Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.062412

ARTICLE

Two-Stage Category-Guided Frequency Modulation for Few-Shot Semantic Segmentation

Yiming Tang* **and Yanqiu Chen**

School of Computer Science, Fudan University, Shanghai, 200438, China

*Corresponding Author: Yiming Tang. Email: ymtang18@fudan.edu.cn

Received: 18 December 2024; Accepted: 18 February 2025; Published: 16 April 2025

ABSTRACT: Semantic segmentation of novel object categories with limited labeled data remains a challenging problem in computer vision. Few-shot segmentation methods aim to address this problem by recognizing objects from specific target classes with a few provided examples. Previous approaches for few-shot semantic segmentation typically represent target classes using class prototypes. These prototypes are matched with the features of the query set to get segmentation results. However, class prototypes are usually obtained by applying global average pooling on masked support images. Global pooling discards much structural information, which may reduce the accuracy of model predictions. To address this issue, we propose a Category-Guided Frequency Modulation (CGFM) method. CGFM is designed to learn category-specific information in the frequency space and leverage it to provide a two-stage guidance for the segmentation process. First, to self-adaptively activate class-relevant frequency bands while suppressing irrelevant ones, we leverage the Dual-Perception Gaussian Band Pre-activation (DPGBP) module to generate Gaussian filters using class embedding vectors. Second, to further enhance category-relevant frequency components in activated bands, we design a Support-Guided Category Response Enhancement (SGCRE) module to effectively introduce support frequency components into the modulation of query frequency features. Experiments on the PASCAL-5^i and COCO-20^i datasets demonstrate the promising performance of our model. The code will be released at https://github.com/tymatfd/CGFM, accessed on 17 February 2025.

KEYWORDS: Few-shot semantic segmentation; frequency feature; category representation

1 Introduction

Computer vision has made significant progress in the era of deep learning [1–3]. Semantic segmentation [4–6] is a fundamental task in computer vision. It assigns a label to every pixel in an image, which allows models to interpret the scene at a fine-grained level. However, training semantic segmentation models is typically resource-intensive and time-consuming, as they rely heavily on carefully annotated large-scale datasets. Additionally, these models struggle to maintain accurate predictions for objects belonging to categories unseen in the training set, limiting their applicability in real-world scenarios. Few-shot semantic segmentation aims to address these problems by providing a few examples (the support set) for each target class and attempting to identify objects from the same class in other images (the query set) based on these examples. To make sure that the trained model can be well generalized to unseen classes, classes (base classes) contained in the train set are designed to have no overlap with those (novel classes) in the test set. Few-shot segmentation has emerged as a promising paradigm across various fields [7–10] demonstrating its versatility and potential for solving real-world segmentation challenges with limited labeled data.

To successfully recognize the target objects in the query set, it is essential to extract effective target class representations from the limited support set examples. Most few-shot segmentation (FSS) methods [11–15] employ masked average pooling (MAP) to obtain the class representation. MAP utilizes target object masks provided by the support set to mask out all background regions and then applies global average pooling to the remaining object regions. The obtained feature vector is subsequently used as the prototype for the target class. Some approaches have attempted to increase the number of prototypes [16–19] using superpixels or clustering, but these methods often face optimization difficulties and high costs on computation, especially with multiple support samples. Besides, authors in [20–22] attempt to replace prototype comparison with point-wise comparsion approaches to identify target objects in the query set, but such methods need to consider each pixel in support samples and are vulnerable to noise interference. Consequently, following approaches [21,23] tend to integrate prototypes into the query and support features before point-wise comparison, which can effectively eliminates class-irrelevant noise. As a result, MAP remains widely used in both prototypical and point-wise FSS methods.

However, the global average pooling operation used in MAP may have some shortages. Study [24] finds that for image features, vectors obtained through global average pooling are equivalent to the first component of image frequency features, which means it discards other frequency components. As shown in Fig. 1, preserving only the first few components of the frequency features may lead to the loss of structure information and failure in perfectly reconstructing the target object. In FSS, this means a certain loss of class-level information and will ultimately affect the model's prediction accuracy. To address this issue, we propose to preserve more frequency components and extract representative class-level features from them. Considering the inter-class variances between base and novel classes, as well as among the novel classes themselves, handling frequency-domain features without guidance may lead the model to directly apply knowledge learned from the training set to novel classes, resulting in poor generalization. Furthermore, the intra-class variances between the support and query set are also not negligible. Segmentation tasks are sensitive to the position information of objects in the input images. The position information in the support set contained in frequency features is quite different from that in the query set. Directly introducing support frequency features may disrupt the position information in the query set and cause severe performance reduction.

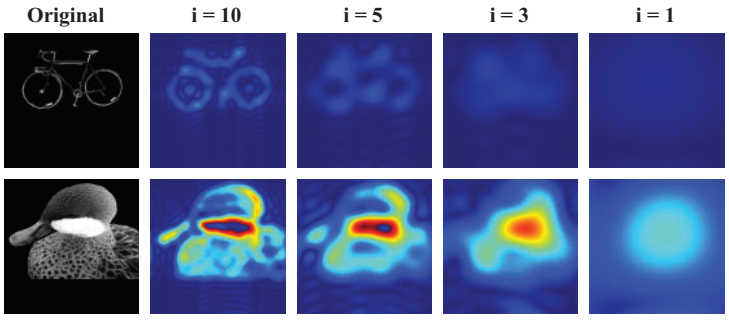

Figure 1: Comparison of features reconstructed using the first i frequency bands, from low to high frequencies. As shown in the last column, when $i = 1$, the remaining frequency components are equivalent to the features obtained through global average pooling, and some crucial structure information is discarded during the pooling operation

To address the aforementioned problems, we propose a two-stage Category-Guided Frequency Modulation (CGFM) approach. In the first stage, we propose a Dual-Perception Gaussian Band Pre-activation (DPGBP) module to preactivate class-relevant frequency bands and minimize the impact of class-irrelevant

signals in the frequency features. This operation is guided by a class embedding vector, initialized with class prototypes, and updated in Transformer blocks to percept and capture both class-level and instance-level information. The dual-perception mechanism makes our model self-adaptive to different target classes and samples. In the second stage, the Support-Guided Category Response Enhancement (SGCRE) module further enhances category-relevant frequency components by introducing support frequency features. In this module, the support and query frequency features are first decomposed into phase and amplitude features to separately process position and intensity information. By introducing position-irrelevant class information, our CGFM model can effectively strengthen the model's response to frequency components most relevant to the target class and alleviate the impact of background noise.

In summary, our contributions can be summarized as follows:

- We propose the Dual-Perception Gaussian Band Pre-activation (DPGBP) module to perceive and capture category and instance information, and then leverage them to guide the activation of class-relevant regions.
- We design the Support-Guided Category Response Enhancement (SGCRE) module to further enhance class-relevant frequency components in query features by introducing the category information decomposed from the support frequency features.
- We evaluated our model on PASCAL-5^i and COCO-20^i, and the results validated the effectiveness of the proposed modules.

2 Related Works

2.1 Few-Shot Semantic Segmentation

Few-shot semantic segmentation was first proposed by OSLSM [11]. As pixel-wise annotations have been provided in the support set, authors in [25] proposed to extract categorical prototypes using masked average pooling (MAP), which masks out background pixels from support features and then makes a global average pooling on them. To get a robust prototype, study [16] leverages a graph attention network to extract more semantic information from unlabeled data, and [17–19,26] extract multiple prototypes for a single category, which are generated using the Expectation-Maximization algorithm, super-pixel segmentation, or self-guided mechanism. PFENet [13] proposed a simple but effective way to improve the model's generalization ability, which constructs prior maps by calculating the cosine similarity between each support and query pixel pair and picking the maximum scores for query pixels. The prior map mechanism does not introduce extra training and largely improves the model's generalization ability, which makes it widely used in the following FSS methods. CyCTR [23] fuses the prior map with corresponding support and query features, then it aggregates support features into query ones using cross-attention between support and query features. To filter out irrelevant pixel pairs between support and query images, it implements a cycle-consistent attention mechanism to suppress possible harmful support features. HDMNet [14] proposed a hierarchical transformer framework by distilling category information between multi-scale features.

