbone	CIFAR-FS	FC100	*mini*Imagenet	*tiered*Imagenet
Proto Nets [20]	C64E	68.78±0.77	48.30±0.76	64.69±0.69	61.50±0.74
Matching Nets [19]	C64E	68.93±0.74	48.28±0.73	62.75±0.75	60.74±0.81
Relation Nets [21]	C64E	65.01±0.79	48.02±0.72	63.96±0.69	60.35±0.75
MAML [17]	C64E	70.30±0.77	46.75±0.72	63.37±0.70	62.31±0.78
DKT [59]	C64E	67.81±0.73	47.70±0.74	62.43±0.72	59.58±0.80
FEAT [60]	C64E	72.43±0.77	47.61±0.69	63.18±0.69	65.42±0.79
S2M2 [61]	C64E	71.01±0.74	47.90±0.72	64.70±0.66	63.53±0.76
CSS [32]	C64E	74.59±0.72	49.72±0.69	68.08±0.73	67.80±0.77
Masked Soft *k*-Means [62]	C64E	67.46±0.78	49.06±0.75	62.68±0.72	60.77±0.79
PRWN [63]	C64E	70.16±0.75	49.50±0.73	64.83±0.71	63.58±0.75
RNNP [13]	C64E	68.69±0.74	45.91±0.74	63.88±0.79	63.73±0.87
RapNet [12]	C64E	67.84±0.76	47.86±0.75	63.53±0.68	61.56±0.79
IDEAL-B (Ours)	C64E	74.63±0.70	**51.95±0.72**	67.82±0.67	67.30±0.76
IDEAL-T (Ours)	C64E	**74.64±0.71**	51.76±0.69	**68.10±0.62**	**67.93±0.72**
Masked Soft *k*-Means [62]	ResNet-12	75.33±0.71	50.53±0.71	69.50±0.73	66.12±0.78
PRWN [63]	ResNet-12	81.43±0.67	54.15±0.74	73.83±0.69	69.57±0.75
RNNP [13]	ResNet-12	75.56±0.77	49.12±0.77	65.88±0.78	69.10±0.86
RapNet [12]	ResNet-12	80.06±0.68	54.67±0.74	72.86±0.64	69.44±0.75
IDEAL-B (Ours)	ResNet-12	83.06±0.68	57.33±0.74	**75.90±0.62**	**77.02±0.75**
IDEAL-T (Ours)	ResNet-12	**83.86±0.61**	**57.87±0.73**	75.26±0.66	76.25±0.77

TABLE 2: Comparisons on OFSL tasks with varying noise rates of type I ID noise (5-way 5-shot, Acc.%).

Methods	Backbone	CIFAR-FS		FC100		*mini*Imagenet		*tiered*Imagenet	
		20%	40%	20%	40%	20%	40%	20%	40%
Proto Nets [20]	C64E	63.61±0.82	49.66±0.84	43.79±0.72	36.83±0.69	58.40±0.76	47.07±0.75	55.80±0.84	44.57±0.84
Matching Nets [19]	C64E	59.75±0.78	48.77±0.77	43.15±0.70	36.32±0.61	56.55±0.71	44.78±0.72	54.48±0.74	45.19±0.75
Relation Nets [21]	C64E	63.98±0.80	51.30±0.85	41.54±0.74	34.54±0.67	57.06±0.70	45.87±0.77	56.19±0.81	45.01±0.82
MAML [17]	C64E	61.96±0.84	48.58±0.80	40.50±0.69	33.75±0.63	55.77±0.70	45.00±0.71	54.19±0.79	41.90±0.77
DKT [59]	C64E	61.13±0.81	49.64±0.80	44.38±0.73	36.95±0.68	56.68±0.72	46.13±0.73	56.63±0.77	46.14±0.75
FEAT [60]	C64E	66.88±0.78	52.64±0.90	43.29±0.68	35.66±0.64	59.16±0.75	46.82±0.81	60.98±0.81	47.24±0.80
S2M2 [61]	C64E	66.45±0.80	52.95±0.87	43.76±0.71	36.49±0.70	60.38±0.74	47.41±0.84	58.86±0.81	46.89±0.87
CSS [32]	C64E	65.39±0.82	52.44±0.91	43.86±0.73	36.59±0.71	59.82±0.75	47.78±0.78	59.45±0.83	47.18±0.82
Masked Soft *k*-Means [62]	C64E	59.18±0.85	41.77±0.84	43.95±0.75	36.08±0.78	56.97±0.74	43.30±0.80	54.66±0.85	35.42±0.81
PRWN [63]	C64E	62.36±0.83	45.88±0.83	45.39±0.73	36.67±0.72	57.97±0.72	44.09±0.77	56.10±0.76	42.32±0.83
RNNP [13]	C64E	65.59±0.85	52.59±1.17	42.24±0.75	35.57±0.77	59.97±0.86	47.74±1.05	60.07±0.93	47.44±1.13
RapNet [12]	C64E	64.40±0.83	45.62±0.89	45.71±0.74	35.92±0.75	59.01±0.75	43.98±0.79	56.28±0.80	42.46±0.85
IDEAL-B (Ours)	C64E	69.03±0.77	**55.17±0.95**	47.10±0.76	37.60±0.77	61.54±0.79	**49.87±0.83**	61.10±0.81	**48.16±0.88**
IDEAL-T (Ours)	C64E	**69.15±0.80**	53.52±0.82	**47.40±0.76**	**37.80±0.67**	**61.70±0.73**	48.06±0.78	**61.89±0.81**	47.86±0.85
Masked Soft *k*-Means [62]	ResNet-12	72.97±0.83	43.40±0.93	45.53±0.73	36.19±0.75	67.07±0.72	47.32±0.88	62.65±0.83	41.95±0.85
PRWN [63]	ResNet-12	75.28±0.75	53.00±0.93	48.45±0.73	38.14±0.69	67.91±0.72	49.89±0.84	63.41±0.85	46.55±0.83
RNNP [13]	ResNet-12	73.01±0.81	59.07±1.25	46.29±0.74	37.02±0.77	64.78±0.81	50.62±1.06	65.36±0.89	51.84±1.22
RapNet [12]	ResNet-12	74.61±0.74	53.35±0.93	47.36±0.76	36.81±0.72	67.09±0.72	50.21±0.86	64.91±0.81	46.45±0.92
IDEAL-B (Ours)	ResNet-12	**80.95±0.72**	**64.33±1.06**	52.68±0.80	40.91±0.76	**71.36±0.69**	**57.35±0.91**	71.72±0.81	**55.15±1.04**
IDEAL-T (Ours)	ResNet-12	80.44±0.71	62.79±0.96	**53.48±0.76**	**40.94±0.71**	70.20±0.70	55.73±0.84	**71.84±0.80**	53.56±0.91

5.1 Experimental settings

In experiments, we adopt four standard few-shot classification datasets, including CIFAR-FS [64], FC-100 [65], *mini*Imagenet [19] and *tiered*Imagenet [62].

