

# Relevant Intrinsic Feature Enhancement Network for Few-Shot Semantic Segmentation

Xiaoyi Bao,<sup>\*1,2,3</sup> Jie Qin,<sup>\*1,2</sup> Siyang Sun,<sup>3</sup> Yun Zheng,<sup>3</sup> Xingang Wang<sup>2†</sup>

<sup>1</sup>School of Artificial Intelligence, University of Chinese Academy of Sciences

<sup>2</sup>Institute of Automation, Chinese Academy of Sciences <sup>3</sup>Alibaba group

{baoxiaoyi2021, qinjie2019}@ia.ac.cn, {siyang.ssy,zhengyun.zy}@alibaba-inc.com, xingang.wang@ia.ac.cn

## Abstract

For few-shot semantic segmentation, the primary task is to extract class-specific intrinsic information from limited labeled data. However, the semantic ambiguity and inter-class similarity of previous methods limit the accuracy of pixel-level foreground-background classification. To alleviate these issues, we propose the Relevant Intrinsic Feature Enhancement Network (RiFeNet). To improve the semantic consistency of foreground instances, we propose an unlabeled branch as an efficient data utilization method, which teaches the model how to extract intrinsic features robust to intra-class differences. Notably, during testing, the proposed unlabeled branch is excluded without extra unlabeled data and computation. Furthermore, we extend the inter-class variability between foreground and background by proposing a novel multi-level prototype generation and interaction module. The different-grained complementarity between global and local prototypes allows for better distinction between similar categories. The qualitative and quantitative performance of RiFeNet surpasses the state-of-the-art methods on *PASCAL-5<sup>i</sup>* and *COCO* benchmarks.

## 1 Introduction

As a fundamental and crucial task in the field of computer vision, semantic segmentation is widely applied in visual tasks such as medical image understanding, industrial defect inspection, virtual reality games, autonomous driving, etc. With the development of convolutional neural networks, fully-supervised semantic segmentation has achieved remarkable success (?). Subsequent transformer-based methods also greatly improve the segmentation performance (???). However, it is still arduous to acquire pixel-level annotations that require a huge amount of manual effort and costs. To alleviate such expensive and unwieldy annotations, many works tend to resort to a few-shot semantic segmentation paradigm. In this setting, the model learns to segment with a few limited labeled data as the support set and then transferred to test the query inputs.

Previous approaches (??) focus on making full use of limited support data. The training data is divided into a support

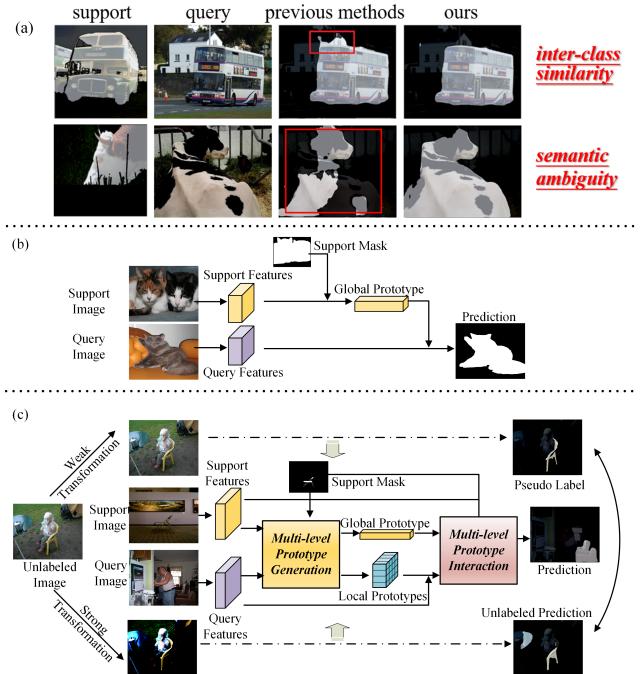


Figure 1: Comparison of RiFeNet and other works. (a) Two foreground-related issues that limit the effectiveness of previous research. (b) Schedule of previous prototype-based methods. The two-branch structure uses a single global prototype as the only medium for information interaction. (c) Framework of RiFeNet. An additional unlabeled branch is used for feature mining during training, with multi-granularity prototypes extracted from different branches.

set and a query set, which are processed using two different branches. Relationship prompt and information interaction between the two branches are achieved by proposing superior matching strategies (?) or generating representative prototypes (????). The former class of methods is represented by HSNet (?), which usually uses a cost volume method to aggregate dense correlation scores in high-dimensional space by 4D convolution. Subsequent researches improve the network using a swin transformer (?) or a doubly deformable transformer (?) in 4-dimension space. With much

<sup>\*</sup>These authors contributed equally.