The representation ability of class prototype may be restrained by the scarcity of samples for target classes. To address this issue, study [27] uses superpixel-based pseudo-labels to train a few-shot medical image segmentation model without manual annotations. Study [28] proposes a self-supervised tuning framework for few-shot segmentation, which dynamically adjusts the distribution of latent features across different episodes using a self-supervised inner-loop base learner. Study [29] designs another self-supervised meta-learning framework to create training pairs without manual annotations by leveraging unsupervised saliency estimation and image augmentations. Study [30] estimates target distributions through graph

partitioning and Laplacian matrix eigenvectors. Then they adaptively predicts the query mask using the eigenvectors from the support images without the need for manual annotation.

2.2 Frequency Learning

Frequency learning has served as a classic tool in digital image processing [31,32] for a long time. Frequency-based methods have proven effective in improving the representation learning in various tasks. Authors in [33] proposed a fast and privacy-preserving framework to evaluate CNN models using Fourier Transform and fully homomorphic encryption to efficiently evaluate CNN models, and enable secure predictions without compromising model or input privacy. Authors in [34] propose to mix tokens in the frequency space by learning instance-adaptive masks for semantic filtering. They achieve remarkable improvements in lightweight neural networks for visual recognition. Study [35] jointly exploit global frequency information and local spatial dependencies to improve the quality of face super-resolution. Their method highlights the importance of phase information in the frequency domain for preserving facial structures, combining both frequency and spatial learning for superior performance. Authors in [36] introduce deep frequency filtering into domain generalization to suppress domain-specific features and enhance cross-domain transferability. Study [37] presents a novel transformer-based deep learning method for medical image denoising signals, addressing the limitations of existing techniques that often fail to handle complex noise patterns and generate artifacts.

For segmentation tasks, authors in [38] introduced a novel self-attention mechanism in the frequency domain that decouples high and low-frequency components. They significantly reduce computational complexity while achieving state-of-the-art segmentation performance with improved edge preservation and object consistency. Study [39] removes the decoder and uses an adaptive frequency filter to reduce complexity while preserving high-resolution semantics. However, these methods are fully supervised segmentation models trained on large datasets. These methods do not consider few-shot scenarios with significant class variances. They fail to include the inherent characteristics of each target class and sample in their features, which may lead to a lack of self-adaption to target classes.

3 Problem Formulation

Few-shot semantic segmentation aims to segment objects belonging to a specific class using only a few annotated examples. As defined in [11], a few-shot segmentation episode splits the data into two parts: the support set and the query set. Both sets contain images with objects from the same specific class. The support set provides a few example images I_s along with their respective binary masks M_s, which annotate the objects belonging to the target class in the current episode. The query set consists of images I_q and the ground-truth mask M_q. The segmentation model needs to make predictions on I_q, which segments objects belonging to the specified class based on I_s and M_s. Few-shot segmentation aims to maintain robust generalization ability when dealing with novel classes. To ensure this, the training and testing classes must have no overlap $C_{\text{train}} \cap C_{\text{test}} = \varnothing$.

4 Method

4.1 Dual-Perception Gaussian Band Pre-activation Module

Samples from the support set and query set, denoted as I_s and I_q, are processed by a pre-trained encoder to obtain the support and query features F_s and $F_q \in \mathbb{R}^{H\times W\times C}$, where H, W, and C represent the spatial and channel dimensions of the features, respectively. Few-shot segmentation (FSS) models need to handle a variety of target classes, and they may encounter significant overfitting due to large inter-class variances

and class-irrelevant noise. To address this issue, we propose a self-adaptive frequency band pre-activation module that activates class-relevant frequency bands and suppresses class-irrelevant noise for different target categories, as illustrated in Fig. 2.

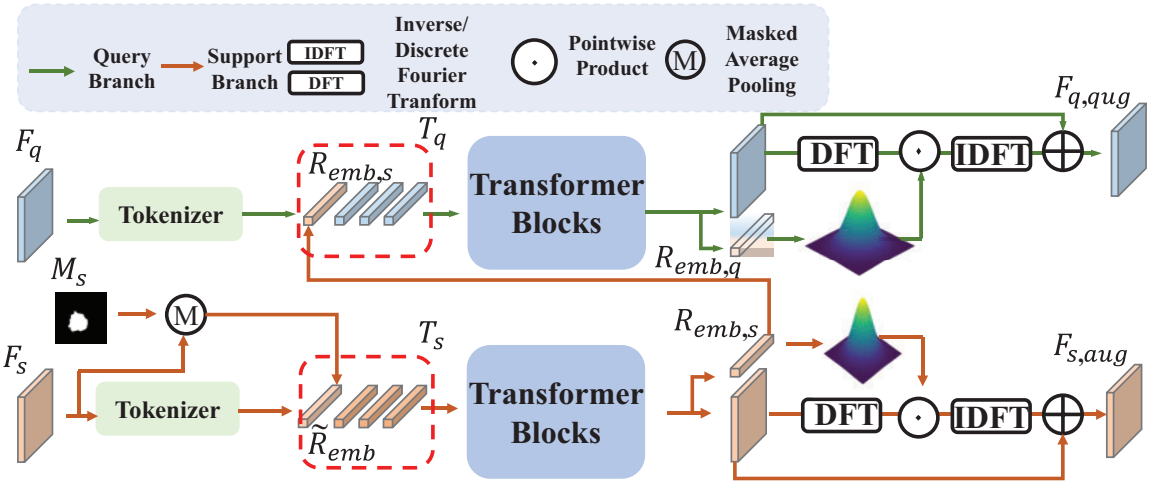

Figure 2: Dual-perception gaussian band pre-activation module

The Dual-Perception Gaussian Band Pre-activation (DPGBP) module modulates frequency features using a series of Gaussian filters, which are generated by perciving category and instance information. We utilize a class embedding vector to capture the intrinsic class attributes and guide the filter generation. The class embedding vector is initialized using the class prototype. Specifically, we perform masked global average pooling (GAP) on the support set features to obtain the class prototype:

$$\tilde{R_{emb}} = \mathrm{GAP}(F_s \cdot M_s). \tag{1}$$

$\tilde{R_{emb}} \in \mathbb{R}^{1\times1\times C}$ is the initial class embedding, and GAP is Global Average Pooling. Subsequently, the class embedding vector is updated in Transformer blocks [14] through the attention pooling mechanism [40], along with tokens extracted from the support and query samples. By integrating the intrinsic instance information extracted from these samples with the previously obtained target category information, our model will be able to adaptively activate class-relevant frequency bands through the Gaussian filters generated using the class embedding vector. Taking the query features as an example, we first concatenate it with the patch embeddings of F_q to get a token sequence: $T_q = [\mathrm{PE}(F_q), \tilde{R_{emb}}]$, where 'PE' denotes the patch embedding operation in Transformer blocks [14]. Then T_q is processed in the Transformer blocks to update the class embedding feature contained in it through the self-attention mechanism:

$$T_q = \mathrm{MHA}(W_Q T_q, W_K T_q, W_V T_q), \tag{2}$$

where W_Q, W_K, and W_V are the linear embedding layer parameters in the Transformer blocks. 'MHA' denotes multi-head attention module. The output features from Transformer blocks are then split to obtain updated query features F_q and the class embedding vector $R_{emb,q}$. Compared to the original $\tilde{R_{emb}}$, $R_{emb,q}$ is updated along with tokens from the query set in transformer blocks and captures the instance information from the query samples in this process. We get the updated class embedding vector $R_{emb,s}$ for the support set using a similar way as the query set, which has been shown in Fig. 2.

