Bidirectional Reciprocative Information Communication for Few-Shot Semantic Segmentation

Yuanwei Liu [1] **Junwei Han** [1 2] **Xiwen Yao** [1] **Salman Khan** [3 4] **Hisham Cholakkal** [3] **Rao Muhammad Anwer** [3] **Nian Liu** [3] **Fahad Shahbaz Khan** [3 5]

Abstract

Existing few-shot semantic segmentation methods typically rely on a one-way flow of category information from support to query, ignoring the impact of intra-class diversity. To address this, drawing inspiration from cybernetics, we introduce a Query Feedback Branch (QFB) to propagate query information back to support, generating a query-related support prototype that is more aligned with the query. Subsequently, a Query Amplifier Branch (QAB) is employed to amplify target objects in the query using the acquired support prototype. To further improve the model, we propose a Query Rectification Module (QRM), which utilizes the prediction disparity in the query before and after support activation to identify challenging positive and negative samples from ambiguous regions for query self-rectification. Furthermore, we integrate the QFB, QAB, and QRM into a feedback and rectification layer and incorporate it into an iterative pipeline. This configuration enables the progressive enhancement of bidirectional reciprocative flow of category information between query and support, effectively providing query-adaptive support information and addressing the intra-class diversity problem. Extensive experiments conducted on both PASCAL-5^i and COCO-20^i datasets validate the effectiveness of our approach. The code is available at https://github.com/LIUYUANWEI98/IFRNet.

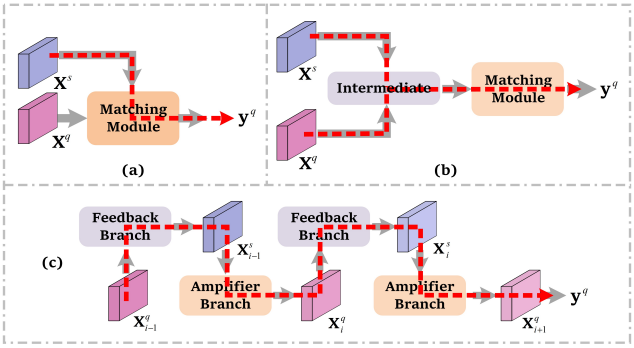

Figure 1. **Comparison between existing framework (a)&(b) and our approach (c) for FSS.** The red dotted line indicates the information flow of category information. (a) The category information is extracted from support and flows into the query to identify target objects. (b) The category information simultaneously flows from support and query into the intermediate representation. However, in our method (c), the category information circulates bidirectionally between query and support, which could be more compatible with target object in the query.

1. Introduction

Few-shot semantic segmentation (FSS) poses a significant challenge in computer vision, aiming to segment objects in images with a minimal number of labeled examples. This concept aligns with the human capability to comprehend new concepts based on limited instances, providing flexibility and practicality, particularly in scenarios where collecting an extensive amount of annotated data is impractical.

Following a typical paradigm proposed by (Shaban et al., 2017), most FSS methods (Zhang et al., 2021a; Li et al., 2021; Liu et al., 2020b; Yang et al., 2020; Liu et al., 2022b; Fan et al., 2022a; Gao et al., 2022; Moon et al., 2023; Tian et al., 2020; MAO, 2022; Nguyen & Todorovic, 2019; Dong

[1]Northwestern Polytechnical University [2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center [3]Mohamed bin Zayed University of Artificial Intelligence [4]Australian National University [5]CVL, Linkoping University. Correspondence to: Nian Liu <liunian228@gmail.com>, Junwei Han <junweihan2010@gmail.com>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

& Xing, 2018; Okazawa, 2022; Wang et al., 2019; Zhang et al., 2019; Rakelly et al., 2018; Yang et al., 2021; Liu et al., 2020a; Siam et al., 2019; Xie et al., 2021; Wang et al., 2020; Fan et al., 2022a; Lang et al., 2022; Fan et al., 2022b; Min et al., 2021; Hong et al., 2022; Shi et al., 2022; Zhang et al., 2021b; Liu et al., 2023b) initially extract category information from the support and subsequently employ it to activate target objects in the query image using a matching module, as depicted in Figure 1 (a). The flow path of the category information in these methods is 'support $\rightarrow$ query'. However, these approaches overlook the intra-class diversity between the query and support, treating the support as an idealized category representation, which is impractical. Other methods (Liu et al., 2022c; Hu et al., 2019; Wu et al., 2021) propose to collect the category information from support and query simultaneously, aggregating them into an intermediate representation, as illustrated in Figure 1 (b). The category information flow path is 'support $\rightarrow$ intermediate $\leftarrow$ query'. Although these methods can mitigate in-class diversity by fusing both query and support information, the provided support information is still derived from the original support feature, whose intra-class diversity with the query persists. As a consequence, the obtained category information from support remains incompatible with the query, making the intermediate representation sub-optimal.

To address the weakness discussed above, we draw inspiration from cybernetics and tackle the FSS problem from a new perspective. Specifically, we introduce information feedback to create a network for facilitating a bidirectional reciprocative flow of category information, as shown in Figure 1 (c). In our framework, the category information propagation path is 'query $\rightarrow$ support $\rightarrow$ query $\rightarrow$ support $\rightarrow$ ... $\rightarrow$ query'. As such, the support feature is no longer an independent entity. Instead, it establishes a dependency relationship with the query by feeding back the query information. This allows the support to dynamically adjust and focus more on parts related to the query, thereby producing a more compatible activation signal.

Specifically, we achieve this by designing two parallel branches: a Query Feedback Branch (QFB) and a Query Amplifier Branch (QAB). In the QFB, the support feature receives feedback information from the query and is used to generate a query-related support prototype that incorporates adaptive query knowledge. In the QAB, which operates in parallel, the query-related support prototype is employed as a control signal to activate the target objects once more. This dual-branch design ensures a bidirectional and constructive exchange of category information, which initially flows from the query to the support, and then back from the support to the query.

After the QAB, the target objects in the query are significantly activated. However, we noticed that some ambiguous regions still pose challenges for accurate segmentation. To address this, we introduce a Query Rectification Module (QRM). The QRM identifies these regions by comparing the differences between the predictions before and after activation by the query-related support prototype. These regions are categorized into hard positive and negative samples and are extracted as a supplementary prototype and an exclusion prototype, respectively. By utilizing these two prototypes, we further refine the query feature by reactivating and suppressing corresponding regions, thereby enhancing the segmentation accuracy.

By integrating the QFB, QAB, and QRM, we establish a Feedback and Rectification Layer (FRL) as the foundation of our model. We stack multiple FRLs to construct an Information Feedback and Rectification Network (IFRNet), facilitating the bidirectional reciprocative exchange of category information between the support and query. This process helps the support provide category information better aligned with the query, and rectify the query by itself.

Our main contributions can be summarized as:

- We introduce the concept of bidirectional reciprocative information communication into FSS based on cybernetics, aiming to reduce the intra-class gap between query and support.
- We introduce a Query Feedback Branch (QFB) to feed query information back to the support, and within the Query Amplifier Branch (QAB), support information is reciprocally transferred to the query. This establishes a bidirectional information flow structure between support and query in our IFRNet.
- We leverage the prediction difference to identify ambiguous regions and propose a Query Rectification Module (QRM) for self-rectifying query features based on inferred hard positive and negative samples.
- Extensive experiments on PASCAL-5^i and COCO-20^i datasets demonstrate that our proposed framework achieves state-of-the-art performance.

2. Related works

Few-shot semantic segmentation (FSS) aims to predict pixel labels for new classes using only a few annotated support images. PFENet (Tian et al., 2020) generates a prior mask and designs a feature enrichment module for multi-level feature matching. CWT (Lu et al., 2021) simplifies meta-learning by focusing on the classifier, while SCL (Zhang et al., 2021a) proposes self-guidance modules to retrieve critical information that may have been lost. To address the limitations of a single prototype, PPNet (Liu et al., 2020b) and PMM (Yang et al., 2020) propose multi-prototype approaches. Furthermore, ASGNet (Li et al., 2021) suggests an adaptive prototype learning strategy for FSS, using super pixel-guided clustering to obtain multiple prototypes. SCCA

(Xu et al., 2023) designs a patch alignment module to align each query patch with its most similar support patch for improved cross-attention. HDMNet (Peng et al., 2023) introduces self-attention modules and matching modules to mine pixel-level support correlation. MIANet (Yang et al., 2023) leverages semantic word embeddings as general knowledge for accurate segmentation. DPCN (Liu et al., 2022a) generates dynamic kernels from the support foreground and then uses these kernels for information interaction through convolution operations over the query.

