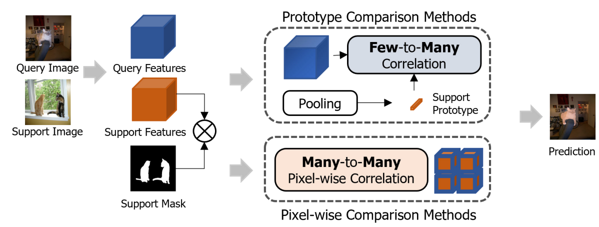
PRFormer: Matching Proposal and Reference Masks by Semantic and Spatial Similarity for Few-Shot Semantic Segmentation

Guangyu Gao *Member, IEEE*, Anqi Zhang, Jianbo Jiao, *Member, IEEE*, Chi Harold Liu, *Senior Member, IEEE*, and Yunchao Wei *Member, IEEE*.

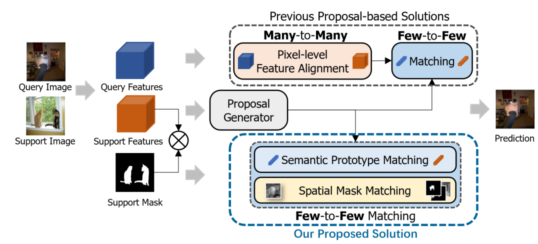
***Abstract*—Few-shot Semantic Segmentation (FSS) aims to accurately segment query images with guidance from only a few annotated support images. Previous methods typically rely on pixel-level feature correlations, denoted as the many-to-many (*pixels-to-pixels*) or few-to-many (*prototype-to-pixels*) manners. Recent mask proposals classification pipeline in semantic segmentation enables more efficient few-to-few (*prototype-to-prototype*) correlation between masks of query proposals and support reference. However, these methods still involve intermediate pixel-level feature correlation, resulting in lower efficiency. In this paper, we introduce the *Proposal and Reference masks matching transFormer (PRFormer)*, designed to rigorously address mask matching in both spatial and semantic aspects in a thorough few-to-few manner. Following the mask-classification paradigm, PRFormer starts with a class-agnostic proposal generator to partition the query image into proposal masks. It then evaluates the features corresponding to query proposal masks and support reference masks using two strategies: semantic matching based on feature similarity across prototypes and spatial matching through mask intersection ratio. These strategies are implemented as the *Prototype Contrastive Correlation (PrCC)* and *Prior-Proposals Intersection (PPI)* modules, respectively. These strategies enhance matching precision and efficiency while eliminating dependence on pixel-level feature correlations. Additionally, we propose the category discrimination NCE (cdNCE) loss and IoU-KLD loss to constrain the adapted prototypes and align the similarity vector with the corresponding IoU between proposals and ground truth. Given that class-agnostic proposals tend to be more accurate for training classes than for novel classes in FSS, we introduce the *Weighted Proposal Refinement (WPR)* to refine the most confident masks with detailed features, yielding more precise predictions. Experiments on the popular Pascal-5^i and COCO-20^i benchmarks show that our Few-to-Few approach, PRFormer, outperforms previous methods, achieving mIoU scores of 70.4% and 49.4%, respectively, on 1-shot segmentation. Code is available at https://github.com/ANDYZAQ/PRFormer.**

***Index Terms*—few-shot learning, semantic segmentation, mask matching, proposal masks, and contrastive learning.**

Guangyu Gao, Anqi Zhang and Chi Harold Liu are with School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: guangyugao@bit.edu.cn, andy_zaq@outlook.com, chiliu@bit.edu.cn). Jianbo Jiao is with the School of Computer Science, University of Birmingham, B15 2TT Birmingham, U.K. (e-mail: j.jiao@bham.ac.uk). Yunchao Wei is with the School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China (email: yunchao.wei@bjtu.edu.cn). This work was funded by the National Natural Science Foundation of China (No. 62472033 and U23A20314).



(a) Pixel-level correlation pipeline (*many-to-many* & *few-to-many*).



(b) Proposal-based matching pipeline. (*few-to-few*)

Fig. 1. Comparison of various FSS Pipelines. (a) shows pixel-level correlation pipeline, showcasing both the *many-to-many* manner with dense pixel comparisons and the *few-to-many* manner with prototype-to-pixel comparisons. (b) presents the proposal-based pipeline, where proposals are pooled as prototypes for comparisons. The upper part of (b) depicts the *few-to-few* manner, but with pixel-wise alignment attached in a *many-to-many* fashion. In contrast, the lower part highlights our *thorough few-to-few PRFormer*, which effectively eliminates dependence on pixel-level correlations.

I. Introduction

Unlike traditional semantic segmentation, which is resource-intensive and time-consuming, Few-shot Semantic Segmentation (FSS) utilizes only a few annotated support images for class-agnostic segmentation of novel categories, as first introduced in OSLSM [1]. FSS methods primarily use a pixel prediction paradigm, focusing on feature extraction and pixel-level similarity assessment between query and support images, as shown in Fig. 1a. These methods [2], [3] primarily emphasize exploring pixel-level feature correlations to enhance similarity assessment. Some approaches [4], [5] condense the annotated support features into semantic-level prototypes, correlating them with query pixels in a few-to-many manner, whereas others [6], [7] employ complete many-to-many pixel-wise correlations between query and support features. Attracted by the superficial benefits of many-to-many or few-to-many dense correlation, recent works have focused on *exploring stronger pixel-level feature correlations* in a dense matching manner. However, most of these methods akin to '*robbing Peter to pay Paul*', typically demand substantial training time and resources for robust model development and are more prone to overfitting specific datasets.

Recently, mask proposals, originating in Object Detection, have been adapted for semantic segmentation [8], [9]. Building on MaskFormer [8], various subfields of semantic segmentation, *e.g.*, Open Vocabulary Semantic Segmentation [10], [11], have gained traction. MMFormer [12] advances MaskFormer's concept by presenting a two-stage FSS framework in a few-to-few way, generating mask proposals for query images, and then performing similarity assessment between mask prototypes from query proposals and support reference. However, influenced by most methods' continuous pursuit of performance gains through dense pixel-level matching, MMFormer has not escaped the superficial benefits of pixel-level feature correlation. Fig. 1b shows that its Mask Matching module unintentionally includes a *Feature Alignment Block* to align query and support pixel features in a many-to-many manner, rendering MMFormer a partially few-to-few approach.

To address mask matching with pooled prototypes through a real few-to-few manner, we propose a mask-classification-based approach, *i.e.*, Prototype and Reference masks matching transFormer (PRFormer). In PRFormer, with the class-agnostic proposals, the few-to-few matching is realized in a dual strategy of semantic matching across prototypes, and spatial matching over masks, for more precise similarity assessment. Specifically, a simple yet effective pure prototype-based multi-scale matching module, *i.e.*, *Prototype Contrastive Correlation (PrCC)*, is proposed for similarity assessment in the semantic view. Besides, when describing the similarity between two masks, the extent of overlap in their spatial distribution is the same important as the similarity between the masked features. Therefore, the *Prior-Proposals Intersection (PPI)* module is designed to measure spatial similarity with the ratio of spatial overlap in the proposal and reference masks.

Additionally, the success of proposal-based methods heavily depends on the proposals' quality, yet proposal generators in few-shot scenarios often favor base classes, leading to less accurate proposals for novel classes. To address this, we introduce the *Weighted Proposal Refinement (WPR)*, to meticulously refine the most reliable proposal masks with detailed features for better prediction. We further design the *category discrimination NCE (cdNCE)* loss for PrCC that buffers and updates support prototypes for contrastive learning with the current adapted query prototypes. We also introduce the *IoU Kullback-Leibler Divergence (IoU-KLD)* loss to align the similarity vector close to binary IoU between query proposal masks and ground truth reference masks.

In summary, our contributions are as follows:

- We introduce PRFormer, a few-to-few approach that improves mask similarity assessment in semantic and spatial aspects via Prototype Contrastive Correlation (PrCC) and Prior-Proposals Intersection (PPI) modules.
- To address the tendency of proposal-based FSS methods to favor base classes and produce inaccurate proposals for novel classes, we introduce the Weighted Proposal Refinement (WPR) for refining reliable masks with detailed features, complemented by two specific losses to boost prediction accuracy.
- Extensive evaluations on the Pascal-5^i and COCO-20^i datasets show that our PRFormer achieves state-of-the-art performance with high efficiency.

II. Related Work

A. Semantic Segmentation

Fully Convolutional Networks [13] accelerate the advancement of semantic segmentation. Following that, various techniques have emerged to further enhance semantic segmentation, including encoder-decoder structures [14], dilated convolution [15]–[17], pyramid pooling operation [18], attention mechanism [19], and Transformer modules [20], among others. Recently, inspired by the proposal generation mechanism used in object detection [21], MaskFormer/Mask2Former [8], [9] introduced a two-stage segmentation pipeline, involving proposal generation and classification. The class-agnostic proposal generation process provides accurate mask proposals from the original image, simulating the logic of recognizing objects of humans and revolutionizing the field of semantic segmentation. The recent segmentation foundation model, Segment Anything [22], builds on the concept of Mask2Former by generating high-quality class-agnostic masks from various prompts such as points, boxes, and coarse masks. However, despite these advancements, semantic segmentation methods still struggle with generalizing to novel categories, primarily due to the necessity of obtaining new annotations and retraining models, a labor-intensive endeavor.

