Adaptive FSS: A Novel Few-Shot Segmentation Framework via Prototype Enhancement

Jing Wang[1,2], **Jinagyun Li**[1,2], **Chen Chen**[3], **Yisi Zhang**[1,2], **Haoran Shen**[1,2], **Tianxiang Zhang**[1,2*]

[1]School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, China
[2] Key Laboratory of Knowledge Automation for Industrial Processes, Ministry of Education, Beijing, China
[3] Center for Research in Computer Vision, University of Central Florida, Orlando, USA
m202120718@xs.ustb.edu.cn, leejy@ustb.edu.cn, chen.chen@crcv.ucf.edu,
{m202210579, m202220738}@xs.ustb.edu.cn, txzhang@ustb.edu.cn.

Abstract

The Few-Shot Segmentation (FSS) aims to accomplish the novel class segmentation task with a few annotated images. Current FSS research based on meta-learning focuses on designing a complex interaction mechanism between the query and support feature. However, unlike humans who can rapidly learn new things from limited samples, the existing approach relies solely on fixed feature matching to tackle new tasks, lacking adaptability. In this paper, we propose a novel framework based on the adapter mechanism, namely Adaptive FSS, which can efficiently adapt the existing FSS model to the novel classes. In detail, we design the Prototype Adaptive Module (PAM), which utilizes accurate category information provided by the support set to derive class prototypes, enhancing class-specific information in the multi-stage representation. In addition, our approach is compatible with diverse FSS methods with different backbones by simply inserting PAM between the layers of the encoder. Experiments demonstrate that our method effectively improves the performance of the FSS models (e.g., MSANet, HDMNet, FPTrans, and DCAMA) and achieves new state-of-the-art (SOTA) results (i.e., 72.4% and 79.1% mIoU on PASCAL-5^i 1-shot and 5-shot settings, 52.7% and 60.0% mIoU on COCO-20^i 1-shot and 5-shot settings). Our code is available at https://github.com/jingw193/AdaptiveFSS.

Introduction

As one of the fundamental tasks in the field of computer vision, semantic segmentation has achieved significant improvement driven by the rapid increase of data scale. However, acquiring detailed, pixel-level annotations for images is a well-known challenge, both time-consuming and costly (Everingham et al. 2010; Lin et al. 2014). This complexity compounds the challenge for models when acquiring knowledge about novel categories. To address this, few-shot segmentation (FSS) is proposed to learn new concepts on a few labeled samples (i.e. support images), realizing new class segmentation on unlabeled images (i.e. query images).

Recently, most FSS researches (Kang and Cho 2022; Yang et al. 2020a; Lang et al. 2022) employ a meta-learning episodic training strategy and focus on elaborately designing a base segmentation model, which includes sophisticated feature interaction mechanisms (Zhang et al. 2021; Iqbal, Safarov, and Bang 2022) between query images and support images. Specifically, in each training episode, a query image and a support set are sampled to imitate the situation of segmenting a novel category. During meta-testing, the base segmentation model predicts the query mask without changing any parameters as illustrated in the upper portion of Fig. 1. A fundamental challenge of this pipeline is how to rapidly learn task-specific information corresponding to new concepts and adapt the model to novel classes using limited labeled samples. Simultaneously, in the few-shot classification task (Dhillon et al. 2019; Kang et al. 2021), a common approach to achieve the above objectives and significant performance improvements is to pre-train the entire model and then fine-tune the specific head for new tasks.

However, designing an effective fine-tuning framework suitable for existing FSS methods presents the following challenges. Firstly, determining the appropriate number and

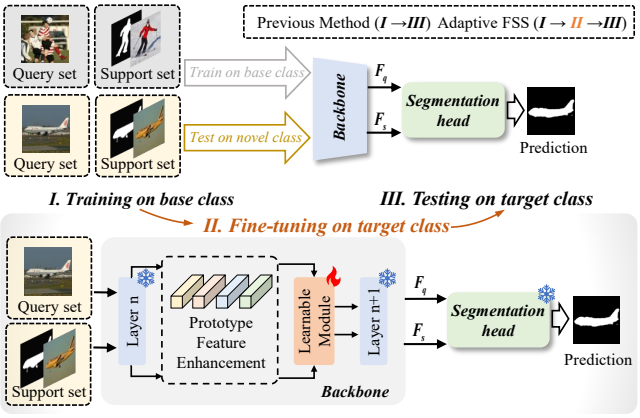

Figure 1: The overview of our Adaptive FSS. Previous works generally train the FSS model on the base classes and directly evaluate it on novel classes. In our framework, we insert the Prototype Adaptive Module (PAM), conducting a fine-tuning step before testing to effectively adapt the model to novel classes through prototype enhancement.

*Corresponding Author.

optimal placement of updated parameters in the FSS model is difficult for ensuring the effectiveness of fine-tuning. For example, in the meta-learning pipeline of FSS, the primary objective of the segmentation head is mainly to distinguish the foreground and the background without a strong relation to the specificity of the category. For dense prediction FSS tasks, the adaptability of fine-tuning a task-specific head like in classification based on image-level semantic matching may be insufficient (related experimental results can be seen in Table 1). On the other hand, fine-tuning the entire model (i.e., encoder or decoder) is prone to over-fitting in the case of limited data.

Secondly, diverse backbones (i.e., ResNet (He et al. 2016), ViT (Dosovitskiy et al. 2020), Swin Transformer (Liu et al. 2021b)) and various types of decoders are employed in existing FSS methods (Peng et al. 2023; Zhang et al. 2022a; Shi et al. 2022), which is an obstacle to design a general fine-tuning architecture. Besides, fine-tuning parameters in the original model will lead to catastrophic forgetting and compression of knowledge about base classes.

