



Universiteit
Leiden
The Netherlands

Exploring deep learning for data-efficient and generalizable visual recognition

He, K.

Citation

He, K. (2026, September 22). *Exploring deep learning for data-efficient and generalizable visual recognition*. Retrieved from <https://hdl.handle.net/1887/4310276>

Version: Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/4310276>

Note: To cite this publication please use the final published version (if applicable).

Chapter 3

ProtoSimNet: Exploiting Prototype Similarity for Few-shot Learning

In the previous chapter we have introduced the background and research questions for this thesis. Starting with this chapter, we begin to answer the research questions with our proposed approaches. In this chapter, we address the question of how to design a unified framework for FSL that simultaneously enhances metric discriminability and ensures training stability (**RQ 1**).

Many few-shot learning (FSL) approaches exploit Euclidean Distance as the metric, as it provides a stable and well-understood geometric structure for prototype-based classification. However, cosine similarity has become increasingly popular in normalized embedding spaces due to its scale invariance and computational efficiency. Meanwhile, with limited supervision, cosine-based scoring often leads to overly smooth similarity distributions and weak angular decision boundaries, resulting in ambiguous predictions when distinguishing fine-grained novel categories. In this work, we explore a new prototype similarity network (ProtoSimNet) for FSL that emphasizes sharpening the angular geometry of the embedding space under data scarcity constraints. ProtoSimNet introduces a cosine sharpening transformation, combined with an elastic regularization, which together enhance discriminability by amplifying inter-class angular differences while stabilizing intra-class embedding consistency. Our method strengthens the reliability of cosine-based classification and improves the robustness of prototype matching when only a few labeled samples are available. Extensive experiments demonstrate our method outperforms other competitors in generalizing evaluations.

This chapter is based on the accepted paper:

- **He, K.**, Pu, N., Bakker, E. M., and Lew, M. S. “ProtoSimNet: Exploiting Prototype Similarity for Few-shot Learning.” accepted by the Neurocomputing journal, 2026.

3.1 Introduction

Over recent years, with the rapid development of big data, remarkable successes have been achieved in deep learning area for various computer vision tasks, such as image classification [229, 230], object detection [231, 232] and semantic segmentation [233, 234]. However, these deep learning models usually require a large number of labeled samples during the training process, which severely restricts its application in scenarios with limited data. Therefore, training a good model with only few labeled samples, named few-shot learning (FSL) [130, 235, 236], has received growing interest and has become an important topic in deep learning.

In general, these FSL methods can be roughly categorized into three types, metric-based [141], model-based [169, 176] and optimized-based methods [2, 115, 237]. Metric-based methods focus on finding a metric space, where query samples can be simply matched with the support set. Compared to other kinds of FSL methods, metric-based methods do not require to learn additional parameters once the metric is learned. Due to the promising performances, metric-based methods are taken as a breakthrough point and a series of studies have been proposed. In order to calculate the similarity between support and query samples, Sung *et al.* [24] introduced RelationNet which utilizes network to learn the relation scores between the support and query samples. Based on a memory module, cosine distance is applied as a metric in MatchingNet [14] to classify query samples. In order to achieve better classification performances, Hou *et al.* designed CAN [22] to highlight the proper regions of interest through an attention module. Besides, Zhang *et al.* [37] also proposed an algorithm named DeepEMD, which applies the Earth Mover’s Distance (EMD) as a similarity metric.

As a typical metric-based method, Prototypical Network achieves promising performance in few-shot classification. However, Euclidean prototypes can be sensitive to the distribution of support samples. When class prototypes are estimated from imbalanced or slightly distribution-shifted support sets, small perturbations in the support distribution may significantly shift the class prototypes in the embedding space and thus change the decision boundary, which potentially degrades generalization performance on new tasks. This motivates us to replace the Euclidean distance with another metric which can preserve stable decision boundaries.

In this paper, we propose a novel prototype similarity network (ProtoSimNet) for FSL tasks. ProtoSimNet consists of two complementary modules: a cosine sharpening transformation module and an elastic regularization module. 1) Cosine sharpening transformation module. We first introduce a prototype-level cosine sharpening transformation to refine similarity estimation. Specifically, we apply a non-linear sharpening function to the cosine similarities between query features and class prototypes, which preserves the original angular relationships while amplifying the gap

between high and low similarity scores. This operation effectively enlarges similarity margins and yields more distinct decision boundaries in the embedding space, thereby mitigating the sensitivity of prototype-based classifiers to imbalanced or mildly shifted support distributions. 2) Elastic regularization module. Although cosine sharpening transformation already brings noticeable improvements, we observe that it is still insufficient to fully address issues such as overfitting to base classes, label noise, and class imbalance. To further enhance robustness, we design an elastic regularization mechanism that explicitly shapes the feature space by encouraging compact within-class clusters and preventing excessive prototype drift across episodes. This regularization term is lightweight and can be seamlessly integrated into the episodic training objective. By jointly optimizing these two modules within ProtoSimNet, we obtain more robust feature representations and more stable decision boundaries, leading to substantial performance gains on both base and novel classes. The whole framework is built on a standard Conv4 backbone network and trained in an end-to-end episodic strategy. The comparison of different metrics is presented in Figure 3.1 and the overall pipeline diagram is shown in Figure 3.2.