The updated class embeddings $R_{emb,q}$ and $R_{emb,s}$ are used to guide the generation of Gaussian filters separately for the query and support branches. The Gaussian kernel construction [41] for images is originally formulated as:

$$G(\sigma,\rho)(i,j) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)}\left[\frac{i^2}{{\sigma_x}^2} + \frac{j^2}{{\sigma_y}^2} - \frac{2\rho ij}{\sigma_x\sigma_y}\right]\right). \tag{3}$$

G denotes the Gaussian filters constructed using σ_r and ρ. i and j denote the position indices on the image. σ, σ_x, $\sigma_y \in \mathbb{R}^{1\times1\times C}$ represent the deviation value and its component along the horizontal and vertical dimension. $\rho \in \mathbb{R}^{1\times1\times C}$ represents the correlation coefficients. In our method, σ and ρ are both computed based on the class embedding. The class embedding is initialized using the class prototype, which is obtained by applying MAP on the support features. Considering that there are a lot of zero values in the mask used in MAP, the class embedding, as well as the obtained deviation and correlation coefficients, may also contain a certain amount of zero values. These values may lead to numerical issues when directly using Eq. (3) to generate Gaussian kernels. To address this problem, we rewrite Eq. (3) into the following form:

$$G(\sigma_r,\rho)(i,j) = \frac{\sigma_{rx}\sigma_{ry}}{2\pi\sqrt{1-\rho^2}} \exp\left(-\left[(\sigma_{rx}^2\cdot i^2 - 2ij\cdot\rho\sigma_{rx}\sigma_{ry} + \sigma_{ry}^2\cdot j^2\right]\right). \tag{4}$$

σ_r denotes the deviation reciprocals. σ_{rx}, $\sigma_{ry} \in \mathbb{R}^{1\times1\times C}$ represent its component along the horizontal and vertical dimensions. We directly calculate the deviation reciprocals instead of itself:

$$\begin{aligned} \sigma_r &= \theta\cdot Sigmoid(f_{var}(R_{emb,q})),\\ \rho &= \gamma\cdot Sigmoid(f_{corr}(R_{emb,q})),\\ F_{q,aug} &= \mathscr{F}^{-1}(Conv_{1\times1}(G(\sigma_r,\rho)\cdot\mathscr{F}(F_q))), \end{aligned} \tag{5}$$

$\mathscr{F}$ and $\mathscr{F}^{-1}$ denote the Discrete Fourier Transform (DFT) and its inverse, respectively. f_{var} and f_{corr} are convolutional layers used to generate the standard deviations and correlation coefficients, respectively. θ and γ are scaling coefficients. A similar calculation process is performed for the support features to obtain the pre-activated support features $F_{s,aug}$.

4.2 Support-Guided Category Response Enhancement Module

4.2.1 Decomposition of Frequency Features

As shown in Fig. 3, $F_{s,aug}$ and $F_{q,aug}$ are first processed by cross-attention Transformer blocks [14] to exchange class information between the query and support set, which helps mitigate the impact of intra-class variances. The features are then transformed into the frequency domain. However, directly applying the support frequency features to the query features may lead to the degradation of position information, significantly reducing model performance. To address this issue, we decompose the frequency features and use only the amplitude component. Specifically, we decompose the frequency features into amplitude and phase components:

$$\begin{aligned} A_q, P_q &= \mathscr{F}(F_{q,aug}),\\ A_s, P_s &= \mathscr{F}(F_{s,aug}\cdot M_s), \end{aligned} \tag{6}$$

where $A_q, P_q, A_s, P_s \in \mathbb{R}^{H\times(\frac{W}{2}+1)\times C}$ denote the amplitude and phase pairs of the query and support samples, respectively. The support amplitude is used to enhance the class-relevant components in the query amplitude, while the query phase is retained to reconstruct the frequency features.

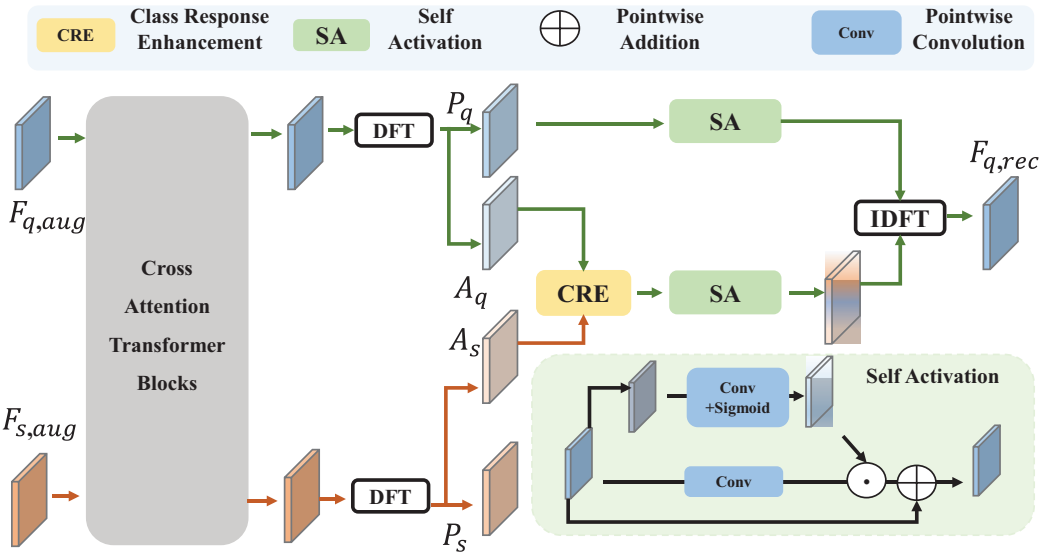

Figure 3: Support-guided category response enhancement module

4.2.2 Category Response Enhancement

We introduce the frequency amplitude of the support set as the target class representation to enhance the class-relevant frequency components in the query set. To achieve this, we first reshape A_s and A_q to $\mathbb{R}^{N\times C}$, N equals $H \times (\frac{W}{2}+1)$, and calculate the amplitude similarity between them:

$$S = Softmax(\frac{A_s^{\top} A_q}{\sqrt{H \times (\frac{W}{2}+1)}}). \tag{7}$$

$S \in \mathbb{R}^{C\times C}$ is then utilized to integrate class-relevant frequency information from the support features into the query frequency features:

$$A_q = A_q + Conv_{1\times1}(A_s S). \tag{8}$$

Subsequently, we perform self-activation on the enhanced frequency features to further strengthen the class-relevant frequency components:

$$\begin{aligned} A_q &= Sigmoid(Conv_{1\times1}(A_q)) \cdot A_q + Conv_{1\times1}(A_q), \\ P_q &= Sigmoid(Conv_{1\times1}(P_q)) \cdot P_q + Conv_{1\times1}(P_q). \end{aligned} \tag{9}$$

The query set features are finally reconstructed with enhanced amplitudes and phases: $F_{q,rec} = \mathscr{F}^{-1}(A_q \exp^{P_q j})$. $F_{q,rec}$ are used in the final classification of foreground and background areas.

4.3 Objective Function and k-shot Extension

The query features output by the SGCRE module are processed by the segmentation decoder to obtain the prediction result $\hat{M}_q$ for the target class region. Binary cross-entropy (BCE) loss is then computed with the true label M_q for each pixel. Additionally, we predict the target class region $\hat{M}_s$ in parallel on the support set samples and calculate BCE loss with the support set labels to guide the model. The descriptions in this article so far are all based on a 1-shot scenario, where only one sample is contained in the support set. In

the k-shot scenario, there are k samples in the support set. Since all frequencies in the frequency domain features are derived by integrating all pixels in the spatial domain, each value can be considered an overall representation of the target class. Therefore, the $\tilde{R_{emb}}$ and A_s corresponding to the support set in our model are obtained by averaging the values of k support samples. Finally, the model combines the cross-entropy losses from both the query and support set into the model's objective function:

$$\mathscr{L}_{seg} = BCE(\hat{M}_q, M_q) + \lambda \cdot \sum_{i}^{k} BCE(\hat{M}_s(i), M_s(i)). \tag{10}$$

where λ is the weight coefficient.