5.1.1 Compared Methods

To comprehensively verify the superiority of our method, we compare our methods IDEAL-B (BiLSTM as intra-class calibration encoder) and IDEAL-T (Transformer as intra-class calibration) with state-of-the-art typical FSL methods modified semi-supervised FSL methods and advanced RFSL methods. The state-of-the-art FSL methods are including Proto Nets [20], Matching Nets [19], Relation Nets [21], MAML [17], Deep Kernel Transfer (DKT) [59], FEAT [60], S2M2 [61], CSS [32]. Two advanced robust FSL methods, i.e., RNNP [13] and RapNet [12], are set as baselines. We also modify two semi-supervised FSL i.e., Masked Soft *k*-Means [62] and PRWN [63] for the setting of OFSL. In OFSL problem, since Masked Soft *k*-Means and PRWN are originally proposed for semi-supervised few-shot learning, we adjust them appropriately to adapt to the setting of OFSL. We use support samples and query samples instead of the union of labeled samples and unlabeled samples as the dataset for an episode, which is the same setting as other methods. At the same time, we take the rehearsal strategy and deliberately add label noise to the support set of D_b to boost the robustness of these methods, which is the same setting as IDEAL and RapNet.

5.1.2 Implementation Details

For a fair comparison, we implement all methods by PyTorch and the code framework proposed in [2]. We adopt two backbones in the experiments, including the four-block-based ConvNet model (C64E) [2] and ResNet-12 [66]. In C64E, each block is comprised of 64-channel 3×3 convolution, batch normalization, ReLU [67] nonlinearity, and 2×2 max-pooling. The feature embedding dimension is set to 1600. The Adam optimizer [68] is used by all methods to

TABLE 3: Comparisons on OFSL tasks with varying noise rates of type II ID noise (5-way 5-shot, Acc.%).

Methods	Backbone	CIFAR-FS		FC100		*mini*Imagenet		*tiered*Imagenet	
		20%	40%	20%	40%	20%	40%	20%	40%
Proto Nets [20]	C64E	64.26±0.81	58.17±0.84	45.12±0.72	40.32±0.72	59.50±0.73	54.10±0.78	57.72±0.80	51.03±0.84
Matching Nets [19]	C64E	63.65±0.78	57.33±0.78	44.76±0.69	40.20±0.67	58.16±0.71	52.54±0.75	57.61±0.80	51.99±0.80
Relation Nets [21]	C64E	65.02±0.78	58.12±0.86	42.94±0.72	39.06±0.75	57.38±0.73	51.46±0.79	56.94±0.81	50.31±0.83
MAML [17]	C64E	65.03±0.81	58.80±0.83	41.99±0.72	37.25±0.71	58.57±0.74	51.11±0.79	57.02±0.82	50.58±0.86
DKT [59]	C64E	62.36±0.83	56.30±0.85	44.91±0.75	39.95±0.69	58.04±0.72	52.53±0.73	57.99±0.80	51.10±0.78
FEAT [60]	C64E	67.71±0.78	61.35±0.86	44.85±0.70	39.49±0.66	60.11±0.73	53.87±0.78	61.70±0.81	55.10±0.90
S2M2 [61]	C64E	67.57±0.79	61.16±0.86	44.93±0.73	40.69±0.74	61.35±0.75	54.15±0.80	60.04±0.82	54.12±0.89
CSS [32]	C64E	66.29±0.80	60.45±0.89	45.00±0.73	39.98±0.69	60.59±0.76	54.70±0.84	60.24±0.81	54.64±0.89
Masked Soft *k*-Means [62]	C64E	62.47±0.81	51.51±0.87	44.14±0.70	37.92±0.73	58.39±0.71	51.04±0.78	57.13±0.80	46.14±0.85
PRWN [63]	C64E	65.59±0.83	55.22±0.87	43.72±0.73	38.24±0.74	59.61±0.76	51.69±0.79	58.12±0.78	49.54±0.81
RNNP [13]	C64E	64.57±0.87	57.10±0.92	43.32±0.77	38.51±0.76	59.65±0.85	53.44±0.93	59.44±0.91	53.62±0.98
RapNet [12]	C64E	64.94±0.83	56.24±0.82	45.37±0.74	39.97±0.73	62.02±0.85	52.31±0.76	57.43±0.84	49.90±0.81
IDEAL-B (Ours)	C64E	70.64±0.79	**64.04±0.83**	47.90±0.72	**42.36±0.76**	**63.95±0.67**	**56.99±0.80**	62.88±0.83	**56.34±0.86**
IDEAL-T (Ours)	C64E	**70.90±0.76**	63.13±0.85	**48.19±0.78**	41.98±0.76	63.27±0.71	55.60±0.79	**62.97±0.84**	56.11±0.78
Masked Soft *k*-Means [62]	ResNet-12	66.78±0.85	57.12±0.93	47.15±0.73	39.64±0.81	67.25±0.73	57.35±0.82	62.18±0.83	52.09±0.85
PRWN [63]	ResNet-12	76.58±0.75	65.48±0.88	46.79±0.74	39.99±0.74	68.28±0.69	57.72±0.76	66.27±0.82	55.70±0.85
RNNP [13]	ResNet-12	71.50±0.81	63.88±0.95	47.28±0.78	41.68±0.80	63.62±0.80	56.60±0.91	64.50±0.88	58.14±0.97
RapNet [12]	ResNet-12	76.27±0.73	65.94±0.83	50.40±0.71	41.96±0.76	68.77±0.71	60.86±0.79	65.97±0.77	58.59±0.89
IDEAL-B (Ours)	ResNet-12	81.28±0.71	**70.80±0.87**	53.55±0.72	45.32±0.71	71.90±0.68	**66.03±0.79**	71.99±0.82	**67.46±0.91**
IDEAL-T (Ours)	ResNet-12	**81.79±0.68**	69.60±0.94	**53.67±0.74**	**47.42±0.82**	**72.61±0.69**	64.32±0.77	**72.42±0.80**	65.01±0.87

TABLE 4: Comparisons on OFSL tasks with varying noise rates of OOD noise (5-way 5-shot, Acc.%).