B. Few-shot Semantic Segmentation

Few-shot Semantic Segmentation (FSS) infers the pixel-level prediction of novel categories with a few annotated samples. Previous methods relying on pixel-level feature comparison are mainly divided into two groups: prototype comparison methods [23]–[29] and pixel-wise comparison methods [6], [7], [30], [31]. Prototype comparison methods, inherited from the few-shot learning [32], use semantic prototypes to facilitate interactions between the query and support samples. These methods employ Masked Average Pooling [5] to aggregate support features based on their corresponding masks, creating prototypes. Query features are then compared to these prototypes using cosine similarity or learnable convolutional operations. While support prototypes capture the global features of the support object, they may overlook internal variations. In contrast, pixel-wise comparison methods, largely embodied in HSNet [6], focus on intra-class differences using operations like the 4D Hypercorrelation operation for more detailed excavation. Subsequent methods [7], [33] have further introduced transformer-based structures to enhance 4D pixel-wise comparisons. Recent approaches [34]–[37] combine both prototype and pixel-wise methods for a more comprehensive feature comparison. Additionally, the proposal-based structure [12] has emerged as a novel pipeline in FSS, which contains a proposal generator for generating a bunch of class-agnostic masks, followed by a few-to-few

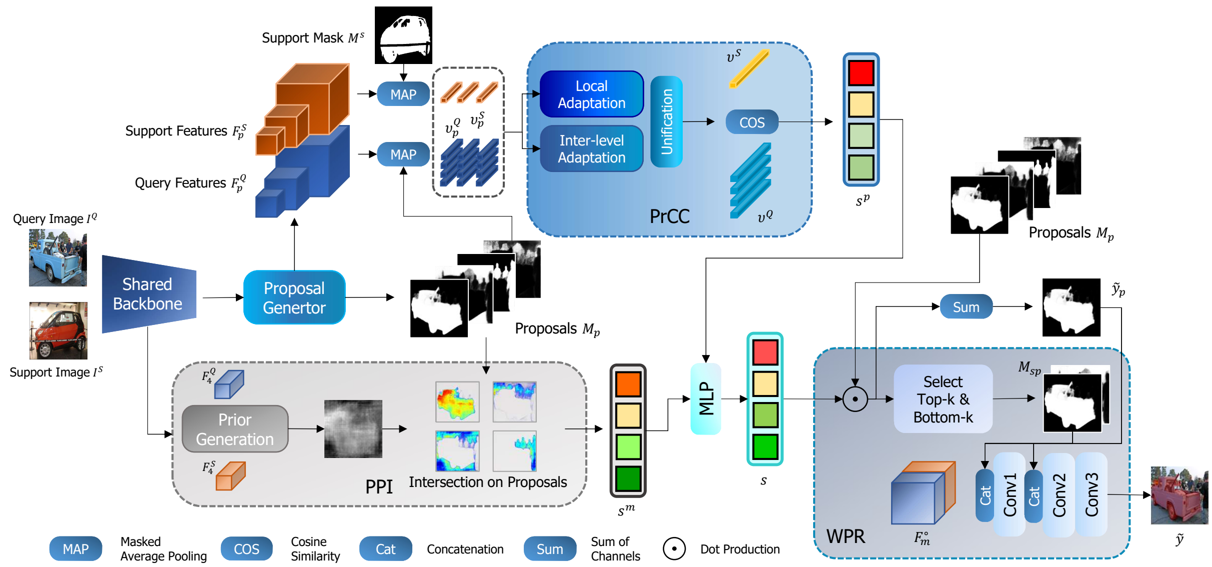


Fig. 2. Overall framework of the proposed PRFormer. PRFormer primarily comprises a ResNet-based backbone and a proposal generator for feature extraction, the Prototype Contrastive Correlation (PrCC) and Prior-Proposals Intersection (PPI) modules for mask similarity assessment, alongside the Weighted Proposal Refinement (WPR) module for prediction refinement. Conv1 and Conv3 denote 1×1 Convolution blocks, while Conv2 refers to 3×3 Convolution block.

prototype comparisons for proposal selection. However, the proposal selection process still relies on pixel-level dense feature alignment, thus not fully embodying a pure few-to-few method. Considering these developments, our approach employs a proposal-based structure with a duplex prototype and mask matching stream, merging semantic and spatial similarity for thoroughly few-to-few proposal selection.

III. Preliminaries

The Few-shot Semantic Segmentation (FSS) task aims to enable segmentation with guidance from a few annotated samples. In the standard FSS task, to evaluate the generalization ability of meta-learning approaches, datasets are divided into the training and test sets, denoted as $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$, with disjoint categories. The categories are correspondingly divided into two groups: training classes $\boldsymbol{C}_{train}$ and testing classes $\boldsymbol{C}_{test}$, aligning with $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$, respectively. Each set comprises episodes containing a query set $\boldsymbol{Q}$ and a support set $\boldsymbol{S}$. The query set $\mathcal{Q} = \{(\boldsymbol{I}^Q, \boldsymbol{M}^Q)\}$ includes a query image $\boldsymbol{I}^Q$ and its corresponding ground truth segmentation mask $\boldsymbol{M}^Q$. The support set $\mathcal{S} = \{(\boldsymbol{I}_i^S, \boldsymbol{M}_i^S)\}_{i=1}^K$ contains K pairs of the support image $\boldsymbol{I}_i^S$ and its mask $\boldsymbol{M}_i^S$. Importantly, the query set $\mathcal{Q}$ and the support set $\mathcal{S}$ belong to the same category. During each training iteration, a group of query set Q and support set S belonging to $\mathcal{D}_{train}$ is applied. The support images $\boldsymbol{I}_i^S$, accompanied with their corresponding support masks $\boldsymbol{M}_i^S$, provide the reference for the target category. Guided by support set $\mathcal{S}$, the learnable parameters of the model are optimized through the loss between predictions for query images $\boldsymbol{I}^Q$ and the ground truth $\boldsymbol{M}^Q$. After the training episodes, the model's performance is assessed on $\mathcal{D}_{test}$. The inference of a query image $\boldsymbol{I}^Q$ is conducted with the reference of a support set S containing the object of the same category, following the training process. The predictions for the query image $\boldsymbol{I}^Q$ are evaluated across all testing episodes. The testing samples are selected from the categories that do not exist in the $\boldsymbol{C}_{train}$, which ensures the evaluation result is not influenced by overfitting on $\boldsymbol{C}_{train}$.

IV. Approach

Our approach, *Proposal and Reference masks matching transFormer (PRFormer)*, is mainly composed of the ResNet backbone, the proposal generator, as well as the mask similarity assessment between masks of query proposals and support reference that combines the Prototype Contrastive Correlation (PrCC), Prior-Proposals Intersection (PPI), and the Weighted Proposal Refinement (WPR) modules, as shown in Fig. 2. The proposal generator is built upon the architecture of Mask2Former [9] and mainly includes the pixel decoder and transform decoder. Subsequent similarity assessment operations involve pure few-to-few mask matching in both the semantic and spatial views, departing from the dense pixel-wise feature matching used in traditional methods. The PrCC module conducts semantic affinity and compatibility assessment, while the PPI module introduces parameter-free spatial overlap assessment on masks. The WPR module further refines the prediction result with selected confident proposals and pixel-level features.

A. Feature Extraction

We adopt ResNet [38] as the backbone to extract features for input support and query images, denoted as $\boldsymbol{F}^S$ and $\boldsymbol{F}^Q$, respectively. Here, $\boldsymbol{F} = \{\boldsymbol{F}_l\}$, where $l \in \{0, 1, 2, 3, 4\}$ represents the block index in the backbone. While the pixel decoder of the proposal generator takes $\boldsymbol{F}_2$, $\boldsymbol{F}_3$, and $\boldsymbol{F}_4$ as input, it produces three semantically enriched multi-scale feature maps $\boldsymbol{F}_{p2}$, $\boldsymbol{F}_{p3}$, and $\boldsymbol{F}_{p4}$, which are further used by PrCC. Meanwhile, the transformer decoder of the proposal generator partitions the query image into N proposals, which are represented as masks $\boldsymbol{M}_q = \{\boldsymbol{M}_q^n\}_{n=1}^N \in [0, 1]^{N \times H \times W}$.

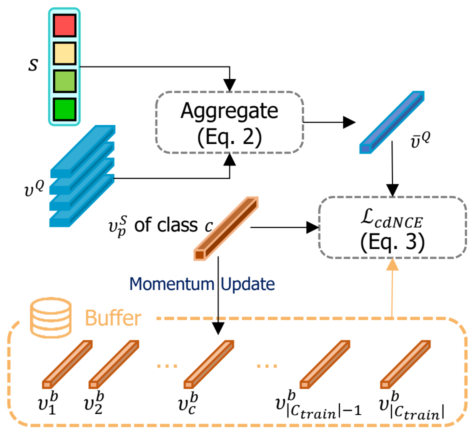


Fig. 3. Illustration of $\mathcal{L}_{cdNCE}$. The query prototypes from the proposals are weighted and aggregated according to the similarity vector, as shown in Eq. 2, then $\mathcal{L}_{cdNCE}$ is computed following Eq. 3. The corresponding buffer prototype is updated with momentum by the current support prototype.

B. Matching Process

1) Prototype Contrastive Correlation (PrCC): The PrCC module is designed to facilitate matching between semantic-level prototype features in multi-scale, thereby eliminating the need for dense pixel-level matching. Previous methods like MMFormer [12] utilize backbone-derived features $\boldsymbol{F}_2^\circ$, $\boldsymbol{F}_3^\circ$, and $\boldsymbol{F}_4^\circ$ for matching, where the placeholder '$\circ$' denotes either the support (S) or query (Q) image. The PrCC module, however, leverages the features from the proposal generator for matching, *i.e.*, $\boldsymbol{F}_{p2}^\circ$, $\boldsymbol{F}_{p3}^\circ$, and $\boldsymbol{F}_{p4}^\circ$. It not only harnesses richer semantic features from the proposal generator but also markedly enhances efficiency with fewer feature dimensions, from 512, 1024, and 2048 channels of the backbone to 256 channels of the proposal generator. In detail, with the query proposals $\boldsymbol{M}_q \in \mathbb{R}^{N\times H\times W}$ and the support reference mask $\boldsymbol{M}^S \in \mathbb{R}^{H\times W}$, we respectively apply the Masked Average Pooling [5] on the support and query features from proposal generator for corresponding support prototype vector $\boldsymbol{v}_i^S \in \mathbb{R}^{1\times C}$, and N query prototype vectors $\boldsymbol{v}_i^Q = \{\boldsymbol{v}_{i,n}^Q\} \in \mathbb{R}^{N\times C}$, where $i \in \{p2, p3, p4\}$.