Currently, adapter (Houlsby et al. 2019; Chen et al. 2022b) is a commonly used parameter-efficient transfer learning method, and is utilized to accommodate few-shot classifiers (Zhang et al. 2022c; Li, Liu, and Bilen 2022) to new tasks and domains. During training, it only updates the parameters in the additionally inserted adapter module and freezes the rest of the model. By flexibly adjusting the parameter scale, insertion position, and design ideas of the adapter, the model can effectively adapt to new tasks while mitigating over-fitting and preventing catastrophic forgetting. Therefore, adapter tuning may be an appropriate way to solve the aforementioned problems. However, for the FSS task, the existing adapter cannot effectively exploit the provided support set to extract class-specific features. Besides, it is difficult to model general representations for different classes with limited samples.

In this paper, we propose the Adaptive FSS framework, exploring the potential of the fine-tuning process based on the adapter mechanism in the FSS task. *To address the challenge of applying the existing adapter module to the FSS task, we meticulously design a powerful module named the Prototype Adaptive Module (PAM), as shown in Fig. 1.* Specifically, it consists of a prototype enhancement module (PEM) and a learnable adaptive module (LAM). In the PEM, the support set is utilized to encode a class prototype, enhancing the information associated with the new class in the features. Besides, the class prototype bank updated by momentum is defined to improve the quality of prototypes and acquire the general class representation. For LAM design, we employ a simple set of projection layers to further model the task-specific information. From the perspective of feature adaptation, our approach only inserts the PAM into the backbone to obtain multi-stage category-specific features for improving model adaptability. Importantly, this method does not necessitate specialized decoder designs, making it universally applicable to any existing FSS model with various backbones. By fine-tuning PAM accounting for a small proportion (0.5% on average) of the whole network, the base segmentation model can rapidly adapt to the new category.

We conduct extensive experiments on two benchmarks (i.e. PASCAL-5^i (Everingham et al. 2010) and COCO-20^i (Lin et al. 2014)) based on four well-performed FSS models (i.e. MSANet (Iqbal, Safarov, and Bang 2022), HDMNet (Peng et al. 2023), FPTrans (Zhang et al. 2022a), and DCAMA (Shi et al. 2022)) to prove the effectiveness of the proposed Adaptive FSS. The results demonstrate that our approach is a feasible and effective solution for enhancing new class adaption to boost the SOTA performance in FSS tasks. Our main contributions could be summarized as follows:

- We propose a novel framework Adaptive FSS on applying the adapter mechanism to the few-shot segmentation task. This general architecture can be integrated into various FSS methods, facilitating effective adaptation of the base segmentation model to novel classes.
- We design a novel Prototype Adaptive Module (PAM) to realize the enhancement of refined features for specific categories during fine-tuning. Moreover, it is plug-and-play and well-suitable for the FSS task.
- The experimental results on two benchmark datasets demonstrate that our proposed method achieves superior performance over SOTA approaches (with an average $\uparrow$ 2.8% mIoU on PASCAL-5^i in the 1-shot setting, requiring only a 0.5% parameters increase).

Related Work

Few-Shot Learning

Few-shot learning (FSL) aims to enable models to learn and adapt, generalizing to novel domains based on the hints of a few labeled samples. Existing FSL research mainly focuses on image classification (Kang et al. 2021; Zheng et al. 2022) and other visual tasks (Kang et al. 2023), and have led to several primary categories of solutions in this field, including fine-tuning (Ravi and Larochelle 2016; Finn, Abbeel, and Levine 2017), metric learning (Yoon, Seo, and Moon 2019; Schwartz et al. 2018) and meta-learning (Zhang et al. 2022b). Among them, fine-tuning-based methods involve training models on a substantial amount of source samples and fine-tuning them on a small set of task-specific samples.

Few-Shot Segmentation

The mainstream training pipeline for few-shot segmentation(FSS) is based on episodic training, which is a typical form of meta-learning. Early methods (Tian et al. 2020; Rakelly et al. 2018; Zhang et al. 2020; Zhang, Xiao, and Qin 2021) for FSS generally follow a classic dual-branch structure proposed by (Shaban et al. 2017), the pioneering work of FSS. The trend of recent methods is the single-branch structure based on prototypical network (Yang et al. 2020a; Kang and Cho 2022).

Rather than constructing a prototype feature extractor, some works (Vinyals et al. 2016; Zhang et al. 2019; Yang et al. 2020b; Lu et al. 2021; Liu et al. 2021a) are devoted to capturing correspondence between the query image and support set. HSNet (Min, Kang, and Cho 2021) exploits multi-level feature relevance and efficient 4D convolution to filter the query-support correlation map. APANet (Chen

et al. 2021) introduces an adaptive prototype representation to regulate incomplete feature interaction. Yet, designs based on meta-learning suffer from severe loss of spatial structure and limitations of the inherent priors. CRNet (Liu et al. 2020) opens up another way, proposing a combination of conditional network, Siamese network encoder, cross-inference module, and mask refinement to achieve FSS. RePRI (Boudiaf et al. 2021) abandons the strong assumption in meta-learning that base set and novel set have similar distributions and resorts to simply supervised base training by cross-entropy loss.

However, current few-shot segmentation approaches are limited by trading model simplicity for generalization to unseen classes. To this end, we propose a framework to implement FSS through Parameter-Efficient-Tuning, empowering models to learn new classes in a stable and parameter-efficient way.

Parameter-Efficient Transfer Learning

In the FSS task, learning new classes using standard fine-tuning methods leads to over-fitting or catastrophic forgetting. To address this, Parameter-Efficient Tuning (PET) technique emerges as an effective solution. There has been a growing demand for PET and several alternative directions have emerged, including Adapter (Houlsby et al. 2019; Hu et al. 2023), Prompt-tuning (Zhu et al. 2022; Liu et al. 2022), Prefix-tuning (Li and Liang 2021), and LoRA (Hu et al. 2021). In computer vision, Adapter (Chen et al. 2022b,a; Pan et al. 2022) and Prompt-tuning (Wang et al. 2021; Jia et al. 2022) exhibit great performance in transfer learning.