<sup>†</sup>Corresponding author.

lower computational costs, prototype-based methods also achieve good segmentation results. As illustrated in Fig. 1 (a), the single or multiple prototypes integrating semantic class-specific features are extracted from the support branch and used to activate query features. Researchers have proposed different activation methods in PPNet (?), PFENet (?), CyCTR (?), etc., to fully exploit and utilize the limited number of category semantic features.

The object to be segmented is called the foreground while the background refers to the rest of the stuff and things. Despite the prevailing success of these methods, there are still two main issues affecting the segmentation effect of previous methods, *i.e.* **semantic ambiguity** and **inter-class similarity**, as shown in Fig. 1 (a). For the foreground itself, semantic ambiguity appears for different instances of the same class. The intra-class differences between the two branches lead to semantic mistakes on query images. Further to the problem of distinguishing between foreground and background, inter-class similarity is specifically the difficulty in pixel-level binary classification. When objects of different classes with similar textures appear simultaneously, the local features of foreground and background become confusing.

The semantic ambiguity of the foreground is caused by the poor intra-class generalization of the model. Previous models mainly explore how to match the support set and the query set. With little labeled data in few-shot tasks, their models are prone to extract semantically ambiguous shallow features such as shape and pose, and thus unable to learn robust class representations for highly diverse query appearances. This distribution discrepancy makes the model identify the semantic parts of the foreground objects incompletely and segment them inappropriately. As for the inter-class similarity between foreground and background, the lack of discriminative features for query data is to blame for this problem. Specifically, previous methods extract global prototypes or prototypes with a single level of granularity. The information they carry is inadequate and monolithic, thus leading to pixel misidentify.

To address the above problems, we propose a novel relevant intrinsic feature enhancement network, as shown in Fig. 1 (c). For foreground objects, we enhance the semantic consistency of the same class, thus improving the intra-class generalization of features. We incorporate a novel unlabeled branch into the support-query pair branches to teach the model how to mine intrinsic robustness behind appearance variability. The additional unlabeled data branch serves as a regularization term during training that enhances the semantic consistency of few-shot segmentation models.

We go one step further in achieving an overall enhancement of the distinction between foreground and background. A novel multi-level prototype generation and interaction module is proposed to ensure inter-class variability. The multi-level prototype features contain a global prototype from the support data and a local prototype from the query data. The former represents the high-level semantic abstraction about the entire structure of the target category, while the latter captures the fine-grained appearance concepts providing details for category discrimination. The global and local prototype interaction can complement each other to

ensure that the network extracts the corresponding category features. This multi-level prototype feature interaction module can greatly widen the inter-class differences, benefiting the identification between foreground and background.

Our contributions can be summarized as follows:

- We propose a relevant intrinsic feature enhancement network. By alleviating semantic ambiguity and inter-class similarity, our model improves the forehead segmentation performance in this few-shot task.
- To maintain the foreground semantic consistency, we propose an unlabeled branch as an efficient data utilization method, which teaches the model how to extract intrinsic features robust to intra-class differences.
- To further achieve an overall effective distinction between foreground and background objects, we propose a novel multi-level prototype generation and interaction module to extend the inter-class variability. With different-grained semantic information from different sources, the mutual complementarity of information is facilitated and the discriminative representation ability of confusing foreground and background objects is boosted.
- Extensive experiments demonstrate the effectiveness of our proposed method, which achieves state-of-the-art accuracy on both PASCAL-5<sup>i</sup> and COCO benchmarks.

## 2 Related Work

### 2.1 Semantic Segmentation

Rapid progress has been made on semantic segmentation tasks since fully convolutional networks (FCN) transformed them into pixel-level classification (?). After that, the corresponding encoder-decoder architecture has been widely used (??). Recent research mainly focuses on multi-scale feature fusion (??), insertion of attention modules (?????), and context prior (???).

Inspired by the success of the vision transformer (?), researchers have carried out attempts to apply the transformer structure to segmentation (????). These works perform well under the task setting of semantic segmentation. However, in practical application scenarios, they are unable to cope with sparse training data and thus often fail on categories that they have not seen during the training process.

### 2.2 Few-Shot Segmentation

A two-branch structure is widely used in few-shot segmentation, *i.e.*, a support branch and a query branch. Mainstream few-shot methods are divided into two categories: prototype extraction and spatial correlation. With the development of image-matching, spatial correlation-based methods have regained attention in few-shot segmentation. HSNet (?) applies 4D convolution to 4D correlation tensors in a pyramid structure, and the following ASNet (?) squeezes semantic correlation into the foreground map. VAT (?) and DACM (?) use cost volume aggregation-based models with 4D transformers.

