CGPRNET: CROSS-SCALE GRAPH PROTOTYPE REFINEMENT FOR ROBUST FEW-SHOT SEMANTIC SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Few-shot semantic segmentation (FSS) learns to segment novel classes with only a few annotated support images. Despite the promising results of recent methods, there remain three critical limitations: (1) current FSS methods do not explicitly model the cross-scale semantic dependencies between support and query; (2) they generate prototypes by simple embedding statistics while disregarding the progressive refinement ability of graph; (3) they fuse the multi-scale information in a principled manner. To address these challenges, We propose CGPR, a framework that jointly integrates query-conditioned graph prototype refinement, adaptive multi-scale matching, and dual-branch feature encoding to improve prototype quality, feature representation, and robustness to cross-scale variation. Concretely, we design a Cross-Attention Graph Prototype Generator to iteratively refine the prototypes by aggregating the cross-scale support-query relationships, and a Multi-Scale Prototype Matching module to compute the similarity at different receptive fields in an adaptive way. Additionally, we introduce a dual-branch feature encoder with frozen generic features and trainable task-specific features to prevent overfitting while maintaining discriminative power. We conduct extensive experiments on PASCAL-5^i, COCO-20^i, and FSS-1000 datasets to show the superiority of CGPRNet over the previous state-of-the-art methods. Under both 1-shot and 5-shot settings, our CGPRNet obtains better mIoU, FB-IoU and Dice scores on these three benchmarks, demonstrating the effectiveness of cross-scale and query-conditioned prototype refinement learning for robust few-shot segmentation.

1 INTRODUCTION

Semantic segmentation has achieved impressive success by using fully supervised deep learning methods Chen et al. (2014); Eigen & Fergus (2015); Eigen et al. (2014). However, the practical applications of segmentation models still suffer from the need to collect large-scale pixel-level annotations for every object category, which is often prohibitively expensive.Most existing FSS methods Tian et al. (2020); Wang et al. (2019); Xie et al. (2022) use single-scale features or fuse multiple scales naively through concatenation or averaging (Fig. 1(a)), leaving cross-scale semantic relations unmodeled. Coarse scales encode global object context but lose local detail, while fine scales preserve detail but lack context—complementary cues that are rarely combined in a principled way, despite the large scale variation between support and query objects in FSS. Such fixed-weight or concatenation-based fusion Tian et al. (2020); Li et al. (2025); Zhang et al. (2024) also cannot adaptively select the most informative scale for each spatial location, which matters when objects of different sizes appear in the same scene.

Feature Overfitting to Base Classes. FSS models are commonly trained with a fully trainable backbone, which overfits base classes and generalizes poorly to novel ones Tian et al. (2020); Yang et al. (2020b); Zhang et al. (2019). Freezing the backbone preserves generic pre-trained knowledge but leaves it insufficiently adapted to the few-shot task. Finding the right trade-off between knowledge preservation and task-specific tuning remains an open question.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To address these issues, we present CGPR (Fig. 1(b)), a few-shot segmentation framework built on cross-scale prototype refinement and matching. (1) Unlike methods with fixed prototypes, our Cross-Attention Graph Prototype Generator (CAGPG) iteratively refines support prototypes through query-conditioned graph reasoning, enabling adaptive prototype refinement for support–query interactions, complementing the feature-level improvements provided by the dual-branch encoder and adaptive multi-scale matching. (2) Our **Multi-Scale Prototype Matching (MSPM)** module computes similarity maps at 3×3, 5×5, and 7×7 receptive fields and uses end-to-end learnable gating to assign adaptive weights per scale. (3) A **two-branch architecture** retains generic pre-trained features in a frozen branch while learning task-specific features in a trainable branch, fusing them via a learned gate to mitigate overfitting without sacrificing discriminative power.

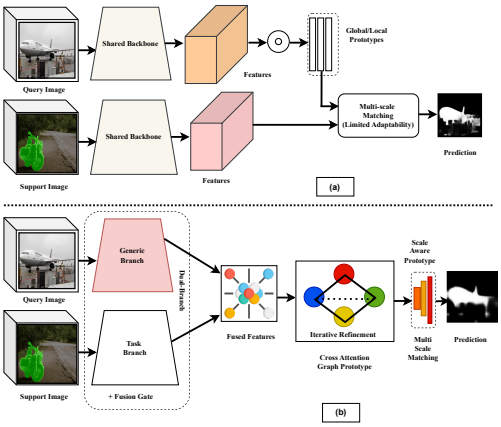

Figure 1: (a) Existing methods match static support prototypes with query features via naive multi-scale fusion. (b) Our CGPRNet uses fusion gates, cross-attention graph prototypes, and scale-aware iterative refinement.

CGPR achieves state-of-the-art performance on FS-1000, PASCAL-5^i, and COCO, with the largest gains in the 1-shot and 5-shot settings where prototype quality matters most. Ablations validate each component and highlight the importance of cross-scale reasoning for FSS. For fair comparison, all models use standard backbones (VGG-16, ResNet-50, ResNet-101). Our main contributions can be summarized as follows:

- Cross-Attention Graph Prototype Generator (CAGPG) – prototypes are treated as nodes of a fully connected graph and refined over T iterations by alternating spatially-normalised cross-attention to the query with inter-node message passing, replacing static masked average pooling. A background node is included, which both models non-target regions explicitly and keeps the attention non-degenerate in the 1-shot setting.
- Multi-Scale Prototype Matching (MSPM) – Calculates adaptive similarity across multiple receptive fields (3×3, 5×5, 7×7) with learnable scale-specific weights for handling multi-scale objects.
- Dual-Branch Feature Encoder – Merges frozen generic features with trainable task-specific features through learned fusion gates to avoid overfitting while maintaining discriminative power.
- State-of-the-art Performance – Comprehensive experiments on PASCAL-5^i, COCO-20^i, and FSS-1000 demonstrate superior performance across all settings.