However, certain works (Liu et al., 2022c; Hu et al., 2019; Wu et al., 2021; Lu et al., 2023) may share similar high-level ideas to ours, and we aim to provide clarification and comparison. The proposal in (Hu et al., 2019) involves aggregating query and support simultaneously into intermediate features across multiple scales, which are then used for segmentation. Similarly, (Liu et al., 2022c) advocates for collecting category information from both support and query concurrently, aggregating them into an intermediate representation. This representation subsequently replaces the support prototype to match with the query. Meanwhile, (Wu et al., 2021) introduces memory to gather meta-class information from both query and support features simultaneously. After training, the memory encompasses information from base classes and is transferred to novel classes during inference. Distinctively, (Lu et al., 2023) utilizes the similarity between query and support as a condition to guide the support information.

In contrast, our approach draws inspiration from cybernetics to establish a bidirectional reciprocative information exchange framework between query and support. This framework achieves a category information propagation path of 'query $\rightarrow$ support $\rightarrow$ query $\rightarrow$ support $\rightarrow$ ... $\leftarrow$ query'. Additionally, we employ the prediction difference before and after support activation to identify ambiguous regions and achieve query self-rectification.

3. Method

3.1. Problem Definition

Our work follows the meta-learning pipeline with episodic model training. The dataset is divided into a training set $\mathcal{D}_{train}$ and a test set $\mathcal{D}_{test}$, with the base categories $\mathcal{C}_{train}$ and the novel categories $\mathcal{C}_{test}$, respectively, where $\mathcal{C}_{train} \cap \mathcal{C}_{test} = \varnothing$. The model learns from $\mathcal{D}_{train}$ and is evaluated on $\mathcal{D}_{test}$. During training, episodes are created from $\mathcal{D}_{train}$, where $K+1$ image-mask pairs of the same base category form one episode. Among them, K pairs are treated as the support set $\mathcal{S}$, while the remaining pair is used as the query set $\mathcal{Q}$. The model uses both $\mathcal{S}$ and the query image $\mathbf{I^q}$ to predict the mask of the query. Model parameters are optimized under the supervision of the query mask $\mathbf{M^q}$. The testing phase is similar but uses data from $\mathcal{D}_{test}$, and the query mask $\mathbf{M^q}$ is used to assess the model performance on novel categories.

3.2. Overview

Here, we provide a brief overview of our IFRNet. As depicted in Figure 2, after extracting features from the backbone, both query and support features are passed through our proposed Feedback and Rectification Layers (FRL), which consists of three components: Query Feedback Branch (QFB), Query Amplifier Branch (QAB), and Query Rectification Module (QRM). The QFB gathers query category information and feeds it back to the support to activate and emphasize the foreground areas related to the query. We utilize *query-related mask pooling* to extract query-related category information from the support images and generate a query-related support protoype. The QAB enhances query category information and activates target objects using itself and the support prototype. Furthermore, we identify challenging samples in ambiguous regions within the query to enhance its own rectification in the QRM. Our iterative feedback mechanism of stacking FRL progressively facilitates bidirectional reciprocative exchange of category information and query self-rectification, ultimately enhancing segmentation performance.

3.3. Feedback and Rectification Layer

In contrast to previous works that employed a one-way flow, our objective is to introduce an additional branch that transfers category information from the query as feedback to the support. This facilitates a bidirectional flow of category information between the query and support.

The prerequisite for the feedback branch is to extract category information from the query. For simplicity, we use the query prototype to represent this information. Specifically, given the query feature $\boldsymbol{X}_i^q \in \mathbb{R}^{H\times W\times C}$ from the i-th layer, we initially generate a prediction using a basic segmentation network $\boldsymbol{Seg}$. This prediction is then used to generate a query prototype $\boldsymbol{P}_i^q \in \mathbb{R}^{1\times 1\times C}$ via masked average pooling. The process can be summarized as follows:

$$\boldsymbol{P}_i^q = \mathcal{F}_{pool}(\boldsymbol{X}_i^q \odot \mathcal{B}(\boldsymbol{Seg}(\boldsymbol{X}_i^q))), \tag{1}$$

where $\boldsymbol{Seg}$ consists of two 3x3 convolutional layers with a sigmoid activation function. The function $\mathcal{B}$ converts predictions into binary values of either 0 or 1, by setting the threshold as 0.5. $\mathcal{F}_{pool}$ means average pooling and $\odot$ means pixel-wise multiplication.

3.3.1. Query Feedback Branch

Given the support feature $\boldsymbol{X}_i^s \in \mathbb{R}^{1\times 1\times C}$ from the i-th layer, we utilize the query prototype $\boldsymbol{P}_i^q$ as category cues to emphasize the shared category information in the support feature and suppress dissimilar information. Furthermore, we employ this activated support feature $\boldsymbol{X}_{i+1}^s$ to segment the target objects in the support using another $\boldsymbol{Seg}$ network.

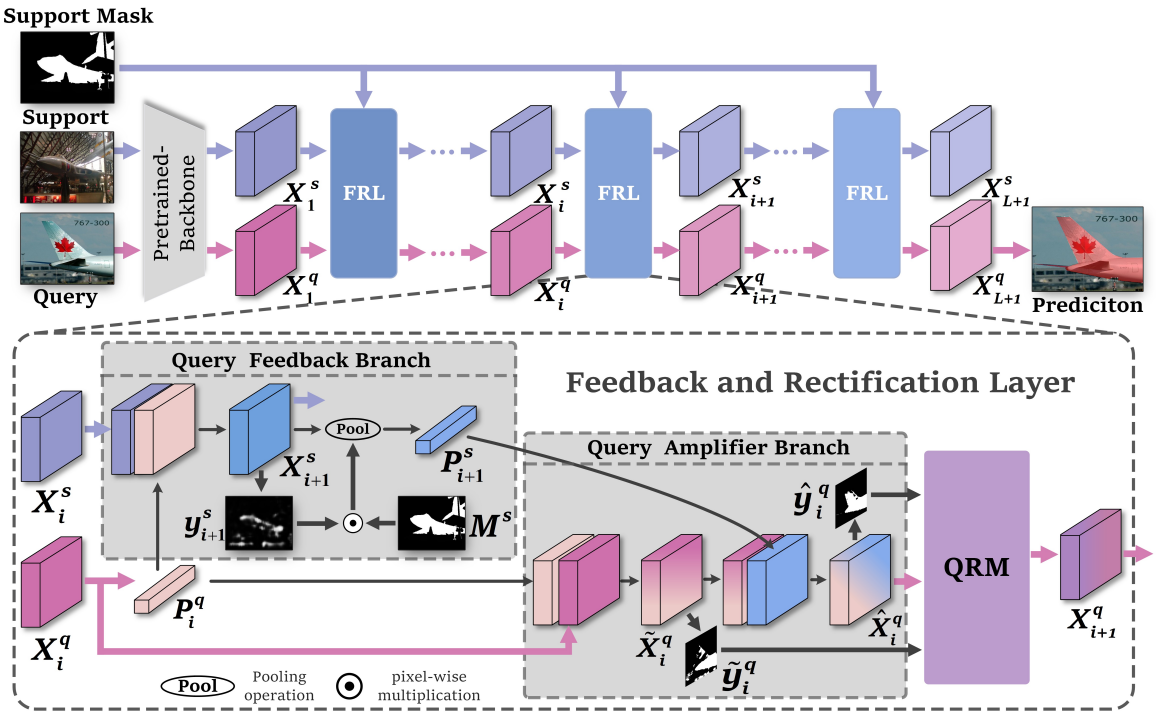

Figure 2. **Overall architecture of the proposed IFRNet.** Support and query images are first fed into the pre-trained backbone encoder to extract initial features, denoted as $\mathbf{X}_1^s$ and $\mathbf{X}_1^q$. Then, we feed them and the support mask $\mathbf{M}^s$ into our feedback and rectification layers (FRLs) to iteratively update support and query features. After L iterations, the final query feature $\mathbf{X}_{L+1}^{\mathbf{q}}$ is used to obtain the final segmentation result.

These procedures can be summarized as:

$$\boldsymbol{X}_{i+1}^s = \mathcal{F}_{1\times1}(\boldsymbol{X}_i^s \oplus \mathcal{O}(\boldsymbol{P}_i^q)) + \boldsymbol{X}_i^s, \tag{2}$$

$$\boldsymbol{y}_{i+1}^s = \boldsymbol{Seg}(\boldsymbol{X}_{i+1}^s), \tag{3}$$

where $\mathcal{F}_{1\times1}(\cdot)$ denotes a 1×1 convolutional layer that reduces the channel dimension from $2C$ to C, with the aim of at activating the related objects with the prototype. Concatenation $\oplus$ is performed along the channel dimension. $\mathcal{O}$ is an expansion function defined as $\mathcal{O}$: $\mathbb{R}^{1\times1\times C} \rightarrow \mathbb{R}^{H\times W\times C}$. Please note that the activated support feature $\boldsymbol{X}_{i+1}^s$ will be input to the next layer. In this way, a connection is established between the query and support, reducing the gap between them.