The above methods benefit from the retained spatial structure. However, their high computational complexity and the excessive number of parameters are the drawbacks, making

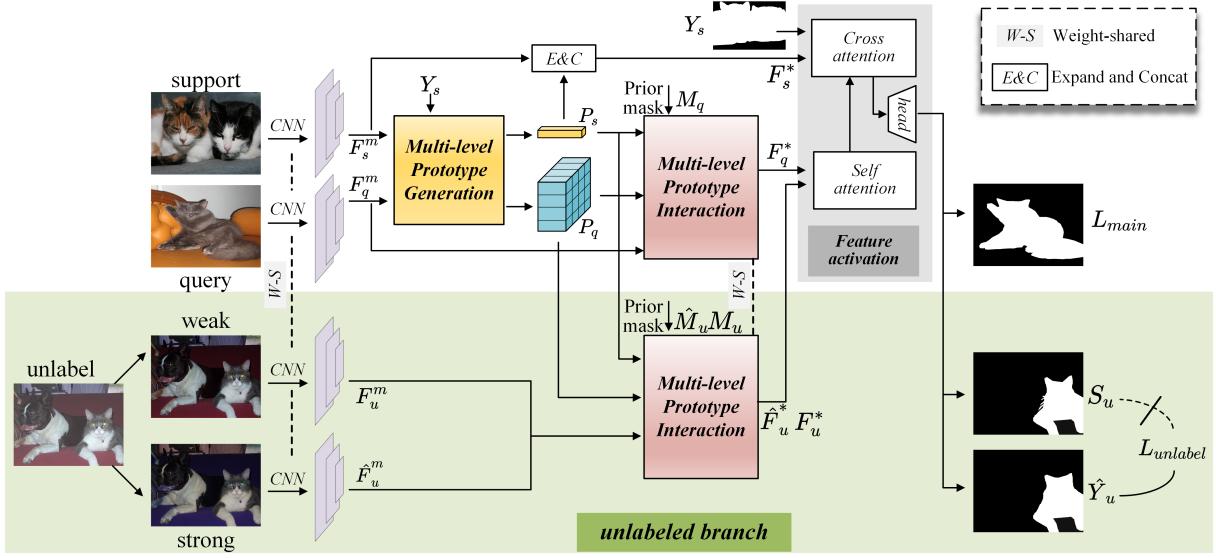


Figure 2: Overall architecture of RiFeNet. An additional branch using unlabeled inputs is attached to the traditional two-branch structure, as shown in green. The forward process of RiFeNet consists of three main blocks. The multi-level prototype generation block generates global and local prototypes. The multi-level prototype interaction module allows for interaction and integration between prototypes. A feature activation block is used consequently for obtaining final segmentation results.

training slow and failing to generalize. Since PL (?) introduced prototype learning to few-shot segmentation, most of the following research focus on prototype-based methods (??). SG-ONE (?) proposes to use masked average pooling to extract a prototype from support features carrying the category information. Dense matching computation is performed between query features and this prototype. Tian proposes the PFENet (?), adding a train-free prior mask to the feature enrichment process. Multi-prototypes are generated in PPNet (?), ASGNet (?), and RPMMs (?) to include more local information with different generating methods, while the use of multiple prototypes in these networks does not bring them more competitive results. IPMT (?) proposes an intermediate prototype and uses the transformer to update the prototype iteratively.

SSP (?) also proposes the idea of the self-support prototype, but their single spacial-agnostic prototype is generated for self-matching. Although also generated from query branches, our multiple local prototypes carry discriminative and comprehensive spatial information that helps to provide details for intra-class discrimination.

### 2.3 Unlabeled Usage in Few-Shot Segmentation

Very few researchers have explored the leverage of unlabeled data in few-shot tasks. PPNet (?) is one of the cases, which incorporates 100 unlabeled data into every support input, with the use of a graph neural network (GNN). Based on it, Soopil proposes an uncertainty-aware model to make more adequate use of unlabeled data without using graph neural networks (?). However, the above two methods also require additional unlabeled data of the current novel class, which is inconsistent with the original few-shot task setting but relevant to the semi-supervised paradigm. And the num-

ber of unlabeled images used in every meta-training process far exceeds that of support and query images. In contrast, our work removes the dependence of the testing process on unlabeled data. During training, a tiny amount of unlabeled data is used to constrain the foreground semantic similarity.

## 3 Method

### 3.1 Problem Definition

The few-shot semantic segmentation task is defined as the segmentation of novel-class objects, based on a very small number of images with pixel-level annotations. According to this definition, the dataset for training  $D_{train}$  and the one for testing  $D_{test}$  have no overlap in their categories.

In each meta-training process of  $K$ -shot setting, conventional methods randomly sample  $K + 1$  image pairs of the same class  $j$ :  $\{(I^i, Y^i), i \in \{1, 2, \dots, K + 1\} | (I^i, Y^i) \in D_{train}, C^i = j\}$ .  $I^i$ ,  $Y^i$ , and  $C^i$  refer to the RGB image, its segmentation mask, and the class of the segmented object. Among them,  $K$  pairs are sent to the support branch, and the last is used as the query input and ground truth. The difference in our network is that we take  $K + 1 + M$  pairs of images from  $D_{train}$  at a time, and  $I^i$  of the extra  $M$  pairs are used as input to the unlabeled branch.