The main contributions of our work can be concluded as follows: 1) We identify and systematically analyze a previously overlooked metric over-smoothing phenomenon in prototypical few-shot learning, where conventional distance metrics produce overly similar prototype scores and thus weak decision margins. To explicitly address this issue, we design a new metric-based framework, ProtoSimNet, specifically designed to alleviate metric over-smoothing and enhance the separability of prototypes. 2) We introduce a non-linear cosine sharpening transformation for prototype similarity, which induces an implicit margin without altering the classifier architecture. The transformation delivers stronger discriminability and improves model sensitivity to subtle inter-class differences. 3) We incorporate an elastic regularization that adaptively balances ℓ_1 sparsity constraints and ℓ_2 weight decay, effectively mitigating overfitting in data-scarce scenarios. This regularization strategy demonstrates particular efficacy in enhancing model robustness against class imbalance and label noise. 4) Extensive experiments on standardized FSL benchmarks demonstrate that our method achieves superior performance across various task configurations, including both 5-way 1-shot and 5-way 5-shot scenarios. The consistent improvements over conventional cosine and Euclidean distance metrics validate the effectiveness of our proposed approach.

3.2 Related Work

FSL Methods: Metric-based methods have witnessed great success due to their strong representation ability. These methods employ a learning-to-compare paradigm, matching features between support and query samples through a suitable metric function. As one of the most representative metric-based methods, Prototypical

3. PROTOSIMNET: EXPLOITING PROTOTYPE SIMILARITY FOR FEW-SHOT LEARNING

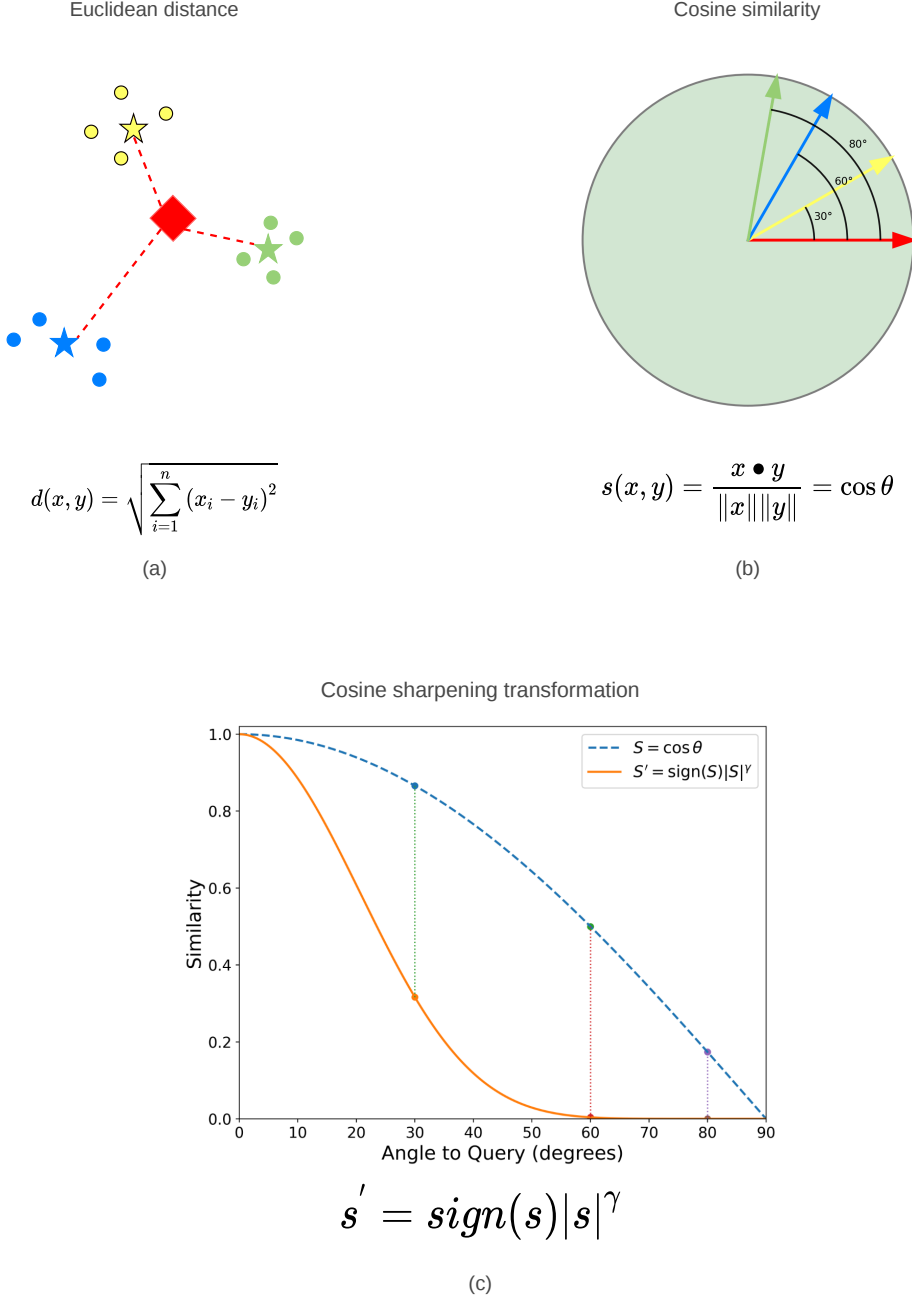


Figure 3.1: The comparison of different metrics. (a) Euclidean distance; (b) Cosine similarity; (c) Cosine sharpening transformation.