Then, we design a simple yet efficient adaptation structure to regulate these multi-scale prototypes with the Multi-Layer Perceptron (MLP). On one side, we separately adapt each prototype for *local adaptation* and combine them via concatenation, *i.e.*, $\widehat{\boldsymbol{v}}^\circ = [MLP(v_{p2}^\circ), MLP(v_{p3}^\circ), MLP(v_{p4}^\circ)]$, where $[\cdot]$ means concatenation. On the other side, these prototypes are first concatenated and then regulated by *inter-level adaptation*, *i.e.*, $\check{\boldsymbol{v}}^\circ = MLP([\boldsymbol{v}_{p2}^\circ, \boldsymbol{v}_{p3}^\circ, \boldsymbol{v}_{p4}^\circ])$. After that, a Linear layer unifies $\widehat{\boldsymbol{v}}^\circ$ and $\check{\boldsymbol{v}}^\circ$ as $\boldsymbol{v}^\circ \in \mathbb{R}^{N\times C}$. Transitioning the placeholder $\circ \in \{S, Q\}$, we get $\boldsymbol{v}^S$ and $\boldsymbol{v}^Q = \{\boldsymbol{v}^Q(n)\}_{n=1}^N$ for the adapted support prototype and N adapted query prototypes respectively. We measure the cosine similarity between N query prototypes and the support prototype as $\boldsymbol{s}_1 = \{s_1(n)\}_{n=1}^N \in \mathbb{R}^{N\times 1}$ for prototype based semantic matching, where

$$s_1(n) = \frac{\boldsymbol{v}^Q(n)(\boldsymbol{v}^S)^T}{\|\boldsymbol{v}^Q(n)\|\|\boldsymbol{v}^S\|}. \tag{1}$$

Current FSS methods have to prevent overfitting to the data of training classes $\boldsymbol{C}_{train}$, since the test classes $\boldsymbol{C}_{test}$ do not exist in $\boldsymbol{C}_{train}$, so as our PRFormer. The PrCC module, while efficient in matching, is prone to overfitting to the data of $\boldsymbol{C}_{train}$ due to its high-level simplification of features into prototypes. Thus, restricting the adaptation with some specific loss is warranted, especially ensuring that similar category prototypes are closer while different category prototypes are farther apart. In response, we introduce the category discrimination NCE (*cdNCE*) loss $\mathcal{L}_{cdNCE}$ to constrain the adaptation in PrCC. Specifically, we registered a buffer $\boldsymbol{v}^{\text{buf}} \in \mathbb{R}^{r\times C}$ for storing support prototypes of r seen categories during training. Whenever the prototypes are adapted, the query prototypes $\boldsymbol{v}^Q$ are weighted and aggregated as a single average prototype $\bar{\boldsymbol{v}}^Q$ by the similarity vector $\boldsymbol{s}_1$:

$$\bar{\boldsymbol{v}}^Q = \frac{1}{N}\sum_{n=1}^{N} s_1(n)\cdot \boldsymbol{v}^Q(n). \tag{2}$$

With dot production-based similarity, we formulate *cdNCE* loss based on the registered buffer and average prototype as:

$$\mathcal{L}_{cdNCE} = -log\frac{exp(\bar{\boldsymbol{v}}^Q\cdot \boldsymbol{v}^S)}{\sum_{i=0}^{|C_{train}|} exp(\bar{\boldsymbol{v}}^Q\cdot \boldsymbol{v}_i^{\text{buf}})}, \tag{3}$$

where the support prototype $\boldsymbol{v}^S$ serves as the positive sample, whereas the buffered prototypes of other categories act as negative ones. Meanwhile, the buffer is updated by the current support prototype $\boldsymbol{v}^S$ in a momentum way, so that the buffer prototypes can continually represent the feature of C_{train}:

$$\boldsymbol{v}_i^b = (1-\alpha)\cdot \boldsymbol{v}_i^{\text{buf}} + \alpha\cdot \boldsymbol{v}^S, \tag{4}$$

where α represents the update momentum. The whole process is illustrated in Fig. 3.

2) Prior-Proposal Intersection (PPI): In the PrCC module, prototype matching efficiently captures semantic features yet lacks spatial information, which is another crucial factor for segmentation. To address this issue while maintaining an efficient few-to-few approach, we further introduce the parameter-free Prior-Proposal Intersection (PPI) module. Given that intra-class variations are prevalent within the same object or category, our PPI module evaluates the spatial correlation between two types of pseudo masks for the query image. One of the pseudo masks is the query proposal masks $\boldsymbol{M}_p$ mentioned in Sec. IV-A. The other pseudo mask is the prior mask $\widetilde{\boldsymbol{M}}_p$, generated by utilizing features from the backbone, including query features $\boldsymbol{F}_4^Q$, support features $\boldsymbol{F}_4^S$, and support mask $\boldsymbol{M}^S$. These fine-grained semantic features are then converted into a prior mask $\widetilde{\boldsymbol{M}}_p \in \mathbb{R}^{H\times W}$, widely used in previous FSS approaches [23], [39], [40]:

$$\widetilde{\boldsymbol{M}}_p(i,j) = \max_{t\in\{1,2,\ldots,HW\}}\left(\frac{\mathcal{I}(\boldsymbol{F}_4^Q(d))^T\mathcal{I}(\boldsymbol{F}_4^{S^+}(t))}{\|\mathcal{I}(\boldsymbol{F}_4^Q(d))\|\|\mathcal{I}(\boldsymbol{F}_4^{S^+}(t))\|}\right), \tag{5}$$

where $\mathcal{I}$ represents flattening the spatial dimensions from $h\times w$ to hw, $\boldsymbol{F}_4^{S^+}(t)$ is the foreground part of $\boldsymbol{F}_4^S(t)$ according to $\boldsymbol{M}^S$, $d = i\times W + j$ and s denote the index of pixel in $\boldsymbol{F}_4^Q$ and $\boldsymbol{F}_4^S$, respectively. This prior mask $\widetilde{\boldsymbol{M}}_p$ serves to provide a rough probability estimate for pixels of the query image belonging to the target class, as the similarity value of a pixel in $\boldsymbol{F}_4^Q$ largely depends on its most similar part in $\boldsymbol{F}_4^S$.

Hitherto, we have two groups of potential prompts for the query image, namely the proposal masks $\boldsymbol{M}_p = \{\boldsymbol{M}_p^n\}_{n=1}^N$ and the prior mask $\widetilde{\boldsymbol{M}}_p$. The former summarizes the internal features from query samples, while the latter emphasizes the category features from the target objects in the support samples. While explaining the resemblance between two masks, the proportion of their overlapping region in the spatial arrangement is a significant measure of similarity, as most semantic segmentation evaluations utilize Intersection over Union (IoU) as the metric. Therefore, we design an efficient mask-matching measurement on these two types of prompts as spatial similarity. For proposal mask $\boldsymbol{M}_p^n$ and the prior mask $\widetilde{M}_p$, we estimate the influence of the high-probability region with the proportion of the intersection area over the proposal area as spatial similarity vector $\boldsymbol{s}_2 = \{s_2(n)\}_{n=1}^N \in \mathbb{R}^{N\times 1}$:

$$s_2(n) = \frac{sum(\boldsymbol{M}_p(n) \odot \widetilde{\boldsymbol{M}}_p)}{sum(\boldsymbol{M}_p(n))}, \tag{6}$$

where $\odot$ represents the Hadamard production, and sum means to sum the value over all pixels. The proportions are ensembled as $\boldsymbol{s}_2 = \{s_2(n)\}_{n=1}^N$, which squeezes the pixel-level mask-matching process into a concise similarity vector to present the level of spatial overlap.

3) Initial Segmentation Prediction: The PrCC module generates a semantic similarity vector $\boldsymbol{s}_1$ across prototypes, while the PPI module produces a spatial similarity vector $\boldsymbol{s}_2$ across proposal masks. These similarity vectors are combined into a unified similarity vector $\boldsymbol{s}$. We concatenate $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ into a $2N$-channel vector, and then use an MLP layer to squeeze them into a unified similarity vector $\boldsymbol{s} = \{s(n)\}_{n=1}^N \in \mathbb{R}^{1\times N}$. This unified vector offers a more precise measure of the similarity between each query proposal and the support mask. Subsequently, the initial segmentation prediction is derived by applying a weighted sum of the proposal masks, where the similarity vector $\boldsymbol{s}$ serves as the weights. The proposal-based initial segmentation prediction is defined as $\tilde{y}_{in} \in \mathbb{R}^{1\times H\times W}$:

$$\tilde{y}_{in} = \sum_{n=1}^{N} s(n)\boldsymbol{M}_p(n). \tag{7}$$

C. Refinement and Optimization

1) Weighted Proposal Refinement (WPR): The precision of proposal-based prediction $\tilde{y}_{in}$ is highly related to the precision of the similarity vector $\boldsymbol{s}$, and the quality of proposal masks $\boldsymbol{M}_p$. However, due to the lack of intersection between training classes $\boldsymbol{C}_{train}$ and novel classes $\boldsymbol{C}_{test}$, the proposal generator tends to accurately segment the training classes, resulting in less accurate proposals for novel classes. Recognizing the synchronized improvement or deterioration of mask-level predictions, we introduce a lightweight post-process module, the Weighted Proposal Refinement (WPR).