In each testing process, the  $K + 1$  pairs from the  $j$ -th novel classes  $\{(I^i, Y^i), i \in \{1, 2, \dots, K + 1\} | (I^i, Y^i) \in D_{test}, C^i = j\}$  are sampled and assigned to support and query branch as before.

### 3.2 Relevant Intrinsic Feature Enhancement

As a pixel-level binary classification task, semantic segmentation needs to identify the foreground, *i.e.* object to be segmented, and the background, *i.e.* the rest of the stuff and

things. To alleviate the semantic ambiguity and inter-class similarity in Fig.1, we propose this relevant intrinsic feature enhancement network. We present the complete framework under the 1-shot setting in Fig. 2. The proposed RiFeNet consists of three branches with a shared backbone. The extra unlabeled branch helps the traditional support-query framework learn how to assure semantic consistency.

The forward process consists of three main modules: the multi-level prototype generation module, the multi-level prototype interaction module, and the feature activation module. The first two modules are used to provide multi-grained evidence for better inter-class discrimination. The feature activation module activates pixels containing objects of the target class and deactivates the others, providing the final segmentation result.

### 3.3 Feature Enhancement with Unlabeled Data

There are large intra-class distributional differences in the foreground objects that need to be segmented. Therefore, the training purpose is to ensure the semantic consistency of features extracted from different instances of the foreground category, rather than focusing only on semantically ambiguous appearance features.

Therefore, we introduce an auxiliary unlabeled branch as an effective data utilization method to aid model learning. Without adding training data, we re-sample a subset of training samples as unlabeled data, with the same segmentation loss applied. By augmenting the sample diversity, it teaches the model to avoid learning sample-specific biases of labeled inputs, even in the absence of unlabeled data during testing.

Our unlabeled branch shares parameters with the query branch to align features and enhance relevance. It is trained with pseudo-labels generated from each other. Specifically, the initial unlabeled input goes through two versions of data augmentation, different in their kinds and intensities. We use  $I_u$  and  $\hat{I}_u$  to refer to the weakly and strongly transformed unlabeled input. Both are sent to the backbone simultaneously with an identical forward process, thus we use symbols without a hat in all subsequent equations for simplicity.

$$F_u^m = \text{ReLU}(\text{Conv}(\text{CAT}(F^1(I_u), F^2(I_u)))). \quad (1)$$

$F_u^m$  refers to merged unlabeled feature maps. For the sake of simplicity, we omit the above processing in Fig.2. The support and query branches are handled in the same way, and all parameters in Eq.1 are shared with query branches.  $F^j(I_u)$  corresponds to the feature map from the  $j$ -th backbone layer.

As for the prototype generation block, if we follow the query paradigm to generate local prototypes from the unlabeled branch, the self-supervised training may be off-target in a few extreme cases. In complex environments with multiple objects, the segmented class in the generated pseudo mask may not be inconsistent. This off-track pseudo-truth may worsen the expression capability of the model.

We introduce query prototypes here to provide class-specific prior. Only intrinsic features with intra-class relevance are extracted when processing unlabeled data.

The subsequent prototype interaction is the same as the query one. Augmented unlabeled features are obtained and

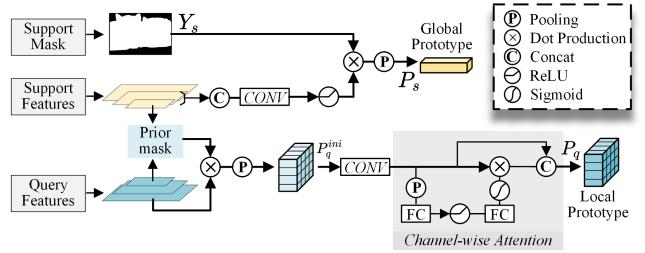


Figure 3: Visual illustration of the multi-level prototype generation block that extracts global and local prototypes.

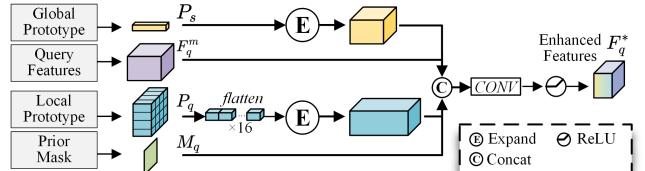


Figure 4: Schedule of the multi-level prototype interaction merging relevant intrinsic information for enhancement.

sent to the activation block for prediction.  $S^u$  and  $\hat{Y}^u$  represent the predicted results of weakly and strongly transformed inputs, respectively. The former output  $S^u$  serves as a pseudo label for the latter one. The loss of this self-supervised process is calculated as follows, with dice loss (?) as its training objective.

$$\mathcal{L}_{unlabel} = \text{DICE}(S^u, \hat{Y}^u). \quad (2)$$

### 3.4 Multi-level Prototype Processing

While the unlabeled branch ensures intra-class semantic consistency of the foreground, the background noise it introduces may worsen existing inter-class similarity between foreground and background. Therefore, we propose the multi-level prototype generation and interaction blocks aim to capture representative information in different granularity and augment the discrimination of foreground and background in confusing areas.

