Tailoring Embedding Function to Heterogeneous Few-Shot Tasks by Global and Local Feature Adaptors*

Su Lu, Han-Jia Ye, De-Chuan Zhan
State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, 210023, China
{lus, yehj}@lamda.nju.edu.cn, zhandc@nju.edu.cn

Abstract

Few-Shot Learning (FSL) is essential for visual recognition. Many methods tackle this challenging problem via learning an embedding function from seen classes and transfer it to unseen classes with a few labeled instances. Researchers recently found it beneficial to incorporate task-specific feature adaptation into FSL models, which produces the most representative features for each task. However, these methods ignore the diversity of classes and apply a global transformation to the task. In this paper, we propose **G**lobal and **Lo**cal **F**eature **A**daptor (GLoFA), a unifying framework that tailors the instance representation to specific tasks by global and local feature adaptors. We claim that class-specific local transformation helps to improve the representation ability of feature adaptor. Global masks tend to capture sketchy patterns, while local masks focus on detailed characteristics. A strategy to measure the relationship between instances adaptively based on the characteristics of both tasks and classes endow GLoFA with the ability to handle mix-grained tasks. GLoFA outperforms other methods on a heterogeneous task distribution and achieves competitive results on benchmark datasets.

Introduction

Modern deep learning systems have achieved unprecedented success in various fields. Their requirements for a large amount of labeled data impedes deep models' applications when limited examples are available. Few-shot learning (FSL) aims to endow a learner with the ability to generalize well from a small number of training examples. In FSL, we often assume that a sizeable related dataset which contains SEEN classes is available. After extracting some transferable knowledge from this dataset, the model can identify UNSEEN classes with only a few labeled instances.

Many FSL methods try to learn a generalizable embedding function from SEEN classes (Koch, Zemel, and Salakhutdinov 2015; Vinyals et al. 2016; Snell, Swersky, and Zemel 2017; Ye, Lu, and Zhan 2020). The main limitation of these methods is that a single embedding space shared by all tasks may not work well when target tasks differ a lot from each other. We should emphasize different feature dimensions when solving different tasks, which is the reason that many recent methods, including our proposed GLoFA, focus on task-specific features. Given tasks sampled from a latent distribution $p(\mathcal{T})$, we can learn a feature adaptor $p(\mathcal{M}|\mathcal{T})$, which captures the characteristics of $\mathcal{T}$ and outputs the most relevant features $\mathcal{M}$ to this task. Researchers implement the feature adaptor in different ways, such as category traversal module (Li et al. 2019), task encoding network (Oreshkin, López, and Lacoste 2018), set-to-set func-

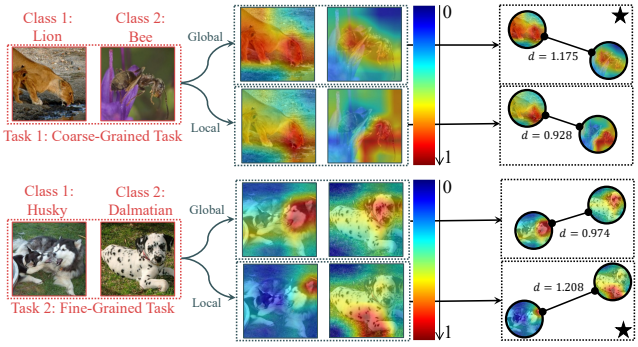


Figure 1: Different effects of global and local masks on a coarse-grained task and a fine-grained task. In this experiment, we train our model on the *meta-training* set of *mini*ImageNet. We sample two tasks from the *meta-testing* set. One task is differing a lion from a bee, and another is differing a husky from a dalmatian. We apply global masks and local masks generated by GLoFA to the instances. Masks are normalized to $[0, 1]$, and highlight warm-colored regions. We can see that global masks tend to focus on sketchy patterns, *e.g.*, the whole body of lion and bee. These patterns are enough to discriminate two classes when there is a large semantic gap between them. Local masks catch detailed characteristics, *e.g.*, ear of the husky and leg of the dalmatian. For a fine-grained task, local masks separate two similar concepts. For each task, we calculate the euclidean distance between two masked instances. Global masks push the bee far from the lion, while local masks work better to distinguish two dogs. Since global and local masks have different effects, we combine them in GLoFA and fuse them adaptively based on the target task so that GLoFA emphasizes appropriate masks for different tasks.

*Han-Jia Ye is the corresponding author. This work is supported by NSFC (61773198, 6163000043, 61921006, 62006112).

tion (Ye et al. 2020), and dynamic subspaces (Simon et al. 2020).

Whatever the task descriptor is implemented as, existing methods apply a shared transformation to the entire task. As indicated by (kyun Noh, tak Zhang, and Lee 2018; Wang, Kalousis, and Woznica 2012), the discriminatory power of the features might vary between different classes, and a global metric space may not fit the distance over the data manifold. Inspired by this, we propose global and local feature adaptors to capture the characteristics of both entire task and each class. We generate task-level and class-level feature masks for each task. Class-wise masks project instances of each class into several local spaces. In GLoFA, global spaces and class-wise local spaces are learned simultaneously and fused adaptively. In Figure 1, we show that different masks are emphasized by GLoFA for different tasks.

GLoFA contains three components. Firstly, there is an embedding network to extract vector features from raw data. All the downstream operations are performed on these extracted features. Secondly, there are two feature adaptors for tailoring embeddings to heterogeneous tasks at task-level and class-level. At each level, the feature adaptor is implemented as a permutation-invariant function. Thirdly, a mask combiner automatically fuses global and local masks based on the target task context. The outputs of feature adaptors are balanced by this mask combiner.

On tasks with mixed granularity, GLoFA outperforms existing methods because global and local feature masks are optimized, and the importance of general patterns and details is appropriately adjusted. GLoFA also achieves competitive performance on several FSL benchmark datasets.

In summary, our contributions are threefold:

- Different from existing methods, we consider classes' diversity, and apply local transformations to each class.
- We investigate different effects of global and local masks, and propose a mask combiner to adjust their importance.
- We empirically demonstrate the effectiveness of GLoFA on heterogeneous tasks and benchmark datasets.