Query-Related Mask Pooling. The scores in the prediction mask $\boldsymbol{y}_{i+1}^s \in (0,1)^{H\times W\times1}$ not only represent the network's prediction probability for the support foreground object, but also indicate the correlation of the category information associated with the query target objects present in the support. We believe that this correlation can be intuitively utilized to help gather query-compatible category information from support.

To achieve this, we merge the prediction mask $\boldsymbol{y}_{i+1}^s$ with the ground-truth mask $\boldsymbol{M}^s \in \{0,1\}^{H\times W\times1}$ of support to generate a query-related mask. This mask considers the correlation of the target object areas between the query and support. A higher score signifies a stronger relevance to the query, whereas a lower score suggests less relevance. For the background, we set all pixels to 0 following the ground truth, effectively ignoring the background region. Subsequently, we utilize this query-related mask to average pool the support feature, yielding a query-related support prototype $\boldsymbol{P}_{i+1}^s \in \mathbb{R}^{1\times1\times C}$. To summarize, the process can be outlined as follows:

$$\hat{\boldsymbol{M}}_{i+1}^s = \boldsymbol{y}_{i+1}^s \odot \boldsymbol{M}^s, \tag{4}$$

$$\boldsymbol{P}_{i+1}^s = \mathcal{F}_{pool}(\boldsymbol{X}_{i+1}^s \odot \hat{\boldsymbol{M}}_{i+1}^s). \tag{5}$$

As a result, we successfully provide category information in the query to the support, and obtain a support prototype $\boldsymbol{P}_{i+1}^s$ that is specifically tailored to the query.

3.3.2. Query Amplifier Branch

Due to intra-object appearance variation, certain query pixels may contain ambiguous information, making it difficult for $\boldsymbol{P}_{i+1}^s$ obtained from the feedback branch to effectively control the activation of the target object in the query. Therefore, we aim to reinforce the category information in the query feature before being activated by $\boldsymbol{P}_{i+1}^s$.

Concretely, given the collected query information $\boldsymbol{P}_i^q$, we first utilize it to stimulate the target object in the query feature, which can be viewed as a self-activation mechanism. The self-activated query feature $\tilde{\boldsymbol{X}}_i^q$ is then inputted into a

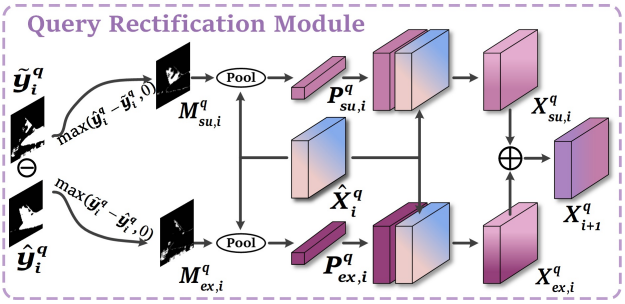

Figure 3. **Illustration of the proposed query rectification module.** Given the $\hat{\boldsymbol{y}}_i^q$ and $\tilde{\boldsymbol{y}}_i^q$, we compare the disparity between them and identify the challenging positive samples and negative samples, respectively, to rectify the query feature.

segmentation network to generate a self-activated prediction $\tilde{\boldsymbol{y}}_i^q$.

This procedure can be expressed as:

$$\tilde{\boldsymbol{X}}_i^q = \mathcal{F}_{1\times1}(\boldsymbol{X}_i^q \oplus \mathcal{O}(\boldsymbol{P}_i^q)) + \boldsymbol{X}_i^q, \tag{6}$$

$$\tilde{\boldsymbol{y}}_i^q = \boldsymbol{Seg}(\tilde{\boldsymbol{X}}_i^q). \tag{7}$$

Furthermore, following (Zhang et al., 2021b), we suggest incorporating deformable self-attention to enhance the representation of the target object in $\tilde{\boldsymbol{X}}_i^q$. As a result, the category information in the query is effectively enhanced.

Subsequently, $\boldsymbol{P}_{i+1}^s$ obtained from (5) is used as the category cue to control the activation of query target objects within it, resulting in a support-activated prediction $\hat{\boldsymbol{y}}_i^q$:

$$\hat{\boldsymbol{X}}_i^q = \mathcal{F}_{1\times1}(\tilde{\boldsymbol{X}}_i^q \oplus \mathcal{O}(\boldsymbol{P}_{i+1}^s)) + \tilde{\boldsymbol{X}}_i^q, \tag{8}$$

$$\hat{\boldsymbol{y}}_i^q = \boldsymbol{Seg}(\hat{\boldsymbol{X}}_i^q). \tag{9}$$

As such, the category information is transferred back to the query. Therefore, we successfully have constructed the category information flow path 'query → support → query'. Please note that $\boldsymbol{Seg}$ in (9) and (7) do not share weights.

3.3.3. Query Rectification Module

Although the target objects in $\hat{\boldsymbol{X}}_i^q$ are significantly activated, some regions are still difficult to segment. To enhance our model's performance, we utilize the difference between the two masks (i.e., $\tilde{\boldsymbol{y}}_i^q$ and $\hat{\boldsymbol{y}}_i^q$) generated from query features before and after support activation to identify these regions for further refinement of $\hat{\boldsymbol{X}}_i^q$, as depicted in Figure 3. Our module is designed assuming that $\hat{\boldsymbol{y}}_i^q$ is more accurate than $\tilde{\boldsymbol{y}}_i^q$, and we demonstrate it in Section 4.4.3. Next, we analyze this difference from two perspectives.

Supplement Prototype. When certain regions are predicted in $\hat{\boldsymbol{y}}_i^q$ but not captured in $\tilde{\boldsymbol{y}}_i^q$, they can be regarded as hard positive samples since $\hat{\boldsymbol{y}}_i^q$ is assumed to be more accurate than $\tilde{\boldsymbol{y}}_i^q$. We extract category information from the query feature using these regions and create a supplement prototype (SuP), which is used to compensate for regions not identified as target objects in $\tilde{\boldsymbol{y}}_i^q$. Here's a summary of the entire process:

$$\boldsymbol{M}_{su,i}^q = Max\{\mathcal{B}(\hat{\boldsymbol{y}}_i^q) - \mathcal{B}(\tilde{\boldsymbol{y}}_i^q), 0\}, \tag{10}$$

$$\boldsymbol{P}_{su,i}^q = \mathcal{F}_{pool}(\hat{\boldsymbol{X}}_i^q \odot \boldsymbol{M}_{su,i}^q), \tag{11}$$

$$\boldsymbol{X}_{su,i}^q = \mathcal{F}_{1\times1}(\hat{\boldsymbol{X}}_i^q \oplus \mathcal{O}(\boldsymbol{P}_{su,i}^q)), \tag{12}$$

where Max ignores negative values, focusing on predicted regions in $\hat{\boldsymbol{y}}_i^q$ but not obtained in $\tilde{\boldsymbol{y}}_i^q$. $\boldsymbol{P}_{su,i}^q \in \mathbb{R}^{1\times1\times C}$ is used to activate the supplement regions in $\hat{\boldsymbol{X}}_i^q$ and obtain the refined feature $\boldsymbol{X}_{su,i}^q \in \mathbb{R}^{H\times W\times C}$.

Exclusion Prototype. On the other hand, certain regions may be identified in $\tilde{\boldsymbol{y}}_i^q$ but not in $\hat{\boldsymbol{y}}_i^q$. These regions are probably hard negative examples. They are often found at the edges of the target objects and can lead to incorrect activation and predictions. To mitigate this issue, we extract an exclusion prototype (ExP) from the query feature using these regions. The ExP is then used to suppress and exclude query regions that are similar to it. This process can be summarized as follows:

$$\boldsymbol{M}_{ex,i}^q = Max\{\mathcal{B}(\tilde{\boldsymbol{y}}_i^q) - \mathcal{B}(\hat{\boldsymbol{y}}_i^q), 0\}, \tag{13}$$

$$\boldsymbol{P}_{ex,i}^q = \mathcal{F}_{pool}(\hat{\boldsymbol{X}}_i^q \odot \boldsymbol{M}_{ex,i}^q), \tag{14}$$

$$\boldsymbol{X}_{ex,i}^q = \mathcal{F}_{1\times1}(\hat{\boldsymbol{X}}_i^q \oplus \mathcal{O}(\boldsymbol{P}_{ex,i}^q)), \tag{15}$$

where $\boldsymbol{M}_{ex,i}^q \in \mathbb{R}^{H\times W\times1}$ indicates the regions that are predicted in $\tilde{\boldsymbol{y}}_i^q$ but do not appear in $\hat{\boldsymbol{y}}_i^q$. $\boldsymbol{X}_{ex,i}^q \in \mathbb{R}^{H\times W\times C}$ is obtained by activating $\boldsymbol{X}_i^q$ with $\boldsymbol{P}_{ex,i}^q \in \mathbb{R}^{1\times1\times C}$.