**Global support prototype generation.** To capture category information at the high-dimensional category level, we extract global prototypes from support features.

As shown in Fig.3,  $P_s \in \mathbb{R}^{1 \times 1 \times C}$  is obtained through masked average pooling on the global feature.  $w$  and  $h$  are the width and height of  $F_s^m$ .  $\odot$  means dot production.

$$P_s = \frac{\sum_{x=1,y=1}^{w,h} (F_s^m \odot Y_s)^{x,y}}{\sum_{x=1,y=1}^{w,h} Y_s^{x,y}}. \quad (3)$$

**Local query prototype generation.** In some cases, the similarities between different classes may be more significant than the differences within classes (?). Meanwhile, the prototypes obtained by global average pooling completely discard spatial structure information, making the decoupling of local components unrealistic.

To provide discriminative details, we additionally extract local prototypes from the query branch. These prototypes facilitate binary classification by prompting fine-grained information. Fig.3 illustrates their generation process.

$$P_q^* = \text{Pooling}_{avg}(F_q^m \odot M(F^3(I_q))). \quad (4)$$

$M(F^3(I_q))$  is the prior mask proposed in PFENet (?), as a form of similarity measure between high-level feature maps of support and query branches. Local average pooling is done on the multiplication of prior masks and query feature maps.

To squeeze the channel and enhance the representation of local prototypes,  $1 \times 1$  convolution and channel-wise attention are used. The refined local prototypes  $P_q \in \mathbb{R}^{m \times m \times C'}$  from the query branch itself are obtained as follows. The symbol  $\text{Att}_{chn}$  refers to channel-wise attention.

$$P_q = \text{Att}_{chn}(\text{Conv}^{1 \times 1}(P_q^*)). \quad (5)$$

For the generation of prior masks, fixed high-level feature maps  $F_s^4$  and  $F_q^4$  are more suitable. But when generating local query prototypes, superficial partly appearance information counts a lot. With the need for pixel-level correspondence, we use feature map  $F_q^1$  from the first layer to extract locally consistent appearance information in the local area, as shown in Fig.3.

**Multi-level prototype interaction.** To emphasize the intrinsic class information, we build interactions between different-grained prototypes, thus enhancing the feature-mining ability for identification.

Fig.4 shows detailed steps of our interaction block. The generated global and local prototypes are expanded to the size of feature maps and then concatenated with query features and the prior mask. After a  $1 \times 1$  convolution with activation, augmented query features  $F_q^*$  are obtained.

$$F_q^* = \text{ReLU}(\text{Conv}(\text{CAT}(P_s, P_q, F_q^m, M(F^3(I_q))))). \quad (6)$$

### 3.5 Feature activation

For deeper information interaction between the query and the support branch, we use an n-layer transformer encoder for feature activation. In this process, information with relevance is activated, enhancing the total feature map.

Augmented query features  $F_q^*$  are processed by self-attention and then cross-attention with support features  $F_s^*$ . Each block is made up of a multi-head attention and a feed-forward network. The setting of multi-head attention for cross-attention follows the one in CyCTR (?), with the segmentation mask of the support image  $Y_s$  used to maintain the cycle consistency.

$$F_q^{final} = \text{Att}_{cross}(\text{Att}_{self}(F_q^*), F_s^*, Y_s). \quad (7)$$

$\text{Att}_{cross}$  and  $\text{Att}_{self}$  represent the cross-attention and self-attention block, respectively. The output of the transformer  $F_q^{final}$  is resized and passed to the classification head to obtain the final pixel-by-pixel segmentation result, as shown in the following equation.

$$Y_q = \text{CLS}(F_q^{final} + \text{ReLU}(\text{Conv}(\text{CAT}(F_q^{final}))). \quad (8)$$

$Y^q$  represents the predicted result of the query input and  $\text{CLS}$  refers to the segmentation head.

The loss of RiFeNet is calculated between predictions of query inputs and their corresponding ground truth. We choose dice loss as the loss function, thus the loss for each meta-learning task can be represented as:

$$\mathcal{L}_{main} = \text{DICE}(\hat{Y}^q, Y^q) + \mathcal{L}_{aux}. \quad (9)$$

where  $\hat{Y}^q$  and  $Y^q$  are the prediction and ground truth of query images.  $\mathcal{L}_{aux}$  refers to an auxiliary loss attached for feature alignment, following the common setting (?).

$$\mathcal{L}_{final} = \mathcal{L}_{main} + \beta \mathcal{L}_{unlabel}. \quad (10)$$

As the above equation shows, the final loss is a weighted sum of the main loss and the self-supervised loss  $\mathcal{L}_{unlabel}$ . The weight  $\beta$  is set to 0.5 empirically.