The WPR module enhances performance by adjusting the representative similarity-weighted proposals and the predictions using detailed features. We first multiply the similarity vector $\boldsymbol{s}$ with the proposals $\boldsymbol{M}_p$ to obtain the weighted proposals $\boldsymbol{M}_{wp}$. Then, proposals in $\boldsymbol{M}_{wp}$ are sorted based on the similarity values from the similarity vector $\boldsymbol{s}$. However, within the set of N proposals, many may not cover the desired regions, resulting in considerable redundancy. Consequently, after sorting, we keep only two groups of proposals: the top-k most likely to contain the target and the bottom-k least likely to do so, for subsequent prediction. We merge these $2k$ proposal masks as $\boldsymbol{M}_{sp} \in \mathbb{R}^{2k\times H\times W}$ with detailed features, producing more precise predictions. Specifically, we compress the concatenated features of middle-level features $\boldsymbol{F}_2^{\circ}$ and $\boldsymbol{F}_3^{\circ}$ to $\boldsymbol{F}_m^{\circ} \in \mathbb{R}^{C\times H\times W}$ with C channels. The support middle-level features $\boldsymbol{F}_m^S$ are transformed into global support features $\boldsymbol{F}_g^S \in \mathbb{R}^{C\times H\times W}$ via MAP and feature expansion. Leveraging these middle-level features, the proposal-based prediction is then refined for a more robust prediction by

$$\tilde{\boldsymbol{y}} = \mathcal{F}_{refine}(\tilde{y}_{in}, \boldsymbol{F}_m^Q, \boldsymbol{F}_g^S, \boldsymbol{M}_{sp}), \tag{8}$$

where $\mathcal{F}_{refine}$ denotes the lightweight refinement module with a group of 1×1 and 3×3 convolutional blocks.

2) Objective Function: We follow the Mask2Former [9] and MMFormer [12] settings on the Proposal Generator and apply both Binary Cross-Entropy loss $\mathcal{L}_{ce}$ and dice loss $\mathcal{L}_{dice}$. To optimize the predictions $\tilde{\boldsymbol{y}}_{in}$ and $\tilde{\boldsymbol{y}}$, we adopt dice loss [50] with guidance from the ground truth mask $\boldsymbol{M}^Q$ as $\mathcal{L}_p$ and $\mathcal{L}_{fp}$. In the prediction generation, condensation of the proposals and similarities leads to coarse learning for the prediction. To precisely optimize the similarity vector, we introduce a specialized IoU Kullback-Leibler Divergence (IoU-KLD) loss:

$$\mathcal{L}_{IoU-KLD} = \sum_{n=1}^{N}(\boldsymbol{s}^{IoU}(n)\cdot log\frac{\boldsymbol{s}^{IoU}(n)}{\boldsymbol{s}(n)}), \tag{9}$$

where $\boldsymbol{s}^{IoU}$ means the IoU between the ground truth mask $\boldsymbol{M}^Q$ and the proposal masks $\boldsymbol{M}_p$. The IoU-KLD loss $\mathcal{L}_{IoU-KLD}$ seeks to align the similarity scores with the IoU, considering all associated similarities for each proposal. Overall, the loss function can be unified as

$$\begin{aligned}\mathcal{L} =& \lambda_1(\mathcal{L}_{ce} + \mathcal{L}_{dice}) + \lambda_2(\mathcal{L}_p + \mathcal{L}_{fp}) \\ &+ \lambda_3\mathcal{L}_{cdNCE} + \lambda_4\mathcal{L}_{IoU-KLD},\end{aligned} \tag{10}$$

where λ_1, λ_2, λ_3, λ_4 are specified in the experiments.

V. Experiments

A. Datasets and Evaluation Metrics

We evaluate our PRFormer on two benchmark datasets: PASCAL-5^i [1] and COCO-20^i [52]. PASCAL-5^i is an extension of PASCAL VOC 2012 [53], supplemented with additional annotations from SDS [54], encompassing 20 categories. COCO-20^i is derived from COCO [55] with 80 categories. We adopt the cross-validation by dividing the datasets into 4 folds, each containing 5 categories for PASCAL-5^i and 20 for COCO-20^i. We split the PASCAL-5^i following [1], where the categories are divided in sequential order, *i.e.* categories of $\{5\cdot i+1, 5\cdot i+2, \ldots, 5\cdot i+5\}$ belong to the i-th fold. For COCO-20^i, we follow [52] and pick one category out of every three in sequential order for each fold, *i.e.* categories of $\{4\cdot 0+i, 4\cdot 1+i, \ldots, 4\cdot 19+i\}$ belong to the i-th fold. Three of the four folds are used for training, while the remaining fold is randomly sampled into 1000 episodes for evaluation. Consistent with most previous methods, we employ mean Intersection over Union (mIoU) as the evaluation metric.

TABLE I
PERFORMANCE COMPARISONS WITH THE SOTA METHODS FOR 1-SHOT AND 5-SHOT SEGMENTATION ON PASCAL-5^i IN MIOU. THE RESULTS IN **BOLD** REFER TO THE BEST RESULT AMONG ALL METHODS. †: WE EVALUATED MMFORMER WITH RESNET-101 BASED ON ITS OPEN-SOURCED CODE.

Method	1 shot					5 shot				
	$Fold^0$	$Fold^1$	$Fold^2$	$Fold^3$	**Mean**	$Fold^0$	$Fold^1$	$Fold^2$	$Fold^3$	**Mean**
Pixel-level feature correlation methods with ResNet-50										
PANet [ICCV19] [41]	44.0	57.5	50.8	44.0	49.1	55.3	67.2	61.3	53.2	59.3
PGNet [ICCV19] [42]	56.0	66.9	50.6	50.4	56.0	57.7	68.7	52.9	54.6	58.5
PPNet [ECCV20] [43]	48.6	60.6	55.7	46.5	52.8	58.9	68.3	66.8	58.0	63.0
PFENet[TPAMI20] [23]	61.7	69.5	55.4	56.3	60.8	63.1	70.7	55.8	57.9	61.9
RePRI [CVPR21] [44]	59.8	68.3	62.1	48.5	59.7	64.6	71.4	71.1	59.3	66.6
CWT [CVPR21] [2]	56.3	62.0	59.9	47.2	56.4	61.3	68.5	68.5	56.6	63.7
ASGNet [CVPR21] [24]	58.8	67.9	56.8	53.7	59.3	63.7	70.6	64.2	57.4	63.9
HSNet [ICCV21] [6]	64.3	70.7	60.3	60.5	64.0	70.3	73.2	67.4	67.1	69.5
CyCTR [NeurIPS21] [45]	65.7	71.0	59.5	59.7	64.0	69.3	73.5	63.8	63.5	67.5
SSP [ECCV22] [46]	60.5	67.8	66.4	51.0	61.4	68.0	72.0	**74.8**	60.2	68.8
DCAMA [ECCV22] [7]	67.5	72.3	59.6	59.0	64.6	70.5	73.9	63.7	65.8	68.5
VAT [ECCV22] [33]	67.6	72.0	62.3	60.1	65.5	**72.4**	73.6	68.6	65.7	68.5
BAM [CVPR22] [39]	69.0	73.6	**67.6**	61.1	67.8	70.6	75.1	70.8	67.0	70.9
QCLNet [TCSVT23] [30]	65.2	70.3	60.8	61.0	64.3	70.6	73.5	66.7	67.1	69.5
MIANet [CVPR23] [47]	68.5	**75.8**	67.5	63.2	68.7	70.2	**77.4**	70.0	**68.8**	71.6
ABCNet [CVPR23] [48]	68.8	73.4	62.3	59.5	66.0	71.7	74.2	65.4	67.0	69.6
RPMG-FSS [TCSVT23] [31]	64.4	72.6	57.9	58.4	63.3	65.3	72.8	58.4	59.8	64.1
SCCAN [ICCV23] [49]	67.5	72.6	67.2	60.5	67.0	69.9	74.3	70.1	66.9	70.3
DRNet [TCSVT24] [37]	66.1	68.8	61.3	58.2	63.6	69.2	73.9	65.4	65.3	68.5
Proposal-based methods with ResNet-50										
MMFormer [NeurIPS22] [12]	-	-	-	-	63.3	-	-	-	-	64.9
PRFormer [Ours]	**70.2**	75.0	67.3	**65.4**	**69.5**	**72.4**	76.8	70.4	68.3	**71.9**
Pixel-level feature correlation methods with ResNet-101										
PFENet [TPAMI20] [23]	60.5	69.4	54.4	55.9	60.1	62.8	70.4	54.9	57.6	61.4
RePRI [CVPR21] [44]	59.6	68.6	57.8	51.6	58.2	57.9	69.0	60.1	54.9	60.5
HSNet [ICCV21] [6]	67.3	72.3	62.0	63.1	66.2	71.8	74.4	67.0	68.3	70.4
CyCTR [NeurIPS21] [45]	69.3	72.7	56.5	58.6	64.3	73.5	74.0	58.6	60.2	66.6
NTRENet [CVPR22] [40]	65.5	71.8	59.1	58.3	63.7	67.9	73.2	60.1	66.8	67.0
VAT [ECCV22] [33]	70.0	72.5	64.8	64.2	67.9	75.0	75.2	68.4	69.5	72.0
IPMT [NeurIPS22] [34]	71.6	73.5	58.0	61.2	66.1	75.3	76.9	59.6	65.1	69.2
DCAMA [ECCV22] [7]	65.4	71.4	63.2	58.3	64.6	70.7	73.7	66.8	61.9	68.3
QCLNet [TCSVT23] [30]	67.9	72.5	64.3	63.4	67.0	72.5	74.8	68.5	68.9	71.2
RPMG-FSS [TCSVT23] [31]	63.0	73.3	56.8	57.2	62.6	67.1	73.3	59.8	62.7	65.7
SCCAN [ICCV23] [49]	69.1	74.0	66.3	61.6	67.7	71.6	75.2	**69.5**	66.5	70.7
DRNet [TCSVT24] [37]	66.4	70.7	64.9	59.8	65.3	69.3	74.1	66.7	66.5	69.2
Proposal-based methods with ResNet-101										
MMFormer† [NeurIPS22] [12]	70.2	74.6	64.6	61.8	67.8	74.6	76.2	64.8	66.6	70.6
PRFormer [Ours]	**72.0**	**76.3**	**66.6**	**66.9**	**70.4**	**76.4**	**78.0**	67.0	**70.5**	**73.0**

B. Implementation Details

We use ImageNet-pretrained ResNet-50 and ResNet-101 as the backbone networks for feature extraction. The parameters of the backbone are fixed to prevent overfitting and promote efficiency. Training data is augmented through random horizontal flipping and cropping. The input image size is 480×480 for both PASCAL-5^i and COCO-20^i datasets. During training, the batch size is set to 8, and we use the AdamW optimizer [56] with an initial learning rate of 0.0001. The weight decay is set to 0.05. We employ the poly learning rate strategy, with a factor of 0.9 and a constant ending learning rate of 0.00001. Following MMFormer [12], we implement a two-stage training strategy. First, the proposal generator is trained for 30,000 iterations on PASCAL-5^i and 80,000 iterations on COCO-20^i. Then, the entire network, with the proposal generator frozen, is trained for 15,000 iterations on PASCAL-5^i and 30,000 iterations on COCO-20^i. λ_1 is set to 5.0 for the proposal generator in the 1^{st} stage and deprecated in the 2^{nd} stage, while λ_2, λ_3, and λ_4 are set to 10.0, 1.0, and 20.0, respectively, for the matching process in the 2^{nd} stage and deprecated in the 1^{st} stage. Note that both λ_1 and λ_2 are set according to the settings in MMFormer [12]. For K-shot segmentation, we averaged K support prototypes to reduce intra-category differences. The update momentum α for $\mathcal{L}_{cdNCE}$ is set to 0.5. The number of proposals N is set to 100, following the default setting of Mask2Former [9]. The PRFormer is implemented and trained using PyTorch on the NVIDIA RTX 2080Ti.