Related Work

Meta-learning (Thrun and Pratt 2012) aims at extracting task-level experience (so-called meta-knowledge) from seen data, while generalizing the learned meta-knowledge to unseen tasks efficiently. It acts as one main tool for few-shot learning (Dai et al. 2017; Liu, Wang, and Zhang 2019), where the few-shot facilitated external memory (Graves, Wayne, and Danihelka 2014; Santoro et al. 2016; Munkhdalai et al. 2019), shared embedding (Vinyals et al. 2016; Snell, Swersky, and Zemel 2017; Lee et al. 2019) or optimization strategy (Finn, Abbeel, and Levine 2017) are meta-learned and reused. Among these algorithms, metric-based meta-learning achieves promising performance in FSL. This line of works projects instances into a task-specific embedding space by feature adaptation (Oreshkin, López, and Lacoste 2018; Ye et al. 2020; Li et al. 2019; Ravichandran, Bhotika, and Soatto 2019). Although the feature adaptors are implemented differently in existing methods, the learned transformation is shared by all the classes in a task. It is natural to emphasize different feature dimensions for each class, and a local feature mask may be beneficial for capturing detailed characteristics. Different from existing methods, GLoFA selects global and local features simultaneously in a unifying framework.

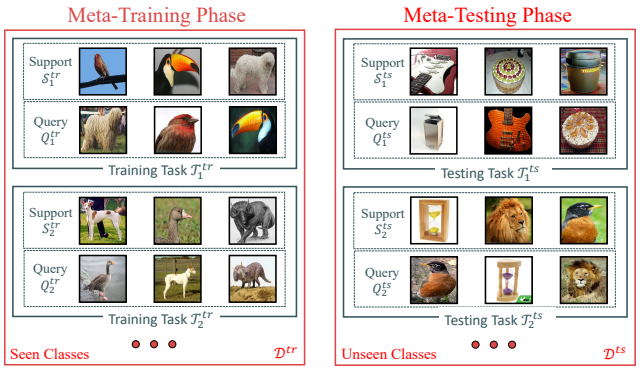


Figure 2: An illustration of the episodic training protocol.

Preliminary

FSL means learning from limited examples. In classification scenario, an N-way K-shot task is composed of N classes and K training examples per class. Another testing set sampled from the same N classes is provided to evaluate the classifier. In FSL literature, the small training set of each task is referred as *support* set $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{NK}$ and the testing set is called *query* set $\mathcal{Q} = \{(\mathbf{x}_i, y_j)\}_{j=1}^{NM}$. That is, a task $\mathcal{T}$ is defined as $\mathcal{T} = (\mathcal{S}, \mathcal{Q})$.

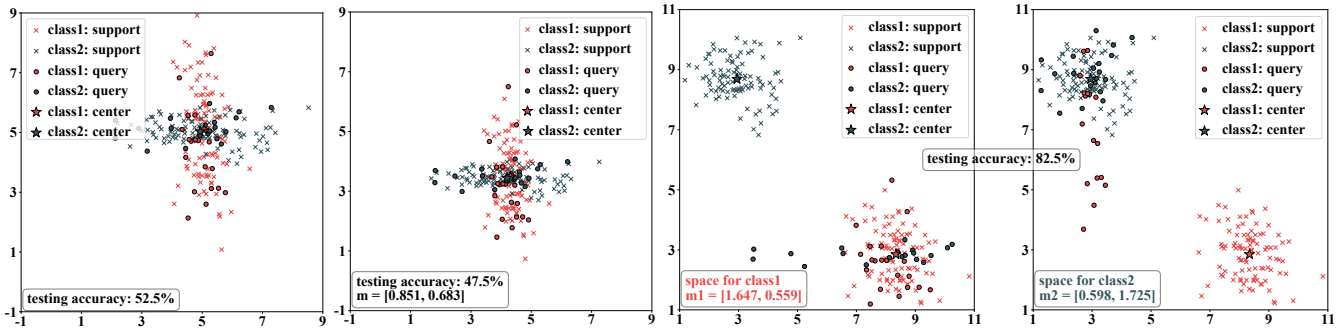
Researchers often utilize meta-learning to tackle FSL problems. A key idea in meta-learning is to mimic *meta-testing* process in *meta-training* phase. Since the learned meta-model is intended for N-way K-shot classification tasks, we sample episodic N-way K-shot tasks from *meta-training* set $\mathcal{D}^{tr}$ (composed of SEEN classes) to optimize our model. The main target is to extract knowledge from sampled tasks and reuse them when a new task comes. In *meta-testing* phase, N-way K-shot tasks are sampled from a *meta-testing* set $\mathcal{D}^{ts}$ (composed of UNSEEN classes). Figure 2 gives an illustration of this episodic training protocol.

A simple solution is to meta-learn an embedding function ϕ, which maps an input object $\mathbf{x}$ to a d-dimensional vector. In a *meta-training* task $\mathcal{T}^{tr}$ sampled from $\mathcal{D}^{tr}$, the label of a *query* instance $\mathbf{x}_j$ could be determined by its distance to each class center in the *support* set $\mathcal{S}^{tr}$ as shown in the following two equations. $[N]$ means $\{1, 2, \ldots, N\}$. $\mathrm{dis}(\cdot, \cdot)$ is some distance function like euclidean distance.

$$p(\hat{y}_j = n|\mathbf{x}_j) = \frac{\exp\{-\mathrm{dis}(\phi(\mathbf{x}_j), \mathbf{c}_n)\}}{\sum_{n'=1}^{N} \exp\{-\mathrm{dis}(\phi(\mathbf{x}_j), \mathbf{c}_{n'})\}} \quad (1)$$

$$\mathbf{c}_n = \frac{1}{K} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{S}^{tr} \wedge y_i = n} \phi(\mathbf{x}_i),\ n \in [N] \quad (2)$$

Cross-entropy loss is optimized on all sampled tasks.



(a) Raw space. (b) Global space. (c) Local spaces for two classes.