Methods	Backbone	CIFAR-FS+*mini*Imagenet		FC100+*mini*Imagenet		*mini*Imagenet+CIFAR-FS		*tiered*Imagenet+CIFAR-FS	
		20%	40%	20%	40%	20%	40%	20%	40%
Proto Nets [20]	C64E	64.48±0.79	58.62±0.83	44.88±0.73	40.03±0.69	61.51±0.68	55.53±0.70	57.58±0.82	53.14±0.84
Matching Nets [19]	C64E	63.64±0.80	58.17±0.81	45.35±0.72	41.62±0.69	59.53±0.73	55.94±0.77	58.48±0.81	54.62±0.82
Relation Nets [21]	C64E	65.02±0.82	58.69±0.88	41.45±0.72	36.92±0.71	58.90±0.73	53.90±0.72	57.68±0.75	53.72±0.78
MAML [17]	C64E	65.01±0.83	57.99±0.84	42.52±0.71	37.76±0.69	59.08±0.74	52.75±0.75	57.41±0.80	51.21±0.80
DKT [59]	C64E	63.33±0.81	56.86±0.87	45.35±0.74	40.95±0.71	58.24±0.68	53.96±0.73	58.54±0.74	54.96±0.80
FEAT [60]	C64E	67.72±0.78	60.82±0.87	43.26±0.69	37.63±0.70	61.43±0.70	56.16±0.74	61.39±0.78	56.63±0.81
S2M2 [61]	C64E	66.83±0.79	59.74±0.84	43.96±0.72	39.58±0.74	62.69±0.70	57.07±0.72	61.53±0.78	57.10±0.84
CSS [32]	C64E	66.37±0.79	60.74±0.90	44.18±0.72	39.14±0.72	61.90±0.71	56.85±0.73	61.65±0.81	56.99±0.82
Masked Soft *k*-Means [62]	C64E	62.70±0.82	52.24±0.86	44.70±0.72	38.90±0.74	58.87±0.68	49.96±0.74	57.88±0.81	46.61±0.84
PRWN [63]	C64E	65.61±0.80	55.13±0.92	44.56±0.74	39.31±0.74	60.06±0.68	52.59±0.73	58.54±0.79	51.21±0.81
RNNP [13]	C64E	63.76±0.90	55.99±0.95	40.83±0.73	34.56±0.74	60.19±0.81	56.71±0.81	60.26±0.90	56.49±0.89
RapNet [12]	C64E	65.56±0.82	55.68±0.85	45.33±0.72	40.50±0.71	56.28±0.80	52.54±0.71	57.92±0.85	49.72±0.83
IDEAL-B (Ours)	C64E	71.22±0.74	**64.92±0.85**	47.43±0.74	**42.35±0.76**	**64.37±0.72**	57.36±0.78	62.97±0.80	**57.97±0.85**
IDEAL-T (Ours)	C64E	**71.39±0.75**	64.30±0.82	**47.50±0.72**	42.02±0.69	63.83±0.70	**57.97±0.74**	**63.52±0.80**	57.67±0.79
Masked Soft *k*-Means [62]	ResNet-12	68.15±0.89	58.25±0.91	47.85±0.72	40.51±0.79	68.05±0.71	56.88±0.81	63.18±0.84	53.98±0.82
PRWN [63]	ResNet-12	77.09±0.74	67.70±0.86	48.51±0.72	42.37±0.72	68.92±0.70	61.01±0.74	66.51±0.84	58.16±0.88
RNNP [13]	ResNet-12	70.30±0.90	60.10±0.97	45.33±0.78	37.92±0.77	64.18±0.79	59.44±0.81	63.70±0.86	58.24±0.88
RapNet [12]	ResNet-12	74.38±0.77	66.26±0.87	50.05±0.72	41.62±0.74	68.98±0.71	61.04±0.75	66.45±0.78	58.96±0.86
IDEAL-B (Ours)	ResNet-12	81.81±0.69	**74.71±0.80**	**53.20±0.72**	44.22±0.73	73.04±0.66	65.59±0.78	**73.60±0.78**	**68.93±0.87**
IDEAL-T (Ours)	ResNet-12	**81.83±0.67**	72.28±0.86	52.90±0.74	**47.35±0.73**	**73.80±0.66**	**67.77±0.76**	73.21±0.83	67.31±0.85

optimize parameters. The learning rate is set to 10^{-3} for all methods. In IDEAL, g_ξ and h_φ are implemented by C64E backbone. τ in Eq.(6) is set to 0.1 and T in Eq.(23) is set to 0.1. In Eq.(7) and Eq.(8), α is set to 0.9 and β is set to 0.1. In Eq.(29), η is set to 0.1 and γ is set to 0.1. For all methods, in the meta-training stage, the maximum number of training episodes is set to 40000. In the inference stage, all experimental results are averaged accuracy of 600 randomly generated test episodes with 95% confidence intervals.

5.1.3 Datasets

CIFAR-FS [64] is a few-shot dataset created by dividing the 100 classes of CIFAR-100 into 64 base classes, 16 validation classes, and 20 novel test classes.

FC100 [65] is built based on CIFAR100. It contains 100 object classes which have been grouped into 20 superclasses. It uses 60 classes belonging to 12 superclasses for training, 20 classes belonging to four superclasses for validation, and 20 classes belonging to the rest four superclasses for the test.

***mini*ImageNet** [19] is commonly used in evaluating few-shot classification algorithms for object recognition. It consists of a subset of 100 classes taken from the ImageNet dataset and contains 600 images for each class.

***tiered*ImageNet** [62] is a larger few-shot dataset and its categories are selected with hierarchical structure to split base and novel datasets semantically. We follow the split introduced in [62] with base set of 20 superclasses (351 classes), validation set of 6 superclasses (97 classes) and novel set of 8 superclasses (160 classes). Each class contains 1281 images on average.

5.2 Typical Few-Shot Learning Problems

We conduct typical FSL experiments to demonstrate the universality of IDEAL in clean datasets. Average accuracies

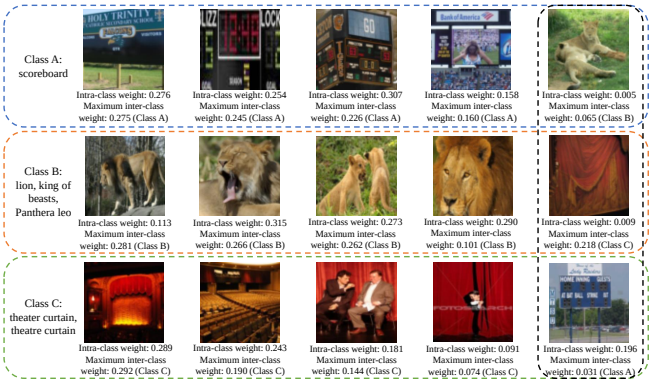


Fig. 5: Intra-class weight and maximum inter-class weight of each sample in the studied OFSL task with type I ID noise. For each support sample, the class with the maximum inter-class weight and its corresponding value are given. The last column denotes the type I ID noise.

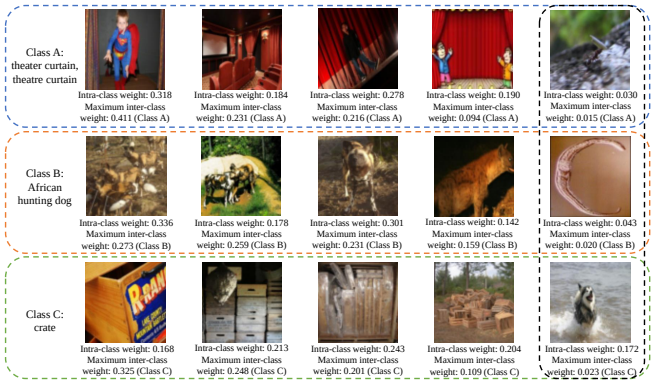


Fig. 6: Intra-class weight and maximum inter-class weight of each sample in the studied OFSL task with type II ID noise. For each support sample, the class with the maximum inter-class weight and its corresponding value are given. The last column denotes the type II ID noise.

and standard deviations of different algorithms for 5-way 5-shot FSL tasks are given in Table 1.

In Table 1, for each comparison on each backbone, the best result among IDEAL, Masked Soft k-Means, PRWN, RNNP and RapNet is highlighted in bold, and the best result among all algorithms is underlined. From Table 1, we can find that in all typical FSL tasks, IDEAL can achieve the best classification performance. The results indicate that our IDEAL can significantly outperform the compared FSL methods in typical few-shot learning problems.

5.3 Open-World Few-Shot Learning Problems

In the open-world, due to unknown sources of data, different types of noise might exist in data which make the FSL problem more difficult. In this section, we compare our methods in OFSL settings.