C. Comparison with State-of-the-Arts

We compare our PRFormer with State-of-the-Art (SOTA) methods on PASCAL-5^i and COCO-20^i datasets, as summarized in Tab. I and Tab. II.

a) PASCAL-5^i.: Tab. I presents the 1-shot and 5-shot performance on PASCAL-5^i. Our PRFormer consistently outperforms other approaches using both ResNet-50 and ResNet-101 backbones. For 1-shot segmentation, PRFormer achieves 69.5% mIoU with ResNet-50 and 70.4% mIoU with ResNet-101, surpassing previous methods by at least 0.8% and 2.5%, respectively. For 5-shot segmentation, PRFormer maintains competitive performance with 71.9% mIoU using ResNet-

TABLE II
PERFORMANCE COMPARISONS WITH LATEST METHODS FOR 1-SHOT AND 5-SHOT SEGMENTATION ON COCO-20^i IN MIOU. THE RESULTS IN **BOLD** REFER TO THE BEST RESULT AMONG ALL THE METHODS. †: WE EVALUATE MMFORMER WITH RESNET-101 BASED ON ITS OPEN-SOURCED CODE.

Method	1 shot					5 shot				
	Fold^0	Fold^1	Fold^2	Fold^3	**Mean**	Fold^0	Fold^1	Fold^2	Fold^3	**Mean**
Pixel-level feature correlation methods with ResNet-50										
PPNet [ECCV20] [43]	28.1	30.8	29.5	27.7	29.0	39.0	40.8	37.1	37.3	38.5
PFENet [TPAMI20] [23]	36.5	38.6	34.5	33.8	35.8	36.5	43.3	37.8	38.4	39.0
RePRI [CVPR21] [44]	32.0	38.7	32.7	33.1	34.1	39.3	45.4	39.7	41.8	41.6
CWT [CVPR21] [2]	32.2	36.0	31.6	31.6	32.9	40.1	43.8	39.0	42.4	41.3
ASGNet [CVPR21] [24]	34.9	36.9	34.3	32.1	34.6	41.0	48.3	40.1	40.5	42.5
HSNet [ICCV21] [6]	36.3	43.1	38.7	38.7	39.2	43.3	51.3	48.2	45.0	46.9
BAM [CVPR22] [39]	43.4	50.6	47.5	43.4	46.2	49.3	54.2	51.6	49.6	51.2
SSP [ECCV22] [46]	35.5	39.6	37.9	36.7	37.4	40.6	47.0	45.1	43.9	44.1
VAT [ECCV22] [33]	39.0	43.8	42.6	39.7	41.3	44.1	51.1	50.2	46.1	47.9
DPCN [CVPR22] [51]	42.0	47.0	43.2	39.7	43.0	46.0	54.9	50.8	47.4	49.8
IPMT [NeurIPS22] [34]	41.4	45.1	45.6	40.0	43.0	43.5	49.7	48.7	47.9	47.5
DCAMA [ECCV22] [7]	41.9	45.1	44.4	41.7	43.3	45.9	50.5	50.7	46.0	48.3
QCLNet [TCSVT23] [30]	39.8	45.7	42.5	41.2	42.3	46.4	53.0	52.1	48.6	50.0
ABCNet [CVPR23] [48]	42.3	46.2	46.0	42.0	44.1	45.5	51.7	52.6	46.4	49.1
MIANet [CVPR23] [47]	42.5	**53.0**	**47.8**	47.4	47.7	45.8	**58.2**	51.3	51.9	51.7
SCCAN [ICCV23] [49]	39.5	49.3	47.3	44.3	45.1	45.7	56.4	**56.5**	50.7	52.3
DRNet [TCSVT24] [37]	42.1	42.8	42.7	41.3	42.2	47.7	51.7	47.0	49.3	49.0
Proposal-based methods with ResNet-50										
MMFormer [NeurIPS22] [12]	40.5	47.7	45.2	43.3	44.2	44.0	52.4	47.4	50.0	48.4
PRFormer [Ours]	**49.6**	50.8	45.2	**50.6**	**49.1**	**54.3**	55.5	49.5	**56.0**	**53.8**
Pixel-level feature correlation methods with ResNet-101										
PFENet [TPAMI20] [23]	36.8	41.8	38.7	36.7	38.5	40.4	46.8	43.2	40.5	42.7
CWT [CVPR21] [2]	30.3	36.6	30.5	32.2	32.4	38.5	46.7	39.4	43.2	42.0
HSNet [ICCV21] [6]	37.2	44.1	42.4	41.3	41.2	45.9	53.0	51.8	47.1	49.5
NTRENet [CVPR22] [40]	38.3	40.4	39.5	38.1	39.1	42.3	44.4	44.2	41.7	43.2
SSP [ECCV22] [46]	39.1	45.1	42.7	41.2	42.0	47.4	54.5	50.4	49.6	50.2
IPMT [NeurIPS22] [34]	40.5	45.7	44.8	39.3	42.6	45.1	50.3	49.3	46.8	47.9
QCLNet [TCSVT23] [30]	40.0	45.5	45.1	43.6	43.6	46.9	55.8	53.6	51.1	51.9
SCCAN [ICCV23] [49]	41.7	51.3	**48.4**	46.7	47.0	49.0	**59.3**	**59.4**	52.7	55.1
DRNet [TCSVT24] [37]	43.2	43.9	43.3	43.9	43.6	52.0	54.5	47.9	49.8	51.1
Proposal-based methods with ResNet-101										
MMFormer† [NeurIPS22] [12]	45.8	45.1	44.5	44.9	45.1	49.5	52.9	46.2	52.8	50.3
PRFormer [Ours]	**47.8**	**51.5**	47.3	**51.2**	**49.4**	**55.3**	58.1	50.9	**57.3**	**55.4**

TABLE III
EFFICIENCY COMPARISON OF PRFORMER AND TWO REPRESENTATIVE PRIOR METHODS ON 1-SHOT PASCAL-5^i WITH RESNET-50.

Methods	HSNet	NTRENet	VAT	BAM	SCCAN	MMFormer	PRFormer
Infer time (ms/it)	734	129	534	106	130	165	**99**

50 and 73.0% mIoU with ResNet-101, demonstrating its effectiveness across different backbones and few-shot settings.

b) COCO-20^i.: The COCO-20^i dataset is considerably more challenging than PASCAL-5^i, due to its four times of categories and more than ten times of samples. Despite this, PRFormer achieves 49.1% mIoU (1-shot) and 53.8% mIoU (5-shot) with the ResNet-50 backbone, which is 1.4% (1-shot) and 1.5% (5-shot) ahead of previous SOTA methods, respectively. With the ResNet-101 backbone, PRFormer achieves 49.4% mIoU for 1-shot and 55.4% mIoU for 5-shot, continuing to lead in performance. Notably, PRFormer outperforms MMFormer by approximately 5% in both 1-shot and 5-shot segmentation with ResNet-50, further unleashing the potential of proposal-based methods. These results establish PRFormer as the new SOTA in proposal-based FSS methods.

c) Efficiency comparison. : We evaluate the inference time of various approaches using the ResNet-50 backbone on the 1-shot task of PASCAL-5^i. For a fair comparison, efficiency experiments are conducted on a single NVIDIA RTX 2080Ti with PyTorch v1.10.1. Tab. III shows that our PRFormer demonstrates superior efficiency compared to previous advanced methods. Pixel-wise comparison methods [6], [33] exhibit the lowest efficiency due to their extensive computation on 4D correlations. Previous prototype comparison methods [39], [40], [47], [49] are more efficient than the earlier proposal-based method MMFormer [12], which still relies on a few-to-many feature alignment process. Our PRFormer achieves an inference time of 99ms per episode, highlighting the effectiveness of few-to-few proposal-based methods in terms of efficiency compared to other advanced approaches.

VI. ABLATION STUDIES

We conduct a series of ablation studies on the PASCAL-5^i dataset to evaluate the contribution of each proposed module and loss in PRFormer. All experiments are performed under 1-shot settings using the ImageNet-pretrained ResNet-50.