Figure 3: Necessity of local masks on a synthetic task. (a) *Support* and *query* instances sampled from two Gaussian distributions $\mathcal{P}^1$ and $\mathcal{P}^2$. Two empirical class centers are close to each other. (b) A global mask $\mathbf{m}$ is learned on *support* set. $\mathbf{m}$ fails to separate two class centers and accuracy drops to 47.5%. (c) Two local masks $\mathbf{m}^1$ and $\mathbf{m}^2$ are learned on *support* set. *Query* instances are projected into class-specific spaces to compute their distances to the corresponding class center.

$$\min_{\phi} \sum_{\mathcal{T}^{tr} \sim \mathcal{D}^{tr}} \sum_{(\mathbf{x}_j, y_j) \in \mathcal{Q}^{tr}} -\log p(\hat{y}_j = y_j | \mathbf{x}_j) \qquad (3)$$

We apply the learned embedding function ϕ to N-way K-shot tasks $\mathcal{T}^{ts}$ sampled from $\mathcal{D}^{ts}$. In this simple approach, all that we can learn from seen tasks is an embedding function. Learning such an embedding to estimate the class prototype in Equation (1) neglects the diversity of tasks. It is natural to emphasize different feature dimensions when solving different tasks, so many recent methods (Oreshkin, López, and Lacoste 2018; Li et al. 2019; Ye et al. 2020) focus on task-specific features.

Main Approach

Existing methods seek task-specific features by applying a shared transformation to all the instances in a task, ignoring the diversity of classes. We claim that each class should be treated differently to capture local characteristics. This section introduces global and local feature adaptors, and then presents a mask combiner to fuse two masks' effects. Next, we describe the implementation of these modules.

Feature Adaptor

In GLoFA, feature masks at two levels, namely task-level and class-level, are simultaneously learned. Corresponding important features are emphasized to adapt the embedding function when dealing with a specific task. Denote $\mathcal{F} = \{f^{\text{task}}(\cdot), f^{\text{cls}}(\cdot)\}$ be the set of feature adaptors.

Task-level feature adaptation. Embedding function $\phi(\cdot)$ is not ideal because the representation output by it dose not necessarily highlight the most discriminative feature dimensions. To this end, we set $\mathbf{m}^{\text{task}} = f^{\text{task}}(\{\phi(\mathbf{x}_i)\}_{i=1}^{NK})$ where $f^{\text{task}}(\cdot)$ is the task-level feature adaptor. $\mathbf{m}^{\text{task}}$ is a d-dimensional vector and encodes the excess importance of each dimension, *i.e.*, $\mathbf{1} + \mathbf{m}^{\text{task}}$ will be multiplied to $\phi(\mathbf{x})$ to highlight important dimensions and eliminate irrelevant dimensions. Based on the *support* set $\mathcal{S}$,[1] the task-level feature mask $\mathbf{m}^{\text{task}}$ is output and applied to both *support* instances and *query* instances, making our feature adaptor inductive rather than transductive.

Class-level feature adaptation. Class-specific local modeling is the main difference of GLoFA from existing methods. We set $\mathbf{m}_n^{\text{cls}} = f^{\text{cls}}(\{\phi(\mathbf{x}_i) | y_i = n\}_{i=1}^{NK}),\ n \in [N]$ where $f^{\text{cls}}(\cdot)$ is the class-level feature adaptor. Class-level masks encode excess importance of each dimensions within the scope of corresponding class. The n-th class-level mask is computed based on the *support* instances of n-th class.

How to apply class-level masks to *query* instances? For a *query* instance $\mathbf{x}_j$, although its class label is not available, we can project it into the class-specific space by n-th local mask when computing its distance to n-th class center. Let $\{\mathbf{e}_n\}_{i=n}^{N}$ be the N empirical class centers masked by corresponding feature masks, for a *query* instance $\mathbf{x}_j$, its distance to the n-th class center is computed as $d_n = \text{dis}(\phi(\mathbf{x}_j) \odot (1 + \mathbf{m}_n^{\text{cls}}), \mathbf{e}_n)$. $\odot$ means element-wise multiplication. $\mathbf{x}_j$ will be classified into the same category of its nearest center. In *meta-training* phase, these distances are normalized to a label posterior probability, which is then optimized using cross-entropy loss. The training procedure automatically adjusts the scale of each class-specific space, and makes $\{d_n\}_{n=1}^{N}$ comparable to each other.

Importance of class-level masks. To seize the heterogeneity of tasks, existing methods project instances into task-specific spaces and use some distance function to determine the label posterior probability. We can view these methods as finding a metric space shared by all the instances. However, a global metric does not necessarily fit well the distance over the data manifold. Consider a simple

[1] We omit the super-script tr and ts when the notation applies to both *meta-training* set and *meta-testing* set.

task where instances are sampled from two Gaussian distributions. Let $\mathcal{P}^1 = \mathcal{N}(\boldsymbol{\mu}^1, \boldsymbol{\Sigma}^1)$ and $\mathcal{P}^2 = \mathcal{N}(\boldsymbol{\mu}^2, \boldsymbol{\Sigma}^2)$ be the distributions of two classes where $\boldsymbol{\mu}^1, \boldsymbol{\mu}^2 \in \mathbb{R}^2$ and $\boldsymbol{\Sigma}^1, \boldsymbol{\Sigma}^2 \in \mathbb{R}^{2\times 2}$. For each class, we sample 100 instances as *support* set, $\{\mathbf{x}_s^1\}_{s=1}^{100} \overset{\text{i.i.d.}}{\sim} \mathcal{P}^1, \{\mathbf{x}_s^2\}_{s=1}^{100} \overset{\text{i.i.d.}}{\sim} \mathcal{P}^2$, and 20 instances as *query* set, $\{\mathbf{x}_u^1\}_{u=1}^{20} \overset{\text{i.i.d.}}{\sim} \mathcal{P}^1, \{\mathbf{x}_u^2\}_{u=1}^{20} \overset{\text{i.i.d.}}{\sim} \mathcal{P}^2$. Nearest Center Mean (NCM) classifier is used to predict the label of an instance, *i.e.*, $p(\hat{y} = n|\mathbf{x}) = \frac{\exp\{-\text{dis}(\mathbf{x},\mathbf{e}^n)\}}{\exp\{-\text{dis}(\mathbf{x},\mathbf{e}^1)\}+\exp\{-\text{dis}(\mathbf{x},\mathbf{e}^2)\}}$ where $\mathbf{e}^n$ is the empirical class center of n-th class. We set $\boldsymbol{\mu}^1, \boldsymbol{\mu}^2, \boldsymbol{\Sigma}^1, \boldsymbol{\Sigma}^2$ as follows:

$$\boldsymbol{\mu}^1 = \boldsymbol{\mu}^2 = \begin{bmatrix}5\\5\end{bmatrix}, \boldsymbol{\Sigma}^1 = \begin{bmatrix}0.2 & 0\\0 & 2\end{bmatrix}, \boldsymbol{\Sigma}^2 = \begin{bmatrix}2 & 0\\0 & 0.2\end{bmatrix}$$

In this task, the two class means $\boldsymbol{\mu}^1$ and $\boldsymbol{\mu}^2$ are equal, making it difficult for the NCM classifier to distinguish them in raw feature space. As shown in Figure 3a, directly using NCM to predict the labels of *query* instances achieves an accuracy of 52.5%. Next, we optimize a global mask $\mathbf{m}$ to minimize the cross-entropy loss on the *support* set. We then apply $\mathbf{m}$ to both *support* and *query* instances in the inference phase. Figure 3b is a visualization of the instances masked by $\mathbf{m}$. Two class centers are still close to each other, and accuracy drops to 47.5%. The feature mask encodes the importance of each dimension and projects instances into a new space. But in this case, whichever dimension we focus on, the two class centers cannot be separated, which is the reason that the global mask fails. As an alternative, we optimize two local masks $\mathbf{m}^1$ and $\mathbf{m}^2$ for each class and project instances into class-specific spaces. For each *query* instance $\mathbf{x}_u$, we mask it by $\mathbf{m}^n$ when computing its distance to n-th class center. In Figure 3c, we show the two class-specific spaces. Two class centers are far from each other, and accuracy rises to 72.5%. Local masks significantly improve the representation ability of feature adaptors.

Mask Combiner

As is indicated before, different masks should be emphasized for different tasks. Thus, we propose a mask combiner to balance the strengths of two feature adaptors.

Mask fusion by smoothing parameter. Since both task-level masks and class-level masks encode excess importance, we can divide them by two positive scalar parameters α^{task} and α^{cls} to adjust their strengths. For a mask $\mathbf{m}$, we have $\lim_{\alpha\to\infty} \phi(\mathbf{x}) \odot \left(\mathbf{1} + \frac{\mathbf{m}}{\alpha}\right) = \phi(\mathbf{x})$ and large α tends to eliminate the effect of $\mathbf{m}$. As indicated before, which mask should be amplified depends on the task itself. Thus, we learn from the task two smoothing parameters, *i.e.*, $[\alpha^{\text{task}}; \alpha^{\text{cls}}] = g(\{\phi(\mathbf{x}_i)\}_{i=1}^{NK})$ where $g(\cdot)$ is the task-adaptive balance module. α^{task} and α^{cls} play the role of mask balancer. When α is small, the effect of corresponding mask $\mathbf{m}$ is amplified because the differences between values in $\mathbf{m}$ are enlarged, making the distribution over excess importance sharper and more informative.

GLoFA Framework

Main objective. We compute masked class center $\mathbf{e}_n$ as Equation (4) and Equation (5).

$$\mathbf{e}_n = \frac{1}{K} \sum_{(\mathbf{x}_i, y_i)\in\mathcal{S}^{tr} \wedge y_i = n} \mathbf{z}_i,\ n \in [N] \tag{4}$$

$$\mathbf{z}_i = \phi(\mathbf{x}_i) \odot \left(\mathbf{1} + \frac{\mathbf{m}^{\text{task}}}{\alpha^{\text{task}}}\right) \odot \left(\mathbf{1} + \frac{\mathbf{m}^{\text{cls}}_{y_i}}{\alpha^{\text{cls}}}\right) \tag{5}$$

To infer the class label of *query* instance $\mathbf{x}_j$, we need to compute the posterior probability $p(\hat{y}_j = n|\mathbf{x}_j)$, as shown in Equation (6) and Equation (7).

$$p(\hat{y}_j = n|\mathbf{x}_j) = \frac{\exp\left\{-\text{dis}(\mathbf{z}_j^n, \mathbf{e}_n)\right\}}{\sum_{n'=1}^{N} \exp\left\{-\text{dis}(\mathbf{z}_j^{n'}, \mathbf{e}_{n'})\right\}} \tag{6}$$

$$\mathbf{z}_j^n = \phi(\mathbf{x}_j) \odot \left(\mathbf{1} + \frac{\mathbf{m}^{\text{task}}}{\alpha^{\text{task}}}\right) \odot \left(\mathbf{1} + \frac{\mathbf{m}^{\text{cls}}_n}{\alpha^{\text{cls}}}\right) \tag{7}$$

$\mathbf{z}_j^n$ is the masked representation of *query* instance $\mathbf{x}_j$ for computing its distance to the n-th class center. This class-specific operation makes it possible to use class-level masks for *query* instances without knowing their class labels. α^{task} and α^{cls} are two balance parameters conditioned on the task context, as shown in Equation (8).

$$[\alpha^{\text{task}}, \alpha^{\text{cls}}] = g(\{\phi(\mathbf{x}_i)\}_{i=1}^{NK}) \tag{8}$$

The main objective of our GLoFA framework is:

$$\min_{\phi,\mathcal{F},g} \sum_{\mathcal{T}^{tr}\sim\mathcal{D}^{tr}} \sum_{(\mathbf{x}_j, y_j)\in\mathcal{Q}^{tr}} -\log p(\hat{y}_j = y_j|\mathbf{x}_j) \tag{9}$$

Figure 4 shows the whole framework of GLoFA. There remain two details in our framework, *i.e.*, how to generate masks with $\mathcal{F}$ and how to implement $g(\cdot)$. We instantiate the feature adaptors and the balance module as set functions.