2 Proposed Method

2.1 Problem Definition and Motivation

Few-shot semantic segmentation (FSS) seeks to segment images with respect to novel object classes in the query images given only K annotated support samples, without re-training the model. In line with the common protocols, we separate the dataset into disjoint base classes C_{base} for training, and novel classes C_{novel} for testing, where $C_{\text{base}} \cap C_{\text{novel}} = \emptyset$. During training, the model learns from episodes, each containing K support pairs $S = \{(I_k^s, M_k^s)\}_{k=1}^K$ and one query pair (I^q, M^q) of a randomly sampled class from C_{base}. At test time, the trained model segments query images using support samples from C_{novel}.

Existing prototype-based methods have two main limitations: (1) Static prototypes are extracted by simply using masked average pooling. These static prototypes are not query-specific, and do not generalize well across query appearance variations. (2) Single-scale matching is insufficient

to model multi-scale objects and different spatial contexts. These problems are exacerbated in the 1-shot setting, where only a single piece of support information is available. To overcome these limitations, we introduce Cross-Attention Graph Prototype Refinement (CGPR) Network to refine the support prototypes in a dynamic way. In particular, the prototypes are refined by iterative interaction between the query and prototypes, and multi-scale matching is conducted. Our central observation is that prototypes should not be static embeddings but rather should be progressively updated by attending to the query-specific information.

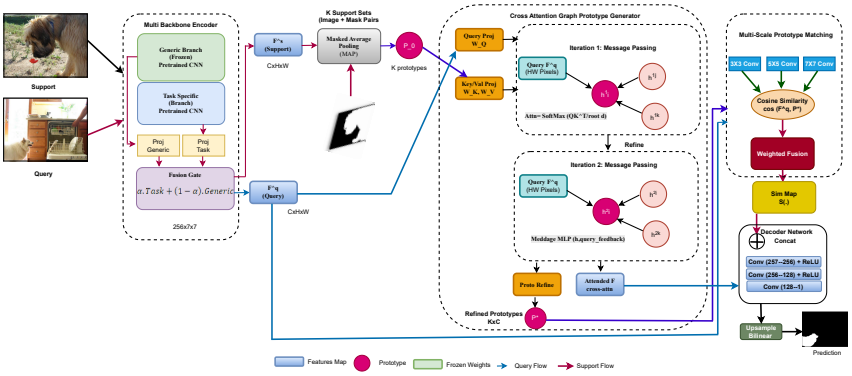

Figure 2: CGPRNet Architecture.

2.2 Overall Architecture

Fig. 2. Overview of our CGPR Network. It can be divided into four main modules: (1) A two-branch feature encoder that generates complementary generic and task-specific features with an adaptive fusion unit; (2) A cross-attention graph prototype generator that refines the support prototypes with a T-step query-prototype interaction; (3) A multi-scale matching module that measures reliable similarity maps with various receptive fields; (4) A light-weight decoder that integrates the attended features and similarity maps to generate the final prediction. Given query image $I^q \in \mathbb{R}^{H\times W\times 3}$ and support image-mask pairs $\{(I_k^s, M_k^s)\}_{k=1}^K$, we first extract features using the dual-branch encoder (Section 3.3). The cross-attention graph module then generates refined prototypes $\{\tilde{P}_k^s\}_{k=1}^K$ by iteratively incorporating query feedback (Section 3.4). Multi-scale matching computes the final similarity map S^q (Section 3.5), which is fed to the decoder along with attended query features to produce prediction $\hat{M}^q$ (Section 3.6).

2.3 Dual-Branch Feature Encoder

Pre-trained backbones provide a rich pool of generic visual knowledge, which may lack task-specific discriminative cues, while the task-specific branch is prone to overfitting to the base classes. We address the problem with a two-branch architecture that adaptively fuses the complementary representations. We use a single ImageNet-pretrained backbone E, kept entirely frozen and shared across both branches, so no second backbone is trained. Given an input image I, we extract shared features $F = E(I) \in \mathbb{R}^{h\times w\times d}$, with $d = 2048$ for ResNet and $d = 512$ for VGG-16. The generic branch forwards these frozen features directly, while the task-specific branch applies a lightweight trainable adapter A_{task} (two 3×3 convolutions):

$$F_{\text{gen}} = F, \qquad F_{\text{task}} = A_{\text{task}}(F).$$

We project both to a unified dimension C (default 256):

$$F'_{\text{gen}} = \phi_{\text{gen}}(F_{\text{gen}}), \quad F'_{\text{task}} = \phi_{\text{task}}(F_{\text{task}}),$$

where $\phi(\cdot) = \text{ReLU}(\text{BN}(\text{Conv}_{1\times 1}(\cdot)))$ and $F'_{\text{gen}}, F'_{\text{task}} \in \mathbb{R}^{h\times w\times C}$. Rather than static fusion (for example concatenation or averaging), we introduce a learned gating mechanism to dynamically balance the two branches:

$$G = \sigma\big(\psi([F'_{\text{gen}}; F'_{\text{task}}])\big), \quad F = G \odot F'_{\text{task}} + (1 - G) \odot F'_{\text{gen}},$$

where $\psi(\cdot)$ is a 1×1 convolution, $[\cdot;\cdot]$ denotes channel concatenation, σ is sigmoid, $\odot$ is element-wise multiplication, and $G \in \mathbb{R}^{h \times w \times C}$ is the spatial gating map. This allows the network to emphasize task-specific features in discriminative regions while relying on generic features in ambiguous areas. We apply this encoder to both query and support images to obtain F^q and $\{F^s_k\}_{k=1}^K$.