A. Ablation Study on Component Integration

In this section, we evaluate the effectiveness of key components in PRFormer, including the PrCC, PPI, WPR modules, and the IoU-KLD loss $\mathcal{L}_{IoU-KLD}$. Tab IV summarizes the

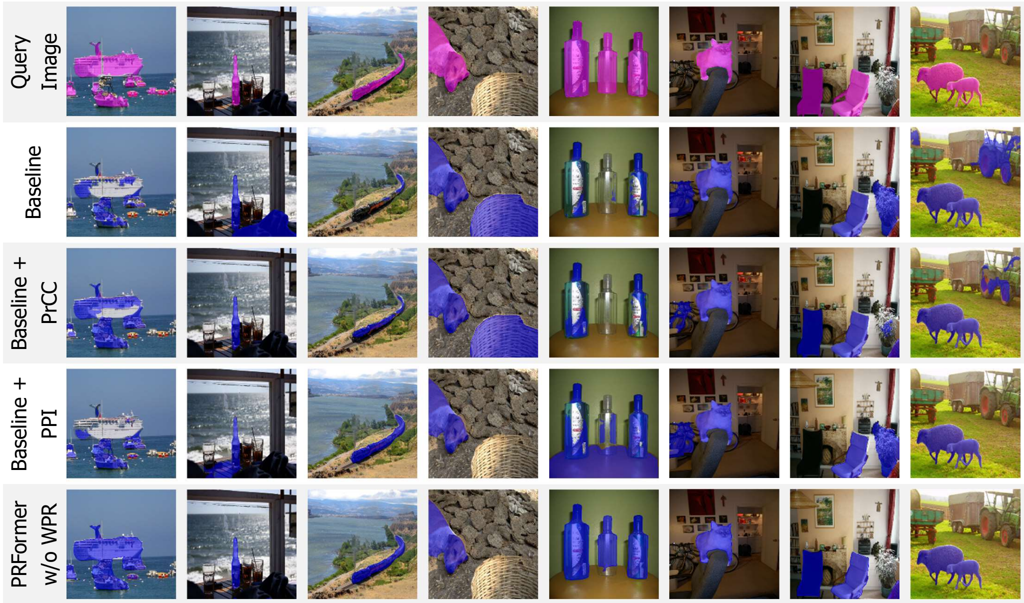


Fig. 4. Qualitative results of different module combinations on Pascal-5^i and COCO-20^i. The baseline represents direct prototype matching without adaptation.

TABLE IV
ABLATION STUDY ON COMPONENT INTEGRATION. $\mathcal{L}_{IoU-KLD}$ REFERS TO THE *IoU-KLD* LOSS WITH THE SIMILARITY VECTOR. NOTE THAT THE PRCC MODULE IS GUIDED BY THE *cdNCE* LOSS.

PrCC	PPI	$\mathcal{L}_{IoU-KLD}$	WPR	mIoU (%)
✓				66.6
	✓			65.9
✓		✓		67.9
	✓	✓		67.5
✓	✓	✓		68.5
✓	✓	✓	✓	**69.5**

TABLE V
ABLATION STUDY ON THE COEFFICIENT OF $\mathcal{L}_{IoU-KLD}$.

λ_4	mIoU (%)	Δ
0	67.5	0.0
10	68.1	+0.6
20	**68.5**	+1.0
30	68.1	+0.6
50	68.1	+0.6

TABLE VI
ABLATION STUDY ON THE UTILIZATION OF $\mathcal{L}_{IoU-KLD}$ ON THE SIMILARITY VECTOR.

Losses Selection	mIoU (%)
$\mathcal{L}_{IoU-KLD}$	**68.5**
$\mathcal{L}_{co}$	67.7

results from various combinations of these modules and loss functions. Each experiment consistently integrates $\mathcal{L}_{cdNCE}$ loss with the PrCC module. While maintaining an MLP structure for similarity vectors, we avoid using both the PrCC and PPI modules simultaneously. Isolated evaluations of the PrCC and PPI modules yield mIoUs of 66.6% and 65.9%, respectively, confirming their efficacy in selecting appropriate mask proposals for accurate segmentation. Incorporating $\mathcal{L}_{IoU-KLD}$ increases mIoU by 1.3% for PrCC and 1.6% for PPI, underscoring this loss's role in refining similarity vectors based on IoU values between proposal masks and ground truth. The synergy of the PrCC and PPI modules elevates mIoU to 68.5%, outperforming their individual contributions by 0.6% and 1.0%, respectively, highlighting the benefit of leveraging their complementary strengths. To address inaccuracies in proposal generation, we employ the WPR module, which refines predictions using mid-level features and sorted weighted proposals, achieving an additional 1.5% mIoU improvement. We further assess the impact of different coefficient values for $\mathcal{L}_{IoU-KLD}$. As shown in Tab. V, setting the coefficient λ_4 to 20 maximizes the loss's potential for precise optimization of similarity vectors. Moreover, compared to the cross-alignment loss $\mathcal{L}_{co}$ from MMFormer [12], which only selects the mask proposals with maximum and minimum values in the similarity vector for computing dice loss, our $\mathcal{L}_{IoU-KLD}$ offers a 0.8% improvement according to Tab. VI.

The qualitative analysis of different module combinations is presented in Fig. 4. The baseline method, matching prototypes without adaptation, is depicted in the 1st row of Tab. VII. Methods from the 3rd to 5th rows correspond to those in Tab. IV. Typically, PrCC-only and PPI-only methods outperform the baseline, particularly in object localization and coverage enhancement. The PrCC module excels with objects having minimal internal diversity, such as the *bottle* in the 2nd column and the *chair* in the 7th column, while the PPI module is better suited for objects with significant internal diversity, such as the *train* in the 3rd column, the *dog* in the 4th column, and the *sheep* in the last column. The qualitative assessment underscores the distinct advantages of the PrCC and PPI modules: PrCC compresses spatial information into vectors, reducing internal diversity, whereas PPI retains internal diversity through mask prompts but lacks channels for precise global information. By combining these two modules, we integrate their strengths, enhancing predictions in more complex scenarios and illustrating the efficiency and effectiveness of their integration in a few-to-few-matching manner.

TABLE VII
ABLATION STUDY ON THE ADAPTATION SCHEME IN THE LOCAL ADAPTATION AND INTER-LEVEL ADAPTATION.

Adaptation scheme	mIoU (%)
None	43.4
Linear layer	65.1
MLP	**66.6**
MLP *w.* Residual	65.9

TABLE VIII
ABLATION STUDY ON THE PRCC MODULE. PR_4 REFERS TO THE METHOD USING PROTOTYPES FROM THE 4TH-LEVEL BACKBONE FEATURES. $\mathrm{PR}_{\widehat{v}}$ AND $\mathrm{PR}_{\widetilde{v}}$ DENOTE METHODS WITH SOLELY LOCAL AND INTER ADAPTED PROTOTYPES IN PRCC, RESPECTIVELY. THE 5_{th} ROW, WHICH COMBINES $\mathrm{PR}_{\widehat{v}}$ AND $\mathrm{PR}_{\widetilde{v}}$, DEMONSTRATES THE METHOD WITH PRCC THAT UNIFIES PROTOTYPES ACROSS BOTH OF THEM. $\mathcal{L}_{cdNCE}$ MEANS THE USE OF *cdNCE* LOSS IN THE ADAPTATION PROCESS.

Pr_4	$\mathrm{Pr}_{\widehat{v}}$	$\mathrm{Pr}_{\widetilde{v}}$	$\mathcal{L}_{cdNCE}$	mIoU(%)	Δ
✓				57.0	0.0
✓			✓	62.8	+5.8
	✓		✓	64.9	+7.9
		✓	✓	65.5	+8.5
	✓	✓	✓	**66.6**	+9.6

B. Effectiveness of PrCC Module

The PrCC module matches proposals by adapting various prototypes guided by $\mathcal{L}_{cdNCE}$. We define the baseline as the method that matches prototypes directly obtained from the features F_4° of the last block of the ResNet backbone, as fine-grained features offer greater inter-category distinctiveness. Note that the baseline method includes an MLP for prototype adaptation. Building on this baseline, we introduce different PrCC designs using prototypes from the proposal generator and the cdNCE loss $\mathcal{L}_{cdNCE}$. The experiment results are shown in Tab. VIII. The baseline method, denoted as Pr_4, achieves a mIoU of 57.0%. Incorporating $\mathcal{L}_{cdNCE}$ with the adaptation module yields a 4.5% improvement, demonstrating the effectiveness of $\mathcal{L}_{cdNCE}$ in promoting category distinctiveness and preventing overfitting. Unlike previous FSS pipelines, which rely on few-to-many prototype comparisons and many-to-many pixel-wise comparisons, we utilize features from the Proposal Generator in the second stage to derive prototypes with finer global information. This approach is more efficient, using features with 256 channels compared to the 2048 channels of F_4°. We explore two adaptation methods to optimally utilize these prototypes: local adaptation as $\mathrm{Pr}_{\widehat{v}}$ and inter-level adaptation as $\mathrm{Pr}_{\widetilde{v}}$. Comparing the experimental results in the third and fourth rows of Tab. VIII with those in the second row, our two adaptation methods yield performance gains of 2.1% and 2.7%, respectively, indicating that the prototypes from the proposal generator are more suitable for the matching process. Moreover, we enhance the benefits of both methods by combining the two adapted prototypes ($\widehat{\boldsymbol{v}}$ and $\widetilde{\boldsymbol{v}}$) through a linear layer, achieving a remarkable mIoU of 66.6%.

To further demonstrate the reliability of the prototype selection and adaptation method, we visualize the compactness of each category in COCO-20^i, as shown in Fig. 5. For each category, we use average variance to measure the internal differences of prototypes. Note that the average variance is calculated on the prototypes of categories from their corresponding test folds, meaning these unseen categories are not specifically trained for evaluation. The prototypes Pr_4 from the backbone exhibit significantly greater variance compared to Pr_v, which is extracted from the proposal generator. Our adaptation strategy effectively enhances the compactness of prototypes from the same category, demonstrating that the adaptation positively impacts unseen classes instead of overfitting seen classes.

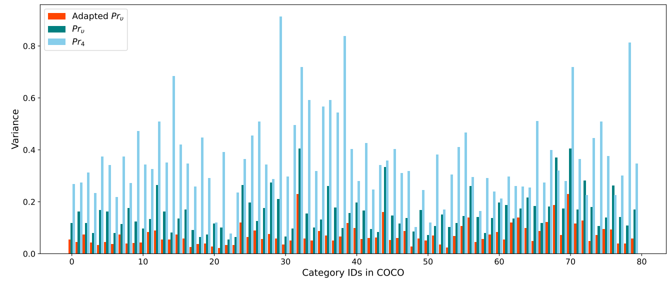


Fig. 5. The visualization of intra-category compactness via average variance of the categories in COCO-20^i. Note that the variance is calculated on the prototypes of categories from their corresponding test folds.