Recently, adapter-based methods have also been introduced into the FSL task. Researchers have explored adapting models to unseen classes by attaching sets of additive parameters called adapters into original models and fine-tuning these sets. FSL algorithms based on adapter-tuning (Li, Liu, and Bilen 2022; Zhang et al. 2022c) represent striking classification accuracy and flexibility.

Methodology

This section first introduces the primary setup of regular FSS (Few-shot segmentation). Then, the details of the proposed Adaptive FSS architecture are presented. After that, the Prototype Adaptive Module (PAM) structure is elaborated from the two components of the Prototype Enhancement Module (PEM) and the Learnable Adaptive Module (LAM).

Problem Setup

Few-Shot Segmentation Setup. Few-shot segmentation (FSS) focuses on tackling the semantic segmentation problem on novel classes with only the corresponding few samples. For an n-way k-shot FSS task, current research (Zhang et al. 2021) widely adopts the meta-learning paradigm called episodic training (Vinyals et al. 2016), where each episode is associated with a single category becoming a 1-way k-shot task. In general, the dataset is first split into D_{tr} and D_{ts} for training and testing. Further, D_{tr} and D_{ts} are divided into $\{D_{tr}^0, D_{tr}^1, ..., D_{tr}^{c-1}\}$ and $\{D_{ts}^0, D_{ts}^1, ..., D_{ts}^{c-1}\}$ (c denotes the number of classes) by classes respectively. For training each episode, a query sample $\{I_q, M_q\}$ and K support samples $\{I_s^k, M_s^k\}_{k=1}^K$ are selected from $\{D_{tr}^i\}$ and the model is expected to predict query mask M_q according to $\{I_s^k, M_s^k\}_{k=1}^K$ and I_q. For testing, the query-support images are sampled from D_{ts}^j ($j \neq i$) to achieve inference of novel classes. In this paper, we follow this popular episodic training and testing scheme.

Adaptive FSS Framework

Overview. An overview of our proposed Adaptive FSS framework is presented in Fig. 2. Given a query image I_q and a support set $\{I_s^k, M_s^k\}_{k=1}^K$, the encoder first extracts query feature F_q and support feature F_s as the previous methods. Then, F_s, M_s, and F_q are input to our proposed PAM to obtain the class-specific features F_s^* and F_q^* by PEM. Further, we feed the F_s^* and F_q^* into LAM to learn special information for novel tasks, generating $\hat{F}_s$ and $\hat{F}_q$. After that, $\hat{F}_s$ and $\hat{F}_q$ are injected into the original feature F_s and F_q, which are employed in the downstream decoder to achieve more precise segmentation. When fine-tuning, we only update the parameters of the PAM and freeze the rest of the network, making the base segmentation model adapt to the new category efficiently.

Prototype Enhancement Module. It consists of prototype generation to get high-quality class prototypes and feature enhancement to reinforce class-specific information according to prototypes, which are elaborated as follows.

Prototype Generation. As shown in Fig. 2, for an n-way k-shot task, the support feature $F_s \in \mathbb{R}^{k\times d\times h\times w}$ and mask $M_s \in \mathbb{R}^{k\times h\times w}$ (k denotes the number of support samples, d presents the corresponding feature dimension, h and w denote feature height and width. M_s is down-sampled to the same resolution as the features.) are utilized to obtain higher-quality category prototypes. Specifically, we first define a class prototype bank $P = \{P_1, P_2, ...P_n\} \in \mathbb{R}^{n\times d}$ in each PAM, where n denotes the number of novel classes and $P_{i\in[1:n]} \in \mathbb{R}^d$ presents the prototype of target class. Then, a temporary prototype $P_t \in \mathbb{R}^d$ of the target class is obtained by calculating the spatial-wise multiplication between the support feature F_s and masks M_s and averaging nonzero feature embedding. The above mathematical expression is defined as follows:

$$P_t = \mathrm{Mean}(F_s \circ M_s), \tag{1}$$

where $\circ$ presents spatial-wise multiplication and Mean denotes calculating the mean in the spatial dimension only on the nonzero position of M_s. This way can avoid the influence of the proportion difference of the object on the whole image. With the temporary prototype P_t, we update and select the corresponding class prototype P_i from the previous class prototype bank P during training and testing respectively. In the training phase, when adapting target class i, P_i is located from P according to i for precisely improving the representational quality of the class prototype. Then, we momentum update P_i by:

$$P_i = (1-\alpha)\times l_2(P_t) + \alpha \times l_2(P_i), \tag{2}$$

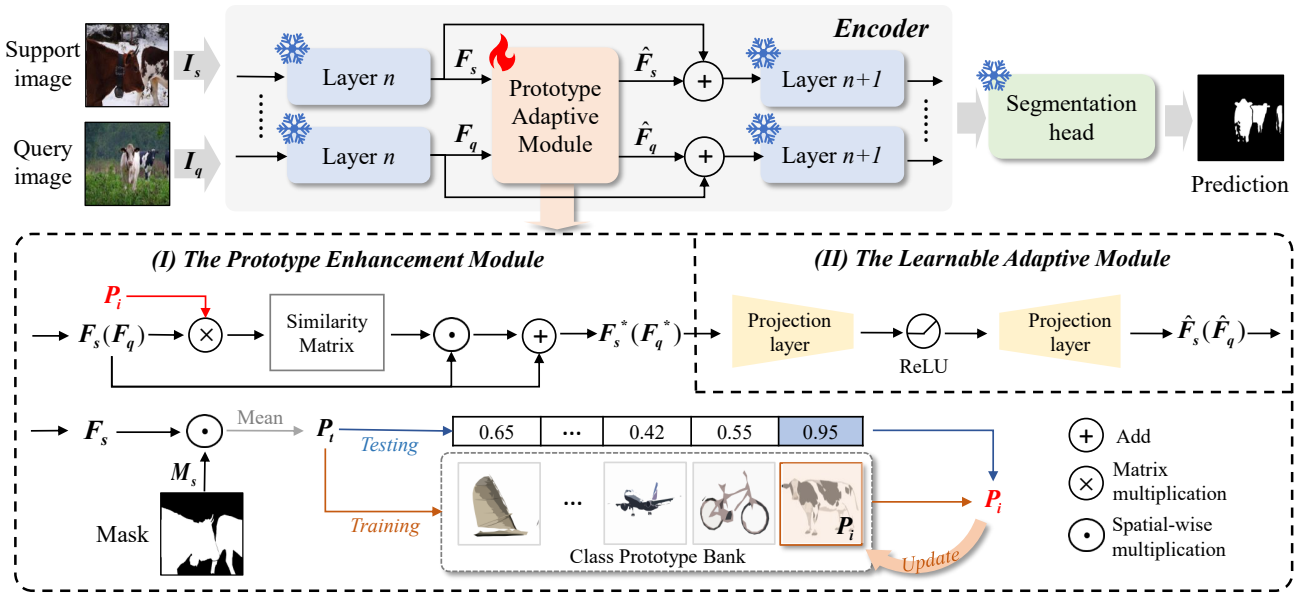

Figure 2: The overall architecture of our proposed Adaptive FSS. Given a support set $\{I_s, M_s\}$, the image I_s is fed into the encoder and generates feature F_s (F_q). In each PAM, with calculation between F_s and mask M_s, the temporary prototype P_t is firstly obtained for the selection of prototype P_i and bank update. After that, the corresponding class prototype P_i and feature F_s (F_q) are combined to generate the class-specific feature F_s^* (F_q^*). Finally, F_s^* (F_q^*) is sent into the Learnable Adaptive Module, leading to the acquired $\hat{F}_s$ ($\hat{F}_q$), which are injected into the encoder.

where l_2 and α denote L_2 normalization and momentum ratio respectively. In the testing phase, the prototype P_i is selected by similarity matching between P_t and P.

$$P_i = P_{\text{argmax}(l_2(P)\cdot l_2(P_t))}, \tag{3}$$

In this manner, each P_i can well represent the semantics of the corresponding category in the current feature representation space.

Feature Enhancement. As shown in Fig. 2, given a P_i, we can accurately enhance the part of the feature that associates with the new class to make the model easier to distinguish between the foreground and background. Firstly, we calculate the similarity map $S_s \in \mathbb{R}^{k\times h\times w}$ between F_s and P_i, namely

$$\begin{aligned} S_s &= l_2(F_s)\cdot l_2(P_i), \\ \text{ReLU6}(x) &= \text{Min}(6, \text{Max}(0, x)), \\ E_m &= \text{ReLU6}(S_s \times \sqrt{d}), \end{aligned} \tag{4}$$

where L_2 normalization and dot multiply are employed at d dimension. By utilizing the ReLU6, dissimilar positions are suppressed and similar points are retained while avoiding the impact of excessive values. With the enhancement matrix E_m, we can generate the class-specific feature F_s^*:

$$F_s^* = E_m \circ F_s + F_s. \tag{5}$$

To be emphasized, the query F_q feature is also enhanced by the above process with the same prototype P_i. The above only takes the support feature F_s as an example to illustrate.

The Learnable Adaptive Module. After the above process, the enhanced feature F_s^* and F_q^* can be obtained. However, the ability of the model to encode task-specific information is insufficient. Moreover, there is a problem of distribution differences between enhanced features F_s^* (F_q^*) and original features F_s (F_q) and it will increase cumulatively as the number of layers deepens. To solve these problems, a learnable module is adopted in our PAM. Since the samples available for training are very limited, the parameters of this module are very few that can be ignored compared with the entire model to alleviate the over-fitting phenomenon. Specifically, it includes a down-projection linear layer with parameters $W_{down} \in \mathbb{R}^{d\times\frac{d}{\gamma}}$ for compressing feature dimensions and an up-projection linear layer with parameters $W_{up} \in \mathbb{R}^{\frac{d}{\gamma}\times d}$ to recover feature dimensions. γ presents the hidden dimensions ratio and a ReLU layer is placed between two layers to complement the non-linear properties. The task-specific features $\hat{F}_s$ and $\hat{F}_q$ can be obtained as

$$\begin{aligned} \hat{F}_s &= \text{ReLU}(F_s^* \cdot W_{down})\cdot W_{up}, \\ \hat{F}_q &= \text{ReLU}(F_q^* \cdot W_{down})\cdot W_{up}. \end{aligned} \tag{6}$$

Then, we inject features $\hat{F}_s$ and $\hat{F}_q$ into the encoder as shown in Fig. 2. This process of injection is written as:

$$\begin{aligned} F_s &= \hat{F}_s \times \beta + F_s, \\ F_q &= \hat{F}_q \times \beta + F_q, \end{aligned} \tag{7}$$

where β denotes the scaling factor.