2.4 Cross Attention Graph Prototype Generator

Conventional masked average pooling (MAP) operation yields fixed prototypes that lack query-aware context. To address this issue, we develop an iterative refinement mechanism, inspired by graph neural networks and cross-attention, where support prototypes are recursively updated by attending to the query features. For each support sample k, we extract an initial prototype via MAP:

$$P^s_k = \text{MAP}(F^s_k, M^s_k) = \frac{\sum_{i,j} F^s_k(i,j) \odot \bar{M}^s_k(i,j)}{\sum_{i,j} \bar{M}^s_k(i,j)},$$

where $\bar{M}^s_k \in \mathbb{R}^{h \times w}$ is the support mask resized to feature resolution, and $P^s_k \in \mathbb{R}^C$ is the initial prototype. We refine prototypes through T iterations (default $T = 2$) of cross-attention with query features. Let $Q = \text{Reshape}(F^q) \in \mathbb{R}^{N \times C}$ where $N = h \times w$. At iteration $t \in \{1, \ldots, T\}$

Background node. We additionally construct a background prototype by applying masked average pooling over the complement of the support mask and averaging across the K support images:

$$P^s_{\text{bg}} = \frac{1}{K} \sum_{k=1}^{K} \frac{\sum_{i,j}(1 - \bar{M}^s_k(i,j))F^s_k(i,j)}{\sum_{i,j}(1 - \bar{M}^s_k(i,j))} \in \mathbb{R}^C. \tag{1}$$

The background prototype is included as an additional graph node, resulting in $K_{\text{eff}} = K + 1$. This explicitly models non-target regions and prevents degenerate prototype-axis softmax in the 1-shot setting.

Multi-shot handling. Each support image contributes one prototype node P^s_k, computed independently from (I^s_k, M^s_k). Together with the background node, the graph contains $K_{\text{eff}} = K + 1$ nodes, which are jointly refined for T iterations. Inter-shot interaction occurs through graph message passing, where each prototype aggregates information from all other nodes, and through the final similarity fusion, which combines the refined per-prototype similarity maps. **Step 1: Prototype–Query Affinity.** For each prototype node k, we compute its affinity with every query pixel:

$$e^{(t)}_{k,i} = \frac{\big(P^{(s,(t))}_k W_Q\big) \cdot \big(Q(i) W_K\big)^\top}{\sqrt{C}}, \qquad e^{(t)} \in \mathbb{R}^{K_{\text{eff}} \times N}, \tag{2}$$

where $N = h \times w$ and $K_{\text{eff}} = K + 1$ includes the background node.

Step 2: Query Feedback Aggregation. The aggregation weights are normalized over the *spatial* dimension so that each prototype receives a convex combination of query features:

$$A^{(t)}_{k,i} = \frac{\exp\big(e^{(t)}_{k,i}\big)}{\sum_{i'=1}^{N} \exp\big(e^{(t)}_{k,i'}\big)}, \tag{3}$$

$$M^{(t)}_k = \sum_{i=1}^{N} A^{(t)}_{k,i} \big(Q(i)\, W_V\big) \ \in \mathbb{R}^C. \tag{4}$$

Normalizing over N rather than over K is essential. Under the 1-shot protocol, a prototype-axis softmax is identically one, reducing the module to an unweighted global pooling of the query feature map and producing a spatially constant attended feature map. Spatial normalization preserves a genuine attention distribution for any $K \geq 1$, while the convex combination keeps $\|M_k\|$ independent of the feature-map resolution.

Step 3: Graph Message Passing. Prototype nodes additionally exchange messages along fully connected edges, with self-loops excluded:

$$G_{k,l}^{(t)} = \frac{\exp\big(P_k^{(t)} W_Q^g (P_l^{(t)} W_K^g)^\top / \sqrt{C}\big)}{\sum_{l' \neq k} \exp\big(P_k^{(t)} W_Q^g (P_{l'}^{(t)} W_K^g)^\top / \sqrt{C}\big)}, \tag{5}$$

$$\Gamma_k^{(t)} = \sum_{l \neq k} G_{k,l}^{(t)} \big(P_l^{(t)} W_V^g\big). \tag{6}$$

where W_Q^g, W_K^g, and $W_V^g \in \mathbb{R}^{C \times C}$ are learnable graph-specific projection matrices, and $l \in \{1, \ldots, K_{\text{eff}}\} \setminus \{k\}$ indexes all nodes except the k-th node.

Step 4: Residual Update.

$$P_k^{(s,(t+1))} = \mathrm{LN}\Big(P_k^{(s,(t))} + \mathrm{MLP}^{(t)}\big([P_k^{(s,(t))};\, M_k^{(t)};\, \Gamma_k^{(t)}]\big)\Big). \tag{7}$$

Attended Query Features. The complementary normalization of the same affinities over the prototype dimension produces a spatially varying attention map:

$$B_{i,k}^{(T)} = \frac{\exp\big(e_{k,i}^{(T)}\big)}{\sum_{k'=1}^{K_{\text{eff}}} \exp\big(e_{k',i}^{(T)}\big)}, \tag{8}$$

$$\tilde{F}^q = \mathrm{Reshape}\big(B^{(T)} (\tilde{P}^s W_V)\big) \in \mathbb{R}^{h \times w \times C}, \tag{9}$$

where $N = h \times w$, $K_{\text{eff}} = K+1$ includes the background node defined above, $B^{(T)} \in \mathbb{R}^{N \times K_{\text{eff}}}$, and $\tilde{P}^s = [\tilde{P}_1^s; \ldots; \tilde{P}_K^s; \tilde{P}_{\text{bg}}^s] \in \mathbb{R}^{K_{\text{eff}} \times C}$ stacks the refined nodes. The value projection W_V is shared with Eq. (**??**). Because the background node guarantees $K_{\text{eff}} \geq 2$, $B^{(T)}$ remains non-degenerate for every shot setting.