After combining $\boldsymbol{X}_{su,i}^q$ and $\boldsymbol{X}_{ex,i}^q$, we obtain the rectified query feature. The process is summarized in this equation:

$$\boldsymbol{X}_{i+1}^q = \boldsymbol{X}_{su,i}^q + \boldsymbol{X}_{ex,i}^q, \tag{16}$$

where $\boldsymbol{X}_{i+1}^q$ represents the final rectified query feature of this layer and will be used in the next layer.

3.3.4. Iterative Query Information Feedback

One Feedback and Rectification Layer (FRL) integrates the Query Feedback Branch (QFB), Query Amplifier Branch (QAB), and Query Rectification Module (QRM) to complete a two-way information exchange and query self-rectification. By iteratively executing this process, we can enhance our method further by facilitating bidirectional reciprocative information exchange between query and support, progressively rectifying the query feature, and ultimately achieving superior segmentation results. Assuming we have L FRLs, and for each layer i, we have:

$$\boldsymbol{X}_{i+1}^q, \boldsymbol{X}_{i+1}^s, \hat{\boldsymbol{y}}_i^q, \tilde{\boldsymbol{y}}_i^q = \boldsymbol{FRL}(\boldsymbol{X}_i^q, \boldsymbol{X}_i^s, \boldsymbol{M}^s), \tag{17}$$

Table 1. **Performance comparison on PASCAL-5^i in terms of mIoU under 1-shot and 5-shot settings.** The results of 'Mean' are the averaged class mIoU scores of all four folds. Results in **bold** denote the best performance. '*' denotes that we follow BAM (Lang et al., 2022) and use the ensemble module to remove the impact of base classes.

Backbone	Method	Venue	1-shot					5-shot				
			fold-0	fold-1	fold-2	fold-3	mean	fold-0	fold-1	fold-2	fold-3	mean
VGG-16	PFENet(Tian et al., 2020)	TPAMI'20	56.9	68.2	54.4	52.4	58.0	59.0	69.1	54.8	52.9	59.0
	PMMs(Yang et al., 2020)	ECCV'20	47.1	65.8	50.6	48.5	53.0	50.0	66.5	51.9	47.6	54.0
	HSNet(Hu et al., 2018)	ICCV'21	59.6	65.7	59.6	54.0	59.7	64.9	69.0	64.1	58.6	64.1
	APANet(Chen et al., 2022)	TMM'22	58.0	68.9	57.0	52.2	59.0	59.8	70.0	62.7	57.7	62.6
	NTRENet(Liu et al., 2022b)	CVPR'22	57.7	67.6	57.1	53.7	59.0	60.3	68.0	55.2	57.1	60.2
	DPCN (Liu et al., 2022a)	CVPR'22	58.9	69.1	63.2	55.7	61.7	63.4	70.7	68.1	59.0	65.3
	IFRNet(ours)		**64.9**	**72.4**	**67.6**	**66.2**	**67.8**	**67.0**	**73.9**	**69.0**	**69.2**	**69.8**
	BAM*(Lang et al., 2022)	CVPR'22	63.2	70.8	66.1	57.5	64.4	67.4	73.0	70.6	64.0	68.8
	FECANet*(Liu et al., 2023a)	TMM'23	66.5	68.9	63.6	58.3	64.3	68.6	70.8	66.7	60.7	66.7
	MIANet*(Yang et al., 2023)	CVPR'23	65.4	**73.6**	67.7	61.6	67.1	69.0	**76.1**	**73.2**	69.6	**72.0**
	InPNet*(Luo et al., 2023)	SP'23	61.3	71.6	**69.8**	60.9	65.9	67.9	73.7	72.3	63.5	69.4
	MVPNet*(Wang et al., 2023)	APIN'23	60.6	69.5	65.1	56.3	62.9	65.6	72.8	69.7	64.7	68.2
	HDMNet*(Peng et al., 2023)	CVPR'23	64.8	71.4	67.7	56.4	65.1	68.1	73.1	71.8	64.0	69.3
	IFRNet*(ours)		**67.0**	72.9	68.2	**66.9**	**68.7**	**69.7**	75.4	70.9	**71.2**	71.8
ResNet-50	PMMs(Yang et al., 2020)	ECCV'20	55.2	66.9	52.6	50.7	56.3	56.3	67.3	54.5	51.0	57.3
	PFENet(Tian et al., 2020)	TPAMI'20	61.7	69.5	55.4	56.3	60.8	63.1	70.7	55.8	57.9	61.9
	HSNet(Hu et al., 2018)	ICCV'21	64.3	70.7	60.3	60.5	64.0	70.3	73.2	67.4	67.1	69.5
	IPMT(Liu et al., 2022c)	NeurIPS'22	**72.8**	73.7	59.2	61.6	66.8	73.1	74.7	61.6	63.4	68.2
	DCAMA (Shi et al., 2022)	ECCV'22	67.5	72.3	59.6	59.0	64.6	70.5	73.9	63.7	65.8	68.5
	SCCAN(Xu et al., 2023)	ICCV'23	67.5	72.6	67.2	60.5	67.0	69.9	74.3	**70.1**	66.9	70.3
	MSI (Moon et al., 2023)	ICCV'23	71.0	72.5	63.8	65.9	68.3	73.0	74.2	66.6	70.5	71.1
	IFRNet(ours)		71.4	**73.7**	**67.4**	**70.3**	**70.7**	**71.0**	**74.8**	69.9	**72.7**	**72.1**
	BAM*(Lang et al., 2022)	CVPR'22	69.0	73.6	67.6	61.1	67.8	70.6	75.1	70.8	67.2	70.9
	FECANet*(Liu et al., 2023a)	TMM'23	69.2	72.3	62.4	65.7	67.4	72.9	74.0	65.2	67.8	70.0
	MIANet*(Yang et al., 2023)	CVPR'23	68.5	**75.8**	67.5	63.1	68.7	70.2	**77.4**	70.0	68.8	71.6
	MVPNet*(Wang et al., 2023)	APIN'23	70.0	73.4	67.2	65.2	68.9	70.7	75.6	70.2	68.9	71.4
	InPNet*(Luo et al., 2023)	SP'23	69.3	74.4	68.8	62.3	68.7	71.2	75.5	**74.8**	68.3	72.5
	HDMNet*(Peng et al., 2023)	CVPR'23	71.0	75.4	68.9	62.1	69.4	71.3	76.2	71.3	68.5	71.8
	IFRNet*(ours)		**74.3**	74.2	**69.1**	**70.9**	**72.1**	**75.4**	77.2	71.3	**74.1**	**74.5**

which can be broken down into the following steps:

$$\boldsymbol{P}_i^q = \mathcal{F}_{pool}(\boldsymbol{X}_i^q \odot \mathcal{B}(\boldsymbol{Seg}(\boldsymbol{X}_i^q))), \quad (18)$$

$$\boldsymbol{P}_{i+1}^s, \boldsymbol{X}_{i+1}^s = \boldsymbol{QFB}(\boldsymbol{P}_i^q, \boldsymbol{X}_i^s, \boldsymbol{M}^s), \quad (19)$$

$$\hat{\boldsymbol{X}}_i^q, \hat{\boldsymbol{y}}_i^q, \tilde{\boldsymbol{y}}_i^q = \boldsymbol{QAB}(\boldsymbol{P}_i^q, \boldsymbol{X}_i^q, \boldsymbol{P}_{i+1}^s), \quad (20)$$

$$\boldsymbol{X}_{i+1}^q = \boldsymbol{QRM}(\hat{\boldsymbol{X}}_i^q, \hat{\boldsymbol{y}}_i^q, \tilde{\boldsymbol{y}}_i^q). \quad (21)$$

After L iterations, the final query feature $\boldsymbol{X}_{L+1}^q$ is used to predict the segmentation result $\boldsymbol{y}_{final}^q$.