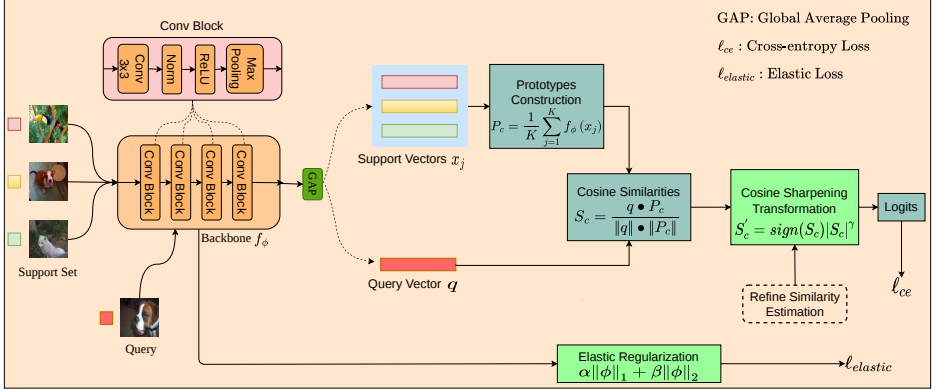


Figure 3.2: The overall pipeline diagram of our framework. The class prototypes, cosine similarities, cosine sharpening transformation, and elastic regularization are defined in Eq. 3.1, 3.2, 3.3, and 3.7, respectively. Their detailed implementations are illustrated in Figure 3.3

Network [1] adopts the mean of intra-class examples as the class prototype, classifying the query sample according to the Euclidean Distance between the query and class prototypes. Model-based methods encourage networks to quickly update the parameters through the design of the model structure. For example, Santoro *et al.* [169] explored a memory-augmented neural network (MANN) which uses an explicit storage buffer to learn unseen classes via an LSTM-based meta-learner. For rapid learning and generalization, Munkhdalai and Yu [176] introduced an advanced weight update scheme by generating fast and slow weights and incorporated them into a neural network. Optimized-based methods try to learn a model with a good initialization from base classes and then adapt the learned model to novel classes through few optimization steps. Model-Agnostic Meta learning (MAML) [2] aims to learn a good parameter initialization by involving the second-order gradients, enabling the model to rapidly adapt to new tasks with a few gradient steps. In order to obtain better generalization for novel classes, Lee *et al.* [237] proposed to utilize properties of linear classifiers to learn feature representations.

Prototypical Network: As one of the most representative methods among metric-based learning, Prototypical Network denotes the mean embeddings of each support class as class prototypes, and the evaluation rule is based on the Euclidean Distance to the nearest class prototype. Despite the outstanding performance of Prototypical Network, several studies have argued that the learned class prototypes may be biased and not efficient when given only few labeled examples. To alleviate the influence caused by biased data distribution, Ji *et al.* [238] proposed IPN, which uses a softmax function to generate weights for the intra-class feature vectors. A metric scaling method [23] is also adopted into the objective function to obtain a task-dependent metric space. Ye *et al.* [239] hold the view that backbone networks may accidentally select uncharacteristic features since classification tasks are agnostic to the current

model. Hence, they utilized a self-attention mechanism to the mean prototypes to yield task-specific prototypes to address the issue. Huang *et al.* [240] believe that a single prototype is insufficient to represent the characteristics of the class. Based on this point, they proposed to learn multiple prototypes of each class via a learnable squeeze and spatial excitation module (sSE) [241].

3.3 Proposed Method

To address the limitations of common similarity metrics in FSL, we propose a novel framework that integrates cosine sharpening transformation with elastic regularization. Our approach employs a unified episodic training strategy with a standard Conv4 backbone, eliminating the need for complex multi-stage optimization while enhancing the discriminative ability of prototype-based classification.

3.3.1 Problem Formulation

In the FSL paradigm, we take a base dataset as $D^{base} = \{x_i^{base}, y_i^{base}\}_{i=1}^{N^{base}}$ with labels $y_i^{base} \in C^{base}$, where both the number of classes $|C^{base}|$ and instances N^{base} are sufficiently large to learn transferable feature representations. For evaluation, we utilize a novel dataset $D^{novel} = (x_j^{novel}, y_j^{novel})_{j=1}^{N^{novel}}$ with $y_j^{novel} \in C^{novel}$, where $C^{base} \cap C^{novel} = \emptyset$ ensures disjoint class spaces between training and testing phases.

Both the training and testing process follow an episodic scheme, where each episode E_i consists of a support set $S_i = (x_i^{t_{support}}, y_i^{t_{support}})_{t=1}^{C \times K}$ and a query set $Q_i = (x_i^{t_{query}}, y_i^{t_{query}})_{t=1}^{C \times K'}$, containing C classes with K examples per class from D^{base} and K' examples per class from D^{novel} , respectively.

3.3.2 Cosine Sharpening Transformation

To enhance the metric’s sensitivity, we introduce a novel cosine sharpening transformation that nonlinearly amplifies the distinction between strong and weak similarities.

Given K support samples from the c -th class, based on the backbone f_ϕ , the c -th class prototype P_c can be calculated as:

$$P_c = \frac{1}{K} \sum_{j=1}^K f_\phi(x_j), \quad (3.1)$$

For any query embedding q , the traditional cosine similarity S_c between the query sample q and the class prototype P_c is computed as:

$$S_c = \frac{q \bullet P_c}{\|q\| \bullet \|P_c\|}, \quad (3.2)$$

where $\|\cdot\|$ is the Euclidean norm and $\bullet$ represents the standard dot product. The cosine similarity measures the angular alignment between the query representation and each class prototype, while reducing the influence of feature magnitude.