5 Experiment

All experiments in this section are based on the PASCAL-5^i, COCO-20^i, and FSS-1000 [42] datasets. Following most current methods, we use mean Intersection over Union (mIoU) and Foreground-Background IoU (FB-IoU) to evaluate our model. PASCAL-5^i is constructed based on the PASCAL VOC 2012 dataset [43] and further enriched with annotations from the SBD dataset [44]. It includes 20 categories, which are divided into four folds for cross-validation. In each fold, 15 categories are used for training, while the remaining 5 are reserved for testing. For evaluation, 1000 support-query pairs are randomly selected from the test categories. Similarly, COCO-20^i is derived from the MS COCO dataset [45], which consists of over 80,000 images spanning 80 categories. These categories are also divided into four folds for cross-validation, with 60 categories used for training and the remaining 20 for testing in each fold. The category splits for PASCAL-5^i and COCO-20^i follow the same protocols as described in [11] and [46], respectively. In the testing stage, we follow the sampling settings used in most FSS methods [13,21,47], randomly sampling 1000 support-query pairs to calculate mIoU and FB-IoU. FSS-1000 is another large-scale dataset designed for few-shot semantic segmentation, comprising 1000 object classes with 10 images per class, totaling 10,000 images. Each image is annotated with pixel-wise binary segmentation masks, focusing on the primary object. The dataset was constructed by collecting images from multiple search engines, ensuring diversity by including both natural and artificial objects, many of which are not present in PASCAL-5^i and COCO-20^i.

5.1 Implementation Details

For a fair comparison, we follow the preprocessing steps proposed in [13], which includes mirroring and random rotation within a range of -10 to 10 degrees during the training phase. The input images are obtained by cropping 473 × 473 patches from the augmented images to serve as training samples. In the evaluation phase, each input image is resized to match the training patch size while preserving its original aspect ratio by padding with zeros. Our model uses VGG16, ResNet50, and ResNet101 pre-trained on ImageNet as the backbone network for feature extraction. The model is trained for 200 epochs on PASCAL-5^i and 50 epochs on COCO-20^i, with a batch size of 4. During model training, all weights in the backbone network remain fixed. Our model is trained using the AdamW optimizer. The learning rate is set to 1e-4. All experiments were conducted on NVIDIA RTX 4090 GPUs.

5.2 Metrics

To evaluate the performance of our model, we primarily use the mean intersection over union (mIoU) as the main evaluation metric, complemented by the foreground-background IoU (FB-IoU) for additional

insights. The mIoU metric is defined as:

$$\text{mIoU} = \frac{1}{C}\sum_{i=1}^{C}\text{IoU}_i,$$

where C is the total number of classes in each fold, and IoU_i represents the intersection-over-union score for the i-th class. This metric comprehensively evaluates the model's segmentation accuracy across all classes. It averages the IoU values to ensure balanced assessment.

The FB-IoU metric is calculated as:

$$\text{FB-IoU} = \frac{1}{2}(\text{IoU}_F + \text{IoU}_B),$$

where IoU_F and IoU_B represent the IoU values for the foreground and background regions, respectively. Unlike mIoU, FB-IoU ignores class-specific distinctions and focuses on the overall segmentation performance by computing the average IoU for foreground and background.

5.3 Comparison with Other FSS Methods

As shown in Table 1, CGFM demonstrates consistent and robust performance across the three backbones: VGG16, ResNet50, and ResNet101. Using the VGG16 backbone, CGFM achieves an average mIoU of 67.3% in the 1-shot setting and 71.0% in the 5-shot setting, showing substantial improvement compared to prior methods such as PFENet (58.0% and 59.0%, respectively). With the ResNet50 backbone, CGFM achieves an average mIoU of 70.0% in the 1-shot setting and 72.7% in the 5-shot setting, consistently outperforming other methods on both metrics. In ResNet101, CGFM achieves an average mIoU of 71.2% in the 1-shot setting and 73.9% in the 5-shot setting, demonstrating its ability to generalize across folds effectively. In addition to mIoU, CGFM achieves high FBIoU scores, with results of 75.5%, 78.5%, and 79.5% in the 1-shot setting and 78.3%, 80.1%, and 81.1% in the 5-shot setting for VGG16, ResNet50, and ResNet101, respectively. These results highlight the consistent performance of CGFM across various backbones and few-shot settings and demonstrate its ability to handle different scenarios effectively. Based on Table 2, CGFM shows strong performance in the COCO-20^i dataset in both the 1-shot and 5-shot settings with the ResNet50 backbone. In the 1-shot setting, CGFM achieves an average mIoU of 50.4%, which is the highest among all methods. Furthermore, CGFM achieves the best FBIoU score of 74.0%, further confirming its ability to accurately recognize foreground objects from the background in complex scenarios. In the 5-shot setting, CGFM maintains its advantage with an average mIoU of 56.5%. Its FBIoU also improves to 78.0%, which shows that CGFM can effectively leverage additional support examples to improve segmentation accuracy. To validate the generalization ability of our model on a larger number of categories, we conducted experiments on FSS-1000. The results in Table 3 demonstrate that our model maintains promising performance, further confirming its generalization capability across different target categories.

In summary, the experimental results demonstrate that CGFM performs well across various experimental settings and datasets. CGFM not only achieves promising results in average mIoU but also exhibits strong consistency across different folds, highlighting its potential as an effective few-shot segmentation method. Furthermore, CGFM's performance on the challenging COCO-20^i dataset further confirms its generalization ability and robustness in handling complex scenarios.

Table 1: Comparison of our method and other FSS methods on PASCAL-5^i. The best results are marked in bold

Backbone	Methods	1-shot						5-shot					
		Fold-0	Fold-1	Fold-2	Fold-3	Mean	FBIoU	Fold-0	Fold-1	Fold-2	Fold-3	Mean	FBIoU
VGG16	OSLSM [11]	33.6	55.3	40.9	33.5	40.8	–	35.9	58.1	42.7	39.1	43.9	–
	co-FCN [48]	31.7	50.6	44.9	32.4	41.1	–	37.5	50.0	44.1	33.9	41.4	–
	SG-one [25]	40.2	58.4	48.4	38.4	46.3	–	41.9	58.6	48.6	39.4	47.1	–
	PANet [12]	42.3	58.0	51.1	41.2	48.1	–	51.8	64.6	59.8	46.5	55.7	–
	FWB [46]	47.0	59.6	52.6	48.3	51.9	–	50.9	62.9	56.5	50.1	55.1	–
	RPMM [17]	47.1	65.8	50.6	48.5	53.0	–	50.0	66.5	51.9	47.6	54.0	–
	PFENet [13]	56.9	68.2	54.4	52.4	58.0	72.0	59.0	69.1	54.8	52.9	59.0	72.3
	CGFM	**68.0**	**72.8**	**67.5**	**61.0**	**67.3**	**75.5**	**70.5**	**74.4**	**72.0**	**67.0**	**71.0**	**78.3**
ResNet50	CANet [49]	52.5	65.9	51.3	51.9	55.4	–	55.5	67.8	51.9	53.2	57.1	–
	PGNet [50]	56.0	66.9	50.6	50.4	56.0	–	54.9	67.4	51.8	53.0	56.8	–
	RPMM [17]	55.2	66.9	52.6	50.7	56.3	–	56.3	67.3	54.5	51.0	57.3	–
	PFENet [13]	62.7	69.5	55.4	56.3	60.8	73.3	63.1	70.7	55.8	57.9	61.9	73.9
	HSNet [21]	64.3	70.7	60.3	60.5	64.0	76.7	70.3	73.2	67.4	67.1	69.5	80.6
	CyCTR [23]	65.7	71.0	59.5	59.7	64.0	–	69.3	73.5	63.8	63.5	67.5	–
	DPCN [47]	65.7	71.6	69.1	60.6	66.7	78.0	70.0	73.2	70.9	65.5	69.9	80.7
	HDMNet [19]	71.0	75.4	68.9	62.1	69.4	–	71.3	76.2	71.3	68.5	71.8	–
	RiFeNet [51]	68.4	73.5	67.1	59.4	67.1	–	70.0	74.7	69.4	64.2	69.6	–
	NTRENet++ [52]	66.8	73.0	61.0	60.3	65.3	78.6	66.7	73.2	62.6	63.0	66.4	79.3
	DCP [53]	67.2	72.9	65.2	59.4	66.1	77.6	70.5	75.3	68.0	67.7	70.3	**81.5**
	CGFM	**71.3**	**75.8**	**70.0**	**63.0**	**70.0**	**78.5**	**72.5**	**76.4**	**73.0**	**69.0**	**72.7**	80.1
ResNet101	FWB [46]	51.3	64.5	56.7	52.2	56.2	–	54.8	67.4	62.2	55.3	59.9	–
	DAN [20]	54.7	68.6	57.8	51.6	58.2	–	57.9	69.0	60.1	54.9	60.5	–
	PFENet [13]	60.5	69.4	54.4	56.0	60.1	72.9	62.8	70.4	54.9	57.6	61.4	73.5
	CyCTR [23]	67.2	71.1	57.6	59.0	63.7	73.0	71.0	75.0	58.5	65.0	67.4	75.4
	AAFormer [54]	69.9	73.6	57.9	59.7	65.3	74.9	**75.0**	75.1	59.0	63.2	67.4	77.3
	ABCNet [14]	65.3	71.3	**76.2**	59.3	68.0	–	71.4	75.0	68.2	63.1	69.4	–
	NTRENet++ [52]	67.2	72.3	60.0	59.6	64.8	76.8	69.2	75.6	62.5	68.8	69.0	79.0
	DCP [53]	68.9	74.2	63.3	62.7	67.3	78.5	72.1	77.1	66.5	70.5	71.5	**82.7**
	CGFM	**72.2**	**77.1**	71.1	**64.3**	**71.2**	**79.5**	73.7	**77.3**	**74.2**	**70.2**	**73.9**	81.1