## 4 Experiments

We conduct experiments on the *PASCAL – 5<sup>i</sup>* and *COCO* datasets. The primary evaluation metric is mean intersection-over-union (mIoU). The foreground-background IoU (FB-IoU) is used as an auxiliary indicator.

### 4.1 Process of training and testing

The number of support: query: unlabeled images is 1 : 1 : 2 or 5 : 1 : 2 for each meta-training task in the one-shot and five-shot settings. The unlabeled input branch comes from the same training dataset. We resample  $M$  extra images of the same class without mask truth, rather than adding another dataset. For a fair comparison, we exclude the unlabeled images during testing.

Pre-trained ResNet50 and ResNet101 (?) are used as the backbone of the model, frozen without parameter update. The baseline method refers to the CyCTR. Local query prototypes are generated by the multi-level prototype generation. Concatenated with one global support prototype,  $m^2$  local query prototype, and the prior mask, query features are sent to the transformer after a merging and flattening operation. We set  $m = 4$  in the main experiments. All data in the following tables are the average results of five experiments with the same settings. Specific implementation details are presented in the Appendix.

### 4.2 Comparison with State-of-the-Arts

We compare the performance of RiFeNet with other classical or effective methods on *PASCAL – 5<sup>i</sup>*, as is shown in Tab.1 and Tab. A of the Appendix. RiFeNet outperforms the best method under most of the experimental scenarios. RiFeNet outperforms CyCTR by about 3.5% under the 1-shot setting and about 2% for the 5-shot one. Compared with the existing state-of-the-art DCAMA in the 1-shot setting with ResNet50 backbone, it surpasses by 2.5%, rising to 2.7% with the use of ResNet101. As for the larger gain in the 1-shot than in the 5-shot setting, we attribute this to the decreasing impact weight due to the constant amount of unlabeled data, as the ratio of unlabeled to labeled images decreases from 2 to 0.4. As labeled images increase,

Method	Backbone	1-shot					5-shot					#learnable params
		split0	split1	split2	split3	mean	split0	split1	split2	split3	mean	
PPNet ?	Res-50	52.7	62.8	57.4	47.7	55.2	60.3	70.0	69.4	60.7	65.1	31.5M
PMM ?		52.0	67.5	51.5	49.8	55.2	55.0	68.2	52.9	51.1	56.8	-
PFENet ?		61.7	69.5	55.4	56.3	60.8	63.1	70.7	55.8	57.9	61.9	10.8M
CyCTR ?		65.7	71.0	59.5	59.7	64.0	69.3	73.5	63.8	63.5	67.5	7.4M
HSNet ?		64.3	70.7	60.3	<b>60.5</b>	64.0	<u>70.3</u>	73.2	67.4	<b>67.1</b>	<u>69.5</u>	2.6M
ASGNet ?		58.8	67.9	56.8	53.7	59.3	63.7	70.6	64.1	57.4	63.9	10.4M
SSP ?		61.4	67.2	<u>65.4</u>	49.7	60.9	68.0	72.0	<b>74.8</b>	60.2	68.8	-
DCAMA ?		<u>67.5</u>	72.3	59.6	59.0	<u>64.6</u>	<b>70.5</b>	<u>73.9</u>	63.7	<u>65.8</u>	68.5	-
<b>RiFeNet (Ours)</b>		<b>68.4</b>	<b>73.5</b>	<b>67.1</b>	59.4	<b>67.1</b>	70.0	<b>74.7</b>	<u>69.4</u>	64.2	<b>69.6</b>	7.7M
DAN ?	Res-101	54.7	68.6	57.8	51.6	58.2	57.9	69.0	60.1	54.9	60.5	-
PMM ?		54.7	68.6	57.8	51.6	58.2	57.9	69.0	60.1	54.9	60.5	-
PFENet ?		60.5	69.4	54.4	55.9	60.1	62.8	70.4	54.9	57.6	61.4	10.8M
CyCTR ?		67.2	71.1	57.6	59.0	63.7	<u>71.0</u>	<u>75.0</u>	58.5	65.0	67.4	7.4M
HSNet ?		<u>67.3</u>	<u>72.3</u>	62.0	<b>63.1</b>	<u>66.2</u>	<b>71.8</b>	74.4	67.0	<b>68.3</b>	<u>70.4</u>	2.6M
ASGNet ?		59.8	67.4	55.6	54.4	59.3	64.6	71.3	64.2	57.3	64.4	10.4M
SSP ?		63.7	70.1	<b>66.7</b>	55.4	64.0	70.3	<b>76.3</b>	<b>77.8</b>	<u>65.5</u>	<b>72.5</b>	-
DCAMA ?		65.4	71.4	63.2	58.3	64.6	70.7	73.7	66.8	61.9	68.3	-
<b>RiFeNet (Ours)</b>		<b>68.9</b>	<b>73.8</b>	<u>66.2</u>	<u>60.3</u>	<b>67.3</b>	70.4	74.5	<u>68.3</u>	63.4	69.2	7.7M

Table 1: Performance comparison on PASCAL-5<sup>i</sup> in terms of mIoU (%).