The cosine sharpening transformation exploits a signed power operation to reshape the cosine similarity distribution:

$$S'_c = \text{sign}(S_c) |S_c|^\gamma, \quad (3.3)$$

where $\gamma > 1$ is a sharpening factor. This transformation performs a non-linear contrast adjustment on the original cosine similarity as it nonlinearly enhances the contrast between highly relevant and marginally relevant pairs. Specifically, the sign function is introduced to preserve the sign of the original cosine similarity and γ controls the strength of the non-linear calibration. The sharpened similarities are subsequently calibrated through temperature scaling τ to form the final logits for classification:

$$\text{logits}_c = \tau S'_c, \quad (3.4)$$

where τ is a parameter that controls the temperature scaling of the distribution sharpness, and class posteriors follow can be obtained via the following softmax function:

$$p(y = c | x) = \frac{\exp(\text{logits}_c)}{\sum_{j=1}^N \exp(\text{logits}_j)}, \quad (3.5)$$

The query loss in an episode is defined as:

$$\ell_{ce} = -\frac{1}{|Q_i|} \sum_{(x,y) \in Q_i} \log p(y | x). \quad (3.6)$$

The whole workflow of cosine sharpening transformation is presented in Figure 3.3a. This formulation allows the module to emphasize discriminative similarity patterns without introducing excessive computational source, making it particularly suitable for data-scarce FSL scenarios.

3.3.3 Elastic Regularization

To alleviate the overfitting problem in data-scarce scenarios and to obtain robust feature representations, we further introduce an elastic regularization term into our framework. In FSL, the backbone is trained on a limited set of base classes. This often leads to unstable prototypes and poor generalization to novel classes when there exists imbalanced distribution in the base set. Elastic regularization aims to mitigate this problem by explicitly constraining the parameter space during episodic training.

Specifically, elastic regularization combines the complementary effects of ℓ_2 smoothness and ℓ_1 sparsity. The ℓ_1 component enables many parameters to become exactly or approximately zero, effectively performing feature selection by suppressing

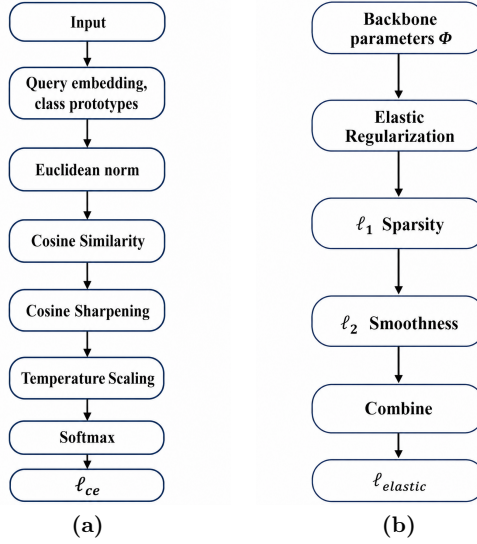


Figure 3.3: Module-level visualization. (a) Cosine sharpening module. (b) Elastic regularization.

uninformative or redundant feature dimensions in the embedding space. The ℓ_2 component, on the other hand, penalizes excessively large weights and promotes smoothness of the learned mapping, which stabilizes optimization and prevents the model from overfitting to noise or outliers in the support set. For the backbone parameters ϕ , the regularizer is:

$$\ell_{elastic} = \alpha \|\phi\|_1 + \beta \|\phi\|_2, \quad (3.7)$$

where $\|\phi\|_1 = \sum_{\phi_i \in \phi} |\phi_i|$, $\|\phi\|_2 = \left(\sum_{\phi_i \in \phi} \phi_i^2 \right)^{1/2}$. α and β are hyperparameters controlling the relative contributions of the ℓ_1 and ℓ_2 regularization terms. The workflow of elastic regularization is presented in Figure 3.3b.

Within the proposed ProtoSimNet framework, the elastic regularization term is optimized jointly with the standard episodic classification loss. This joint optimization gradually shapes the embedding space into one with tighter intra-class clusters and clearer inter-class separation, while also reducing redundancy across feature dimensions. As a result, the learned representations exhibit stronger robustness to label noise, class imbalance, and base/novel distribution mismatch, and the resulting prototypes become more stable across episodes and more discriminative on novel few-shot tasks. The overall objective of ProtoSimNet can be written as:

$$\ell = \ell_{ce} + \ell_{elastic}, \quad (3.8)$$

which enables tighter intra-class clusters and reduces redundancy in the embedding

space.

3.4 Experiments

In this section, we elaborate on the experimental configuration and evaluation. The description of the datasets and implementation details are first presented, followed by experimental results of our approach on popular benchmark datasets. Finally, an ablative analysis is presented.

3.4.1 Datasets and Implementation details

Datasets: We evaluate our approach on three widely used few-shot learning benchmarks: mini-ImageNet [14], tieredImageNet [62] and CUB-200 [113]. The mini-ImageNet dataset is a subset of ImageNet, consisting of 100 categories, each with 600 images. We adopt the data splitting protocol proposed in [115], where 64 classes are allocated for training, 16 classes for validation, and the remaining 20 classes for testing. The tieredImageNet dataset comprises 608 classes, which are further organized into 34 high-level categories. We use 20 categories (351 classes) for training, 6 categories (97 classes) for validation, and 8 categories (160 classes) for testing. The CUB dataset was originally introduced for fine-grained bird classification and contains 11,788 images from 200 classes. We use 100 classes for training, 50 classes for validation, and 50 classes for testing.