Implementation. In this part we specify the concrete implementation of $\mathcal{F}$ and $g(\cdot)$. In GLoFA, we generate masks as excess importance to highlight relevant features. Whether a particular dimension is important is jointly related to the task or class context. Hence we use a set function to implement $\mathcal{F}$, where the outputs are permutation invariant w.r.t the context elements. According to (Zaheer et al. 2017), we denote the set to determine the feature masks as $\mathcal{A}$, then implement $f \in \mathcal{F}$ as a deep-set function:

$$f(\mathcal{A}) = h_\delta\left(\mathbf{MLP}\left(\sum_{\mathbf{x}\in\mathcal{A}} \left[\mathbf{MLP}\left(\phi(\mathbf{x})\right); \phi(\mathbf{x})\right]\right)\right) \tag{10}$$

$\mathbf{MLP}(\cdot)$ is a multi-layer linear network with $\tanh(\cdot)$ activation. $h_\delta(\cdot) = \min(\delta, \max(0, \cdot))$ is a function that projects its input to $[0, \delta]$, ensuring that the excess importance is positive but not too large. After transforming the embedding $\phi(\mathbf{x})$ by first $\mathbf{MLP}(\cdot)$, we concatenate it with $\phi(\mathbf{x})$, and then input

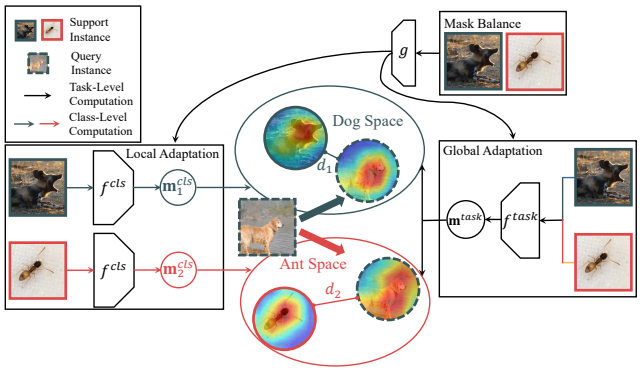


Figure 4: An illustration of GLoFA framework. We take a 2-way 1-shot classification task as an example. In Global Adaptation, GLoFA generates 1 task-level mask $\mathbf{m}^{\text{task}}$. $\mathbf{m}^{\text{task}}$ is shared by two classes. In Local Adaptation, GLoFA generates 2 class-level masks $\mathbf{m}_1^{\text{cls}}$ and $\mathbf{m}_2^{\text{cls}}$ for each class. Based on the task context, GLoFA outputs two smoothing parameters α^{task} and α^{cls}, which adjust the strengths of global and local masks. For each class, $\mathbf{m}^{\text{task}}$ and the corresponding $\mathbf{m}^{\text{cls}}$ induce a class-specific space. The *support* instances of each class are projected into the corresponding space. For a *query* instance, it is projected into the n-th class-specific space when computing its distance to n-th class center.

the joint representation to the second transformation function. This can be seen as a kind of residual operation which is beneficial to training process. At task-level, $\mathcal{A}$ consists of all the *support* instances of a task. At class-level, $\mathcal{A}$ is composed of the *support* instances from a specific class. We implement the mask combiner as a set function too. As indicated before, whether global features or local features should be emphasized depends on the task. Equation (11) defines the mask combiner. $\mathcal{S}$ is the *support* set of a task. ϵ is a small positive value that prevents α^{task} and α^{cls} from being 0.

$$\left[\alpha^{\text{task}}; \alpha^{\text{cls}}\right] = \text{ReLU}\left(\mathbf{MLP}\left(\sum_{\mathbf{x}\in\mathcal{S}}\left[\mathbf{MLP}\left(\phi(\mathbf{x})\right); \phi(\mathbf{x})\right]\right)\right) + \epsilon \tag{11}$$

Experiments

There are three parts of experiments in this section. In the first part, we construct a mixed dataset and sample heterogeneous tasks from it. We show that global and local masks can capture the heterogeneity better than existing methods. In the second part, we test our method on two widely used benchmark datasets *mini*ImageNet and *tiered*ImageNet. GLoFA achieves competitive performances with recent state-of-the-art methods. The third part presents further analyses of GLoFA.

Heterogeneous Tasks

Datasets. We construct a dataset mixed by 5 fine-grained classification sub-datasets, namely AirCraft (Maji et al. 2013), Car-196 (Krause et al. 2013), CUB-200-2011 (Wah et al. 2011), Stanford Dog (Khosla et al. 2011), and Indoor Scenes (Quattoni and Torralba 2009). For each sub-dataset, we randomly extract 20 classes from it, and then split the 20 classes into 3 parts: 10 classes for *meta-training*, 5 classes for *meta-validating*, and 5 classes for *meta-testing*.

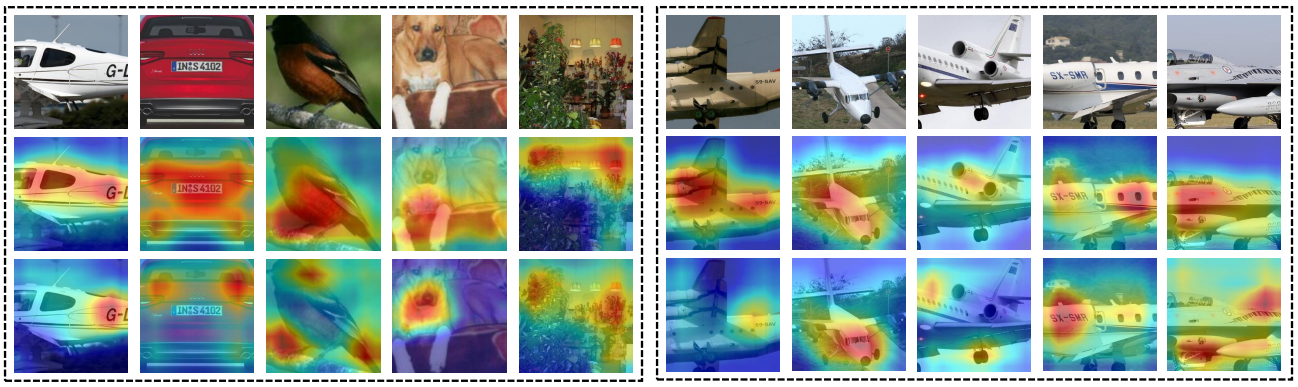
G	ProtoNet	TADAM	CTM	GLoFA
1	49.92±0.29	50.04±0.29	50.45±0.38	**53.14±0.46**
2	57.83±0.40	58.17±0.40	58.34±0.33	**61.07±0.34**
3	59.55±0.38	60.46±0.39	60.28±0.42	**62.49±0.40**
4	75.43±0.34	75.92±0.41	76.47±0.36	**76.56±0.32**
5	77.82±0.29	78.37±0.33	79.03±0.34	**79.92±0.33**
#	67.46±0.32	67.99±0.37	68.42±0.22	**70.60±0.29**