Table 2: Comparison of our method and other FSS methods on COCO-20^i. The best results are marked in bold

BackBone	Methods	1-shot						5-shot					
		Fold-0	Fold-1	Fold-2	Fold-3	Mean	FBIoU	Fold-0	Fold-1	Fold-2	Fold-3	Mean	FBIoU
VGG-16	PANet [12]	–	–	–	–	20.9	–	–	–	–	–	29.7	–
	FWB [46]	18.4	16.7	19.6	25.4	20.0	–	20.9	19.2	21.9	28.4	22.6	–
	PFENet [13]	35.4	38.1	36.8	34.7	36.3	–	38.2	42.5	41.8	38.9	40.4	–
	BAM [55]	36.4	47.1	43.3	41.7	42.1	–	42.9	51.4	48.3	46.6	47.3	–
	HDMNet [14]	40.7	50.6	48.2	44.0	45.9	–	47.0	56.5	54.1	51.9	52.4	**76.7**
	CGFM	**41.9**	**51.3**	**49.0**	**45.3**	**46.9**	**72.3**	**48.4**	**57.1**	**55.3**	**52.7**	**53.4**	–
ResNet50	PPNet [16]	28.1	30.8	29.5	27.7	29.0	–	39.0	40.8	37.1	37.3	38.5	–
	ASGNet [18]	–	–	–	–	34.6	–	–	–	–	–	42.5	
	CyCTR [23]	38.9	43.0	39.6	39.8	40.3	–	41.1	48.9	45.2	47.0	45.6	–
	DPCN [47]	42.0	47.0	43.2	39.7	43.0	63.2	46.0	54.9	50.8	47.4	49.8	67.4
	HSNet [21]	36.3	43.1	38.7	38.7	39.2	68.2	43.3	51.3	48.2	45.0	46.9	70.7
	ABCNet [19]	42.3	46.2	46.0	42.0	44.1	69.9	45.5	51.7	52.6	46.4	49.1	72.7

(Continued)

Table 2 (continued)

BackBone	Methods	1-shot						5-shot					
		Fold-0	Fold-1	Fold-2	Fold-3	Mean	FBIoU	Fold-0	Fold-1	Fold-2	Fold-3	Mean	FBIoU
	DiffSeg [56]	**45.2**	54.1	47.9	48.3	48.9	69.0	**50.7**	58.9	51.6	52.4	53.4	72.6
	CGFM	42.9	**57.1**	**52.8**	48.7	**50.4**	**74.0**	49.8	**63.2**	**57.4**	55.6	**56.5**	**78.0**
ResNet101	DCAMA [22]	41.5	46.2	45.2	41.3	43.5	–	48.0	58.0	54.3	47.1	51.9	–
	SSP [57]	39.1	45.1	42.7	41.2	42.0	–	47.4	54.5	50.4	49.6	50.2	–
	IPMT [58]	40.5	45.7	44.8	39.3	42.6	–	45.1	50.3	49.3	46.8	47.9	–
	NTRENet++ [52]	43.5	46.8	47.3	43.5	45.3	–	45.7	52.6	50.5	47.5	49.1	–
	CGFM	**44.4**	**57.1**	**53.8**	**49.7**	**51.3**	**74.5**	**50.9**	**64.2**	**58.4**	**56.7**	**57.6**	**79.1**

Table 3: Comparison of our method and other FSS methods on FSS-1000. The best results are marked in bold

Backbone	Methods	1-shot	5-shot
ResNet-50	HSNet [21]	85.5	86.5
	DCAMA [22]	88.2	88.8
	NTRENet++ [52]	89.3	90.4
	CGFM	**90.7**	**91.5**
ResNet-101	DAN [20]	85.2	88.1
	HSNet [21]	86.5	88.5
	DCAMA [22]	88.3	89.1
	NTRENet++ [52]	89.7	91.2
	CGFM	**91.2**	**92.7**

5.4 Ablation Study

We conducted ablation experiments on COCO-20^i to validate the effectiveness of the DPGBP module and the SGCRE module. Results in Table 4 show that the average mIoU of the baseline model (without DPGBP and SGCRE) under 1-shot and 5-shot settings are 47.4% and 53.8%, respectively. Performance improves significantly for 2.2% and 1.8% when DPGBP is added into the baseline model and improves for 1.3% and 1.0% when SGCRE is added into the baseline model. Best performances are obtained when DPGBP and SGCRE modules are both added into the baseline model, which achieves an improvement of 3.0% and 2.7% over the baseline model. In Fig. 4, we visualize the comparison between the baseline model and CGFM prediction results, with green highlighted areas representing the foreground region masks derived from ground truth and predictions. The results show that CGFM can effectively alleviate the impact of class-irrelevant noise in the baseline model and significantly enhance the model's prediction.

Table 4: Ablation studies on our proposed modules

DPGBP	SGCRE	1-shot mIoU (%)	5-shot mIoU (%)
		47.4	53.8
✓		49.6	55.6
	✓	48.7	54.8
✓	✓	50.4	56.5