Implementation details: in this work, we use Conv4 as our backbone, which is composed of four convolutional blocks. Each block comprises a 64-filter 3×3 convolution, batch normalization layer, a ReLU layer and a 2×2 max-pooling layer. We use Adam with an initial learning rate of $1e-3$ and cut the learning rate in half every 2000 episodes. Each episode includes 60 classes, with 5 support samples and 5 query samples. Besides, we use random cropping, color jittering and random horizontal flip for data augmentation during the whole process.

3.4.2 Evaluations

miniImageNet and tieredImageNet. Table 3.1 presents a comparison between our approach and the state-of-the-art methods in FSL tasks on the two ImageNet-based benchmarks. Our method is denoted as ProtoSimNet. On both datasets and in both 1-shot and 5-shot scenarios, our approach yields state-of-the-art results. On miniImageNet, ProtoSimNet achieves 59.48% and 79.75% in 5-way 1-shot and 5-way 5-shot tasks, respectively. ProtoSimNet achieves improvements over the best-reported numbers by 0.52% under 5-way 1-shot and 4.58% under 5-way 5-shot settings. On tieredImageNet, ProtoSimNet achieves 59.71% and 79.40% in the 1-shot and 5-shot tasks, respectively. ProtoSimNet achieves improvements over the best-reported numbers by 1.51% under 5-way 1-shot and 3.02% under 5-way 5-shot settings.

3. PROTO SIMNET: EXPLOITING PROTOTYPE SIMILARITY FOR FEW-SHOT LEARNING

Table 3.1: Comparison of ProtoSimNet (Conv4-64) to prior works on miniImageNet and tieredImageNet datasets, with mean accuracy (%) and 95% confidence interval.

Model	Backbone	miniImageNet 5-way 1-shot	miniImageNet 5-way 5-shot	tieredImageNet 5-way 1-shot	tieredImageNet 5-way 5-shot
ProtoNet [1]	Conv4-64	49.42 $\pm$ 0.78	68.20 $\pm$ 0.66	53.31 $\pm$ 0.89	72.69 $\pm$ 0.74
MatchingNet [14]	Conv4-64	43.56 $\pm$ 0.84	55.31 $\pm$ 0.73	–	–
RelationNet [24]	Conv4-64	50.44 $\pm$ 0.82	65.32 $\pm$ 0.70	54.48 $\pm$ 0.93	71.31 $\pm$ 0.78
CovaMNet [151]	Conv4-64	51.19 $\pm$ 0.76	67.65 $\pm$ 0.63	54.98 $\pm$ 0.90	71.51 $\pm$ 0.75
DN4 [116]	Conv4-64	51.24 $\pm$ 0.74	71.02 $\pm$ 0.64	52.89 $\pm$ 0.23	73.36 $\pm$ 0.73
MetaOptNet [237]	Conv4-64	52.87 $\pm$ 0.57	68.76 $\pm$ 0.48	54.71 $\pm$ 0.67	71.79 $\pm$ 0.59
DeepEMD [37]	Conv4-64	51.72 $\pm$ 0.20	65.10 $\pm$ 0.39	51.22 $\pm$ 0.14	65.81 $\pm$ 0.68
RFS-Simple [110]	Conv4-64	55.25 $\pm$ 0.58	71.56 $\pm$ 0.52	56.18 $\pm$ 0.70	72.99 $\pm$ 0.55
RFS-Distill [110]	Conv4-64	55.88 $\pm$ 0.59	71.65 $\pm$ 0.51	56.76 $\pm$ 0.68	73.21 $\pm$ 0.54
FRN [122]	Conv4-64	54.87	71.56	55.54	74.68
DMN4 [242]	Conv4-64	55.77	74.22	56.99	74.13
ATL-Net [243]	Conv4-64	54.30 $\pm$ 0.76	73.22 $\pm$ 0.63	–	–
TALDS-Net [244]	Conv4-64	56.78 $\pm$ 0.37	74.63 $\pm$ 0.13	57.54 $\pm$ 0.71	75.79 $\pm$ 0.43
MCL [245]	Conv4-64	57.88	74.03	58.20	74.77
HTS [246]	Conv4-64	58.96 $\pm$ 0.18	75.17 $\pm$ 0.14	53.20 $\pm$ 0.22	72.38 $\pm$ 0.19
TADNet [247]	Conv4-64	56.14 $\pm$ 0.20	74.68 $\pm$ 0.15	57.88 $\pm$ 0.21	75.98 $\pm$ 0.17
DCRNet [248]	Conv4-64	56.35 $\pm$ 0.31	74.93 $\pm$ 0.17	58.04 $\pm$ 0.21	76.38 $\pm$ 0.23
ProtoSimNet	Conv4-64	59.48 $\pm$ 0.63	79.75 $\pm$ 0.42	59.71 $\pm$ 0.74	79.40 $\pm$ 0.47

CUB-200 dataset. As shown in Table 3.2, our method also achieves state-of-the-art performance on CUB-200 dataset. In the 5-way 1-shot and 5-shot settings, when using Conv4-64 as the backbone, ProtoSimNet improves by 0.33% and 0.42%, respectively, compared to the state-of-the-art method based on HTS [246]. To evaluate the cross-domain generalization of ProtoSimNet, we conducted experiments under the miniImageNet→CUB setting (see Table 3.3) and compared it with state-of-the-art methods. The model is trained on 64 base classes of miniImageNet and evaluated on 50 novel classes in the CUB-200 test set. Our ProtoSimNet demonstrates significant advantages in this cross-domain scenario. Using the ResNet12 backbone, ProtoSimNet maintains a lead in both 5-way 1-shot and 5-way 5-shot settings compared to other methods.