3.4. Total Loss

Our method generates multiple query predictions from segmentation networks. To ensure the network learns as intended, we calculate three binary cross-entropy losses that supervise the predictions ($\boldsymbol{y}_{final}^q$, $\tilde{\boldsymbol{y}}_i^q$, $\hat{\boldsymbol{y}}_i^q, i \in \{1, 2, ..., L\}$). These losses guide the learning process and ensure the accuracy of the predictions.

$$L = \beta \text{BCE}(\boldsymbol{y}_{final}^q, \boldsymbol{M}^q) + \lambda \sum_i^L \text{BCE}(\tilde{\boldsymbol{y}}_i^q, \boldsymbol{M}^q) + \gamma \sum_i^L \text{BCE}(\hat{\boldsymbol{y}}_i^q, \boldsymbol{M}^q), \quad (22)$$

where β, λ, and γ are used to balance the losses.

4. Experiment

4.1. Datasets and Evaluation Setting

Datasets. To ensure a fair assessment, our model is evaluated on two benchmark datasets for FSS: the PASCAL-5^i dataset (Shaban et al., 2017) and the COCO-20^i dataset (Nguyen & Todorovic, 2019). The PASCAL-5^i is built upon the PASCAL VOC 2012 dataset (Everingham et al., 2010) with additional annotations from SDS (Hariharan et al., 2011), containing 20 categories across four folds. The COCO-20^i dataset, a larger dataset derived from MSCOCO (Lin et al., 2014), consists of 80 categories divided into four folds. Our model is trained on three folds and evaluated on the remaining fold, enabling us to perform cross-validation.

Evaluation Metrics. To assess the effectiveness of the model, we use the class mean intersection over union (mIoU) as the primary metric, consistent with previous methods. We also provide the foreground-background intersection over union (FB-IoU) results in the appendix. This comparison focuses on the target and non-target areas rather than on specific categories, providing a more comprehensive analysis.

4.2. Implementation Details

For a fair comparison with previous works, we use VGG-16 (Simonyan & Zisserman, 2014) and ResNet-50 (He et al., 2016) as encoder backbones. These are initialized with

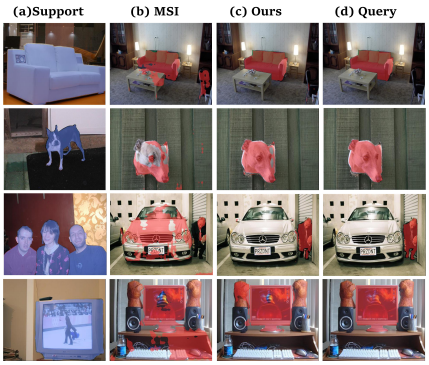

Figure 4. **Comparison between existing method and our method for FSS.** From left to right: support images, MSI predictions, our predictions, and query ground-truth.

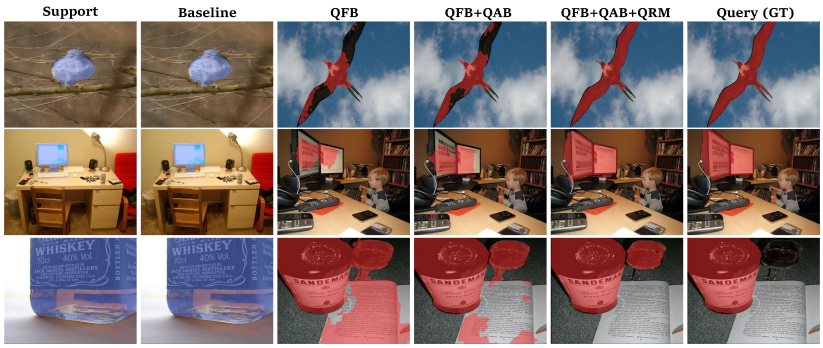

Figure 5. **Visualization of different ablative results.** From left to right: Support images, the results of baseline, the results of only using QFB, the results of using QFB+QAB, the results of using QFB+QAB+QRM (*i.e.*, full model), Ground truth.

weights pretrained on ImageNet and remain fixed during the training process. All experiments are conducted using PyTorch on an NVIDIA RTX 2080 TI GPU for PASCAL-5^i, and four GPUs for COCO-20^i. We augment our training dataset by utilizing a variety of techniques, including random scaling, horizontal flipping, and random rotations between -10 and +10 degrees. Images and masks are randomly cropped to a consistent size of 473×473.

For training, we use the stochastic gradient descent (SGD) optimizer, with a batch size of 4, a learning rate of 0.025, a weight decay of 0.0001, and a momentum value of 0.9. β, λ, and γ are all set as 1.0 for simplicity. The model is trained for 200 epochs on PASCAL-5^i and 50 epochs on COCO-20^i, with a polynomial annealing policy for learning rate reduction, using a power factor of 0.9. During the evaluation, we follow (Yang et al., 2021) and randomly select 1000 support-query pairs from PASCAL-5^i and 4000 pairs from COCO-20^i.

4.3. Comparison with State-of-the-art Methods

4.3.1. Quantitative Analysis

PASCAL-5^i. In Table 1, our IFRNet outperforms all other methods with VGG-16 and ResNet-50 backbones on the PASCAL-5^i dataset. In the 1-shot setting, our method achieves an average mIoU of 67.8% and 70.7% with VGG-16 and ResNet-50 backbones, exceeding previous state-of-the-art by 6.1% and 2.4%, respectively. In the 5-shot setting, it also attains state-of-the-art mIoU scores of 69.8% and 72.1% with VGG-16 and ResNet-50 backbones, outperforming previous state-of-the-art by 4.5% and 1.0%, respectively.

In addition, we use the ensemble model (Lang et al., 2022) to eliminate base class influence for a fair comparison with BAM-based methods. In the 1-shot setting, average mIoU results are 68.7% with VGG-16 and 72.1% with ResNet-50, surpassing previous best performance by 3.6% and 2.7% respectively. In the 5-shot setting, we exceed previous top results by 2.5% with VGG-16 and 2.7% with ResNet-50.

COCO-20^i. Our method also demonstrates impressive performance on the COCO dataset. In Table 2, our method surpasses the previous best methods in both 1-shot and 5-shot settings. In the 1-shot setting, our method shows a significant improvement, achieving 5.1% and 0.2% higher mIoU compared to the previous best methods utilizing the VGG-16 and Resnet-50 backbones, respectively. Furthermore, by eliminating the influence of base classes, our method results in a further outperformance of the previous method by 4.0% and 1.7%, respectively. Similarly, in the 5-shot setting, our method demonstrates a significant advancement, with 5.7% and 0.3% higher mIoU compared to the previous best methods utilizing the VGG-16 and Resnet-50 backbones, respectively. By removing base classes, it has surpassed the previous methods by 3.0% and 2.5%, respectively, further validating the effectiveness of our method.

4.3.2. Qualitative Comparison

In Figure 4, we present the results of our prediction. We observe that our method reduces intra-class diversity, enabling accurate segmentation even with very different support and query (see rows 3 and 4). It is minimally affected by confusing objects and achieves state-of-the-art performance.

4.4. Ablation Study

4.4.1. Effectiveness of FRL

To assess the effectiveness of our FRL, including QFB, QAB, and QRM, we conduct ablation studies on PASCAL-5^i using the 1-shot setting. A baseline model without these modules is initially designed, directly segmenting the target object using the support prototype.

Results presented in Table 4 demonstrate that integrating QFB into the baseline yields a 3.1% mIoU improvement. Moreover, replacing the original support matching process with QAB can further enhance the results to 70.8% mIoU. Additionally, we conduct separate experiments for QRM by decomposing it into QRM-su and QRM-ex, which means

Table 2. **Performance comparison on COCO-20^i in terms of mIoU under 1-shot and 5-shot settings.** The results of 'Mean' are the averaged class mIoU scores of all four folds. Results in **bold** denote the best performance. '*' denotes that we follow BAM (Lang et al., 2022) and use the ensemble module to remove the impact of base classes.