TABLE IX
ABLATION STUDY ON THE COEFFICIENT OF $\mathcal{L}_{cdNCE}$.

λ_3	mIoU (%)	Δ
0	64.3	0.0
0.5	65.8	+1.5
1	**66.6**	+2.3
2	65.9	+1.6
5	64.9	+0.6
10	55.3	-9.0

TABLE X
ABLATION STUDY ON THE MOMENTUM FACTOR α FOR UPDATING THE BUFFER IN THE PRCC MODULE.

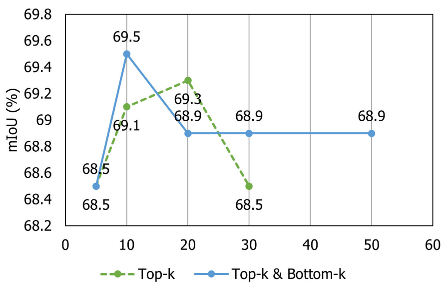
α	0.2	0.4	0.5	0.6	0.8
mIoU	65.1	65.8	**66.6**	66.3	66.1

To analyze the necessity of the current adaptation scheme in the local and inter-level adaptation of the PrCC module, we conduct the ablation study on several adaptation schemes, as shown in Tab. VII. Without any external adaptation, the proposal selection result from the matching process with the original prototypes reaches only a mIoU of 43.4%. A simple linear layer for prototype adaptation enhances the effectiveness of prototype matching, achieving a mIoU of 65.1%, demonstrating the necessity of further adaptation on the prototypes. The MLP layer improves the performance of the PrCC module to a mIoU of 66.6%, as the multi-layer structure enables more precise adaptation. Besides, we evaluate the additional residual operation on the MLP, yet the performance is 0.8% lower than not using residual operation.

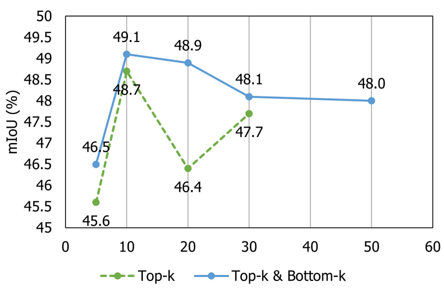
Furthermore, we analyze the influence of different hyperparameter values involved in the PrCC module, including the coefficient λ_4 of $\mathcal{L}_{cdNCE}$ and the momentum factor α for updating the buffer prototypes. The experimental results of hyperparameter λ_4 are shown in Tab. IX. While setting λ_4 to 1, the PrCC-only method achieves the highest mIoU with 2.3% of promotion compared to not using $\mathcal{L}_{cdNCE}$. However, higher values of λ_4 result in a negative effect on maintaining the semantic distinctiveness of unseen categories. The experiment results in Tab. X show that $\mathcal{L}_{cdNCE}$ has the best performance when α is set to 0.5, demonstrating

TABLE XI
ABLATION STUDY ON VARIOUS PROPOSAL SELECTION IN WPR.

Proposal Refinement	mIoU (%)	Δ
None	68.5	0.0
Proposals	68.2	-0.3
Weighted Proposals	68.7	+0.2
Sorted Weighted Proposals	69.1	+0.6
Top 10 & Bottom 10 Proposals	**69.5**	+1.0



(a) The results on Pascal-5^i with different Top-k or Top-k & Bottom-k settings.



(b) The results on COCO-20^i with different Top-k or Top-k & Bottom-k settings.

Fig. 6. Ablation study on Top-k vs. Top-k & Bottom-k weighted proposals with ResNet-50 backbone. Both (a) and (b) demonstrate the effectiveness of selecting Top-10 & Bottom-10 weighted proposals for generating the prediction mask.

that a moderate momentum factor is suitable for maintaining representative buffer prototypes.

C. Effectiveness of WPR Module

The WPR module refines proposal-based predictions for precise segmentation, addressing the issue that the proposal generator, trained on the training set, often generates inferior quality proposals for novel classes. The WPR module requires these proposals to guide the refinement process. However, the original mask proposals lack a clear sequence, with similar proposals randomly positioned. This disorder significantly disrupts the refinement process because proposals in the same channel can have reverse contributions for different query samples. Experiment results, shown in Tab. XI, indicate that disordered proposals negatively impact the mIoU by -0.3% compared to the method without WPR, demonstrating that unprocessed original proposals can hinder refinement. To address this, we introduce a weighting process on the proposals, giving more influence to those similar to the initial prediction. By multiplying the similarity vector s with the corresponding mask proposal, PRFormer's performance marginally improves to a mIoU of 68.7%, which is 0.2% better than the method without WPR. Additionally, to address the random order of mask proposals, we sort them based on their corresponding value in the similarity vector s. This sorting ensures an explicit sequence and stable significance at specific positions. As a result, a substantial 0.6% improvement in mIoU is observed through sorting the weighted proposals.

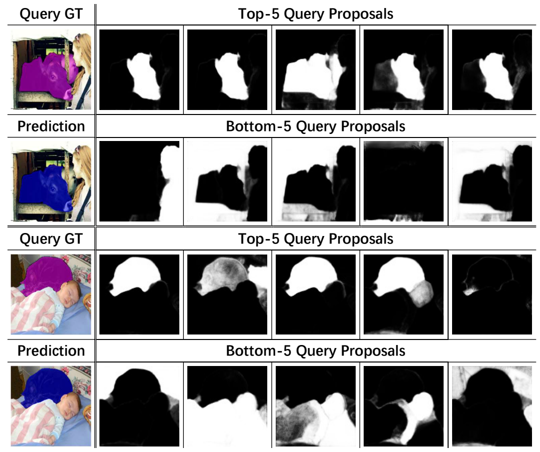


Fig. 7. The visualization of the Top-5 & Bottom-5 mask proposals.

Although the previous processing of proposals enhances the effectiveness of the WPR module, there is still redundancy in proposal usage. Typically, the most similar samples are repositioned to the front, while the least similar ones are moved to the back, providing clear positive or negative guidance. However, proposals with unclear similarities accumulate in the middle positions. Due to the randomness of proposal generation and matching errors, positive, negative, and ambiguous proposals can coexist in this mid-range, complicating the learning process for proposal contributions. We are considering removing a specific range of proposals to address this issue and improve efficiency. We experimented with the WPR module using top-k, top-k & bottom-k proposals. Fig. 6a and Fig. 6b illustrate the overall performance as the value of k changes along the horizontal axis, comparing the methods using only top-k proposals versus both top-k & bottom-k proposals. Among the evaluated methods, using both top-10 & bottom-10 proposals, as implemented in PRFormer, achieved the highest mIoU of 69.5%. The top-k selection performs best when k is set to 20, trailing the top-10 & bottom-10 method by only 0.2% in mIoU. By selecting the top 10 and bottom 10 proposals, PRFormer gains an additional 0.4% mIoU improvement while eliminating redundancy. Furthermore, we visualize the top 5 and bottom 5 mask proposals in Fig. 7. The most similar mask proposals highlight the potential region of the target object, while the least similar ones depict background objects.

D. Qualitative Results

To highlight the effectiveness of PRFormer, we have visualized the segmentation outcomes in Fig. 8. The first three columns illustrate the significant improvements achieved by our proposed PRFormer and the WPR module compared to MMFormer. The 4^{th} to 6^{th} columns demonstrate that even without the WPR module, PRFormer exhibits superior performance over MMFormer. The last two rows reveal that the

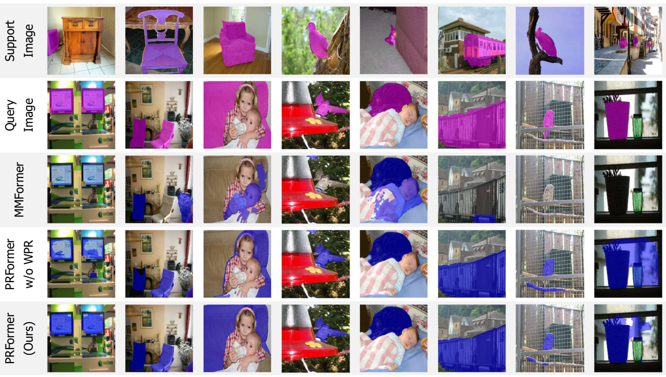


Fig. 8. Qualitative result of MMFormer, PRFormer without WPR and PRFormer on Pascal-5^i and COCO-20^i.

subsequent incorporation of the WPR module further refines details, leading to more accurate predictions.

VII. Conclusion

In this work, we proposed Prototype and Mask Matching transFormer (PRFormer) with several components, to enhance the performance of two-stage proposal-based methods for Few-shot Semantic Segmentation. The PrCC module, accompanied by the cdNCE loss, adjusted feature prototypes for reliable semantic similarity assessment. The parameter-free PPI module efficiently and effectively enhanced spatial similarity assessment regarding spatial overlap, while the IoU-KLD loss sufficiently supervised the similarity value corresponding to each proposal. Moreover, the WPR module refined predictions using weighted proposals and middle-level features. Overall, the experimental results demonstrate that PRFormer achieves state-of-the-art performance among other methods. One limitation is that the proposals do not precisely fit the regions of novel categories, which may be a direction for future research.

References

[1] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-shot learning for semantic segmentation," in *British Machine Vision Conference*, 2017, pp. 167.1–167.13.

[2] Z. Lu, S. He, X. Zhu, L. Zhang, Y. Song, and T. Xiang, "Simpler is better: Few-shot semantic segmentation with classifier weight transformer." in *IEEE Int. Conf. Comput. Vis.*, 2021, pp. 8741–8750.