2.5 Multi-Scale Prototype Matching

Objects have multi-scale properties and diverse spatial contexts. Single-scale matching might fail to extract fine-grained details (on small objects) or global context (on large objects). We introduce computing similarities in multiple receptive fields and adaptively fuse them. We transform the query feature F^q using convolutions with different kernel sizes:

$$F_{r_1}^q = \mathrm{Conv}_{3\times3}(F^q),\ F_{r_2}^q = \mathrm{Conv}_{5\times5}(F^q),$$
$$F_{r_3}^q = \mathrm{Conv}_{7\times7}(F^q)$$

where $F_{r_i}^q \in \mathbb{R}^{h \times w \times C}$ captures patterns at receptive field scale r_i.

For each scale r and prototype k, we compute pixel-wise cosine similarity:

$$S_{r,k}(i,j) = \frac{F_r^q(i,j) \cdot \tilde{P}_k^s}{\|F_r^q(i,j)\|_2 \|\tilde{P}_k^s\|_2} \in [-1, 1],$$

where $S_{r,k} \in \mathbb{R}^{h \times w}$ represents the similarity map at scale r for prototype k. Rather than a single global weight per scale, we predict per-location scale weights from the stacked similarity responses:

$$W(i,j) = \mathrm{softmax}_r\Big(\mathrm{Conv}_{3\times3}^{(2)}\big(\mathrm{ReLU}(\mathrm{Conv}_{3\times3}^{(1)}(Z(i,j)))\big)\Big), \quad Z(i,j) = \big[S_{r_1,k}(i,j); S_{r_2,k}(i,j); S_{r_3,k}(i,j)\big], \tag{10}$$

$$S_k(i,j) = \sum_r W_r(i,j)\, S_{r,k}(i,j). \tag{11}$$

For K-shot scenarios, we average over all prototypes:

$$S^q = \frac{1}{K} \sum_{k=1}^{K} S_k \in \mathbb{R}^{h \times w \times 1}.$$

2.6 SEGMENTATION DECODER

The decoder fuses the attended query features $\tilde{F}^q$ and similarity map S^q to produce the final segmentation:

$$\hat{M}^q = \text{Upsample}\big(\sigma(D([\tilde{F}^q; S^q]))\big),$$

where $[\cdot;\cdot]$ denotes channel concatenation, and D is a three-layer convolutional decoder:

$$D(x) = \text{Conv}^{(3)}_{1\times1} \circ \text{ReLU} \circ \text{Conv}^{(2)}_{3\times3} \circ \text{ReLU} \circ \text{Conv}^{(1)}_{3\times3}(x),$$

where the channel progression is $(C+1) \to C \to C/2 \to 1$, σ is the sigmoid activation, and Upsample performs bilinear interpolation to the original resolution $H \times W$. We clamp predictions to $[\epsilon, 1-\epsilon]$ with $\epsilon = 10^{-7}$ for numerical stability.

2.7 LOSS FUNCTION

We supervise the network using binary cross-entropy loss:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{HW}\sum_{i,j}\Big[M^q(i,j)\log\hat{M}^q(i,j) + (1-M^q(i,j))\log(1-\hat{M}^q(i,j))\Big].$$

The entire network (except the frozen generic branch E_{gen}) is trained end-to-end using this loss. The model is trained using the AdamW optimizer with a learning rate of 1×10^{-4} and a weight decay of 1×10^{-4}, along with gradient clipping (maximum norm of 1.0).

3 EXPERIMENTS

3.1 DATASETS AND EVALUATION METRICS

Datasets. We conduct experiments on three benchmark datasets widely used in few-shot segmentation. FSS-1000 Li et al. (2020) is composed of 1,000 object categories with 10 annotated training samples per class and exhibits great intra-class diversity and large variations. PASCAL-5^i Shaban et al. (2017) is created based on PASCAL VOC 2012 by splitting the 20 object categories equally into 4 folds (5 classes per fold) for cross-validation. COCO-20^i Yang et al. (2021) is built from the MS-COCO dataset by dividing 80 classes into 4 groups of 20 categories with a more cluttered background.

Evaluation Metrics. We used mean Intersection over Union (mIoU) Liu et al. (2020a;b); Shaban et al. (2017); Siam et al. (2019) as the main metric for ablation studies and comparisons. To further evaluate the proposed method, we also report Foreground-Background IoU (FB-IoU), which refers to the IoU between target and non-target regions without distinguishing different categories. We also report the Dice coefficient (DICE) between predicted regions and the ground truth, which is complementary to IoU. Finally, we include precision, defined as $\frac{TP}{TP+FP}$, to evaluate our model's effectiveness in reducing false positives.

3.2 IMPLEMENTATION DETAILS

We use ResNet-50, ResNet-101, and VGG-16 backbones pre-trained on ImageNet. The unified feature dimension is $C = 256$, the graph refinement MLP hidden dimension is 128, and the number of refinement iterations is $T = 2$. The backbone E is kept entirely frozen and shared by both branches; only the task adapter A_{task}, the projection and fusion layers, and the newly introduced modules (CAGPG, MSPM, decoder) are trained, totaling 10M trainable parameters. This design

Table 1: **Class mIoU and FB-IoU on PASCAL-5^i.** Mn denotes mean class mIoU over four folds. Best per column in **bold**, second best underlined.