Table 1: Average test accuracies (%) with 95% confidence intervals on tasks sampled from the constructed heterogeneous dataset. G is the granularity factor defined in experiment settings. # means randomly sampling classes from the whole *meta-testing* set.

Settings. Since these sub-datasets have different semantics, it is easy to distinguish classes from different sub-datasets. If classes in a task are all from a common sub-dataset, the task will be fine-grained and extremely hard. In this experiment, we sample 5-way 1-shot tasks from the whole heterogeneous dataset with different granularity. Here we define granularity G as the number of sub-datasets involved in a task. The smaller G is, the more fine-grained the task is. In *meta-training* phase, we sample tasks from the whole *meta-training* set randomly, which means training the model on heterogeneous tasks with different G values. In *meta-testing* phase, we sample tasks with specific G values to check whether our methods can maintain good performance on tasks with different granularity. Some other metric-based few-shot learning methods are compared, *e.g.*, ProtoNet (Snell, Swersky, and Zemel 2017), TADAM (Oreshkin, López, and Lacoste 2018) and CTM (Li et al. 2019).

Implementation details. We take the commonly used ResNet-12 as the embedding network. After training the model, we sample 600 episodes for each G in $\{1, 2, 3, 4, 5\}$ to evaluate it. We also randomly sample 600 episodes from the whole *meta-testing* set to evaluate our model on heterogeneous tasks. We reimplement ProtoNet, TADAM, and CTM with ResNet-12 embedding network for a fair comparison. More details can be found in the supplement.

Results. Table 1 shows experiment results on the heterogeneous dataset. Different G values are used to sample *meta-testing* tasks. As expected, all methods achieve better performance when G is larger since coarse-grained tasks are easier to solve. We can see that GLoFA outperforms other methods on small G values. Unlike these compared methods that only perform global transformations to the task, GLoFA incorporates local feature masks, which tend to capture detailed characteristics and are most helpful in fine-grained tasks.

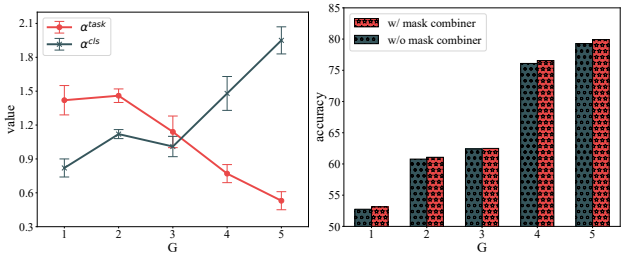


(a) Coarse-grained task. (b) Fine-grained task.

Figure 5: Visualization of *support* instances' feature masks. Raw images, images with global masks, and images with local masks are arranged in the first, the second, and the third row respectively. Task-level masks focus on sketchy patterns while class-level masks pick out detailed characteristics. (a) Visualization of a 5-way 1-shot coarse-grained task. (b) Visualization of a 5-way 1-shot fine-grained task.

Visualization of feature masks. For an instance $\mathbf{x}$ and its d-dimensional feature mask $\mathbf{m}$, we keep 10 largest values in $\mathbf{m}$ and set all other values to zero. After that, $\mathbf{m}$ is multiplied to the penultimate layer's outputs, which contains d semantic feature maps. The weighted sum of feature maps is then applied to the raw image to show what part of an image is highlighted by $\mathbf{m}$. In Figure 5, we can see that global feature masks focus on sketchy patterns while local feature masks catch detailed characteristics.

Effect of α. In this part, we investigate the behaviour of $g(\cdot)$. For G in $\{1, 2, 3, 4, 5\}$, we sample 600 tasks from *meta-testing* set and check the mean and standard deviation of α^{task} and α^{cls}. In Figure 6a, we can see that α^{task} tends to be large on fine-grained tasks. The trend of α^{cls} is opposite to α^{task}. This means the mask combiner trusts local masks more on fine-grained tasks because detailed characteristics are more discriminative. In Figure 6b, we check whether the mask combiner can improve the accuracy. It is shown that the mask combiner improves the model accuracy.



(a) The change of α over G. (b) Improvement in accuracy.

Figure 6: (a) Mean and standard deviation of α^{task} and α^{cls} for different granularity of tasks. α^{cls} tends to be small for fine-grained tasks because detailed characteristics are more important. The trend of α^{task} is opposite to α^{cls}. (b) Average testing accuracies on different granularity of tasks without mask combiner. Global masks and local masks are directly applied to the instance embedding. The model suffers a loss in accuracy on fine-grained tasks and coarse-grained tasks.

Benchmark Evaluations

Datasets. In this part, we test our method on two benchmark dataset, *i.e.*, *mini*ImageNet (Vinyals et al. 2016) and *tiered*ImageNet (Ren et al. 2018). We follow (Ravi and Larochelle 2017) and (Ren et al. 2018) to split *mini*ImageNet and *tiered*ImageNet respectively. More details about these two dataset can be found in the supplement.