3.4.3 Ablative Analysis

Backbone Dependence. To address backbone dependence, we additionally evaluate our method on a stronger backbone, ResNet12. These new results are provided in Table 3.4. Experimental results indicate that the proposed ProtoSimNet is not tied to a specific backbone.

Benefits of Cosine Sharpening Transformation. To study the impact of cosine sharpening transformation, we evaluate our approach with ProtoNet and DeepEMD on miniImageNet. As shown in Table 3.1, compared to ProtoNet which uses Euclidean distance as the metric, ProtoSimNet obtains 10.06% and 11.55% improvements on 5-way 1-shot and 5-way 5-shot tasks, respectively. Compared to DeepEMD, ProtoSimNet also gains 7.76% and 14.65% improvements on 5-way 1-shot and 5-way 5-shot tasks, respectively. Furthermore, we make a comparison

Table 3.2: Comparison of ProtoSimNet (Conv4-64) to prior works on CUB-200 dataset, with mean accuracy (%) and 95% confidence interval.

Model	Backbone	CUB-200 5-way	
		1-shot	5-shot
ProtoNet [1]	Conv4-64	37.36 $\pm$ 1.00	45.28 $\pm$ 1.03
MatchingNet [14]	Conv4-64	45.30 $\pm$ 1.03	59.50 $\pm$ 1.01
CSS [249]	Conv4-64	66.01 $\pm$ 0.90	81.84 $\pm$ 0.59
DN4 [116]	Conv4-64	46.84 $\pm$ 0.81	74.92 $\pm$ 0.62
MADN4 [250]	Conv4-64	57.11 $\pm$ 0.70	77.83 $\pm$ 0.40
HTS [246]	Conv4-64	67.32 $\pm$ 0.24	78.09 $\pm$ 0.15
BDLA [251]	Conv4-64	50.59 $\pm$ 0.97	75.36 $\pm$ 0.72
DLDA [252]	Conv4-64	55.12 $\pm$ 0.86	74.46 $\pm$ 0.65
KLSANet [253]	Conv4-64	66.70 $\pm$ 0.82	83.63 $\pm$ 0.28
FAFD-LDWR [254]	Conv4-64	65.45 $\pm$ 0.67	79.63 $\pm$ 0.33
ProtoSimNet	Conv4-64	67.65 $\pm$ 0.84	84.05 $\pm$ 0.49

Table 3.3: Cross-domain performance comparison of the proposed ProtoSimNet with state-of-the-art methods under miniImageNet $\rightarrow$ CUB-200 setting, with mean accuracy (%) and 95% confidence interval.

Model	Backbone	miniImageNet $\rightarrow$ CUB-200	
		5-way 1-shot	5-way 5-shot
MIFN [255]	ResNet12	48.21 $\pm$ 0.60	65.33 $\pm$ 0.54
KLSANet [253]	ResNet12	48.16 $\pm$ 0.64	67.25 $\pm$ 0.61
FAFD-LDWR [254]	ResNet12	48.64 $\pm$ 0.42	67.36 $\pm$ 0.55
ProtoSimNet	ResNet12	49.38 $\pm$ 0.68	67.83 $\pm$ 0.53

with other cosine-metric methods. The results are presented in Table 3.5. Table 3.5 shows that our ProtoSimNet obtains the best performance on both 5-way 1-shot and 5-way 5-shot tasks on miniImageNet. From the results, we can infer the effectiveness of cosine sharpening transformation during the training process.

Benefits of ElasticNet Loss. To better evaluate the contribution of ElasticNet Loss, we train the model with different loss on miniImageNet. From Table 3.6, we can find that, for the model trained without ElasticNet Loss, it achieves 58.56% and 78.28% in 1-shot and 5-shot tasks, respectively. For the model trained with ℓ_1 or ℓ_2 regularization only, the experimental results demonstrate consistent improvements of varying magnitudes. In particular, the model trained with ElasticNet loss yields the most significant improvement over the prior settings. These results confirm the importance of ElasticNet Loss.

Module contribution. To investigate the contribution of each module, we evaluate our method with different modules. Figure 3.4 shows embedding visualization (t-SNE) on base and novel classes with different models. We can find that samples

3. PROTOSIMNET: EXPLOITING PROTOTYPE SIMILARITY FOR FEW-SHOT LEARNING

Table 3.4: Comparison of ProtoSimNet (ResNet12) to prior works on 5-way 5-shot tasks, with mean accuracy (%) and 95% confidence interval.

Model	Backbone	miniImageNet	tieredImageNet
ProtoNet [1]	ResNet12	80.88 $\pm$ 0.41	84.06 $\pm$ 0.26
MatchingNet [14]	ResNet12	75.99 $\pm$ 0.15	80.60 $\pm$ 0.71
DN4 [116]	ResNet12	81.10	83.41
FEAT [239]	ResNet12	82.05	84.79
BML [256]	ResNet12	83.63 $\pm$ 0.29	85.49 $\pm$ 0.34
FRN [122]	ResNet12	82.83 $\pm$ 0.13	86.89 $\pm$ 0.14
TPMN [257]	ResNet12	83.44 $\pm$ 0.43	86.55 $\pm$ 0.63
DeepEMD [37]	ResNet12	82.41 $\pm$ 0.56	86.03 $\pm$ 0.58
MCL [245]	ResNet12	83.99	86.02
DMN4 [242]	ResNet12	83.52	85.72
DeepBDC [160]	ResNet12	84.46 $\pm$ 0.28	87.31 $\pm$ 0.22
TADNet [247]	ResNet12	84.23 $\pm$ 0.67	86.46 $\pm$ 0.15
Meta-OLE [258]	ResNet12	82.23 $\pm$ 0.67	85.51 $\pm$ 0.59
CORL [259]	ResNet12	83.03 $\pm$ 0.33	86.76 $\pm$ 0.52
LEADNET [260]	ResNet12	83.21 $\pm$ 0.14	86.60 $\pm$ 0.16
ProtoSimNet	ResNet12	84.70 $\pm$ 0.35	88.40 $\pm$ 0.37

Table 3.5: Comparison of ProtoSimNet (Conv4-64) with other cosine-metric methods.