(a) PASCAL-5^i

Backbone	Method	Params	1-shot					5-shot				
			fold-0	fold-1	fold-2	fold-3	mIoU%	fold-0	fold-1	fold-2	fold-3	mIoU%
ResNet 50	BAM	51.63M	69.0	73.6	67.6	61.1	67.8	70.6	75.1	70.8	67.2	70.9
	HDMNet	51.40M	71.0	75.4	68.9	62.1	69.4	71.3	76.2	71.3	68.5	71.8
	MSANet	52.37M	69.3	74.6	67.8	62.4	68.5	72.7	76.3	73.5	67.9	72.6
	MSANet*	52.37M	69.9	74.9	64.8	61.3	67.7	74.0	76.4	69.8	68.0	72.1
	+ **Ours**	0.53M	**71.1**	**75.5**	**67.0**	**64.5**	**69.5**	**74.7**	**78.0**	**75.3**	**70.8**	**74.7**
DeiT-B/16	FPTrans	174.17M	72.3	70.6	68.3	64.1	68.8	76.7	79.0	81.0	75.1	78.0
	FPTrans*	174.17M	73.4	70.3	67.8	63.7	68.8	76.6	79.3	79.4	74.5	77.5
	+ AF	0.49M	74.0	70.6	67.7	66.2	69.8	77.1	79.8	80.0	74.7	77.9
	+ VPT	0.25M	73.8	70.6	67.7	64.2	69.1	76.8	79.6	80.0	75.1	77.9
	+ **Ours**	0.45M	**74.1**	**73.9**	**71.3**	**69.8**	**72.3**	**77.6**	**80.6**	**81.5**	**76.5**	**79.1**
Swin-B	DCAMA	92.97M	72.2	73.8	64.3	67.1	69.3	75.7	77.1	72.0	74.8	74.9
	+ F-Decoder	5.07M	73.2	73.7	65.8	67.5	70.1	76.3	77.6	71.8	74.7	75.1)
	+ F-Head	290	72.5	73.7	64.4	67.0	69.4	75.8	77.2	72.1	74.8	75.0
	+ **Ours**	0.23M	**74.3**	**74.9**	**70.5**	**69.8**	**72.4**	**76.8**	**78.8**	**74.3**	**77.1**	**76.7**

(b) COCO-20^i

Backbone	Method	Params	1-shot					5-shot				
			fold-0	fold-1	fold-2	fold-3	mIoU%	fold-0	fold-1	fold-2	fold-3	mIoU%
ResNet 50	BAM	51.63M	43.4	50.6	47.5	43.4	46.2	49.3	54.2	51.6	49.6	51.2
	MSANet	52.37M	45.7	54.1	45.9	46.4	48.0	50.3	60.9	53.0	50.5	53.7
	MSANet*	52.37M	41.9	53.1	45.5	46.7	46.8	46.7	60.3	53.1	50.4	52.6
	+ **Ours**	0.53M	**44.1**	**55.0**	**46.5**	**48.5**	**48.5**	**48.1**	**60.8**	**54.8**	**51.9**	**53.9**
	HDMNet	51.40M	43.8	55.3	51.6	49.4	50.0	50.6	61.6	55.7	56.0	56.0
	HDMNet*	51.40M	44.0	55.1	50.1	48.7	49.5	52.5	64.5	55.2	55.1	56.8
	+ **Ours**	0.53M	**44.9**	**56.7**	**51.4**	**49.6**	**50.7**	**53.0**	**66.4**	**56.1**	**55.8**	**57.8**
DeiT-B/16	FPTrans	176.66M	44.4	48.9	50.6	44.0	47.0	54.2	62.5	61.3	57.6	58.9
	FPTrans*	176.66M	43.0	49.6	48.0	43.2	46.0	54.5	63.6	59.8	56.9	58.7
	+ AF	0.49M	43.7	50.0	48.4	44.1	46.6	55.0	**64.1**	60.1	58.1	59.3
	+ VPT	0.25M	43.8	50.1	48.5	43.7	46.5	54.9	64.0	59.7	56.8	58.9
	+ **Ours**	0.45M	**45.3**	**53.9**	**49.5**	**44.5**	**48.3**	**57.1**	64.0	**60.7**	**58.2**	**60.0**
Swin-B	DCAMA	92.97M	49.5	52.7	52.8	48.7	50.9	55.4	60.3	59.9	57.5	58.3
	+ F-Decoder	5.07M	49.6	52.4	52.7	51.0	51.4	55.8	60.6	59.7	57.9	58.2
	+ F-Head	290	49.7	52.5	52.7	48.7	50.9	55.4	60.4	59.8	57.8	58.4
	+ **Ours**	0.23M	**51.4**	**53.1**	**54.4**	**51.9**	**52.7**	**57.8**	**61.0**	**60.9**	**58.5**	**59.6**

Table 1: Comparison with state-of-the-art methods on PASCAL-5^i and COCO-20^i. We report 1-shot and 5-shot results using the mIoU (%). $*$ denotes our implemented result. The gray font indicates the existing SOTA method.

Experiments

Implementation Details

Datasets and Metrics. We evaluate our proposed Adaptive FSS on PASCAL-5^i and COCO-20^i, which are two standard FSS benchmarks. PASCAL-5^i consists of PASCAL VOC 2012 (Everingham et al. 2010) and SBD (Hariharan et al. 2014) datasets. It comprises 20 classes that are divided into training and testing splits of 15 and 5 classes. COCO-20^i is created from COCO 2014 (Lin et al. 2014) dataset, which contains 80 classes. We divide them into four splits with each split of 60 classes for training and 20 classes for testing. For evaluation, we use the mean intersection over union (mIoU) to compare with previous methods.

Training Details. Our framework is implemented with Pytorch (Paszke et al. 2019) and trained on RTX 3090 GPUs with 1000 iterations. The cross-entropy loss is employed in our experiments and the batch size is set to 4 for PASCAL-5^i and COCO-20^i. We adopt the SGD optimizer, with a momentum of 0.9, weight decay of 0.001, and learning rate of 0.01. In each fold, two and six samples per novel class are randomly selected from the train set at 1-shot and 5-shot situations respectively for fine-tuning and avoiding image leak. The results of each experiment are the average results obtained from five random sampling trainings.

Comparison With State-of-the-Art Methods

Quantitative Results. To comprehensively evaluate our approach, we conduct experiments on four FSS networks (MSANet, HDMANet, FPTrans, and DCAMA), which adopt three popular backbones (ResNet, ViT, and Swin-B) as shown in Table 1. It is worth emphasizing that we followed

the test setting of DCAMA so that the performance of the other method is slightly different from that in the original paper. As expected, our method consistently improves the performance of existing FSS methods with different encoders on two benchmarks. For example, the result of DCAMA is respectively boosted by 3.1% mIoU and 1.8% mIoU in 1-shot and 5-shot settings on PASCAL-5^i. Besides, the different feature dimensions of multiple stages in various models result in an inconsistent number of parameters.