Method	Backbone	1-shot					5-shot					mean
		split0	split1	split2	split3	mean	split0	split1	split2	split3	mean	
PPNet ?	Res-50	28.1	30.8	29.5	27.7	29.0	39.0	40.8	37.1	37.3	38.5	
PMM ?		29.3	34.8	27.1	27.3	29.6	33.0	40.6	30.3	33.3	34.3	
RPMMs ?		29.5	36.8	28.9	27.0	30.6	33.8	42.0	33.0	33.3	35.5	
CyCTR ?		38.9	43.0	39.6	39.8	40.3	41.1	48.9	45.2	<u>47.0</u>	45.6	
HSNet ?		36.3	43.1	38.7	38.7	39.2	43.3	<u>51.3</u>	48.2	<u>45.0</u>	46.9	
SSP ?		<b>46.4</b>	35.2	27.3	25.4	33.6	<b>53.8</b>	41.5	36.0	33.7	41.3	
DCAMA ?		<u>41.9</u>	<u>45.1</u>	<u>44.4</u>	<u>41.7</u>	<u>43.3</u>	45.9	50.5	<b>50.7</b>	46.0	<u>48.3</u>	
<b>RiFeNet (Ours)</b>		39.1	<b>47.2</b>	<b>44.6</b>	<b>45.4</b>	<b>44.1</b>	44.3	<b>52.4</b>	49.3	<b>48.4</b>	<b>48.6</b>	

Table 2: Performance comparison on COCO in terms of mIoU (%).

the positive effects of the unlabeled branch decrease, with a proportional decline in performance gain in 5-shot.

Similar experiments on COCO support the above conclusion in Tab.2. Faced with a scenario with multiple objects in this dataset and a complex environment, RiFeNet still outperforms the current best DCAMA by 0.8% for almost all splits in the 1-shot setting. The comparison results demonstrate the benefits of RiFeNet. The unlabeled branch provides RiFeNet with richer relevant information, which in turn improves the performance of the model.

Qualitative results also prove the effectiveness of RiFeNet. In Fig.5 and Fig.A of the Appendix, the foreground objects in support and query images vary a lot, with inconsistent postures, appearances, and angles of photography. Despite this large intra-class variability, RiFeNet achieves significant improvement in maintaining foreground semantic consistency compared with the baseline. As for the similarity of background and foreground, the model deals with this binary identification much better even in cases with neighboring objects of similar appearance, with foreground occlusion, and with multiple classes of objects. Looking back to Fig.1, our extracted features are essential to maintain foreground semantic consistency and provide inter-class

distinction for binary classification.

### 4.3 Ablation Studies

**RiFeNet improves the pixel-level binary classification.** To demonstrate the effectiveness of our proposed unlabeled enhancement and multi-level prototypes in RiFeNet, we conduct diagnostic experiments in Tab.3. All comparisons are set under a 1-shot setting, with ResNet50 as the backbone. Using either the unlabeled branch or multi-level prototype interaction results in a boost of approximately 2%. When two strategies work together, RiFeNet improves by 3.1% on top of the baseline.

**Different design choices of multi-level prototypes.** We conduct ablation experiments on the model design details mentioned for the multi-level prototype, as is shown in Tab.4. Consistent with the theoretical analysis in Sec.3.4, it proves that our practices such as adding guidance to unlabeled branches are reasonable and reliable.

**Different design choices of unlabeled branches.** We conduct experiments with different designed unlabeled branches to further explore their effect. As shown in Tab.5, the unlabeled branch without guided query prototypes results in even worse performance than the baseline, which

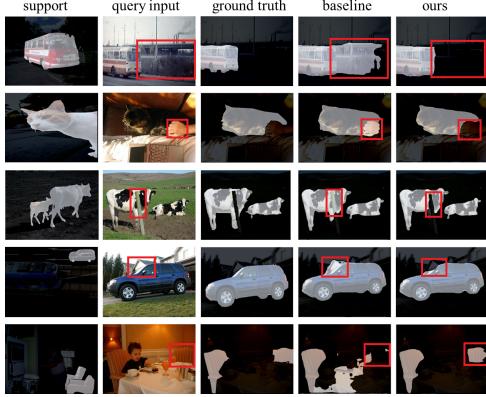


Figure 5: Qualitative segmentation results on novel classes on PASCAL-5<sup>i</sup>. From left to right: support image with mask, query input, query ground-truth mask, query prediction of the baseline, and prediction of RiFeNet.