Model	Metric	miniImageNet 1-shot	5-way 5-shot
DN4 [116]	cosine similarity	51.24 $\pm$ 0.74	71.02 $\pm$ 0.64
DMN4 [242]	cosine similarity	55.77	74.22
TCDSNet [261]	cosine similarity	57.14 $\pm$ 0.22	75.89 $\pm$ 0.35
LDCA [262]	cosine similarity	53.46 $\pm$ 0.63	75.06 $\pm$ 0.48
TALDS-Net [244]	cosine similarity	56.78 $\pm$ 0.37	74.63 $\pm$ 0.13
ProtoSimNet	cosine sharpening transformation	59.48 $\pm$ 0.63	79.75 $\pm$ 0.42

Table 3.6: ProtoSimNet (Conv4-64) performances on miniImageNet, with different regularization terms.

Loss Function	miniImageNet 1-shot	5-way 5-shot
ProtoSimNet (without Elastic Loss)	58.56 $\pm$ 0.64	78.28 $\pm$ 0.43
ProtoSimNet (with ℓ_1 regularization only)	58.68 $\pm$ 0.61	78.53 $\pm$ 0.41
ProtoSimNet (with ℓ_2 regularization only)	59.14 $\pm$ 0.61	78.77 $\pm$ 0.42
ProtoSimNet (with Elastic Loss)	59.48 $\pm$ 0.63	79.75 $\pm$ 0.42

from same classes tend to form tighter, more compact clusters, while different classes become more clearly separated after adding the cosine sharpening transformation and elastic regularization modules. We use the model without any module as the baseline, which uses a Conv4 as the backbone. To obtain better performance, the

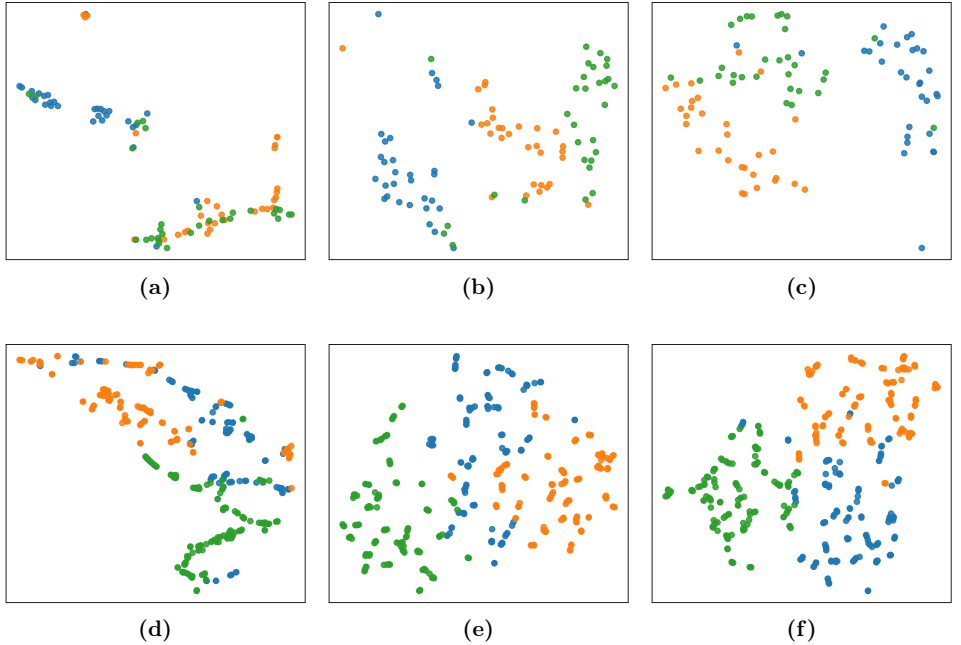


Figure 3.4: Qualitative embedding visualization (t-SNE) on base and novel classes. (a) Base Classes (Cosine Similarity). (b) Base Classes (Cosine Sharpening Transformation). (c) Base Classes (Cosine Sharpening Transformation + Elastic Regularization). (d) Novel Classes (Cosine Similarity). (e) Novel Classes (Cosine Sharpening Transformation). (f) Novel Classes (Cosine Sharpening Transformation + Elastic Regularization).

Table 3.7: ProtoSimNet (Conv4-64) performances on miniImageNet, with different module.

Model	miniImageNet 5-way	
	1-shot	5-shot
baseline	57.98 ± 0.65	77.56 ± 0.44
baseline + cosine sharpening transformation	58.56 ± 0.64	78.28 ± 0.43
baseline + cosine sharpening transformation + elastic regularization	59.48 ± 0.63	79.75 ± 0.42

baseline simulates FSL setting and takes 5 samples per class as the support set and another 5 samples as the query set in each episode during the training process. The cosine similarity is used as the metric for evaluation. From Table 3.7, we find that the baseline achieves 57.98% and 77.56% accuracies on miniImageNet 5-way 1-shot and 5-shot tasks, respectively. By adding the cosine sharpening transformation module, the model obtains 0.58% and 0.72% improvements. The elastic regularization helps the model further get 0.92% and 1.47% gains.