5.3.1 Noise Settings

In the experiments, we set open-world noisy few-shot scenarios built by replacing some images with type I, type II and OOD noise while keeping the labels and the number of images per class unchanged. Type I ID noise is produced via

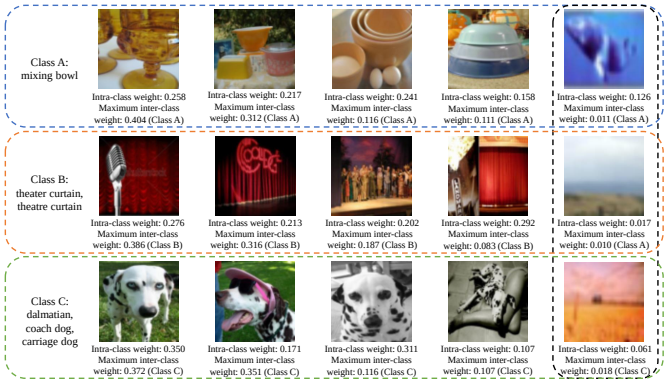


Fig. 7: Intra-class weight and maximum inter-class weight of each sample in the studied OFSL task with OOD noise. For each support sample, the class with the maximum inter-class weight and its corresponding value are given. The last column denotes the OOD noise from CIFAR-FS dataset.

replacing the raw sample with a sample from other classes in the current task, which severely disturbs both distributions of the two classes. Type II ID noise is produced by replacing the raw sample with a sample from other classes in known datasets except for classes in the current task, which disturbs the distribution of the current corrupted class. OOD noise is built by replacing some training samples with an external dataset which leads to unforeseen shift of feature and label distribution.

5.3.2 Experimental Analysis

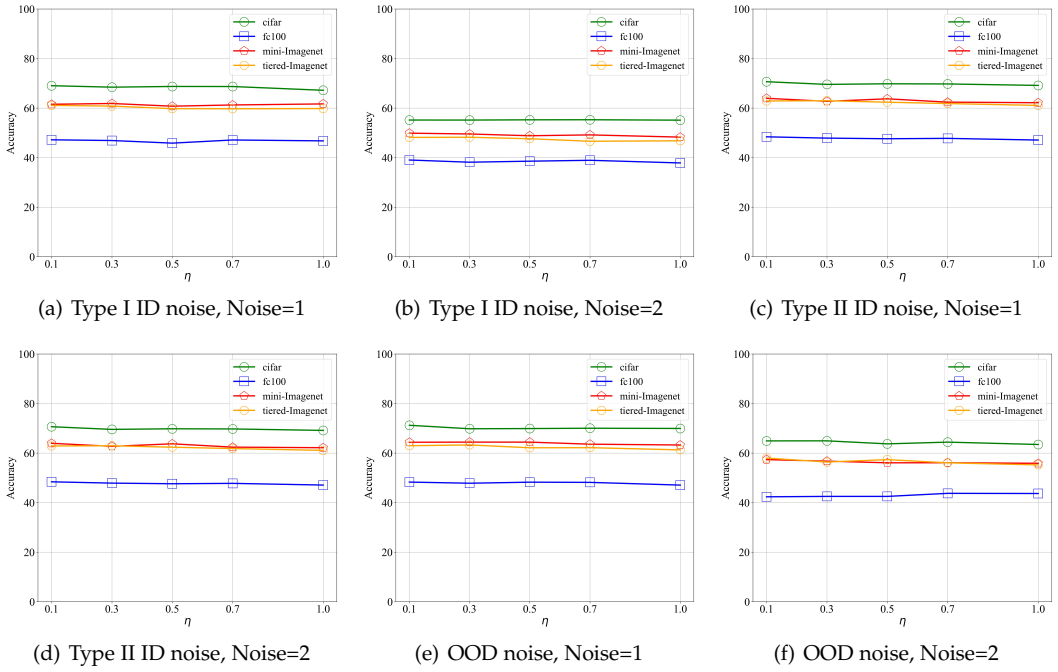
For the 5-way 5-shot problem, Table 2, 3 and Table 4 respectively demonstrate the performance of compared methods faced with various noise rates of type I ID noise, type II ID noise and OOD noise. The second row of tables represents the noise rate. Since the data from unknown domains might not be clearly defined, in this paper, we adopt external datasets to imitate unknown domains. CIFAR-FS+*mini*Imagenet in the second column represents that OOD noise is a sample from *mini*ImageNet while replacing the sample in CIFAR-FS. For each comparison, the best result is underlined, and the best result among IDEAL, Masked Soft k-Means, PRWN, RNNP and RapNet is highlighted in bold. From Tables 2 and 3, we can find that models are more susceptible to type I ID noise than type II noise, which may be because type I noise destroys the decision boundary between the two classes. At the same time, with the increase of the number of noise, the performances of the existing methods become significantly worse in the face of type I noise. However, our IDEAL considering global correlations among all support samples still maintains superior and stable performance faced with two kinds of ID noise and gets rid of the interference of confused representations. Table 4 displays the experimental result with OOD noise which is more consistent with the real-world environment. IDEAL outperforms prior methods with a large margin in all cases. Therefore, our method is more robust to noise in the open world.

5.4 Ablation Study and Model Analysis

In this section, we take IDEAL-B for example to perform ablation experiments to analyze the efficacy of different

TABLE 5: Ablation results on OFSL tasks under different cases on *mini*ImageNet dataset (5-way 5-shot, Acc.%).

Method	Type I ID		Type II ID		OOD	
	20%	40%	20%	40%	20%	40%
(a) w/o instance-wise calibration						
• case a_1: without intra-class calibration	49.71±0.80	34.34±0.85	48.74±0.80	42.42±0.79	48.86±0.81	42.86±0.79
• case a_2: without inter-class calibration	60.54±0.70	49.13±0.87	63.80±0.72	55.46±0.81	62.36±0.71	56.56±0.76
(b) w/o metric-wise calibration						
• case b_1: without $\cos(g_\xi(\boldsymbol{x}))$	57.36±0.70	45.10±0.77	61.70±0.73	54.09±0.80	62.03±0.74	55.56±0.81
• case b_2: without $\cos(h_\varphi(\boldsymbol{x}))$	59.29±0.71	49.53±0.88	61.40±0.74	55.37±0.78	62.10±0.72	55.82±0.77
• case b_3: using Euclidean distance	61.29±0.66	48.92±0.79	63.21±0.70	56.90±0.76	64.23±0.71	56.91±0.77
(c) w/o different losses						
• case c_1: without both losses	61.41±0.75	48.68±0.85	62.72±0.74	55.84±0.79	63.71±0.69	56.60±0.75
• case c_1: without intra-class noise loss	61.43±0.75	49.15±0.84	63.58±0.73	56.80±0.80	64.03±0.70	56.64±0.80
• case c_1: without inter-class noise loss	61.42±0.75	49.49±0.89	63.16±0.75	56.88±0.82	64.25±0.72	56.81±0.80
our model	**61.54±0.79**	**49.87±0.83**	**63.95±0.67**	**56.99±0.80**	**64.37±0.72**	**57.36±0.78**



(a) Type I ID noise, Noise=1 (b) Type I ID noise, Noise=2 (c) Type II ID noise, Noise=1

(d) Type II ID noise, Noise=2 (e) OOD noise, Noise=1 (f) OOD noise, Noise=2

Fig. 8: Comparisons between different values of η in different cases.