Bkb.	Method	1-Shot						5-Shot					
		F0	F1	F2	F3	Mn	FB	F0	F1	F2	F3	Mn	FB
	OSLM Shaban et al. (2017)	33.6	55.3	40.9	33.5	40.8	61.3	35.9	58.1	42.7	39.1	44.0	61.5
	co-FCN Rakelly et al. (2018)	36.7	50.6	44.9	32.4	41.1	60.1	37.5	50.0	44.1	33.9	41.4	60.2
	RPMM Yang et al. (2020a)	47.1	65.8	50.6	48.5	53.0	–	50.0	66.5	51.9	47.6	54.0	–
V16	PFENet Tian et al. (2020)	56.9	**68.2**	54.4	52.4	58.0	72.3	59.0	**69.1**	54.8	52.9	59.0	72.3
	MMNet Wu et al. (2021)	57.1	67.2	56.6	52.3	58.3	–	56.6	67.7	63.6	56.5	58.3	–
	HSNetMin et al. (2021)	59.6	65.7	59.6	54.0	59.7	–	**64.9**	69.0	**64.1**	58.6	**64.1**	–
	NTRENet Liu et al. (2022)	57.7	67.6	57.1	53.7	59.0	**73.1**	60.3	68.0	55.2	57.1	60.2	74.2
	Ours	**60.5**	61.9	**60.7**	**61.3**	**61.1**	70.9	61.8	62.5	61.2	**60.8**	61.6	**74.9**
	CANet Zhang et al. (2019)	52.5	65.9	51.3	51.9	55.4	66.2	55.5	67.8	51.9	53.2	57.1	69.6
	RPMM Yang et al. (2020a)	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)	61.7	69.5	55.4	56.3	60.8	73.3	63.1	70.7	55.8	57.9	61.9	73.9
	SCL Zhang et al. (2021a)	63.0	70.0	57.7	61.8	63.1	71.9	64.5	70.6	57.3	59.1	62.9	72.8
	ASGNet Li et al. (2021)	58.8	67.9	56.8	53.7	59.3	69.2	63.7	70.4	64.2	57.4	63.9	74.2
R50	ReRPI Boudiaf et al. (2021)	59.8	68.3	62.1	48.5	59.7	–	64.6	71.4	71.1	59.3	66.6	–
	SAGNN Xie et al. (2021)	64.7	69.6	57.0	57.2	62.1	73.2	64.9	70.0	57.0	59.3	62.8	73.3
	MLC Yang et al. (2021)	59.2	71.2	65.6	52.6	62.1	–	63.5	71.6	**71.2**	58.1	66.1	–
	HSNetMin et al. (2021)	64.3	70.7	60.3	60.5	64.0	–	70.3	73.2	67.4	**67.1**	69.5	–
	CyCTRZhang et al. (2021b)	65.7	71.0	59.5	59.7	64.0	–	69.3	73.5	63.8	63.5	67.5	–
	DCAMAShi et al. (2022)	67.5	72.3	59.6	59.0	64.6	–	70.5	73.9	63.7	65.8	68.5	–
	NTRENet Liu et al. (2022)	65.4	72.3	59.4	59.8	64.2	77.0	66.2	72.8	61.7	62.2	65.7	78.4
	RiFeNetBao et al. (2024)	68.4	73.5	67.1	59.4	67.1	–	70.0	74.7	69.4	64.2	69.6	–
	Ours	**69.3**	**74.3**	**68.5**	**62.4**	**68.6**	**78.7**	**71.5**	**75.6**	70.9	65.2	**70.8**	**79.3**
	DAN Wang et al. (2020)	54.7	68.6	57.8	51.6	58.2	71.9	57.9	69.0	60.1	54.9	60.5	72.3
	PPNet Liu et al. (2020b)	52.7	62.8	57.4	47.7	55.2	70.9	60.3	70.0	69.4	60.7	65.1	77.5
	PFENet Tian et al. (2020)	60.5	69.4	54.4	55.9	60.1	72.9	61.8	70.4	54.9	57.6	61.4	73.5
	ASGNet Li et al. (2021)	59.8	67.4	56.5	57.0	60.2	71.7	64.6	71.4	64.2	57.4	64.4	75.2
R101	ReRPI Boudiaf et al. (2021)	59.6	68.6	62.2	47.2	59.4	–	66.8	74.9	**71.4**	63.1	68.8	–
	MLC Yang et al. (2021)	60.8	71.3	61.5	56.9	62.6	–	65.8	74.9	**71.4**	63.1	68.8	–
	CyCTRZhang et al. (2021b)	67.2	71.1	57.6	59.0	63.7	–	71.0	75.0	58.5	65.0	67.4	–
	DCAMAShi et al. (2022)	65.4	71.4	63.2	58.3	64.6	–	70.7	73.7	66.8	61.9	68.3	–
	HSNetMin et al. (2021)	67.3	72.3	62.0	63.1	66.2	–	**71.8**	74.4	67.0	**68.3**	**70.4**	–
	NTRENet Liu et al. (2022)	65.5	71.8	59.1	58.3	63.7	75.3	67.9	73.2	60.1	66.8	67.0	78.2
	RiFeNetBao et al. (2024)	**68.9**	**73.8**	66.2	60.3	**67.3**	–	70.4	74.5	68.3	63.4	69.2	–
	Ours	62.1	62.9	**67.1**	**64.9**	64.1	**76.7**	71.2	**75.7**	69.1	62.5	69.6	**78.9**

ensures fairness with single-backbone baselines and avoids the overhead of a second trainable backbone. Training is performed for 50 epochs using AdamW (learning rate 1×10^{-4}, weight decay 1×10^{-4}) with a batch size of 8 and gradient clipping (maximum norm 1.0).

Following PFENet Tian et al. (2020) and NTRENet Liu et al. (2022), all images are resized to 473×473. For ResNet, dilated convolutions are adopted in the last two stages, while the last two pooling layers are removed for VGG-16, yielding an output stride of 8. Support masks are resized to the feature resolution before masked average pooling, and predictions are bilinearly upsampled to the original image size for evaluation.

We follow the standard splits for FSS-1000, PASCAL-5^i, and COCO-20^i. Each episode consists of K support image-mask pairs and one query image from the same class. Performance is evaluated using class mIoU, FB-IoU, and Dice.The network is implemented in PyTorch Paszke et al. (2019) and trained on a single NVIDIA GeForce RTX 5070 GPU.