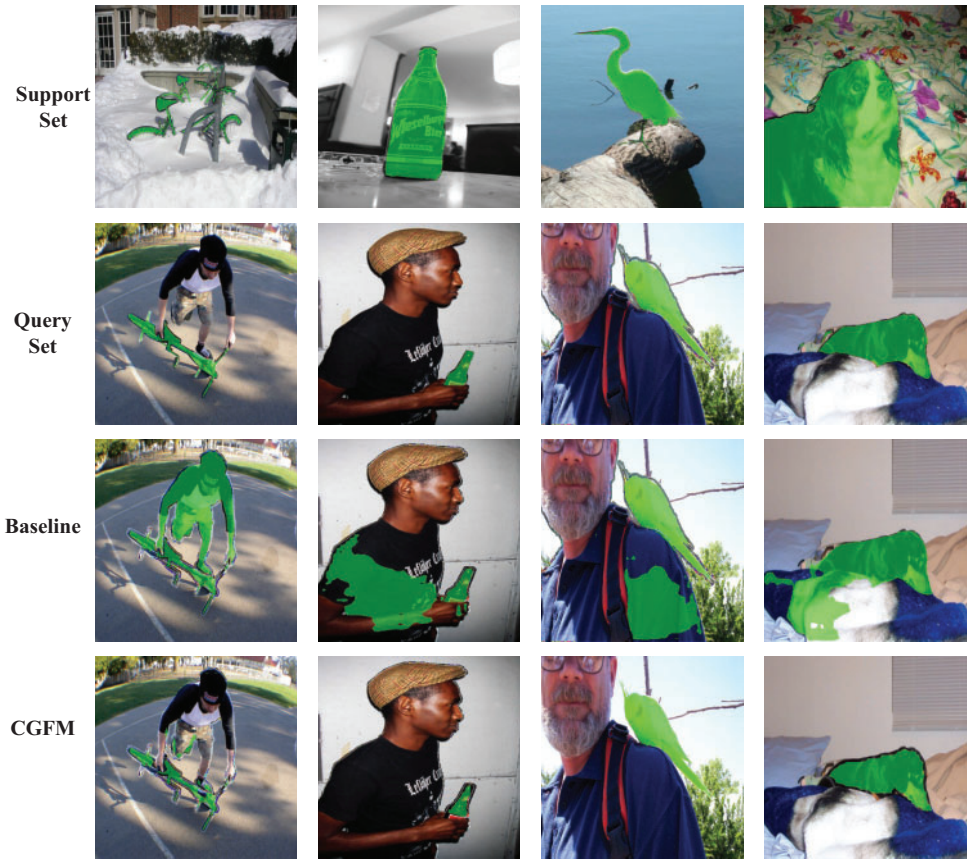

Figure 4: Comparison of CGFM model predictions with the baseline model

5.5 Analysis of DBGBP Module

The third row in Fig. 5 illustrates the effect of the DBGBP module, where the support set annotations and query set ground-truths are highlighted in green in the first row. The remaining three rows display heat maps of the feature activations. The results indicate that, without frequency enhancement, the original feature maps struggle to focus on category-relevant areas and instead show high responses in background and edge regions. After applying the DBGBP module, the model begins to focus on the target objects, and its response to the target class is gradually improved, in contrast to the surrounding background objects.

Table 5 explores the impact of using different class embedding strategies in DBGBP. When only a randomly initialized vector is used, as shown in the 'RI' column, the model's performance decreases by approximately 1.7% compared to the original results, highlighting the importance of category information in the embedding vector. When randomly initialized vectors are updated alongside sample tokens in the Transformer blocks, as shown in the 'RIU' column, the model learns both class and instance information from the query and support features, resulting in a performance improvement of around 1.2%. The 'PI' column shows that using class prototypes to initialize the class embedding vector results in a 1% improvement over the randomly initialized vector. The best performance is achieved in the 'PIU' column, where the prototype-initialized vector is updated in the subsequent Transformer blocks, underscoring the importance of the dual-perception mechanism in DBGBP.

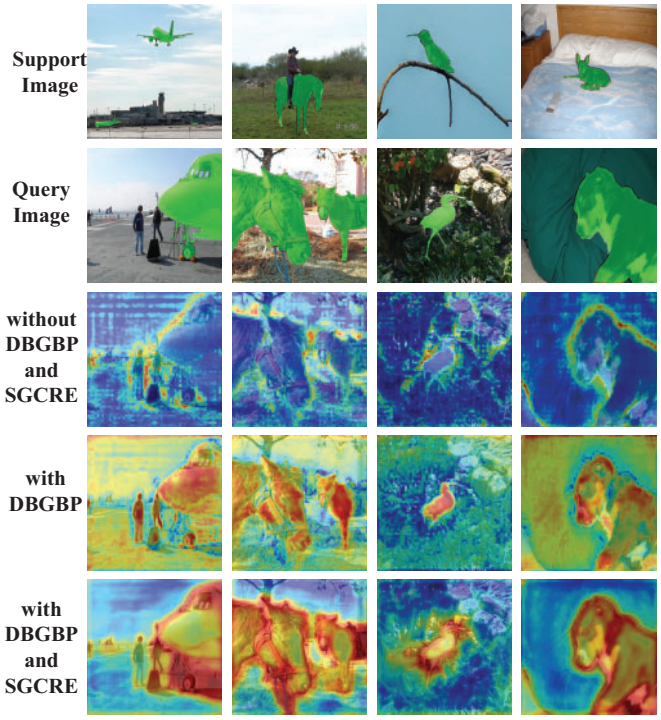

Figure 5: Comparison of the model's response to the target category

Table 5: Comparison of COCO-20^i 1-shot results using different initialization and update strategies for the class embedding vector in DBGBP

Vector type	RI	RIU	PI	PIU
mIoU (%)	48.7	49.9	49.7	50.4

Fig. 6 demonstrates cases where the generated Gaussian kernels change according to different classes and instances. These results show that DBGBP can dynamically adjust the Gaussian kernel generation process based on the inherent characteristics of the target class and instance. This ability ensures the activation of the most relevant frequency bands for the current task.

5.6 Analysis of SGCRE Module

The last row of Fig. 4 shows that, compared to the DBGBP, SGCRE further suppresses class-irrelevant background noise and effectively strengthens the model's response to the target class in fine-grained structures. The two-stage CGFM method gradually directs the model's focus toward the target objects, leading to a notable improvement in the model's performance. We investigated the impact of different scaling coefficients θ and γ in Eq. (5) on model performance, as shown in Fig. 7. The model achieves its best results when θ is 0.7 and γ is 0.5.

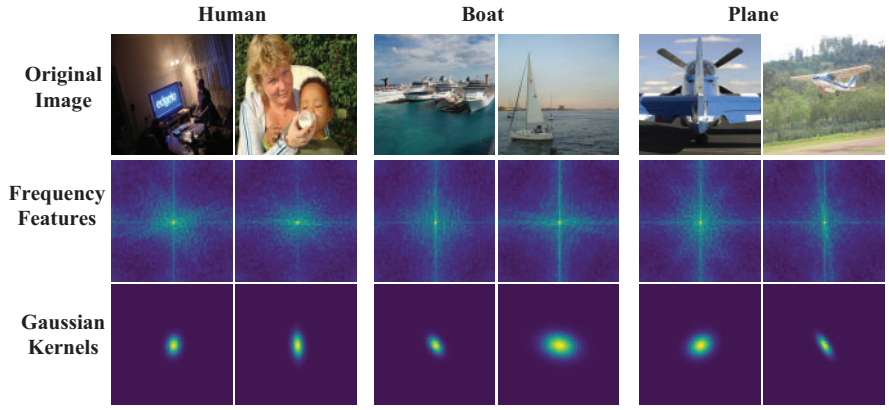

Figure 6: Gaussian kernels generated under different classes and instances

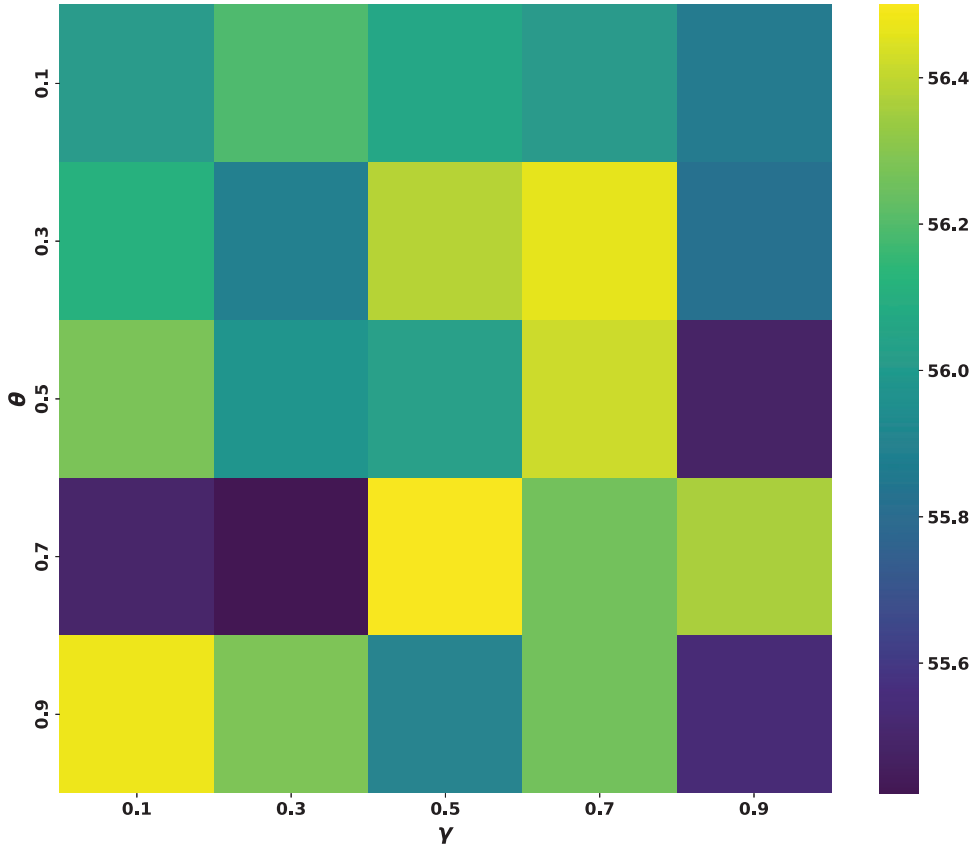

Figure 7: Comparison of model performance under different θ and γ

In Fig. 8, we compare feature activations using different frequency fusion methods. The first and second columns present images from the support and query sets. The target classes are marked by highlighted green areas. The third column displays the query features enhanced directly with the undecomposed support frequency features, while the fourth column shows the query features enhanced by SGCRE, in which support frequency features are decomposed to remove the positional information in the phase spectrum.