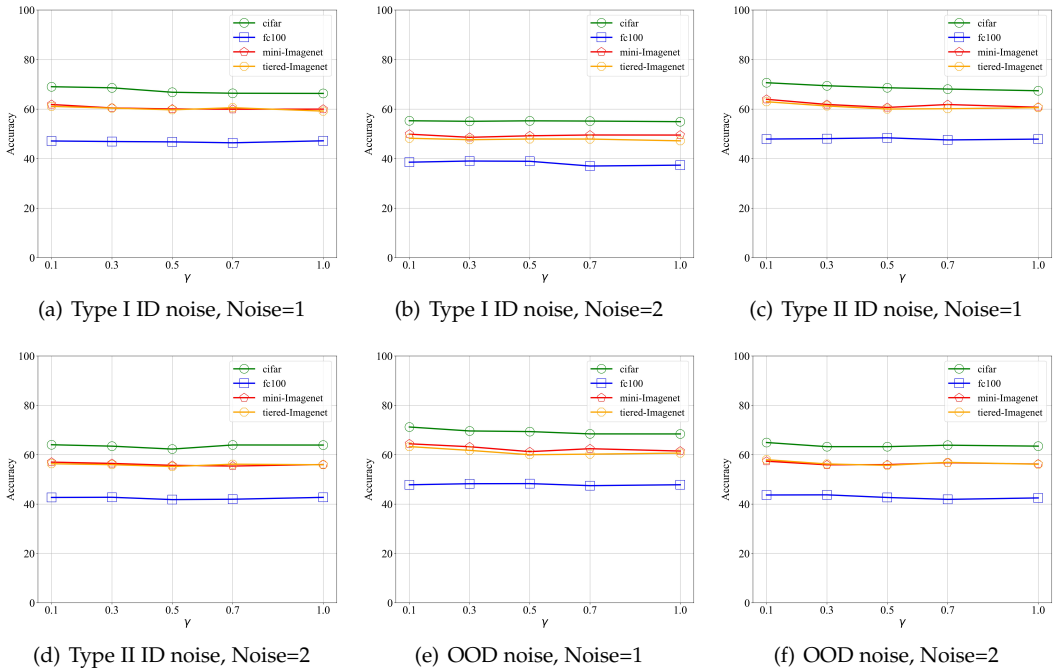
modules and parameters.

5.4.1 Effectiveness of Instance-Wise Calibration and Metric-Wise Calibration

For a comprehensive understanding of our model, we further design five cases to evaluate each module in 5-way 5-shot open-world settings. We also evaluate each module with three kinds of noise, i.e., type I ID, type II ID and OOD noise. For each noise situation, we also add 2 degrees of noise, i.e, 20% and 40% noise rate. The five cases are 1) case a_1, remove the intra-class calibration term in Eq.(7) and Eq.(8), 2) case a_2, remove the inter-class calibration term in Eq.(7) and Eq.(8), 3) case b_1, remove the first-term in the EC similarity of Eq.(24), 4) case b_2, remove the second-term in the EC similarity of Eq.(24) and 5) case b_3, use negative average Euclidean distance to replace the average cosine similarity induced as EC similarity. For all cases, C64E is adopted as the backbone.

Performance drops are observed in Table 5 when any of the instance-wise and metric-wise calibration modules are removed or replaced. These experimental results verify that each module in IDEAL is essential. In cases a_1 and a_2, the combination of intra-class and inter-class calibration can improve the robustness of the model. In cases b_1,b_2 and b_3, the proposed EC similarity is effective to calibrate label confidences. Simply replacing average Cosine similarity with negative average Euclidean distance seriously reduces the model performance due to the lack of properties in EC similarity. In the meantime, we find the performance of the model only using intra-class calibration might be higher than that of the model only using inter-class calibration. This is caused by two reasons: 1) Intra-class calibration is executed by the pre-trained contrastive network, while inter-class calibration is executed by the meta network, which requires being trained from scratch in the meta-training process. Therefore, the parameters of the meta network may fluctuate frequently, which will introduce the uncertainty of optimization and influence the final performance. 2) Since samples are rare, the process of clustering may bring some uncertainty and affect the model performance.

However, although the phenomenon might happen, it



(a) Type I ID noise, Noise=1 (b) Type I ID noise, Noise=2 (c) Type II ID noise, Noise=1

(d) Type II ID noise, Noise=2 (e) OOD noise, Noise=1 (f) OOD noise, Noise=2

Fig. 9: Comparisons between different values of γ in different cases.

does not mean that the intra-class calibration is trivial. In fact, inter-class calibration considers the global correlation among all support samples in a whole task and distinctively tackles these different types of noise, which is ignored by intra-class calibration. To better illustrate the effect of inter-class calibration in dealing with different types of noise, we conduct three kinds of case studies which are shown in Figures 5, 6 and 7. The known domain is from *mini*ImageNet. The last column in each Figure denotes noise. For each sample, the top line represents the intra-class weight of annotated class while the bottom two lines represent the maximum inter-class weight in all classes and the class with the maximum inter-class weight. As displayed in Figure 5, when handling type I ID noise, inter-class calibration can not only correctly refer to the potential correct class according to the class with the maximum inter-class weight, but also ameliorate the class prototype with the corrected noise. From Figures 6 and 7, when handling type II ID noise and OOD noise, the maximum inter-class weights for all noise are smaller than 0.025. At the same time, each correct support samples have a much higher weight, which demonstrates that inter-class calibration can effectively evaluate the inter-class weights of support samples according to their contributions to each cluster to filter out isolated noise. Therefore, inter-class calibration and intra-class calibration can supplement each other to further promote the robustness of the model.

5.4.2 Effectiveness of Intra-Class Noise Loss and Inter-Class Noise Loss

In order to better demonstrate the importance of intra-class noise loss and inter-class noise loss, we conduct further ablation studies to compare the proposed IDEAL with three versions: without intra-class noise loss, without inter-class noise loss and without both losses. The evaluations are conducted on 5-way 5-shot open-world settings on the CIFAR-FS dataset. Table 5 tabulates the experimental results of different versions of IDEAL, and the best performance on each comparison is highlighted in bold. From Table 5, we can find that both intra-class noise loss and inter-class noise loss are beneficial to improving the performance, which proves the validity of the proposed two losses.

5.4.3 Effectiveness of η and γ

In this part, we explore the effectiveness of the hyper-parameters η and γ. We compare the performance of IDEAL with different values of η and γ on the four datasets when facing type I ID, type II ID and OOD noise. The backbone C64E is adopted in the experiments. Fig. 8 illustrates the experimental results of IDEAL with different values of η, and Fig. 9 demonstrates the experimental results of IDEAL with different values of γ.

From Figures 8 and 9, we can find that: 1) IDEAL has stable performances with a wide range of hyperparameter values on all four datasets. 2) The performance of the model is hardly influenced by the changes of η. 3) In general, when the value of γ is between 0.1 and 0.3, the model can achieve better performance. These findings further demonstrate the robustness of the proposed IDEAL in practical application.

5.5 Test on the Union of Base Classes and Novel Classes

We also take Noise=1 for example to test the performance of our proposed IDEAL against compared method on the union of base classes and novel classes. Average accuracies and standard deviations of different algorithms for OFSL tasks with type I ID noise, type II ID noise and OOD noise are given in Table 6. In Table 6, for each comparison on each backbone, the best result among IDEAL, Masked Soft

TABLE 6: Comparisons on OFSL tasks (5-way 5-shot, Acc.%, test on $C_{base} \bigcup C_{novel}$).