Backbone	Method	Venue	1-shot					5-shot				
			fold-0	fold-1	fold-2	fold-3	mean	fold-0	fold-1	fold-2	fold-3	mean
VGG-16	PFENet(Tian et al., 2020)	TPAMI'20	35.4	38.1	36.8	34.7	36.3	38.2	42.5	41.8	38.9	40.4
	SAGNN (Xie et al., 2021)	CVPR'21	35.0	40.5	37.6	36.0	37.3	37.2	45.2	40.4	40.0	40.7
	APANet(Chen et al., 2022)	TMM'22	35.6	40.0	36.0	37.1	37.2	40.1	48.7	43.3	40.7	43.2
	DPCN (Liu et al., 2022a)	CVPR'22	38.5	43.7	38.2	37.7	39.5	42.7	51.6	45.7	44.6	46.2
	IFRNet(ours)		**41.5**	**44.4**	**43.2**	**44.2**	**43.3**	**49.2**	**52.3**	**47.2**	**46.9**	**48.9**
	BAM*(Lang et al., 2022)	CVPR'22	36.4	47.1	43.3	41.7	42.1	42.9	51.4	48.3	46.6	47.3
	FECANet*(Liu et al., 2023a)	TMM'23	34.1	37.5	35.8	34.1	35.4	39.7	43.6	42.9	39.7	41.5
	InPNet*(Luo et al., 2023)	SP'23	41.4	48.2	44.7	41.9	44.1	47.9	53.4	49.2	48.6	49.8
	MIANet*(Yang et al., 2023)	CVPR'23	40.6	**50.5**	**46.5**	**45.2**	45.7	46.2	**56.1**	**52.3**	**49.5**	**51.0**
	IFRNet*(ours)		**43.8**	49.8	46.4	44.4	**46.1**	**49.9**	53.1	49.0	49.1	50.3
ResNet-50	PFENet (Tian et al., 2020)	TPAMI'20	34.3	33.0	32.3	30.1	32.4	38.5	38.6	38.2	34.3	37.4
	PMMs(Yang et al., 2020)	ECCV'20	29.5	36.8	28.9	27.0	30.6	33.8	42.0	33.0	33.3	35.5
	APANet(Chen et al., 2022)	TMM'22	37.5	43.9	39.7	40.7	40.5	39.8	46.9	43.1	42.2	43.0
	DPCN(Liu et al., 2022a)	CVPR'22	42.0	47.0	43.2	39.7	43.0	46.0	54.9	50.8	47.4	49.8
	HSNet (Hu et al., 2018)	ICCV'21	36.3	43.1	38.7	38.7	39.2	43.3	51.3	48.2	45.0	46.9
	NTRENet(Liu et al., 2022b)	CVPR'22	36.8	42.6	39.9	37.9	39.3	38.2	44.1	40.4	38.4	40.3
	IPMT(Liu et al., 2022c)	NeurIPS'22	41.4	45.1	45.6	40.0	43.0	43.5	49.7	48.7	47.9	47.5
	DCAMA(Shi et al., 2022)	ECCV'22	41.9	45.1	44.4	41.7	43.3	45.9	50.5	50.7	46.0	48.3
	SCCAN(Xu et al., 2023)	ICCV'23	39.5	49.3	47.3	44.3	45.1	45.7	56.4	**56.5**	50.7	**52.3**
	MSI (Moon et al., 2023)	ICCV'23	42.4	49.2	**49.4**	46.1	46.8	47.1	54.9	54.1	51.9	52.0
	IFRNet(ours)		**42.6**	**50.9**	47.9	**46.5**	**47.0**	**48.9**	**57.3**	50.3	**52.7**	**52.3**
	BAM*(Lang et al., 2022)	CVPR'22	43.4	50.6	47.5	43.4	46.2	49.3	54.2	51.6	49.6	51.2
	FECANet*(Liu et al., 2023a)	TMM'23	38.5	44.6	42.6	40.7	41.6	44.6	51.5	48.4	45.8	47.6
	InPNet*(Luo et al., 2023)	SP'23	**44.8**	50.4	**49.7**	44.9	47.5	50.9	55.8	52.9	50.1	52.4
	MIANet*(Yang et al., 2023)	CVPR'23	42.5	52.9	47.8	47.4	47.6	45.8	58.2	51.3	51.9	51.7
	MVPNet*(Wang et al., 2023)	APIN'23	42.6	52.9	47.4	43.8	46.7	48.7	56.3	52.8	48.6	51.6
	IFRNet*(ours)		44.4	**54.1**	48.8	46.5	**48.4**	**51.4**	**59.4**	**53.2**	**52.2**	**54.1**

using the supplement prototype and the exclusion prototype, respectively. Including QRM-su contributes another 0.7% mIoU improvement by helping segment challenging positive pixels. Similarly, incorporating QRM-ex results in a 0.8% mIoU improvement by suppressing difficult negative samples. The combination of QFB, QAB, and QRM achieves an outstanding 72.1% mIoU, which gains a larger margin over the baseline model. These findings confirm the effectiveness of our proposed QFB, QAB, and QRM.

4.4.2. Ablation on Different Numbers of FRL

In this study, we investigate the impact of varying the number of FRL layers, ranging from 1 to 5, on the model's performance. The results in Table 5 indicate that the model's performance incrementally improves with an increase in the number of layers. We can observe that increasing the number of layers from 1 to 2 results in a significant performance improvement of 1.3% mIoU. However, adding more than 2 layers does not bring significant gains. Therefore, we use **2** layers for our final model due to the trade-off between performance and computational costs.

4.4.3. Ablation on Different Predictions in FRL

In Section 3.3.3, we speculate that $\hat{\boldsymbol{y}}_i^q$ is more accurate than $\tilde{\boldsymbol{y}}_i^q$. To demonstrate this assumption, we evaluate the mIoU of both $\hat{\boldsymbol{y}}_i^q$ and $\tilde{\boldsymbol{y}}_i^q$ in different layers, and the results are presented in Table 7. The mIoU results obtained by $\hat{\boldsymbol{y}}_i^q$ consistently outperform those of $\tilde{\boldsymbol{y}}_i^q$ across all layers, thus confirming the validity of our hypothesis.

4.4.4. Effectiveness of QRMP

We propose a new technique called Query-Related Mask Pooling (QRMP) to replace the traditional mask average pooling (MAP) in QFB. To evaluate its effectiveness, we conduct ablation studies on PASCAL-5^i in the 1-shot setting using QRMP and MAP respectively in QFB, while keeping the remaining operations consistent with our full model. Results in Table 6 indicate that using QRMP outperforms using traditional MAP in all folds, with a significant improvement of 1.0% mIoU in fold 3.

4.4.5. Qualitative Comparison

In Figure 5, we present additional visual results to showcase the effectiveness of our QFB, QAB, and QRM modules. Column 2 displays the baseline results, while Column 3 shows the results with QFB. A comparison of these two columns clearly indicates that QFB improves segmentation. Column 4 demonstrates that the addition of QAB further enhances the results. The combined use of QFB, QAB, and QRM in Column 5 results in even more accurate predictions by leveraging the perception of challenging regions. These results validate the effectiveness of our proposed modules.

4.4.6. Prototype Comparison with IPMT

In this section, we use the Euclidean distance to quantitatively calculate the distances between the original support

Table 3. **Intra-class diversity measured by Euclidean distances on PASCAL-5^i.** The results of 'Mean' are the averaged class mIoU scores of all classes. We compare the distances between the original support prototypes and the query prototype (D_{ori}), the distances between the query-related support prototypes and the query prototype (D_{our}), and the distance (D_{ipmt}) between the intermediate prototypes and the query prototype in (Liu et al., 2022c). $*$: Please note that we retest the IPMT and normalize prototypes before measurement to ensure fair comparison. Therefore, the reported results may differ from the original results in (Liu et al., 2022c).

class	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16	c17	c18	c19	c20	Mean
D_{ori}	6.52	4.35	6.98	3.10	3.35	7.88	4.44	4.16	4.03	3.51	5.58	4.37	5.28	3.67	3.94	7.82	4.01	5.48	3.87	4.76	4.86
$D_{ipmt}{}^*$	5.86	2.66	3.77	3.77	4.04	6.32	3.02	4.17	3.27	3.94	4.24	2.81	4.91	3.52	3.58	4.46	3.20	5.19	3.51	4.59	4.04
D_{our}	3.56	2.38	3.70	2.19	3.33	2.76	2.29	3.03	3.16	3.14	3.09	2.54	2.86	3.29	2.98	2.80	2.55	2.80	2.81	4.54	2.99

Table 4. **Ablation study on effectiveness of different modules in FRL.** The results of 'mIoU' are the averaged class mIoU scores of all four folds on the PASCAL-5^i dataset.