Moreover, we compare other fine-tuning strategies including finetune decoder (F-Decoder), finetune head (F-Head), and two Parameter-Efficient-Tuning methods (Adaptformer (AF) (Chen et al. 2022b) and VPT (Jia et al. 2022)). Since AF and VPT are well-designed for ViT, we conduct experiments on FPTrans. In particular, the classification head of FPTrans is based on similarity matching without learnable parameters, so the experiment of F-Decoder and F-Head is implemented on DCAMA. Our approach surpasses all fine-tuning strategies by a large margin. We provide comparisons of different models in the **Supplementary Material**.

Results Analysis. The large performance improvement can be explained by three factors. **(1)** Feature adaptation works better. In the existing FSS method based on meta-learning, the classifier is not strongly associated with the specific category. The model relies on correlation matching between the support feature and the query feature at the pixel level to distinguish foreground from background. Therefore, the limited adaptability of F-Head results in a subtle improvement compared to AF, VPT, and F-Decoder. **(2)** The class information contained in the support feature and mask is beneficial to adaptation to novel classes. In each fold, the multiple classes (i.e., 5 classes in PASCAL-5^i) need to be adapted. An effective fine-tuning strategy should help the model extract class-specific features for different test classes by support set. However, the AF and VPT only insert learnable parameters and modules to independently model task-specific features without utilizing the category information. Our approach can greatly achieve this purpose through the well-designed PEM. **(3)** Our method can memorize category-specific knowledge through the class prototype bank (CPB), obtaining high-quality representation for different categories. Ablation studies in Table 2a confirm PEM and CPB as the main reasons for performance improvement.

Qualitative Results. We provide a visual comparison between the baseline and our Adaptive FSS on the PASCAL-5^i as shown in Fig. 3. The FPTrans without finetuning is chosen as the baseline. Our method achieves high-quality segmentation due to adapting the model to new categories effectively.

(a) Support Set (b) Baseline (c) Adaptive FSS (d) Ground Truth

Figure 3: The visual comparison between baseline and our proposed Adaptive FSS on PASCAL-5^i in the 1-shot setting.

Ablation Study

In this subsection, various ablation experiments are conducted on the PASCAL-5^i with the FPTrans, which are summarized in Table 2. We verify the effectiveness of major components in PAM and explore the effect of the insert strategy, momentum ratio, scaling factor, and hidden ratio.

Major Components. As shown in Table 2a, we investigate the effectiveness of key components (i.e., the learnable adaptive module (LAM) and the prototype enhancement module (PEM)) in the designed PAM. Compared to 68.8% mIoU without adaptation on novel classes, the performance is increased by 1.3% mIoU when only using the LAM. Moreover, the result is further improved by 1.3% mIoU, using the temporary prototype P_t to guide subsequent enhancement operations. Finally, with the class prototype bank (CPB), the full PAM promotes the performance to reach 72.3% mIoU, which illustrates that the representation quality of the prototype contributes to precise segmentation.

Insert Strategy. As shown in Table 2b, we take ViT as the backbone to explore the effect of PAM insertion position. Specifically, we explore four ways of inserting, including the front (i.e., '$1 \rightarrow 6$' in Table 2b denotes means inserting PAM in the layers of the model from 1 to 6), middle, back, and all layers of the model. The front and middle approaches achieved similar results. Inserting PAM at the back of the model outperforms others. The reason is that the high-level semantic information, which is conducive to prototype extraction, is located in the deep layer of the network. Higher-quality object prototypes enable more precise association and feature augmentation. Moreover, this characteristic is more prominent for DCAMA, MSANet, and HDMNet based on the hierarchical Swin Transformer and ResNet. Therefore, for the former, we evenly insert the PAM in the last two stages. Due to the different structures between the ResNet-50 and Swin-B, the PAM is only plugged into the last stage at the ResNet-50. Relevant results and detailed explanations are available in the **Supplementary Material**.

Momentum Ratio. As described in Section 3.2, the momentum ratio α is introduced to control the rate of updating

LAE	PEM	mIoU%
		68.8
✓		70.1
✓	✓	71.4
✓	✓(w CPB)	72.3

(a) Major Components.

Strategy	Params	mIoU%
$1 \rightarrow 6$	0.45M	71.8
$3 \rightarrow 8$	0.45M	71.6
$7 \rightarrow 12$	0.45M	72.3
$1 \rightarrow 12$	0.82M	72.1

(b) Insertion Strategy.

α	mIoU%
0.999	72.0
0.99	72.3
0.95	71.9
0.9	71.8

(c) Momentum Ratio α.

β	mIoU%
1	66.7
0.5	69.2
0.1	72.3
0.01	71.0

(d) Scaling Factor β.

γ	mIoU%
4	71.8
8	71.9
16	72.3
32	71.5

(e) Hidden Ratio γ.

Table 2: Ablation study with FPTrans on PASCAL-5^i. The default settings of our method are marked in underline.

prototypes in the class prototype bank. We conduct experiments with the different α as shown in Table 2c. We found that $\alpha = 0.99$ performs the best and set it as the default.

Scaling Factor. The scaling factor β is introduced to balance the original features and task-specific features. We evaluate the performance when selecting different β as shown in Table 2d. Obviously, β = 0.1 achieves the best results. Increasing or decreasing β will bring a performance drop. Therefore, we chose 0.1 as the default setting.

Hidden Ratio. As mentioned in Section 3.2, the hidden ratio γ can control the number of parameters introduced by PAM. A lower ratio means that more parameters are introduced and the task-specific ability is stronger. We study the impact of this hyper-parameter on the performance of the model as shown in Table 2e. It is observed that using 16 as a hidden ratio achieves the highest accuracy on PASCAL-5^i for our method. Increasing γ from 16 to 32 results in a performance drop from 72.3% to 71.5% mIoU.