Table 3 reports the results on **FSS-1000** with different backbones. CGPRNet achieves strong performance across all architectures. ResNet-101 provides the best results with 81.57% mIoU (1-shot) and 82.40% mIoU (5-shot), while ResNet-50 and VGG-16 also maintain competitive accuracy, demonstrating the robustness of our method. Consistent performance across all folds indicates stable generalization. Table 5 further shows that ResNet-101 generally achieves the highest Dice scores, with slight improvements in the 5-shot setting.

Table 2: Class mIoU and FB-IoU on COCO-20^i. Mn denotes mean class mIoU over four folds. Best per column in **bold**, second best underlined.

Bkb.	Method	1-Shot						5-Shot					
		F0	F1	F2	F3	Mn	FB	F0	F1	F2	F3	Mn	FB
R50	PPNet Liu et al. (2020b)	28.1	30.8	29.5	27.7	29.0	–	39.0	40.8	37.1	37.3	38.6	–
	RPMM Yang et al. (2020a)	29.5	36.8	28.9	27.0	30.6	–	33.8	42.0	33.0	33.3	35.5	–
	ASGNet Li et al. (2021)	–	–	–	–	34.6	60.4	–	–	–	–	42.5	67.0
	MMNet Wu et al. (2021)	34.9	41.0	37.2	37.0	37.5	–	37.0	40.3	39.3	36.0	38.2	–
	MLC Yang et al. (2021)	46.8	35.3	26.2	27.1	33.9	–	54.1	41.2	34.1	33.1	40.6	–
	NTRENet Liu et al. (2022)	36.8	42.6	39.9	37.9	39.3	68.5	38.2	44.1	40.4	38.4	40.3	69.2
	HSNetMin et al. (2021)	36.3	43.1	38.7	38.7	39.2	–	43.3	51.3	48.2	45.0	46.9	–
	CyCTRZhang et al. (2021b)	38.9	43.0	39.6	39.8	40.3	–	41.1	48.9	45.2	47.0	45.6	–
	DCAMAShi et al. (2022)	41.9	45.1	44.4	41.7	43.3	–	45.9	50.5	**50.7**	46.0	48.3	–
	RiFeNetBao et al. (2024)	39.1	**47.2**	44.6	45.4	44.1	–	44.3	**52.4**	49.3	48.4	**48.6**	–
	Ours	**48.1**	49.6	**48.9**	**47.5**	**48.5**	**69.7**	**59.7**	59.1	58.4	**57.8**	58.8	**71.7**
R101	DAN Wang et al. (2020)	–	–	–	–	24.4	62.3	–	–	–	–	29.6	63.9
	SCL Zhang et al. (2021a)	36.4	38.6	37.5	35.4	37.0	–	38.9	40.5	41.5	38.7	39.9	–
	PFENet Tian et al. (2020)	34.3	33.0	32.3	30.1	32.4	58.6	38.5	38.6	38.2	34.3	37.4	61.9
	MLC Yang et al. (2021)	50.2	37.8	27.1	30.4	36.4	–	57.0	46.2	37.3	37.2	44.4	–
	SAGNN Xie et al. (2021)	36.1	41.0	38.2	33.5	37.2	60.9	40.9	48.3	42.6	38.9	42.7	63.4
	HSNetMin et al. (2021)	37.2	44.1	42.4	41.3	41.2	–	45.9	53.0	51.8	47.1	49.5	–
	NTRENet Liu et al. (2022)	38.3	40.4	39.5	38.1	39.1	67.5	42.3	44.4	44.2	41.7	43.2	69.6
	Ours	**40.4**	**41.2**	**42.1**	**43.8**	**41.9**	**70.2**	**43.4**	**45.8**	**45.1**	**44.3**	**44.6**	**72.3**
V16	PPNet Liu et al. (2020b)	25.3	27.6	26.1	24.8	26.0	–	35.1	36.9	33.4	35.2	35.2	–
	RPMM Yang et al. (2020a)	27.8	34.2	26.5	25.1	28.4	–	31.5	39.8	30.2	31.1	33.2	–
	NTRENet Liu et al. (2022)	34.1	39.8	37.2	35.6	36.7	65.3	36.4	41.9	38.1	36.0	38.1	66.8
	Ours	**35.6**	**41.0**	**36.2**	**37.9**	**37.7**	**67.8**	**37.8**	**42.2**	**41.5**	**38.9**	**40.1**	**70.4**

Table 3: 1-shot and 5-shot mIoU on FSS-1000. Best in bold.

Backbone	F0	F1	F2	F3	Mean	FB
1-Shot						
ResNet-50	77.18	**82.80**	83.97	80.52	81.12	78.09
VGG-16	74.65	81.69	82.49	79.50	79.58	76.67
ResNet-101	**78.60**	82.72	**84.41**	**80.54**	**81.57**	**78.36**
5-Shot						
ResNet-50	79.85	83.59	**84.39**	80.86	82.17	78.61
VGG-16	78.48	81.23	82.97	79.71	80.60	76.91
ResNet-101	**80.63**	**84.02**	83.18	**81.77**	**82.40**	**79.45**

Table 4: Ablation on FSS-1000 (ResNet-50), mean over 4 folds. Δ = mIoU drop from full model.

Config	1-Shot Dice	1-Shot IoU	1-Shot FB	1-Shot Δ	5-Shot Dice	5-Shot IoU	5-Shot FB	5-Shot Δ
Full	**90.5**	**83.6**	**80.9**	–	**90.8**	**84.0**	**81.0**	–
Core modules								
w/o Dual-Branch	90.2	82.5	80.6	−1.1	85.1	77.1	73.8	−6.9
w/o Graph Refine	90.3	83.2	80.4	−0.4	90.3	83.3	80.5	−0.7
w/o Multi-Scale	90.1	83.0	80.4	−0.6	85.5	77.4	74.2	−6.6
Prototype graph design								
Softmax over K	90.3	83.0	80.6	−0.6	90.6	83.5	80.7	−0.5
w/o BG Node	90.3	82.8	80.5	−0.8	90.4	83.2	80.4	−0.8
w/o Inter-Proto	90.3	82.9	80.5	−0.7	90.5	83.3	80.6	−0.7
Multi-scale design								
Global α_r	90.4	83.1	80.7	−0.5	90.6	83.7	80.8	−0.3

Table 5: Dice on PASCAL-5^i and FSS-1000 (1-shot / 5-shot). Best in bold.