QFB	QAB	QRM-su	QRM-ex	mIoU
				66.8
✓				69.9
✓	✓			70.8
✓	✓	✓		71.5
✓	✓		✓	71.6
✓	✓	✓	✓	**72.1**

Table 5. **Performance comparison of varying the number of FRL layers.**

Layers	1	2	3	4	5
mIoU	70.8	72.1	72.5	72.8	73.0
Pars.(M)	10.6	16.3	22.0	27.7	33.4
GFlops	290.0	310.5	322.5	334.5	350.8

Table 6. **Ablation study on the effectiveness of Query-related Mask Pooling.**

	Fold-0	Fold-1	Fold-2	Fold-3	mIoU
MAP	73.8	74.1	68.5	69.9	71.5
QRMP	**74.3**	**74.2**	**69.1**	**70.9**	**72.1**

Table 7. **Performance comparison of different predictions in FRL.**

Layers	$\tilde{\boldsymbol{y}}_i^q$	$\hat{\boldsymbol{y}}_i^q$	Δ
1	55.9	**62.0**	6.1
2	64.4	**69.7**	5.3

prototypes and the query prototype (D_{ori}) and the distances between the query-related support prototypes and the query prototype (D_{our}), and compare it with the distance (D_{ipmt}) between the intermediate prototypes and the query prototype in (Liu et al., 2022c). From Table 3, we observe that D_{our} is smaller than both D_{ori} and D_{ipmt} for all classes, indicating that our query-related support prototype is more similar to the query prototype than both the original support prototype and the intermediate prototype of IPMT. This further strengthens the effectiveness of our method in reducing the intra-class diversity between the support and the query.

5. Conclusion

In this paper, we propose IFRNet to address the intra-class diversity problem in FSS by introducing information feedback from cybernetics. Our method utilizes QFB to feed category information from query back to support, thereby creating a query-related support prototype. The QAB enhances the target objects through self-activation and support-activation, facilitating bidirectional information exchange. QRM uses prediction differences for self-rectification of the query feature. Combining QFB, QAB, and QRM into FRL and iteratively applying it, we establish a bidirectional reciprocative flow of category information, leading to progressive self-rectification of the query feature. Experiments on two benchmark datasets confirm the superior performance of our method.

Acknowledgment

This work is supported in part by the National Key R&D Program of China under Grant 2022ZD0119004; the National Natural Science Foundation of China under Grants 62071388, 62136007, U21B2048, and 62322605; the Key R&D Program of Shaanxi Province under Grant 2023-YBGY-224; the MBZUAI-WIS Joint Program for AI Research under Grants WIS P008 and P009; the Institute of Artificial Intelligence, Hefei Comprehensive National Science Center Project under Grant 21KT008; the Outstanding Doctoral Dissertation Cultivation Fund of the School of Automation, Northwestern Polytechnical University.

Impact Statement

Positive Impact In data-sensitive fields, this work has the potential to significantly enhance the segmentation and identification of target objects using minimal annotation data. By leveraging advanced algorithms, it enables more accurate and efficient processing of data, leading to improved outcomes in various applications.

Negative Impact However, there are potential negative consequences to consider. The automation enabled by this technology could lead to the displacement of traditional human tasks. For instance, labeling tasks for large datasets that were previously performed by humans may now be automated through algorithms. This shift could impact relevant practitioners by altering the nature of their employment opportunities.

Limitation We have observed that incorporating more than 2 layers does not lead to substantial performance improvements. This suggests that our approach may not enhance model performance through layer stacking, unlike the typical Vision Transformer (ViT) architecture.

References

Task-aware adaptive attention learning for few-shot semantic segmentation. *Neurocomputing*, 494:104–115, 2022. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2022.04.089.

Chen, J., Gao, B.-B., Lu, Z., Xue, J.-H., Wang, C., and Liao, Q. Apanet: Adaptive prototypes alignment network for few-shot semantic segmentation. *IEEE TMM*, 2022.

Dong, N. and Xing, E. P. Few-shot semantic segmentation with prototype learning. In *BMVC*, volume 3, 2018.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

Fan, Q., Pei, W., Tai, Y.-W., and Tang, C.-K. Self-support few-shot semantic segmentation. In *ECCV*, pp. 701–719. Springer, 2022a.

Fan, Q., Pei, W., Tai, Y.-W., and Tang, C.-K. Self-support few-shot semantic segmentation. 2022b.

Gao, G., Fang, Z., Han, C., Wei, Y., Liu, C. H., and Yan, S. Drnet: Double recalibration network for few-shot semantic segmentation. *IEEE TIP*, 31:6733–6746, 2022.

Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., and Malik, J. Semantic contours from inverse detectors. In *ICCV*, pp. 991–998. IEEE, 2011.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Hong, S., Cho, S., Nam, J., Lin, S., and Kim, S. Cost aggregation with 4d convolutional swin transformer for few-shot segmentation. In *ECCV*, pp. 108–126. Springer, 2022.

Hu, J., Shen, L., and Sun, G. Squeeze-and-excitation networks. In *CVPR*, pp. 7132–7141, 2018.

Hu, T., Yang, P., Zhang, C., Yu, G., Mu, Y., and Snoek, C. G. Attention-based multi-context guiding for few-shot semantic segmentation. In *AAAI*, volume 33, pp. 8441–8448, 2019.

Lang, C., Cheng, G., Tu, B., and Han, J. Learning what not to segment: A new perspective on few-shot segmentation. In *IEEE TPAMI*, pp. 8057–8067, 2022.

Li, G., Jampani, V., Sevilla-Lara, L., Sun, D., Kim, J., and Kim, J. Adaptive prototype learning and allocation for few-shot segmentation. In *CVPR*, pp. 8334–8343, 2021.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *ECCV*, pp. 740–755. Springer, 2014.

Liu, H., Peng, P., Chen, T., Wang, Q., Yao, Y., and Hua, X.-S. Fecanet: Boosting few-shot semantic segmentation with feature-enhanced context-aware network. *IEEE TMM*, pp. 1–13, 2023a. doi: 10.1109/TMM.2023.3238521.

Liu, J., Bao, Y., Xie, G.-S., Xiong, H., Sonke, J.-J., and Gavves, E. Dynamic prototype convolution network for few-shot semantic segmentation. In *CVPR*, pp. 11553–11562, 2022a.

Liu, N., Nan, K., Zhao, W., Liu, Y., Yao, X., Khan, S., Cholakkal, H., Anwer, R. M., Han, J., and Khan, F. S. Multi-grained temporal prototype learning for few-shot video object segmentation. In *ICCV*, pp. 18862–18871, 2023b.

Liu, W., Zhang, C., Lin, G., and Liu, F. Crnet: Cross-reference networks for few-shot segmentation. In *CVPR*, pp. 4165–4173, 2020a.

Liu, Y., Zhang, X., Zhang, S., and He, X. Part-aware prototype network for few-shot semantic segmentation. In *ECCV*, pp. 142–158. Springer, 2020b.

Liu, Y., Liu, N., Cao, Q., Yao, X., Han, J., and Shao, L. Learning non-target knowledge for few-shot semantic segmentation. In *CVPR*, pp. 11573–11582, 2022b.

Liu, Y., Liu, N., Yao, X., and Han, J. Intermediate prototype mining transformer for few-shot semantic segmentation. *NeurIPS*, 35:38020–38031, 2022c.

Lu, X., Diao, W., Mao, Y., Li, J., Wang, P., Sun, X., and Fu, K. Breaking immutable: Information-coupled prototype elaboration for few-shot object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 1844–1852, 2023.

Lu, Z., He, S., Zhu, X., Zhang, L., Song, Y.-Z., and Xiang, T. Simpler is better: Few-shot semantic segmentation with classifier weight transformer. In *ICCV*, pp. 8741–8750, 2021.

Luo, X., Duan, Z., and Zhang, T. Intermediate prototype network for few-shot segmentation. *Signal Processing*, 203:108811, 2023.

Min, J., Kang, D., and Cho, M. Hypercorrelation squeeze for few-shot segmentation. In *ICCV*, 2021.

Moon, S., Sohn, S. S., Zhou, H., Yoon, S., Pavlovic, V., Khan, M. H., and Kapadia, M. Msi: Maximize support-set information for few-shot segmentation. In *ICCV*, pp. 19266–19276, October 2023.

Nguyen, K. and Todorovic, S. Feature weighting and boosting for few-shot segmentation. In *ICCV*, pp. 622–631, 2019.

Okazawa, A. Interclass prototype relation for few-shot segmentation. In *ECCV*, pp. 362–378. Springer, 2022.

Peng, B., Tian, Z., Wu, X., Wang, C., Liu, S., Su, J., and Jia, J. Hierarchical dense correlation distillation for few-shot segmentation. In *CVPR*, pp. 23641–23651, 2023.

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

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

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

Siam, M., Oreshkin, B. N., and Jagersand, M. Amp: Adaptive masked proxies for few-shot segmentation. In *ICCV*, pp. 5249–5258, 2019.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Tian, Z., Zhao, H., Shu, M., Yang, Z., Li, R., and Jia, J. Prior guided feature enrichment network for few-shot segmentation. *IEEE TPAMI*, (01):1–1, 2020.