Discussion

Strong Adaptation Ability of Adaptive FSS. In Table 3, we further explore the adaptability of the proposed Adaptive FSS for cross-domain segmentation, following the experimental setup in previous works (Zhang et al. 2022a; Min, Kang, and Cho 2021). We fine-tune and test FPTrans on new classes in PASCAL-5^i, utilizing weights trained on base classes in COCO-20^i. Due to the difference in testing environment, our realized results (68.2% mIoU) are different from 69.7% mIoU in the original paper. Our method obtains significant improvement from 68.2% to 72.5% mIoU and exhibits superiority in the presence of domain gaps.

COCO $\rightarrow$ PASCAL	FPTrans†	FPTrans	Ours
Mean IoU%	68.2	69.7	**72.5 (+4.3)**

Table 3: Evaluation under the domain shift from COCO-20^i to PASCAL-5^i. † denotes our realized result.

Only One Labeled Sample per Class Is Available for Fine-Tuning. To cope with this problem, we explore a compromised training method using only one image simultaneously as the query and support image. Specifically, we utilize the original sample as query and the data-augmented sample as support for fine-tuning. Among them, data enhancement methods such as random rotation, random cropping, horizontal flipping, and vertical flipping are adopted. As shown in Table 4, we use FPTrans as the baseline and experiment on PASCAL-5^i in 1-shot setting. Our method achieves an improvement of 1.6% mIoU, which demonstrates the superiority and robustness of Adaptive FSS.

Method	fold-0	fold-1	fold-2	fold-3	Mean IoU%
FPTrans	73.4	70.3	67.8	63.7	68.8
Ours	**73.8**	**73.2**	**69.4**	**65.0**	**70.4 (+1.6)**

Table 4: Evaluation under only one labeled sample.

The Reason for Performance Improvement. It is the effectiveness of Adaptive FSS rather than extra labeled samples. To prove it, we prevent the influence of additional labeled images by reusing the fine-tuning set as the support set during inference. Specifically, two samples per class act as query and support alternately in fine-tuning, constituting the fine-tuning set. In the inference phase, they are both as support in each inference, which is a 2-shot test setting. Meanwhile, DCAMA employs a 1-shot training and few-shot testing manner, it can flexibly adapt to different numbers of support images. Therefore, we adopt DCAMA as the baseline. As evident from Table 5, our method achieves a result of 75.0 % mIoU with an improvement of 2.4% mIoU.

Method	fold-0	fold-1	fold-2	fold-3	Mean IoU%
DCAMA	72.8	75.2	69.3	73.3	72.6
Ours	**74.6**	**75.9**	**74.9**	**74.4**	**75.0 (+2.4)**

Table 5: Evaluation on the influence of extra labeled sample.

Conclusion

In this paper, we propose a novel FSS framework based on the adapter mechanism that can greatly improve the performance of the current FSS model by adapting it to novel classes. The well-designed PAM could accurately guide the enhancement of features with a high correlation to new objects. The experiments verify that Adaptive FSS can achieve a considerable improvement on diverse FSS networks based on a variety of backbones. We also explore the performance of our method in the presence of domain gaps and only one training sample. Furthermore, we demonstrate that the superiority of Adaptive FSS is independent of additional samples. We hope that our approach can serve as a strong baseline for the novel class adaptation of FSS in future research.

Acknowledgements

This work was supported by National Key Research and Development Program of China (2022YFB3304000), in part by the Natural Science Foundation of China under Grant 42201386, in part by the Fundamental Research Funds for the Central Universities and the Youth Teacher International Exchange and Growth Program of USTB (QNXM20220033), and Interdisciplinary Research Project for Young Teachers of USTB (Fundamental Research Funds for the Central Universities: FRF-IDRY-22-018), and Scientific and Technological Innovation Foundation of Shunde Innovation School, USTB (BK20BE014).

References

Boudiaf, M.; Kervadec, H.; Masud, Z. I.; Piantanida, P.; Ben Ayed, I.; and Dolz, J. 2021. Few-shot segmentation without meta-learning: A good transductive inference is all you need? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13979–13988.

Chen, H.; Tao, R.; Zhang, H.; Wang, Y.; Ye, W.; Wang, J.; Hu, G.; and Savvides, M. 2022a. Conv-adapter: Exploring parameter efficient transfer learning for convnets. *arXiv preprint arXiv:2208.07463*.

Chen, J.; Gao, B.-B.; Lu, Z.; Xue, J.-H.; Wang, C.; and Liao, Q. 2021. Apanet: adaptive prototypes alignment network for few-shot semantic segmentation. *arXiv preprint arXiv:2111.12263*.

Chen, S.; Ge, C.; Tong, Z.; Wang, J.; Song, Y.; Wang, J.; and Luo, P. 2022b. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35: 16664–16678.

Dhillon, G. S.; Chaudhari, P.; Ravichandran, A.; and Soatto, S. 2019. A baseline for few-shot image classification. *arXiv preprint arXiv:1909.02729*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88: 303–338.

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

Hariharan, B.; Arbeláez, P.; Girshick, R.; and Malik, J. 2014. Simultaneous detection and segmentation. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13*, 297–312. Springer.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, 2790–2799. PMLR.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Hu, Z.; Lan, Y.; Wang, L.; Xu, W.; Lim, E.-P.; Lee, R. K.-W.; Bing, L.; and Poria, S. 2023. LLM-Adapters: An Adapter Family for Parameter-Efficient Fine-Tuning of Large Language Models. *arXiv preprint arXiv:2304.01933*.

Iqbal, E.; Safarov, S.; and Bang, S. 2022. Msanet: Multi-similarity and attention guidance for boosting few-shot segmentation. *arXiv preprint arXiv:2206.09667*.

Jia, M.; Tang, L.; Chen, B.-C.; Cardie, C.; Belongie, S.; Hariharan, B.; and Lim, S.-N. 2022. Visual prompt tuning. In *European Conference on Computer Vision*, 709–727. Springer.

Kang, D.; and Cho, M. 2022. Integrative few-shot learning for classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9979–9990.