Methods	Backbone	CIFAR-FS			FC100		
		Type I ID	Type II ID	OOD	Type I ID	Type II ID	OOD
Proto Nets [20]	C64E	64.79±0.82	66.13±0.83	66.82±0.84	67.60±0.88	67.56±0.85	67.94±0.87
Matching Nets [19]	C64E	62.46±0.80	67.10±0.84	67.02±0.77	62.67±0.82	66.73±0.82	68.13±0.81
Relation Nets [21]	C64E	68.32±0.80	68.70±0.79	69.15±0.81	66.02±0.88	66.62±0.87	66.58±0.87
MAML [17]	C64E	65.23±0.77	67.41±0.82	66.87±0.82	64.28±0.94	66.70±0.89	67.16±0.94
DKT [59]	C64E	62.13±0.86	63.80±0.86	64.31±0.82	63.38±0.85	65.16±0.85	65.82±0.81
FEAT [60]	C64E	72.00±0.85	73.67±0.81	72.74±0.79	72.37±0.86	73.02±0.85	74.08±0.88
S2M2 [61]	C64E	70.07±0.84	70.82±0.80	69.58±0.81	69.18±0.84	70.71±0.86	71.48±0.90
CSS [32]	C64E	69.76±0.84	70.25±0.86	70.38±0.84	69.28±0.92	70.09±0.89	70.34±0.91
Masked Soft *k*-Means [62]	C64E	60.51±0.92	63.69±0.84	63.89±0.85	64.96±0.90	67.13±0.87	67.53±0.86
PRWN [63]	C64E	63.65±0.89	67.13±0.81	67.62±0.83	66.20±0.88	67.37±0.82	68.46±0.83
RNNP [13]	C64E	73.47±0.87	71.96±0.89	70.87±0.91	71.64±0.94	69.71±0.94	69.36±0.95
RapNet [12]	C64E	64.56±0.88	65.73±0.80	66.10±0.83	65.26±0.88	69.12±0.87	68.72±0.88
IDEAL-B (Ours)	C64E	73.53±0.79	**75.47±0.80**	75.46±0.79	72.82±0.88	73.36±0.81	**74.54±0.83**
IDEAL-T (Ours)	C64E	**73.64±0.84**	74.91±0.81	**75.47±0.75**	**73.78±0.85**	**73.64±0.85**	74.26±0.86
Masked Soft *k*-Means [62]	ResNet-12	79.25±0.79	72.02±0.83	72.16±0.87	77.41±0.87	79.63±0.81	79.66±0.79
PRWN [63]	ResNet-12	82.66±0.70	84.27±0.68	84.92±0.68	80.36±0.87	82.80±0.87	83.11±0.87
RNNP [13]	ResNet-12	84.23±0.74	81.95±0.79	83.19±0.73	79.75±0.92	77.83±0.96	78.17±0.93
RapNet [12]	ResNet-12	81.57±0.75	83.42±0.71	83.46±0.70	82.33±0.82	83.22±0.76	83.30±0.79
IDEAL-B (Ours)	ResNet-12	92.91±0.49	**93.67±0.47**	93.73±0.46	85.02±0.90	**86.62±0.84**	**86.70±0.82**
IDEAL-T (Ours)	ResNet-12	**93.11±0.52**	93.54±0.49	**93.77±0.46**	**85.44±0.80**	84.95±0.87	86.27±0.78
Methods	Backbone	*mini*Imagenet			*tiered*Imagenet		
		Type I ID	Type II ID	OOD	Type I ID	Type II ID	OOD
Proto Nets [20]	C64E	66.34±0.79	67.58±0.80	68.66±0.81	58.07±0.76	59.22±0.79	60.05±0.82
Matching Nets [19]	C64E	61.67±0.80	66.74±0.81	68.52±0.83	56.71±0.78	58.49±0.75	59.25±0.82
Relation Nets [21]	C64E	64.95±0.83	65.15±0.85	66.11±0.86	59.77±0.81	60.96±0.79	62.05±0.79
MAML [17]	C64E	65.32±0.80	66.85±0.83	68.52±0.82	57.87±0.84	59.62±0.82	60.23±0.81
DKT [59]	C64E	62.25±0.81	63.12±0.79	63.66±0.83	57.88±0.75	59.84±0.81	59.91±0.79
FEAT [60]	C64E	68.25±0.82	68.60±0.86	69.71±0.77	63.30±0.79	64.26±0.76	65.42±0.80
S2M2 [61]	C64E	68.92±0.85	69.35±0.82	70.03±0.86	61.42±0.80	62.52±0.83	63.81±0.79
CSS [32]	C64E	68.98±0.83	70.42±0.75	70.97±0.84	62.38±0.85	62.75±0.81	64.03±0.80
Masked Soft *k*-Means [62]	C64E	63.20±0.88	65.06±0.81	65.67±0.80	56.87±0.81	58.80±0.78	59.32±0.82
PRWN [63]	C64E	64.20±0.84	66.86±0.80	67.81±0.81	57.38±0.78	59.45±0.79	60.29±0.79
RNNP [13]	C64E	72.18±0.87	71.30±0.88	70.68±0.83	64.39±0.91	63.94±0.88	65.18±0.85
RapNet [12]	C64E	65.76±0.85	66.06±0.85	66.63±0.83	57.57±0.85	59.00±0.80	59.48±0.81
IDEAL-B (Ours)	C64E	**73.55±0.80**	75.20±0.81	75.60±0.80	64.64±0.81	66.17±0.79	66.96±0.79
IDEAL-T (Ours)	C64E	72.51±0.80	**75.54±0.77**	**76.29±0.73**	**64.87±0.75**	**66.47±0.79**	**67.12±0.77**
Masked Soft *k*-Means [62]	ResNet-12	78.21±0.75	78.28±0.78	78.84±0.72	66.76±0.80	66.50±0.82	67.08±0.81
PRWN [63]	ResNet-12	80.73±0.75	81.24±0.72	82.25±0.69	69.06±0.83	71.67±0.80	72.37±0.78
RNNP [13]	ResNet-12	80.33±0.79	78.20±0.82	78.27±0.75	74.63±0.85	72.82±0.88	72.48±0.80
RapNet [12]	ResNet-12	79.30±0.76	82.72±0.67	82.82±0.67	70.28±0.79	70.46±0.80	70.48±0.77
IDEAL-B (Ours)	ResNet-12	**91.38±0.54**	**91.55±0.55**	91.67±0.53	80.66±0.73	**82.83±0.68**	82.95±0.69
IDEAL-T (Ours)	ResNet-12	90.51±0.59	91.12±0.51	**91.87±0.53**	**81.37±0.74**	82.03±0.73	**83.00±0.72**

k-Means, PRWN, RNNP and RapNet is highlighted in bold, and the best result among all algorithms is underlined. From Table 6, we can find that in all cases, both IDEAL-B and IDEAL-T can achieve the best classification performance compared with other methods. The results indicate that our IDEAL is superior to the compared few-shot classification methods and achieves plasticity and stability in the open world.

6 Conclusion

In this paper, we propose a novel instance-wise and metric-wise calibration framework (IDEAL) to address the new open-world few-shot learning problem. IDEAL is based on a dual-networks structure. Each network can remedy each other to boost both the feature and label representation capability of the model. In instance-wise calibration, IDEAL leverages intra-class and inter-class sample weights extracted by both networks to obtain rectified prototypes. In metric-wise calibration, we use the proposed EC similarity with fused prototype information to better evaluate the label confidences of query samples. Various experiments demonstrate that our method performs robustly in systemic label-noise few-shot scenarios.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant No. 62076062), Social Development Science and Technology Project of Jiangsu Province (Grant No. BE2022811) and China Postdoctoral Science Foundation (Grant No. 2022M720028). Furthermore, the work was also supported by Collaborative Innovation Center of Wireless Communications Technology.