Implementation details. We use ResNet-12 as embedding network. We pre-train the embedding network on the *meta-training* set of *mini*ImageNet with cross-entropy loss function. Refer to the supplement for more details.

Results. We compare GLoFA to some classic few-shot learning methods and recent state-of-the-art methods. We summarize test accuracies in Table 2. GLoFA achieves competitive performance to state-of-the-art methods.

Evaluation of embedding quality. In this part, we take a closer look at GLoFA to investigate why GLoFA can achieve promising performance on *mini*ImageNet. Since we use NCM classifier, an embedding-based method, to predict the label of each *query* instance, embedding quality may be a key factor to the model accuracy. We perform K-means clustering in the embedding space and use Normalized Mutual Information (NMI) as the criterion to measure the embedding quality. We randomly sample 600 5-way 20-shot tasks from the *meta-testing* set of *mini*ImageNet and perform clustering on their *support* sets. Results are shown in Table 3. GLoFA significantly improves the embedding quality, which results in an increase in model accuracy. The tSNE

method	*mini*ImageNet		*tiered*ImageNet	
	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
MatchNet (Vinyals et al. 2016)	63.08 ± 0.80	75.99 ± 0.60	68.50 ± 0.92	80.60 ± 0.71
ProtoNet (Snell, Swersky, and Zemel 2017)	60.37 ± 0.83	78.02 ± 0.57	65.65 ± 0.92	83.40 ± 0.65
TADAM (Oreshkin, López, and Lacoste 2018)	58.50 ± 0.30	76.70 ± 0.30	63.74 ± 0.45	80.35 ± 0.40
MetaOptNet (Lee et al. 2019)	62.64 ± 0.61	78.63 ± 0.46	65.99 ± 0.72	81.56 ± 0.53
ClassModel (Ravichandran, Bhotika, and Soatto 2019)	60.71	77.26	-	-
CTM* (Li et al. 2019)	62.05 ± 0.55	78.63 ± 0.06	64.78 ± 0.11	81.05 ± 0.52
AFHN (Li et al. 2020)	62.38 ± 0.72	78.16 ± 0.56	-	-
DSN (Simon et al. 2020)	64.60 ± 0.72	79.51 ± 0.50	67.39 ± 0.82	82.85 ± 0.56
MetaVRF** (Zhen et al. 2020)	63.80 ± 0.05	77.97 ± 0.28	-	-
GLoFA	$\mathbf{66.12 \pm 0.42}$	$\mathbf{81.37 \pm 0.33}$	$\mathbf{69.75 \pm 0.33}$	$\mathbf{83.58 \pm 0.42}$

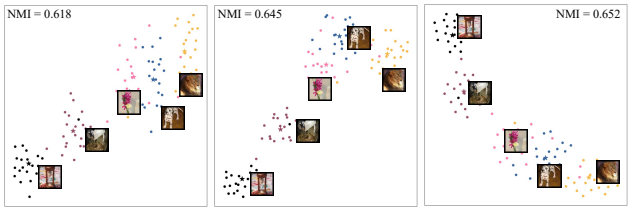
Table 2: Average test accuracies (%) with 95% confidence intervals on tasks sampled from *meta-testing* set of *mini*ImageNet and *tiered*ImageNet. All these methods use ResNet-12 as embedding network except CTM and MetaVRF. (*) CTM uses ResNet-18 as backbone. (**) MetaVRF uses WRN-28-10 as backbone. These two backbones are deeper than ResNet-12.

embedding	no mask	global	local	GLoFA
NMI	0.61±0.08	0.64±0.09	0.65±0.09	**0.66±0.09**

Table 3: Average NMI with 95% confidence intervals on tasks sampled from the *meta-testing* set of *mini*ImageNet. Global masks and local masks both improve the embedding quality. Combing the two masks in GLoFA will further improve the embedding quality.

Model	f^{task}	f^{cls}	g	1-shot	5-shot
0	×	×	×	60.42±0.38	77.29±0.54
1	✓	×	×	65.88±0.29	80.30±0.32
2	×	✓	×	65.72±0.48	80.94±0.35
GLoFA	✓	✓	✓	**66.12±0.42**	**81.37±0.33**

Table 4: Average test accuracies (%) with 95% confidence intervals of several variants on *mini*ImageNet.



(a) w/o masks. (b) Global mask. (c) Local mask.

Figure 7: Visualization of a randomly sampled task. Each color represents a class. ⋆ indicates the class center. (a) tSNE result without any feature masks. (b) tSNE result with global feature mask. (c) tSNE result with local feature mask.

results of a randomly chosen task is shown in Figure 7. We can see that feature masks improve embedding quality.

Further Analyses

Ablation study. In this part, we evaluate the effectiveness of each module in GLoFA. We sample tasks from *mini*ImageNet to train and test several variants of GLoFA. We summarize experiment results in Table 4. By removing global and local feature adaptors in GLoFA, our model degenerates to ProtoNet (Snell, Swersky, and Zemel 2017) and we achieve similar accuracy to that in Table 2. Equipped with feature adaptors and mask combiner, our model outperforms the baseline models by a noticeable margin.

Conclusion

In this paper, we propose GLoFA, a new framework that tailors embedding function to heterogeneous few-shot tasks by global and local feature adaptors. Unlike existing methods that apply a global transformation to all the instances in a task, GLoFA treats each class differently and generates local feature masks. We verify that global masks capture general patterns while local masks focus on detailed characteristics. An adaptive combination of two masks makes GLoFA succeed in learning tasks with mixed granularity. GLoFA also achieves competitive performance on benchmark datasets.

Broader Impact

In this work, we study the problem of few-shot learning, which means extracting concepts from limited labeled examples. The investigation of few-shot learning may ease the model's requirement for large labeled dataset, which expands the application field of deep learning systems. We have not found any negative influences of this technology on human society yet. We believe that demonstrating and developing few-shot learning techniques is vital for robust and universal intelligence. Moreover, advanced few-shot learning algorithms may optimize industrial chain by encouraging practitioners to apply for technology-intensive positions rather than labor-intensive ones like data collectors.