Variations of Hyper-parameters. There are totally four hyper-parameters in

3. PROTOSIMNET: EXPLOITING PROTOTYPE SIMILARITY FOR FEW-SHOT LEARNING

Table 3.8: ProtoSimNet (Conv4-64) performances on miniImageNet, with different γ values.

γ	miniImageNet 5-way	
	1-shot	5-shot
1.2	59.12 ± 0.63	78.49 ± 0.43
1.5	59.48 ± 0.63	79.75 ± 0.42
2	59.43 ± 0.64	79.47 ± 0.42
5	58.45 ± 0.64	79.27 ± 0.43
10	57.63 ± 0.63	79.05 ± 0.43

Table 3.9: ProtoSimNet (Conv4-64) performances on miniImageNet, with different α values.

α	miniImageNet 5-way	
	1-shot	5-shot
0.0001	58.15 ± 0.64	79.21 ± 0.41
0.0005	59.48 ± 0.63	79.75 ± 0.42
0.001	58.90 ± 0.66	78.62 ± 0.42
0.005	58.27 ± 0.62	75.27 ± 0.51
0.01	58.80 ± 0.67	71.46 ± 0.52
0.05	57.86 ± 0.64	59.49 ± 0.62

this work, τ , α , β and γ . τ is a temperature coefficient, γ is a sharpening factor, α and β are the weights of ℓ_1 and ℓ_2 regularization terms, respectively. We investigate the variants of hyper-parameters α , β and γ , where the default values of τ is set to 25.0. Table 3.8 shows the ProtoSimNet performance on miniImageNet by changing γ . It’s obviously that our model obtains over 59% accuracies on 5-way 1-shot tasks when γ range from 1.2 to 2, and then the performance decreases when γ increases from 2 till 10, which indicates the importance of sharpen scale. Table 3.9 shows the ProtoSimNet performance on miniImageNet by changing α . We observe that ProtoSimNet achieves over 58% accuracies on 5-way 1-shot tasks when α increases from 0.0001 till 0.01, while $\alpha = 0.05$ performs the lowest. Table 3.10 presents the ProtoSimNet performance on miniImageNet by changing β . When β ranges from 0.0001 till 0.001, ProtoSimNet can always obtains over 79% accuracies on 5-way 5-shot tasks and achieves the highest accuracy when $\beta = 0.0005$, while $\beta = 0.05$ performs the lowest, indicating a large value of β is not beneficial for the model. We also study ProtoSimNet performance with different values of τ . As shown in Table 3.11, when τ ranges from 10 to 30, ProtoSimNet always obtains over 58% accuracies on miniImageNet 5-way 1-shot tasks and over 76% accuracies on miniImageNet 5-way 5-shot tasks, respectively.

Table 3.10: ProtoSimNet (Conv4-64) performances on miniImageNet, with different β values.

β	miniImageNet 5-way	
	1-shot	5-shot
0.0001	58.94 ± 0.66	79.36 ± 0.43
0.0005	59.48 ± 0.63	79.75 ± 0.42
0.001	59.27 ± 0.65	79.27 ± 0.43
0.005	58.59 ± 0.62	77.77 ± 0.42
0.01	57.90 ± 0.63	77.34 ± 0.43
0.05	57.77 ± 0.62	76.86 ± 0.44

Table 3.11: ProtoSimNet (Conv4-64) performances on miniImageNet, with different τ values.

τ	miniImageNet 5-way	
	1-shot	5-shot
5	56.08 ± 0.63	72.02 ± 0.47
10	58.39 ± 0.62	76.58 ± 0.44
15	58.90 ± 0.63	79.18 ± 0.42
20	59.01 ± 0.64	79.13 ± 0.41
25	59.48 ± 0.63	79.75 ± 0.42
30	58.74 ± 0.65	79.20 ± 0.43

3.5 Chapter Conclusions

In this chapter, we have presented a comprehensive investigation into the limitations of traditional similarity metrics in few-shot learning scenarios, with particular focus on their sensitivity to data distribution shifts and limited discriminative ability when dealing with complex class relationships. In prototype-based few-shot learning, each class prototype acts like a “center of mass” computed from a few support examples, and classification is determined by which prototype is closest. If the support set is imbalanced or slightly shifted, that “center of mass” can move a lot, even when the underlying class has not changed—causing the separating boundary between classes to drift. Correspondingly, we introduce a novel prototype similarity metric framework that integrates two key innovations: a cosine sharpening transformation and an elastic regularization mechanism. The cosine sharpening transformation emphasizes angular similarity between features and encourages clearer separation in the similarity space, while elastic regularization further limits how freely prototypes can shift in response to small support perturbations. In combination, these contributions make the decision boundary less prone to drifting, improving generalization to new tasks.

Future work. We first plan to investigate adaptive mechanisms for automatically

3. PROTOSIMNET: EXPLOITING PROTOTYPE SIMILARITY FOR FEW-SHOT LEARNING

tuning the sharpening factor γ based on task characteristics and data distributions, and then extend the approach to more challenging cross-domain and incremental FSL scenarios. The proposed method provides a solid foundation for developing more effective and practical few-shot learning systems while maintaining the simplicity that makes prototype-based methods appealing.