Appendix A
The Pre-Training of Contrastive Network

The contrastive network g_ξ can be trained with conditional loss and contrastive loss. The contrastive loss in SimSiam [29] is introduced and other contrastive loss is equally acceptable. Given an input image x, two augmented views

x' and x'' are generated according to random augmentation methods. The two views are processed by an encoder network consisting of the contrastive feature extractor g_ξ and a projection MLP head σ. A prediction MLP head, denoted as δ, transforms the output of one view and matches it to the other view. Then, the negative cosine similarity of two embedding vectors is obtained:

$$L_c(x) = -\frac{\delta(\sigma(g_\xi(x')))}{\|\delta(\sigma(g_\xi(x')))\|_2} \cdot \text{stopgrad}\left(\frac{\sigma(g_\xi(x''))}{\|\sigma(g_\xi(x''))\|_2}\right) - \frac{\delta(\sigma(g_\xi(x'')))}{\|\delta(\sigma(g_\xi(x'')))\|_2} \cdot \texttt{stopgrad}\left(\frac{\sigma(g_\xi(x'))}{||\sigma(g_\xi(x'))||_2}\right), \tag{31}$$

where $\|\cdot\|_2$ is l_2-norm and `stopgrad` is the stop-gradient operation [29].

In the meantime, a conditional loss is utilized to guide the learning of contrastive network, which is guided by the feature extractor f_θ [32]:

$$L_o(x) = -\texttt{stopgrad}\left(\frac{f_\theta(x)}{\|f_\theta(x)\|_2}\right) \cdot \frac{g_\xi(x)}{\|g_\xi(x)\|_2}. \tag{32}$$

Combining contrastive loss with conditional loss, the objective function to optimize the contrastive network can be obtained by:

$$L_{pretrain}(x) = \mathop{\mathbb{E}}_{(x_i,y_i)\in D_b}[L_c(x_i) + \gamma L_o(x_i)]. \tag{33}$$

where γ is a positive constant trading off the importance of contrastive loss and the conditional loss.

APPENDIX B
LIMITATIONS OF THE PROPOSED METHOD

The open-world learning is a tremendously difficult open problem, and essential work needs to be done in this direction. In this paper, we mainly focus on the scenario that there are noisy samples from known classes, unknown classes in the known domain, and unknown domains in the support set, which is indeed an open-world scenario. Then, we propose an efficient method with instance-wise calibration and metric-wise calibration to identify the noise from support samples and enhance the robustness of models in the open world. However, when there exists an unknown class in the query set or in both the query set and support set, our method might not be able to operate effectively. Our method could not be updated flexibly when new classes are assumed to appear incremental. Moreover, when there exists a domain shift between support samples and query samples, the proposed method cannot effectively align two spaces and classify query samples. In future work, we will further delve into the robustness, generalization and discrimination ability of the model in various open-world scenarios. We will also explore the dynamic update of the model with the incremental addition of unknown classes for wider open-world applications.

REFERENCES

[1] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 63:1–63:34, 2020.

[2] W. Chen, Y. Liu, Z. Kira, Y. F. Wang, and J. Huang, "A closer look at few-shot classification," in *Proceedings of the 7th International Conference on Learning Representations*, 2019.

[3] C. Ouyang, C. Biffi, C. Chen, T. Kart, H. Qiu, and D. Rueckert, "Self-supervised learning for few-shot medical image segmentation," *IEEE Trans. Medical Imaging*, vol. 41, no. 7, pp. 1837–1848, 2022.

[4] Y. Wang, S. Wang, Y. Li, and D. Dou, "Recognizing medical search query intent by few-shot learning," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 502–512, 2022.

[5] C. Xue, Q. Dou, X. Shi, H. Chen, and P. Heng, "Robust learning at noisy labeled medical images: Applied to skin lesion classification," in *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pp. 1280–1283, 2019.

[6] C. Xue, L. Yu, P. Chen, Q. Dou, and P. Heng, "Robust medical image classification from noisy labeled data with global and local representation guided co-training," *IEEE Trans. Medical Imaging*, vol. 41, no. 6, pp. 1371–1382, 2022.

[7] I. Masi, A. T. Tran, T. Hassner, G. Sahin, and G. G. Medioni, "Face-specific data augmentation for unconstrained face recognition," *Int. J. Comput. Vis.*, vol. 127, no. 6-7, pp. 642–667, 2019.

[8] D. Tsipras, S. Santurkar, L. Engstrom, A. Ilyas, and A. Madry, "From imagenet to image classification: Contextualizing progress on benchmarks," in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, pp. 9625–9635, 2020.

[9] K. J. Liang, S. B. Rangrej, V. Petrovic, and T. Hassner, "Few-shot learning with noisy labels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9089–9098, 2022.

[10] J. Li, C. Xiong, and S. C. Hoi, "Learning from noisy data with robust representation learning," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9485–9494, 2021.

[11] S. Yang, L. Liu, and M. Xu, "Free lunch for few-shot learning: Distribution calibration," in *Proceedings of the 9th International Conference on Learning Representations*, 2021.

[12] J. Lu, S. Jin, J. Liang, and C. Zhang, "Robust few-shot learning for user-provided data," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 32, no. 4, pp. 1433–1447, 2021.

[13] P. Mazumder, P. Singh, and V. P. Namboodiri, "RNNP: A robust few-shot learning approach," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pp. 2663–2672, 2021.

[14] F. Li, R. Fergus, and P. Perona, "A bayesian approach to unsupervised one-shot learning of object categories," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1134–1141, 2003.

[15] R. Zhang, T. Che, Z. Ghahramani, Y. Bengio, and Y. Song, "Metagan: An adversarial approach to few-shot learning," in *Advances in Neural Information Processing Systems 31*, pp. 2371–2380, 2018.

[16] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[17] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, pp. 1126–1135, 2017.

[18] H. Ye and W. Chao, "How to train your MAML to excel in few-shot classification," in *Proceedings of the 9th International Conference on Learning Representations*, 2021.

[19] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems 29*, pp. 3630–3638, 2016.

[20] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems 30*, pp. 4077–4087, 2017.

[21] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018.

[22] V. G. Satorras and J. B. Estrach, "Few-shot learning with graph neural networks," in *Proceedings of the 6th International Conference on Learning Representations*, 2018.

[23] B. Zhang, X. Li, Y. Ye, Z. Huang, and L. Zhang, "Prototype completion with primitive knowledge for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3754–3762, 2021.

[24] W. Xue and W. Wang, "One-shot image classification by learning to restore prototypes," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pp. 6558–6565, 2020.

[25] S. Chen, J. Xue, J. Chang, J. Zhang, J. Yang, and Q. Tian, "SSL++: improving self-supervised learning by mitigating the proxy task-specificity problem," *IEEE Trans. Image Process.*, vol. 31, pp. 1134–1148, 2022.

[26] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. B. Girshick, "Masked autoencoders are scalable vision learners," *CoRR*, vol. abs/2111.06377, 2021.

[27] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9726–9735, 2020.

[28] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, pp. 1597–1607, 2020.

[29] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

[30] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Advances in Neural Information Processing Systems 33*, 2020.