Experimental results reveal that incorporating raw support features without decomposition transfers spatial information from the support set to query features, adversely affecting model performance. As a comparison, when using decomposed support frequency features to enhance query features in SGCRE, the impact from support position information is largely alleviated and class-relevant information is successfully enhanced.

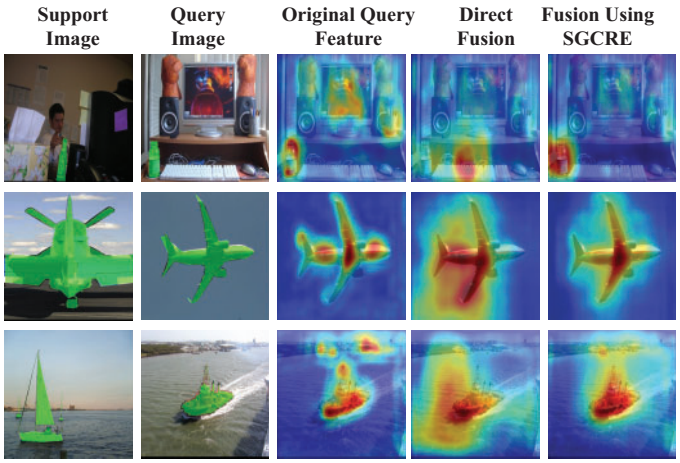

Figure 8: Comparison of query features enhanced using direct frequency fusion and SGCRE

5.7 Comparison of Results Using Different λ

We compared the model's performance using different objective function weights λ based on the 5-shot mIoU on COCO-20^i, as shown in Table 6. The results highlight the importance of λ in training the CGFM model. When λ is 0, the model is trained without any auxiliary supervision, and the average mIoU is approximately 55.3%. As the weight increases, the model's performance gradually improves, reaching a maximum of 56.5% when λ is 1.2. The hyperparameters in our method, including θ, γ, and λ, are all selected according to the performance using different values. To achieve the best performance, we use grid search to find out the optimal hyperparameter group. We compare the θ, γ values under different λ in Fig. 9. Results indicate that theta and gamma exhibit robust stability across a wide range of lambda values. This proves the stability of our method during training.

Table 6: The performance of models trained with different λ

λ	0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0
mIoU (%)	55.3	55.7	55.6	55.9	56.0	56.3	56.5	56.3	56.3	56.4	56.3

5.8 Comparison of Computation Cost

In Table 7, we present a comprehensive performance comparison of various methods in terms of mIoU, the number of parameters and FLOPS. Compared with other non-transformer methods, CGFM achieves a higher mIoU of 50.4% with only 2.3 million parameters. Compared to transformer-based methods (CyCTR and HDMNet), CGFM achieves a comparatively FLOPS amount. The above results demonstrate that our method achieves promising results with few additional computation costs.

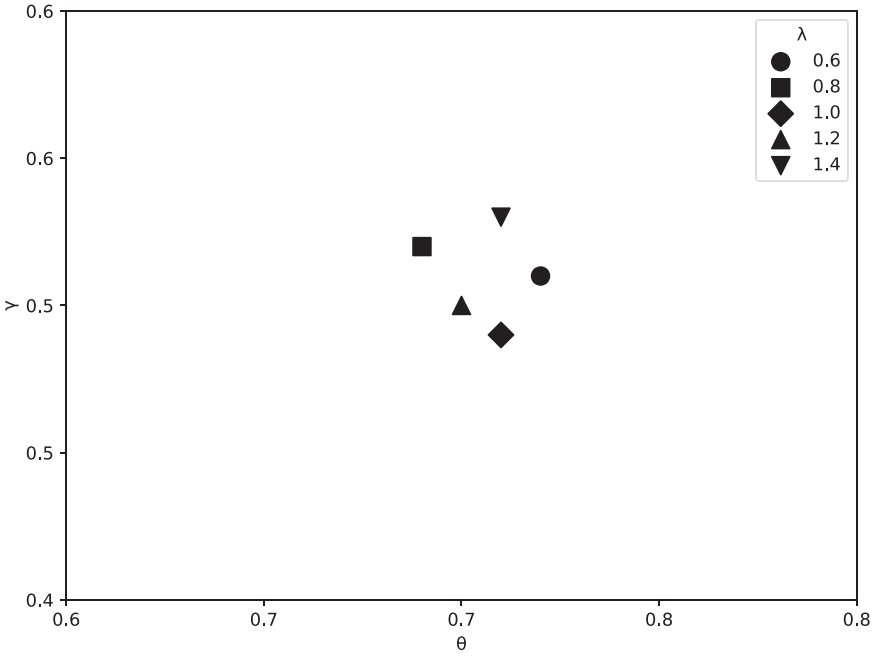

Figure 9: θ and γ value when using differen λ

Table 7: Performance comparison of different decoders

Method	Params (M)	FLOPs (G)
CyCTR [23]	5.6	95.0
HSNet [21]	2.6	20.8
BAM [55]	4.1	22.0
HDMNet [55]	2.7	8.8
DCP [53]	11.3	10.6
CGFM	2.3	10.2

6 Conclusion

We propose a two-stage Category-Guided Frequency Modulation (CGFM) method for Few-Shot Segmentation (FSS). In the first stage, we leverage category embedding vector-guided Gaussian kernels to activate target class-related frequency band regions in the Dual-Perception Gaussian Band Pre-activation (DBGBP) module. This module leverages dynamically generated Gaussian filter kernels to activate class-relevant frequency bands. The kernels are derived from class embedding vectors to ensure robust generalization across various categories and instances. These embeddings are initialized using class prototypes and subsequently refined through transformer blocks by incorporating contextual information from both support and query sets. In the second stage, we further enhance the response to the target class through the Support-Guided Category Response Enhancement (SGCRE) module. SGCRE specifically strengthens class-related components in activated frequency bands and suppresses class-irrelevant background noise. To achieve this and alleviate the impact of the position information, SGCRE decomposes the frequency features from the support set and leverages the amplitude parts as the target class representation to enhance

corresponding components on query features. Experiments show that our CGFM model achieves promising results, demonstrating the effectiveness of our approach.

7 Future Direction

CGFM alleviates the impact of structure information loss caused by class prototypes, but it still suffers from the inherent limitations of few-shot semantic segmentation. It enhances class-relevant information based on extracted example features, but there are only limited examples in the support set. This undermines the representational capability of the extracted class prototypes and further impacts the performance of the information enhancement module. Additionally, few-shot segmentation aims to reduce the annotation cost of the segmentation task. Although existing few-shot semantic segmentation settings have greatly reduced the number of annotated samples needed per class, the masks for the support and query sets used in training still require careful pixel-level annotation, which makes dataset production costly. In the future, on the one hand, we will work on improving feature extractors to obtain more representative example features, which may help to develop more effective frequency enhancement methods. On the other hand, we will resort to semi-supervised or self-supervised methods to further reduce the cost of training few-shot segmentation models and improve the efficiency in leveraging class information contained in provided samples.

Acknowledgement: Not applicable.