Un	MP	split0	split1	split2	split3	mIoU
✓		65.7	71.0	59.5	59.7	64.0
		67.3	71.8	66.2	59.2	66.1
	✓	66.0	72.1	66.2	<b>60.4</b>	66.2
✓	✓	<b>68.4</b>	<b>73.5</b>	<b>67.1</b>	59.4	<b>67.1</b>

Table 3: Ablation studies on the key components of RiFeNet. “Un” and “MP” denote the use of the unlabeled branch and the multi-level prototypes, respectively.

components	split0	split1	split2	split3	mIoU
gp (support-only)	67.3	71.8	66.2	59.2	66.1
gp+gp	67.5	73.1	66.2	58.4	66.3
gp+lp (w/o CA)	68.1	73.2	66.7	59.1	66.8
gp+lp (w/ CA)	<b>68.4</b>	<b>73.5</b>	<b>67.1</b>	<b>59.4</b>	<b>67.1</b>

Table 4: Ablation studies on multi-level prototypes. “gp” and “lp” denote global and local prototypes, respectively. That is, “gp+gp” means extracting both query and support prototypes globally. “CA” refers to channel-wise attention.

is consistent with our analysis in Sec.3.3. On the other hand, because the unlabeled inputs come from resampling the training dataset, we double the training iterations of the baseline for a fair comparison. Increased training iterations have little effect on the baseline due to early convergence. This proves that the effectiveness of our method is not from the multiple sampling of data but from the learned discriminative and semantic features.

**Different hyper-parameters.** We first look into the effect of different numbers of unlabeled input in a single meta-training process. Tab.6 shows the results on *PASCAL* – 5<sup>i</sup> under a 1-shot setting, with ResNet50 as its backbone. The best results are obtained when the number of unlabeled images is set to 2. Initially, the segmentation effect of the model increased as the number of unlabeled images increased. When the number continues to increase, the accuracy de-

components	epoch	split0	split1	split2	split3	mIoU
w/o unlabel	200	66.0	72.1	66.2	<b>60.4</b>	66.2
w/o unlabel	400	66.5	72.4	65.5	59.5	66.0
un (w/o guide)	200	66.9	72.2	65.9	58.3	65.8
un (w/ guide)	200	<b>68.4</b>	<b>73.5</b>	<b>67.1</b>	59.4	<b>67.1</b>

Table 5: Ablation studies on the unlabeled branch. “w/ guide” refers to the use of query local prototypes in the unlabeled branch for guidance, while “w/o guide” means using prototypes generated from the unlabeled branch itself.

num	split0	split1	split2	split3	mIoU
0	66.0	72.1	66.2	60.4	66.2
1	66.8	72.8	66.9	59.8	66.6
2	<b>68.4</b>	<b>73.5</b>	<b>67.1</b>	59.4	<b>67.1</b>
3	65.9	72.6	66.9	<b>59.8</b>	66.0

Table 6: Ablation studies of different numbers of unlabeled images in the single meta-training process.

creases instead. We deem the reason is that when the effect of unlabeled enhancement counts much more than the query branch itself, the attention of feature mining may turn to the unlabeled branch, thus disturbing the query prediction. The segmentation accuracy decreases after the features are blurred. We also conduct detailed ablation experiments with other parameters, which are included in the Appendix.

## 5 Conclusion

In few-shot segmentation, traditional methods suffer from semantic ambiguity and inter-class similarity. Thus from the perspective of pixel-level binary classification, we propose RiFeNet, an effective model with an unlabeled branch constraining foreground semantic consistency. Without extra data, this unlabeled branch improves the in-class generalization of the foreground. Moreover, we propose to further enhance the discrimination of background and foreground by a multi-level prototype generation and interaction module.

Rerum doloribus voluptatem, quibusdam aperiam molitiae culpa? Maiores ipsa quasi, hic aut laborum neque dicta maiores vel? Sequi corporis nulla placeat odit temporibus at reiciendis perspiciatis, molestiae sapiente ad natus molitiae consectetur enim voluptatum tenetur fugiat deserunt maxime, obcaecati error voluptatibus esse, dolor adipisci eligendi, blanditiis assumenda qui veniam? Possimus odit similiique cupiditate, aperiam voluptas iusto eligendi, error id fugiat in odit quia reiciendis, laudantium eaque atque quia fuga odit quam optio itaque iure facere dicta? Ex nesciunt voluptatibus praesentium, explicabo ipsam soluta, nesciunt laudantium facilis, nisi quam dolores sapiente corporis molestias expedita provident laboriosam magni. Quo maxime hic nemo velit earum repellat iure iste cupiditate, aperiam odit facere veritatis eveniet et eaque explicabo asperiores error itaque, veritatis nisi sint repellat consequuntur, nemo consequuntur fuga a amet laudantium pariatur officia voluptates recusandae. Aspernatur ea accusantium vel eum modi, voluptas sapiente magni?