[31] J. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. Á. Pires, Z. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent - A new approach to self-supervised learning," in *Advances in Neural Information Processing Systems 33*, 2020.

[32] Y. An, H. Xue, X. Zhao, and L. Zhang, "Conditional self-supervised learning for few-shot classification," in *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, pp. 2140–2146, 2021.

[33] K. Lee, "Prototypical contrastive predictive coding," in *Proceedings of the 10th International Conference on Learning Representations*, 2022.

[34] Y.-H. H. Tsai, T. Li, M. Q. Ma, H. Zhao, K. Zhang, L.-P. Morency, and R. Salakhutdinov, "Conditional contrastive learning with kernel," in *Proceedings of the 10th International Conference on Learning Representations*, 2022.

[35] B. Frénay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 25, no. 5, pp. 845–869, 2014.

[36] H. Song, M. Kim, D. Park, Y. Shin, and J. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 34, no. 11, pp. 8135–8153, 2023.

[37] B. Han, Q. Yao, T. Liu, G. Niu, I. W. Tsang, J. T. Kwok, and M. Sugiyama, "A survey of label-noise representation learning: Past, present and future," *CoRR*, vol. abs/2011.04406, 2020.

[38] L. Jiang, Z. Zhou, T. Leung, L. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *Proceedings of the 35th International Conference on Machine Learning*, pp. 2309–2318, 2018.

[39] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. W. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Advances in Neural Information Processing Systems 31*, pp. 8536–8546, 2018.

[40] J. Shu, Q. Xie, L. Yi, Q. Zhao, S. Zhou, Z. Xu, and D. Meng, "Meta-weight-net: Learning an explicit mapping for sample weighting," in *Advances in Neural Information Processing Systems 32*, pp. 1917–1928, 2019.

[41] S. E. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," in *Workshop Track Proceedings of the 3rd International Conference on Learning Representations*, 2015.

[42] G. Zheng, A. H. Awadallah, and S. T. Dumais, "Meta label correction for noisy label learning," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pp. 11053–11061, 2021.

[43] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness, "Unsupervised label noise modeling and loss correction," in *Proceedings of the 36th International Conference on Machine Learning*, pp. 312–321, 2019.

[44] Y. Hsu, Y. Shen, H. Jin, and Z. Kira, "Generalized ODIN: detecting out-of-distribution image without learning from out-of-distribution data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10948–10957, 2020.

[45] D. Krueger, E. Caballero, J. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. L. Priol, and A. C. Courville, "Out-of-distribution generalization via risk extrapolation (rex)," in *Proceedings of the 38th International Conference on Machine Learning*, vol. 139, pp. 5815–5826, 2021.

[46] A. Bendale and T. E. Boult, "Towards open world recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1893–1902, 2015.

[47] K. J. Joseph, S. H. Khan, F. S. Khan, and V. N. Balasubramanian, "Towards open world object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5830–5840, 2021.

[48] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, "Large-scale long-tailed recognition in an open world," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2537–2546, 2019.

[49] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *Proceedings of the 37th International Conference on Machine Learning*, pp. 9929–9939, 2020.

[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, pp. 5998–6008, 2017.

[51] A. Lapidoth, *A foundation in digital communication*. Cambridge University Press, 2017.

[52] G. Shalev, Y. Adi, and J. Keshet, "Out-of-distribution detection using multiple semantic label representations," in *Advances in Neural Information Processing Systems 31*, pp. 7386–7396, 2018.

[53] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. US Government printing office, 1964.

[54] H. Ye, D. Zhan, Y. Jiang, and Z. Zhou, "What makes objects similar: A unified multi-metric learning approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 5, pp. 1257–1270, 2019.

[55] C. McDiarmid *et al.*, "On the method of bounded differences," *Surveys in combinatorics*, vol. 141, no. 1, pp. 148–188, 1989.

[56] A. Bellet, A. Habrard, and M. Sebban, "Metric learning," *Synthesis lectures on artificial intelligence and machine learning*, vol. 9, no. 1, pp. 1–151, 2015.

[57] Q. Cao, Z. Guo, and Y. Ying, "Generalization bounds for metric and similarity learning," *Mach. Learn.*, vol. 102, no. 1, pp. 115–132, 2016.

[58] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun, "Learning discriminative reconstructions for unsupervised outlier removal," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1511–1519, 2015.

[59] M. Patacchiola, J. Turner, E. J. Crowley, M. F. P. O'Boyle, and A. J. Storkey, "Bayesian meta-learning for the few-shot setting via deep kernels," in *Advances in Neural Information Processing Systems 33*, 2020.

[60] H. Ye, H. Hu, D. Zhan, and F. Sha, "Few-shot learning via embedding adaptation with set-to-set functions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8805–8814, 2020.

[61] P. Mangla, M. Singh, A. Sinha, N. Kumari, V. N. Balasubramanian, and B. Krishnamurthy, "Charting the right manifold: Manifold mixup for few-shot learning," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pp. 2207–2216, 2020.

[62] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, "Meta-learning for semi-supervised few-shot classification," in *Proceedings of the 6th International Conference on Learning Representations*, 2018.

[63] A. Ayyad, Y. Li, R. Muaz, S. Albarqouni, and M. Elhoseiny, "Semi-supervised few-shot learning with prototypical random walks," in *AAAI Workshop on Meta-Learning and MetaDL Challenge*, vol. 140 of *Proceedings of Machine Learning Research*, pp. 45–57, 2021.

[64] L. Bertinetto, J. F. Henriques, P. H. S. Torr, and A. Vedaldi, "Meta-learning with differentiable closed-form solvers," in *Proceedings of the 7th International Conference on Learning Representations*, 2019.

[65] B. N. Oreshkin, P. R. López, and A. Lacoste, "TADAM: task dependent adaptive metric for improved few-shot learning," in *Advances in Neural Information Processing Systems 31*, pp. 719–729, 2018.

[66] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[67] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pp. 315–323, 2011.

[68] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

Yuexuan An Yuexuan An is currently pursuing the Ph.D. degree in the School of computer science and engineering, Southeast University. She received the B.Sc. in computer science and technology from Jiangsu Normal University in 2015 and M.Sc. degree in computer application technology from China University of Mining and Technology in 2019. Her research interest includes pattern recognition and machine learning.

Hui Xue Hui Xue is currently a professor of School of Computer Science and Engineering at Southeast University, China. She received the B.Sc. degree in Mathematics from Nanjing Norm University in 2002. In 2005, she received the M.Sc. degree in Mathematics from Nanjing University of Aeronautics & Astronautics (NUAA). And she also received the Ph.D. degree in Computer Application Technology at NUAA in 2008. Her research interests include pattern recognition and machine learning.

Xingyu Zhao Xingyu Zhao is currently pursuing the Ph.D. degree in the School of Computer Science and Engineering, Southeast University. He received the B.Sc. and M.Sc. degrees in School of Computer Science and Technology, China University of Mining and Technology in 2016 and 2019, respectively. His research interests mainly include pattern recognition and machine learning.

Jing Wang Jing Wang received the B.Sc. degree in computer science from the Suzhou University of Science and Technology, China, in 2013 and the M.Sc. degree in computer science from Northeastern University, China, in 2015 and the Ph.D. degree in software engineering from Southeast University, China, in 2021. He is currently an assistant professor at the School of Computer Science and Engineering, Southeast University, China. His research interests include pattern recognition and machine learning.