Dataset	Backbone	F0	F1	F2	F3	Mean
	1-Shot					
PASCAL-5^i	ResNet-50	72.34	**75.05**	**74.43**	73.23	73.76
	VGG-16	70.69	73.20	72.50	72.67	72.26
	ResNet-101	**74.10**	74.25	73.35	**74.09**	**73.94**
FSS-1000	ResNet-50	85.99	**90.00**	90.77	**88.15**	88.73
	VGG-16	84.31	89.38	89.82	87.45	87.74
	ResNet-101	**87.16**	89.92	**91.09**	88.10	**89.07**
	5-Shot					
PASCAL-5^i	ResNet-50	**74.69**	**76.17**	75.23	**74.04**	**75.03**
	VGG-16	73.85	72.65	73.27	72.57	73.09
	ResNet-101	73.72	73.86	**75.33**	73.91	74.21
FSS-1000	ResNet-50	87.98	90.29	**91.04**	88.27	89.39
	VGG-16	87.09	88.91	90.08	87.65	88.43
	ResNet-101	**88.56**	**90.83**	89.80	**88.79**	**89.52**

Table 6: Ablation on CAGPG steps T (PASCAL-5^0, ResNet-50).

T	1-Shot mIoU	1-Shot FB-IoU	5-Shot mIoU	5-Shot FB-IoU	GFLOPs	Par.(M)	ms
1	60.21	69.83	60.84	70.11	5.8	4.45	17
2	**68.60**	**78.70**	**70.80**	**79.30**	6.3	4.87	19
3	62.81	71.74	63.90	72.41	6.8	5.28	22
4	62.78	71.70	63.85	72.36	7.3	5.70	25

3.3 Comparison with State-of-the-art Methods

Table 1 shows our method performs strongly on PASCAL-5^i. With ResNet-50, it achieves the best mean mIoU (68.6%/70.8% for 1-/5-shot) and FB-IoU (78.7%/79.3%), surpassing RiFeNet by 1.5%/1.2% mIoU, with the largest gains on the harder folds F2 and F3. It also yields the top VGG-16 1-shot mean mIoU (61.1%) and the best FB-IoU under ResNet-101 (76.7%/78.9%), remaining

competitive in mIoU elsewhere. The consistent FB-IoU advantage indicates cleaner foreground–background separation.

Table 2 confirms our method also generalizes to the more challenging COCO-20^i. With ResNet-50, it obtains the best 1-shot mean mIoU (48.5%) and FB-IoU (69.7%), outperforming RiFeNet by 4.4% mIoU, and attains the highest 5-shot FB-IoU (71.7%). Under ResNet-101 and VGG-16, our model is best in every column, improving 1-/5-shot mean mIoU over NTRENet by 2.8%/1.4% (R101) and 1.0%/2.0% (V16). The consistent FB-IoU gains across all three backbones again point to cleaner foreground–background separation on this large-scale benchmark.

Qualitative comparisons We present the qualitative comparisons of CGPRNet and PFENet few shot segmentation are shown in Fig. 3. CGPRNet generates segmentations with more precise object boundaries for various categories. By contrast, PFENet often over-segments the background (the runway in the first row and the platform in the fourth row). The CGPRNet can better cover the object extent while preserving object boundaries, especially as shown in the airplane and locomotive. As shown in the split-view comparison, CGPRNet can better distinguish the foreground objects from the complex backgrounds.

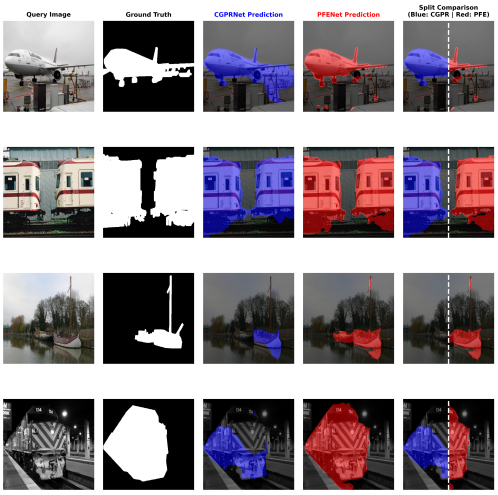

Figure 3: Qualitative results of our proposed CGPRNet with PFENet across diverse object categories. Each row shows: (*left to right*) query image, ground truth segmentation mask, CGPRNet prediction overlaid in blue, PFENet prediction overlaid in red, and a split comparison view.

Table 4 summarizes the mean ablation results on FSS-1000; removing the Dual-Branch encoder causes the largest 5-shot degradation (-6.9% mIoU), Although graph refinement yields smaller aggregate mIoU gains than the dual-branch encoder and multi-scale matching module, the three components operate at different stages of the pipeline and therefore provide complementary benefits. This interpretation is further supported by the iteration-count ablation in Table 6 , where introducing iterative graph refinement ($T = 2$ vs. $T = 1$) substantially improves performance, while additional iterations provide only marginal gains. **Limitations**. Despite our method's strong performance, some limitations are acknowledged. First, performance varies moderately across backbones. On COCO-20^i 1-shot, VGG-16 achieves 47.93% mIoU, compared with 49.62% for ResNet-101, indicating only a modest dependence on the strength of the feature extractor. Secondly, the cross-guidance mechanism incurs additional computational overhead. Lastly, the observed discrepancy in the performance pattern between COCO-20^i and **FSS-1000** indicates sensitivity to the dataset characteristics. In the future, we will address these limitations by exploring prototype aggregation strategies and optimizing the cross-guidance mechanism for computational efficiency.