[3] H. Liu, P. Peng, T. Chen, Q. Wang, Y. Yao, and X.-S Hua, "Fecanet: Boosting few-shot semantic segmentation with feature-enhanced context-aware network." *IEEE Trans. Multimedia*, vol. 25, pp. 8580–8592, 2023.

[4] C. Zhang, G. Lin, F. Liu, R. Yao, and C. Shen, "CANet: Class-Agnostic Segmentation Networks with Iterative Refinement and Attentive Few-Shot Learning." In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 5217–5226, 2019.

[5] X. Zhang, Y. Wei, Y. Yang, and T. S. Huang, "Sg-one: Similarity guidance network for one-shot semantic segmentation," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3855–3865, 2020.

[6] J. Min, D. Kang, and M. Cho, "Hypercorrelation squeeze for few-shot segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 6941–6952, 2021.

[7] X. Shi, D. Wei, Y. Zhang, D. Lu, Munan Ning, J. Chen, K. Ma, and Y. Zheng, "Dense cross-query-and-support attention weighted mask aggregation for few-shot segmentation," in *Proc. Eur. Conf. Comput. Vis.*, pp. 151–168, 2022.

[8] B. Cheng, A. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 17864–17875, 2021.

[9] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1290–1299, 2022.

[10] M. Xu, Z. Zhang, F. Wei, Y. Lin, Y. Cao, H. Hu, and X. Bai, "A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model," in *Proc. Eur. Conf. Comput. Vis.*, pp. 736–753, 2022.

[11] F. Liang, B. Wu, X. Dai, K. Li, Y. Zhao, H. Zhang, P. Zhang, P. Vajda, and D. Marculescu, "Open-vocabulary semantic segmentation with mask-adapted clip," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 7061–7070, 2023.

[12] S. Jiao, G. Zhang, S. Navasardyan, L. Chen, Y. Zhao, Y. Wei, and H. Shi, "Mask matching transformer for few-shot segmentation," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 823–836, 2022.

[13] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 3431–3440, 2015.

[14] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, pp. 801–818, 2018.

[15] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," *arXiv preprint arXiv:1412.7062*, 2014.

[16] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2017.

[17] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[18] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2881–2890.

[19] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 603–612.

[20] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 12077–12090, 2021.

[21] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, pp. 213–229, 2020.

[22] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A.-C. Berg, W. Lo, et al., "Segment anything," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 4015–4026, 2023.

[23] Z. Tian, H. Zhao, M. Shu, Z. Yang, R. Li, and J. Jia, "Prior guided feature enrichment network for few-shot segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 1050–1065, 2020.

[24] G. Li, V. Jampani, L. Sevilla-Lara, D. Sun, J. Kim, and J. Kim, "Adaptive prototype learning and allocation for few-shot segmentation," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 8334–8343, 2021.

[25] A. Okazawa, "Interclass Prototype Relation for Few-Shot Segmentation," in *Proc. Eur. Conf. Comput. Vis.*, pp. 362–378, 2022.

[26] M. Zhang, M. Shi, and L. Li, "MFNet: Multiclass Few-Shot Segmentation Network With Pixel-Wise Metric Learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 12, pp. 8586–8598, 2022.

[27] Z. Zhou, H. Xu, Y. Shu, and L. Liu, "Unlocking the Potential of Pre-trained Vision Transformers for Few-Shot Semantic Segmentation through Relationship Descriptors," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 3817–3827, 2024.

[28] X. Bao, J. Qin, S. Sun, X. Wang, and Y. Zheng, "Relevant Intrinsic Feature Enhancement Network for Few-Shot Semantic Segmentation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 2, pp. 765–773, 2024.

[29] J. Li, K. Shi, G. Xie, X. Liu, J. Zhang, and T. Zhou, "Label-Efficient Few-Shot Semantic Segmentation with Unsupervised Meta-Training," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 4, pp. 3109–3117, 2024.

[30] Z. Zheng, G. Huang, X. Yuan, C. Pun, H. Liu, and W. Ling, "Quaternion-Valued Correlation Learning for Few-Shot Semantic Segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 5, pp. 2102–2115, 2023.

[31] L. Zhang, X. Zhang, Q. Wang, W. Wu, X. Chang, and J. Liu, "RPMG-FSS: Robust Prior Mask Guided Few-Shot Semantic Segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 11, pp. 6609–6621, 2023.

[32] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

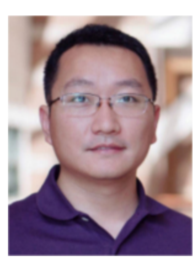
[33] S. Hong, S. Cho, J. Nam, S. Lin, and S. Kim, "Cost aggregation with 4D convolutional swin transformer for few-shot segmentation," in *Proc. Eur. Conf. Comput. Vis.*, pp. 108–126, 2022.
[34] Y. Liu, N. Liu, X. Yao, and J. Han, "Intermediate Prototype Mining Transformer for Few-Shot Semantic Segmentation," in *Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 38020-38031.
[35] B. Peng, Z. Tian, X. Wu, C. Wang, S. Liu, J. Su, and J. Jia, "Hierarchical Dense Correlation Distillation for Few-Shot Segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 23641–23651.
[36] Q. Cao, Y. Chen, C. Ma, and X. Yang, "Break the bias: Delving semantic transform invariance for few-shot segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 5, pp. 3971–3982, 2024.
[37] Z. Chang, X. Gao, N. Li, H. Zhou, and Y. Lu, "Drnet: Disentanglement and recombination network for few-shot semantic segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 7, pp. 5560–5574, 2024.
[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[39] C. Lang, G. Cheng, B. Tu, and J. Han, "Learning what not to segment: A new perspective on few-shot segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8057–8067.
[40] Y. Liu, N. Liu, Q. Cao, X. Yao, J. Han, and L. Shao, "Learning nontarget knowledge for few-shot semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11 573–11 582.
[41] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, "Panet: Few-shot image semantic segmentation with prototype alignment," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9197–9206.
[42] C. Zhang, G. Lin, F. Liu, J. Guo, Q. Wu, and R. Yao, "Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9587–9595.
[43] Y. Liu, X. Zhang, S. Zhang, and X. He, "Part-aware prototype network for few-shot semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 142–158.
[44] M. Boudiaf, H. Kervadec, Z. I. Masud, P. Piantanida, I. Ben Ayed, and J. Dolz, "Few-shot segmentation without meta-learning: A good transductive inference is all you need?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 13 979–13 988.
[45] G. Zhang, G. Kang, Y. Yang, and Y. Wei, "Few-shot segmentation via cycle-consistent transformer," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 21984–21996, 2021.
[46] Q. Fan, W. Pei, Y.-W. Tai, and C.-K. Tang, "Self-support few-shot semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 701–719.
[47] Y. Yang, Q. Chen, Y. Feng, and T. Huang, "Mianet: Aggregating unbiased instance and general information for few-shot semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 7131–7140.
[48] Y. Wang, R. Sun, and T. Zhang, "Rethinking the correlation in few-shot segmentation: A buoys view, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 7183–7192.
[49] Q. Xu, W. Zhao, G. Lin, and C. Long, "Self-calibrated cross attention network for few-shot segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2023, pp. 655–665.
[50] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. Int. Conf. on 3D vis.*, 2016, pp. 565–571.
[51] J. Liu, Y. Bao, G.-S. Xie, H. Xiong, J.-J. Sonke, and E. Gavves, "Dynamic prototype convolution network for few-shot semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11 553–11 562.
[52] K. Nguyen and S. Todorovic, "Feature weighting and boosting for few shot segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 622–631.
[53] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal on Computer Vision*, vol. 88, pp. 303–338, 2010.
[54] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 991–998.
[55] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[56] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent.*, 2018.

Guangyu Gao received the Ph.D. in computer science from the Beijing University of Posts and Telecommunications, Beijing, China, in 2013, and served as a Joint-Ph.D. candidate at the National University of Singapore from 2012 to 2013. He is currently an Associate Professor with the School of Computer Science and Technology, Beijing Institute of Technology. He received the highly competitive IBM Faculty Award in 2016, 2017, and 2019, recognizing his research excellence and industry impact.

Anqi Zhang received the B.Eng. degree in computer science and technology from Beijing Institute of Technology, Beijing, China, in 2023, where he is currently pursuing the master's degree now. His current research interests include computer vision and machine learning.

Jianbo Jiao received the Ph.D. in computer science from the City University of Hong Kong, in 2018. He was a Visiting Scholar with the Beckman Institute, University of Illinois at Urbana–Champaign, from 2017 to 2018. He is currently an Assistant Professor with the School of Computer Science, University of Birmingham, a Royal Society Short Industry Fellow, and a Visiting Researcher with the University of Oxford, U.K. Before joining Birmingham, he was a Postdoctoral Researcher with the Department of Engineering Science, University of Oxford. His research interests include machine learning and computer vision. He was a recipient of Hong Kong Ph.D. Fellowship Scheme (HKPFS).

Chi Harold Liu (Senior Member, IEEE) received the B.Eng. degree from Tsinghua University, Beijing, China, and the Ph.D. degree from the Imperial College London, London, U.K. He is currently a Full Professor and the Vice Dean of the School of Computer Science and Technology, Beijing Institute of Technology, Beijing. Before that, he worked for IBM Research - China and Deutsche Telekom Laboratories, Berlin, Germany, and IBM T. J. Watson Research Center, USA. He is now an Associate Editor for IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING. His research interests include big data analytics, mobile computing, and deep learning. He is a fellow of IET and the Royal Society of the Arts.



Yunchao Wei received the PhD degree from Beijing Jiaotong University, Beijing, China, in 2016. He is currently a professor at the Center of Digital Media Information Processing, Institute of Information Science, Beijing Jiaotong University. He was a Postdoctoral Researcher at Beckman Institute, UIUC, from 2017 to 2019. He is an ARC Discovery Early Career Researcher Award Fellow from 2019 to 2021. His current research interests include computer vision and machine learning.