Kang, D.; Koniusz, P.; Cho, M.; and Murray, N. 2023. Distilling Self-Supervised Vision Transformers for Weakly-Supervised Few-Shot Classification & Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19627–19638.

Kang, D.; Kwon, H.; Min, J.; and Cho, M. 2021. Relational embedding for few-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8822–8833.

Lang, C.; Cheng, G.; Tu, B.; and Han, J. 2022. Learning what not to segment: A new perspective on few-shot segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8057–8067.

Li, W.-H.; Liu, X.; and Bilen, H. 2022. Cross-domain few-shot learning with task-specific adapters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7161–7170.

Li, X. L.; and Liang, P. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

Liu, W.; Zhang, C.; Ding, H.; Hung, T.-Y.; and Lin, G. 2021a. Few-shot segmentation with optimal transport matching and message flow. *arXiv preprint arXiv:2108.08518*.

Liu, W.; Zhang, C.; Lin, G.; and Liu, F. 2020. Crnet: Cross-reference networks for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4165–4173.

Liu, X.; Ji, K.; Fu, Y.; Tam, W.; Du, Z.; Yang, Z.; and Tang, J. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 61–68.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021b. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Lu, Z.; He, S.; Zhu, X.; Zhang, L.; Song, Y.-Z.; and Xiang, T. 2021. Simpler is better: Few-shot semantic segmentation with classifier weight transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8741–8750.

Min, J.; Kang, D.; and Cho, M. 2021. Hypercorrelation squeeze for few-shot segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6941–6952.

Pan, J.; Lin, Z.; Zhu, X.; Shao, J.; and Li, H. 2022. St-adapter: Parameter-efficient image-to-video transfer learning. *Advances in Neural Information Processing Systems*, 35: 26462–26477.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Peng, B.; Tian, Z.; Wu, X.; Wang, C.; Liu, S.; Su, J.; and Jia, J. 2023. Hierarchical Dense Correlation Distillation for Few-Shot Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23641–23651.

Rakelly, K.; Shelhamer, E.; Darrell, T.; Efros, A.; and Levine, S. 2018. Conditional networks for few-shot semantic segmentation.

Ravi, S.; and Larochelle, H. 2016. Optimization as a model for few-shot learning. In *International conference on learning representations*.

Schwartz, E.; Karlinsky, L.; Shtok, J.; Harary, S.; Marder, M.; Pankanti, S.; Feris, R. S.; Kumar, A.; Giryes, R.; and Bronstein, A. M. 2018. RepMet: Representative-Based Metric Learning for Classification and Few-Shot Object Detection. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5192–5201.

Shaban, A.; Bansal, S.; Liu, Z.; Essa, I.; and Boots, B. 2017. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*.

Shi, X.; Wei, D.; Zhang, Y.; Lu, D.; Ning, M.; Chen, J.; Ma, K.; and Zheng, Y. 2022. Dense cross-query-and-support attention weighted mask aggregation for few-shot segmentation. In *European Conference on Computer Vision*, 151–168. Springer.

Tian, P.; Wu, Z.; Qi, L.; Wang, L.; Shi, Y.; and Gao, Y. 2020. Differentiable meta-learning model for few-shot semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 12087–12094.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, 29.

Wang, Y.; Mukherjee, S.; Liu, X.; Gao, J.; Awadallah, A. H.; and Gao, J. 2021. List: Lite prompted self-training makes parameter-efficient few-shot learners. *arXiv preprint arXiv:2110.06274*.

Yang, B.; Liu, C.; Li, B.; Jiao, J.; and Ye, Q. 2020a. Prototype mixture models for few-shot semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, 763–778. Springer.

Yang, X.; Wang, B.; Chen, K.; Zhou, X.; Yi, S.; Ouyang, W.; and Zhou, L. 2020b. Brinet: Towards bridging the intra-class and inter-class gaps in one-shot segmentation. *arXiv preprint arXiv:2008.06226*.

Yoon, S. W.; Seo, J.; and Moon, J. 2019. TapNet: Neural Network Augmented with Task-Adaptive Projection for Few-Shot Learning. *ArXiv*, abs/1905.06549.

Zhang, B.; Xiao, J.; and Qin, T. 2021. Self-guided and cross-guided learning for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8312–8321.

Zhang, C.; Lin, G.; Liu, F.; Yao, R.; and Shen, C. 2019. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5217–5226.

Zhang, G.; Kang, G.; Yang, Y.; and Wei, Y. 2021. Few-shot segmentation via cycle-consistent transformer. *Advances in Neural Information Processing Systems*, 34: 21984–21996.

Zhang, J.-W.; Sun, Y.; Yang, Y.; and Chen, W. 2022a. Feature-proxy transformer for few-shot segmentation. *Advances in Neural Information Processing Systems*, 35: 6575–6588.

Zhang, Q.; Wu, X.; Yang, Q.; Zhang, C.; and Zhang, X. 2022b. HG-Meta: Graph Meta-learning over Heterogeneous Graphs. In *SDM*.

Zhang, R.; Zhang, W.; Fang, R.; Gao, P.; Li, K.; Dai, J.; Qiao, Y.; and Li, H. 2022c. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European Conference on Computer Vision*, 493–510. Springer.

Zhang, X.; Wei, Y.; Yang, Y.; and Huang, T. S. 2020. Sg-one: Similarity guidance network for one-shot semantic segmentation. *IEEE transactions on cybernetics*, 50(9): 3855–3865.

Zheng, W.; Tian, X.; Yang, B.; Liu, S.; Ding, Y.; Tian, J.; and Yin, L. 2022. A few shot classification methods based on multiscale relational networks. *Applied Sciences*, 12(8): 4059.

Zhu, Q.; Li, B.; Mi, F.; Zhu, X.; and Huang, M. 2022. Continual prompt tuning for dialog state tracking. *arXiv preprint arXiv:2203.06654*.