4 Conclusion

We introduced CGPRNet, a few-shot segmentation framework that combines dual-branch feature encoding, query-conditioned graph prototype refinement, and adaptive multi-scale matching. Trained with only 10M parameters on top of a frozen shared backbone, it achieves competitive accuracy on PASCAL-5^i, COCO-20^i, and FSS-1000. Ablations show that the three components provide complementary benefits, with the dual-branch encoder being most critical for discriminative features. Qualitative results further confirm more accurate object boundaries and better background suppression. Future work will extend CGPRNet to video segmentation and cross-domain adaptation.

REFERENCES

Xiaoyi Bao, Jie Qin, Siyang Sun, Xingang Wang, and Yun Zheng. Relevant intrinsic feature enhancement network for few-shot semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 765–773, 2024.

Malik Boudiaf, Hoel Kervadec, Ziko Imtiaz Masud, Pablo Piantanida, Ismail Ben Ayed, and Jose Dolz. Few-shot segmentation without meta-learning: A good transductive inference is all you need? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13979–13988, 2021.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pp. 2650–2658, 2015.

David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014.

Gen Li, Varun Jampani, Laura Sevilla-Lara, Deqing Sun, Jonghyun Kim, and Joongkyu Kim. Adaptive prototype learning and allocation for few-shot segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8334–8343, 2021.

Xiang Li, Tianhan Wei, Yau Pun Chen, Yu-Wing Tai, and Chi-Keung Tang. Fss-1000: A 1000-class dataset for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Yaochen Li, Qiao Li, Cong Gao, Shengjing Gao, Hao Wu, and Rui Liu. Pfenet: Towards precise feature extraction from sparse point cloud for 3d object detection. *Neural Networks*, 185:107144, 2025.

Weide Liu, Chi Zhang, Guosheng Lin, and Fayao Liu. Crnet: Cross-reference networks for few-shot segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4165–4173, 2020a.

Yongfei Liu, Xiangyi Zhang, Songyang Zhang, and Xuming He. Part-aware prototype network for few-shot semantic segmentation. In *European conference on computer vision*, pp. 142–158. Springer, 2020b.

Yuanwei Liu, Nian Liu, Qinglong Cao, Xiwen Yao, Junwei Han, and Ling Shao. Learning non-target knowledge for few-shot semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11573–11582, 2022.

Juhong Min, Dahyun Kang, and Minsu Cho. Hypercorrelation squeeze for few-shot segmenation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6921–6932. IEEE, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alyosha Efros, and Sergey Levine. Conditional networks for few-shot semantic segmentation. *arXiv preprint arXiv:1806.07373*, 2018.

Amirreza Shaban, Shray Bansal, Zhen Liu, et al. One-shot learning for semantic segmentation. bmvc, 2017.

Xinyu Shi, Dong Wei, Yu Zhang, Donghuan Lu, Munan Ning, Jiashun Chen, Kai Ma, and Yefeng Zheng. Dense cross-query-and-support attention weighted mask aggregation for few-shot segmentation. In *European Conference on Computer Vision*, pp. 151–168. Springer, 2022.

Mennatullah Siam, Boris N Oreshkin, and Martin Jagersand. Amp: Adaptive masked proxies for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5249–5258, 2019.

Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia. Prior guided feature enrichment network for few-shot segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 44(2):1050–1065, 2020.

Haochen Wang, Xudong Zhang, Yutao Hu, Yandan Yang, Xianbin Cao, and Xiantong Zhen. Few-shot semantic segmentation with democratic attention networks. In *European conference on computer vision*, pp. 730–746. Springer, 2020.

Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *proceedings of the IEEE/CVF international conference on computer vision*, pp. 9197–9206, 2019.

Zhonghua Wu, Xiangxi Shi, Guosheng Lin, and Jianfei Cai. Learning meta-class memory for few-shot semantic segmentation. pp. 517–526, 2021.

Chenxi Xie, Changqun Xia, Mingcan Ma, Zhirui Zhao, Xiaowu Chen, and Jia Li. Pyramid grafting network for one-stage high resolution saliency detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11717–11726, 2022.

Guo-Sen Xie, Jie Liu, Huan Xiong, and Ling Shao. Scale-aware graph neural network for few-shot semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5475–5484, 2021.

Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Qixiang Ye. Prototype mixture models for few-shot semantic segmentation. pp. 763–778, 2020a.

Lihe Yang, Wei Zhuo, Lei Qi, Yinghuan Shi, and Yang Gao. Mining latent classes for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8721–8730, October 2021.

Xianghui Yang, Bairun Wang, Kaige Chen, Xinchi Zhou, Shuai Yi, Wanli Ouyang, and Luping Zhou. Brinet: Towards bridging the intra-class and inter-class gaps in one-shot segmentation. *arXiv preprint arXiv:2008.06226*, 2020b.

Bingfeng Zhang, Jimin Xiao, and Terry Qin. Self-guided and cross-guided learning for few-shot segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8312–8321, 2021a.

Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5217–5226, 2019.

Gengwei Zhang, Guoliang Kang, Yi Yang, and Yunchao Wei. Few-shot segmentation via cycle-consistent transformer. *Advances in neural information processing systems*, 34:21984–21996, 2021b.

Renjie Zhang, Di Lin, Xin Wang, George Baciu, CL Philip Chen, and Ping Li. Accurate-pgnet: Learning to assemble perceptual body parts for accurate human skeleton establishment. *IEEE Transactions on Multimedia*, 2024.