Wang, H., Zhang, X., Hu, Y., Yang, Y., Cao, X., and Zhen, X. Few-shot semantic segmentation with democratic attention networks. In *ECCV*, pp. 730–746. Springer, 2020.

Wang, H., Cao, G., and Cao, W. A novel inference paradigm based on multi-view prototypes for one-shot semantic segmentation. pp. 1–16, 2023.

Wang, K., Liew, J. H., Zou, Y., Zhou, D., and Feng, J. Panet: Few-shot image semantic segmentation with prototype alignment. In *ICCV*, pp. 9197–9206, 2019.

Wu, Z., Shi, X., Lin, G., and Cai, J. Learning meta-class memory for few-shot semantic segmentation. In *ICCV*, pp. 517–526, 2021.

Xie, G.-S., Liu, J., Xiong, H., and Shao, L. Scale-aware graph neural network for few-shot semantic segmentation. In *CVPR*, pp. 5475–5484, 2021.

Xu, Q., Zhao, W., Lin, G., and Long, C. Self-calibrated cross attention network for few-shot segmentation. In *ICCV*, pp. 655–665, 2023.

Yang, B., Liu, C., Li, B., Jiao, J., and Ye, Q. Prototype mixture models for few-shot semantic segmentation. In *ECCV*, pp. 763–778. Springer, 2020.

Yang, L., Zhuo, W., Qi, L., Shi, Y., and Gao, Y. Mining latent classes for few-shot segmentation. In *ICCV*, pp. 8721–8730, 2021.

Yang, Y., Chen, Q., Feng, Y., and Huang, T. Mianet: Aggregating unbiased instance and general information for few-shot semantic segmentation. In *CVPR*, pp. 7131–7140, 2023.

Zhang, B., Xiao, J., and Qin, T. Self-guided and cross-guided learning for few-shot segmentation. In *CVPR*, pp. 8312–8321, 2021a.

Zhang, C., Lin, G., Liu, F., Yao, R., and Shen, C. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *CVPR*, pp. 5217–5226, 2019.

Zhang, G., Kang, G., Yang, Y., and Wei, Y. Few-shot segmentation via cycle-consistent transformer. *NeurIPS*, 34:21984–21996, 2021b.

Table 8. **Performance comparison of average FB-IoU results on PASCAL-5^i.** Δ denotes the increments over 1-shot results. '*' denotes that we follow BAM (Lang et al., 2022) and use the ensemble module to remove the impact of base classes.

Backbone	Method	Venue	1-shot	5-shot	Δ
VGG-16	PFENet	TPAMI'20	72.0	72.3	0.3
	HSNet	ICCV'21	73.4	76.6	3.2
	NTRENet	CVPR'22	73.1	74.2	1.1
	MVPNet	APIN'23	79.3	80.2	0.9
	FECANet	TMM'23	78.7	80.7	2.0
	IFRNet(ours)		**80.8**	**82.4**	1.6
	BAM*	CVPR'22	77.2	81.1	3.9
	MVPNet*	APIN'23	81.4	82.8	1.4
	IFRNet*(ours)		**83.1**	**83.8**	0.7
ResNet-50	PFENet	TPAMI'20	73.3	73.9	0.6
	HSNet	ICCV'21	76.7	80.6	3.9
	NTRENet	CVPR'22	77.0	78.1	1.1
	MVPNet	APIN'23	79.3	80.2	0.9
	IFRNet(ours)		**81.7**	**82.8**	1.1
	BAM*	CVPR'22	79.7	82.2	2.5
	MVPNet*	APIN'23	81.4	82.8	1.4
	HDMNet*	CVPR'23	72.2	77.7	5.5
	IFRNet*(ours)		**82.2**	**84.6**	2.4

A. Comparison on FB-IoU

As an additional analysis, we compared our method with previous approaches using FB-IoU. Table 8 shows that our method outperforms the previous best results by 2.4% (1-shot) and 2.6% (5-shot) with the ResNet-50 backbone. Furthermore, by incorporating the ensemble module of BAM (Lang et al., 2022), our method surpasses the previous best results by 10.0% and 6.9%, respectively. Using the VGG-16 backbone, we outperform all previous approaches and achieve exceptional results (*i.e.*, 80.8% and 83.1%) in the 1-shot setting, surpassing previous top results by 2.1% and 1.7%, respectively. In the 5-shot setting, we exceed previous results by 2.7% and 1.0%, respectively. These results further confirm the effectiveness of our method.

B. Qualitative prototype comparsion

In this section, we qualitatively visualize the distribution of original support prototypes (red points), query-related support prototypes (green points), and the query prototype (blue point) in Figure 6. We observe that our query-related support prototypes are closer to the query prototype than the original support prototypes in the feature space. Especially in the right subfigure, we can observe that the query prototype is initially positioned at a considerable distance from the support prototypes, suggesting a notable intra-class difference between them. However, our query-related support prototypes successfully close this gap, emphasizing the reduction of intra-class diversity and narrowing the category information disparity between query and support images.

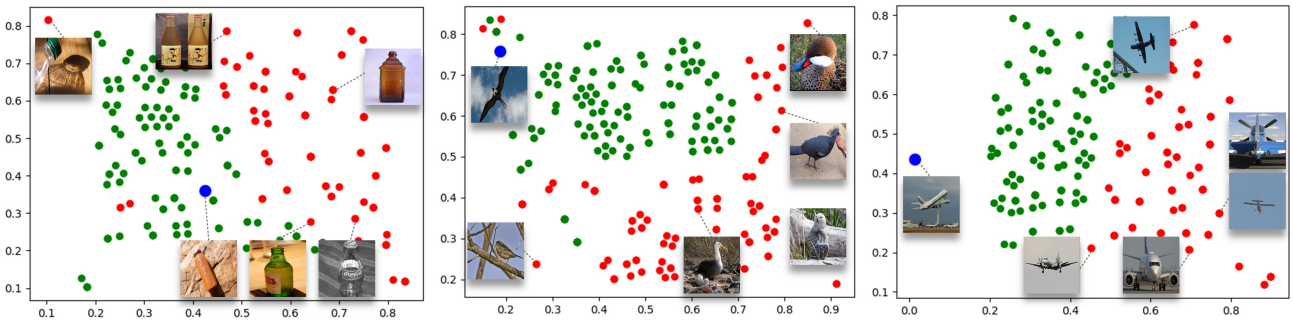

Figure 6. **t-SNE visualization of the prototype distribution.** Our query-related support prototypes are closer to the query prototype than the original support prototypes.

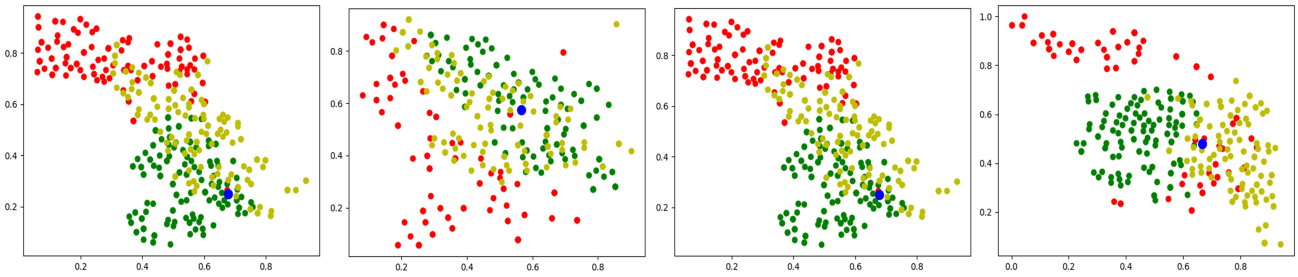

Figure 7. **t-SNE visualization of the prototype distribution.** Our query-related support prototypes are closer to the query prototype than both the intermediate prototypes and the original support prototypes.

C. Qualitative comparison with IPMT

To qualitatively illustrate the superiority of our method over IPMT (Liu et al., 2022c), we incorporate the distribution of the intermediate prototypes (yellow points) and combine it with the original support prototypes (red points), our query-related support prototypes (green points), and the query prototype (blue point) in Figure 7. Upon observation, we notice that our query-related support prototypes, in contrast to intermediate prototypes, are closer to the query prototype and tend to cluster around it. Meanwhile, the position of intermediate prototypes appears more irregular.

D. Additional Qualitative Results

We give more qualitative results in Figure 8 to show the good performance of our IFRNet.

Figure 8. **More qualitative results of our IFRNet.**