Acknowledgements

This work is supported by NSFC (6177319, 616300004, 61921006, 62006112).

References

Dai, W.-Z.; Muggleton, S.; Wen, J.; Tamaddoni-Nezhad, A.; and Zhou, Z.-H. 2017. Logical vision: One-shot meta-interpretive learning from real images. In *Proceedings of International Conference on Inductive Logic Programming*, 46–62.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, 1126–1135.

Graves, A.; Wayne, G.; and Danihelka, I. 2014. Neural Turing Machines. *CoRR* abs/1410.5401.

Khosla, A.; Jayadevaprakash, N.; Yao, B.; and Li, F.-F. 2011. Novel Dataset for Fine-Grained Image Categorization: Stanford Dogs. In *24th IEEE Conference on Computer Vision and Pattern Recognition Workshop*, volume 2.

Koch, G.; Zemel, R.; and Salakhutdinov, R. 2015. Siamese Neural Networks for One-Shot Image Recognition. In *32nd International Conference on Machine Learning Workshop*, volume 2.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d Bbject Representations for Fine-Grained Categorization. In *14th International Conference on Computer Vision Workshop*, 554–561.

kyun Noh, Y.; tak Zhang, B.; and Lee, D. D. 2018. Generative Local Metric Learning for Nearest Neighbor Classification. *IEEE Transcations on Pattern Analysis and Machine Intelligence* 40: 106–118.

Lee, K.; Maji, S.; Ravichandran, A.; and Soatto, S. 2019. Meta-learning with Differentiable Convex Optimization. In *Proceedings of the 32nd IEEE Conference on Computer Vision and Pattern Recognition*, 10657–10665.

Li, H.; Eigen, D.; Dodge, S.; Zeiler, M.; and Wang, X. 2019. Finding Task-Relevant Features for Few-Shot Learning by Category Traversal. In *Proceedings of the 32nd IEEE Conference on Computer Vision and Pattern Recognition*, 1–10.

Li, K.; Zhang, Y.; Li, K.; and Fu, Y. 2020. Adversarial Feature Hallucination Networks for Few-Shot Learning. In *Proceedings of the 33rd IEEE Conference on Computer Vision and Pattern Recognition*, 13470–13479.

Liu, X.-Y.; Wang, S.-T.; and Zhang, M.-L. 2019. Transfer synthetic over-sampling for class-imbalance learning with limited minority class data. *Frontiers of Computer Science* 13(5): 996–1009.

Maji, S.; Rahtu, E.; Kannala, J.; Blaschko, M.; and Vedaldi, A. 2013. Fine-Grained Visual Classification of Aircraft. *CoRR* abs/1306.5151.

Munkhdalai, T.; Sordoni, A.; Wang, T.; and Trischler, A. 2019. Metalearned Neural Memory. In *Advances in Neural Information Processing Systems 32*, 13310–13321.

Oreshkin, B.; López, P. R.; and Lacoste, A. 2018. Tadam: Task Dependent Adaptive Metric for Improved Few-Shot Learning. In *Advances in Neural Information Processing Systems 31*, 721–731.

Quattoni, A.; and Torralba, A. 2009. Recognizing Indoor Scenes. In *Proceedings of the 22nd IEEE Conference on Computer Vision and Pattern Recognition*, 413–420.

Ravi, S.; and Larochelle, H. 2017. Optimization as a Model for Few-Shot Learning. In *Proceedings of the 5th International Conference on Learning Representations*.

Ravichandran, A.; Bhotika, R.; and Soatto, S. 2019. Few-Shot Learning With Embedded Class Models and Shot-Free Meta Training. In *Proceedings of the 17th International Conference on Computer Vision*, 331–339.

Ren, M.; Triantafillou, E.; Ravi, S.; Snell, J.; Swersky, K.; Tenenbaum, J. B.; Larochelle, H.; and Zemel, W. S. 2018. Meta-Learning for Semi-Supervised Few-Shot Classification. In *Proceedings of the 6th International Conference on Learning Representations*.

Santoro, A.; Bartunov, S.; Botvinick, M.; Wierstra, D.; and Lillicrap, T. 2016. Meta-Learning with Memory-Augmented Neural Networks. In *Proceedings of the 33rd International Conference on Machine Learning*, 1842–1850.

Simon, C.; Koniusz, P.; Nock, R.; and Harandi, M. 2020. Adaptive Subspaces for Few-Shot Learning. In *Proceedings of the 33rd IEEE Conference on Computer Vision and Pattern Recognition*, 4136–4145.

Snell, J.; Swersky, K.; and Zemel, R. S. 2017. Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems 30*, 4077–4087.

Thrun, S.; and Pratt, L. 2012. *Learning to Learn*.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; and Wierstra, D. 2016. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems 29*, 3630–3638.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The Caltech-UCSD Birds-200-2011 Dataset .

Wang, J.; Kalousis, A.; and Woznica, A. 2012. Parametric Local Metric Learning for Nearest Neighbor Classification. In *Advances in Neural Information Processing Systems 25*, 1601–1609.

Ye, H.-J.; Hu, H.; Zhan, D.-C.; and Sha, F. 2020. Few-Shot Learning via Embedding Adaptation With Set-to-Set Functions. In *Proceedings of the 33rd IEEE Conference on Computer Vision and Pattern Recognition*, 8808–8817.

Ye, H.-J.; Lu, S.; and Zhan, D.-C. 2020. Distilling Cross-Task Knowledge via Relationship Matching. In *Proceedings of the 33rd IEEE Conference on Computer Vision and Pattern Recognition*, 12396–12405.

Zaheer, M.; Kottur, S.; Ravanbakhsh, S.; Poczos, B.; Salakhutdinov, R. R.; and Smola, A. J. 2017. Deep Sets. In *Advances in Neural Information Processing Systems 30*, 3391–3401.

Zhen, X.; Sun, H.; Du, Y.; Xu, J.; Yin, Y.; Shao, L.; and Snoek, C. 2020. Learning to Learn Kernels with Variational Random Features. In *Proceedings of the 37th International Conference on Machine Learning*, 11626–11636.