Funding Statement: None.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Yiming Tang; data collection: Yiming Tang; analysis and interpretation of results: Yiming Tang, Yanqiu Chen; draft manuscript preparation: Yiming Tang, Yanqiu Chen. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data available within the article.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Zhou B, Zhao H, Puig X, Fidler S, Barriuso A, Torralba A. Scene parsing through ADE20K dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017. p. 633–41.
2. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems 28 (NIPS 2015); 2015. p. 91–9.
3. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, et al. An image is worth 16 x 16 words: transformers for image recognition at scale. arXiv: 2010.11929. 2020.
4. Xie E, Wang W, Yu Z, Anandkumar A, Alvarez JM, Luo P. SegFormer: simple and efficient design for semantic segmentation with transformers. In: Advances in Neural Information Processing Systems 34 (NeurIPS 2021); 2021. Vol. 34, p. 12077–90.
5. Wang X, Zhang X, Cao Y, Wang W, Shen C, Huang T. SegGPT: towards segmenting everything in context. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV); 2023. p. 1130–40.
6. Qureshi AM, Butt AH, Alazeb A, Mudawi NA, Alonazi M, Almujally NA, et al. Semantic segmentation and YOLO detector over aerial vehicle images. Comput Mater Contin. 2024;80(2):3315–32. doi:10.32604/cmc.2024.052582.
7. Zhao N, Chua TS, Lee GH. Few-shot 3D point cloud semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 8873–82.

8. Saha O, Cheng Z, Maji S. Improving few-shot part segmentation using coarse supervision. In: European Conference on Computer Vision; 2022; Cham: Springer Nature Switzerland; p. 283–99.
9. Chen J, Li X, Zhang H, Cho Y, Hwang SH, Gao Z, et al. Adaptive dynamic inference for few-shot left atrium segmentation. Med Image Anal. 2024;98(1):103321. doi:10.1016/j.media.2024.103321.
10. Li X, Chen J, Zhang H, Cho Y, Hwang SH, Gao Z, et al. Hierarchical relational inference for few-shot learning in 3D left atrial segmentation. IEEE Trans Emerg Top Comput Intell. 2024;8(5):3352–67. doi:10.1109/TETCI.2024.3377267.
11. Shaban A, Bansal S, Liu Z, Essa I, Boots B. One-shot learning for semantic segmentation. arXiv:1709.03410. 2017.
12. Wang K, Liew JH, Zou Y, Zhou D, Feng J. PaNet: few-shot image semantic segmentation with prototype alignment. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2019. p. 9197–206.
13. Tian Z, Zhao H, Shu M, Yang Z, Jia J. Prior guided feature enrichment network for few-shot segmentation. IEEE Trans Pattern Anal Mach Intell. 2020;44(2):1050–65. doi:10.1109/TPAMI.2020.3013717.
14. Peng B, Tian Z, Wu X, Wang C, Liu S, Su J, et al. Hierarchical dense correlation distillation for few-shot segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023. p. 23641–51.
15. Zhu L, Chen T, Yin J, See S, Liu J. Addressing background context bias in few-shot segmentation through iterative modulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2024. p. 3370–9.
16. Liu Y, Zhang X, Zhang S, He X. Part-aware prototype network for few-shot semantic segmentation. In: Computer Vision—ECCV 2020: 16th European Conference; 2020 Aug 23–28; Glasgow, UK: Springer International Publishing; 2020. p. 142–58.
17. Yang B, Liu C, Li B, Jiao J, Ye Q. Prototype mixture models for few-shot semantic segmentation. In: Computer Vision–ECCV 2020: 16th European Conference; 2020 Aug 23–28; Glasgow, UK: Springer International Publishing; 2020. p. 763–78.
18. Li G, Jampani V, Sevilla-Lara L, Sun D, Kim J, Kim J. Adaptive prototype learning and allocation for few-shot segmentation. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA: IEEE; 2021. p. 8330–9.
19. Wang Y, Sun R, Zhang T. Rethinking the correlation in few-shot segmentation: a buoys view. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun 17–24; Vancouver, BC, Canada: IEEE; 2023. p. 7183–92.
20. Wang H, Zhang X, Hu Y, Yang Y, Cao X, Zhen X. Few-shot semantic segmentation with democratic attention networks. In: Computer Vision–ECCV 2020: 16th European Conference; 2020 Aug 23–28; Glasgow, UK: Springer International Publishing; 2020. p. 730–46.
21. Min J, Kang D, Cho M. Hypercorrelation squeeze for few-shot segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. p. 6941–52.
22. Shi X, Wei D, Zhang Y, Lu D, Ning M, Chen J, et al. Dense cross-query-and-support attention weighted mask aggregation for few-shot segmentation. In: Computer Vision—ECCV 2022. Cham: Springer Nature Switzerland; 2022. p. 151–68.
23. Zhang G, Kang G, Yang Y, Wei Y. Few-shot segmentation via cycle-consistent transformer. Adv Neural Inform Process Syst. 2021;34:21984–96.
24. Qin Z, Zhang P, Wu F, Li X. FcaNet: frequency channel attention networks. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV); 2021 Oct 10–17; Montreal, QC, Canada: IEEE; 2021. p. 763–72.
25. Zhang X, Wei Y, Yang Y, Huang TS. SG-One: similarity guidance network for one-shot semantic segmentation. IEEE Trans Cybern. 2020;50(9):3855–65. doi:10.1109/TCYB.2020.2992433.
26. Zhang B, Xiao J, Qin T. Self-guided and cross-guided learning for few-shot segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 8312–21.
27. Ouyang C, Biffi C, Chen C, Kart T, Qiu H, Rueckert D. Self-supervision with superpixels: training few-shot medical image segmentation without annotation. In: Computer Vision—ECCV 2020. Cham: Springer International Publishing; 2020. p. 762–80.

28. Zhu K, Zhai W, Zha ZJ, Cao Y. Self-supervised tuning for few-shot segmentation. arXiv:2004.05538. 2020.
29. Amac MS, Sencan A, Baran B, Ikizler-Cinbis N, Cinbis RG. Masksplit: self-supervised meta-learning for few-shot semantic segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision; 2022. p. 1067–77.
30. Karimijafarbigloo S, Azad R, Merhof D. Self-supervised few-shot learning for semantic segmentation: an annotation-free approach. In: International Workshop on Predictive Intelligence In Medicine; 2023; Springer. p. 159–71.
31. Baxes GA. Digital image processing: principles and applications. John Wiley & Sons, Inc.; 1994.
32. Pitas I. Digital image processing algorithms and applications. Vol. 2. John Wiley & Sons, Inc.; 2000. p. 133–8.
33. Li S, Xue K, Zhu B, Ding C, Gao X, Wei D, et al. FALCON: a fourier transform based approach for fast and secure convolutional neural network predictions. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA: IEEE. p. 8702–11.
34. Yu H, Zheng N, Zhou M, Huang J, Xiao Z, Zhao F. Frequency and spatial dual guidance for image dehazing. In: Computer Vision—ECCV 2022. Cham: Springer Nature Switzerland; 2022. p. 181–98.
35. Wang C, Jiang J, Zhong Z, Liu X. Spatial-frequency mutual learning for face super-resolution. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun 17–24; Vancouver, BC, Canada: IEEE; 2023. p. 22356–66.
36. Lin S, Zhang Z, Huang Z, Lu Y, Lan C, Chu P, et al. Deep frequency filtering for domain generalization. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun 17–24; Vancouver, BC, Canada: IEEE; 2023. p. 11797–807.
37. Naqvi RA, Haider A, Kim HS, Jeong D, Lee SW. Transformative noise reduction: leveraging a transformer-based deep network for medical image denoising. Mathematics. 2024;12(15):2313. doi:10.3390/math12152313.
38. Zhang F, Panahi A, Gao G. FsaNet: frequency self-attention for semantic segmentation. IEEE Trans Image Process. 2023;32:4757–72. doi:10.1109/TIP.2023.3305090.
39. Dong B, Wang P, Wang F. Head-free lightweight semantic segmentation with linear transformer. Proc AAAI Conf Artif Intell. 2023;37(1):516–24. doi:10.1609/aaai.v37i1.25126.
40. Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, et al. Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning; 2021; PMLR. p. 8748–63.
41. Gonzalez RC. Digital image processing. India: Pearson education; 2009.
42. Li X, Wei T, Chen YP, Tai YW, Tang CK. FSS-1000: a 1000-class dataset for few-shot segmentation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA: IEEE; 2020. p. 2866–75.
43. Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A. The pascal visual object classes (VOC) challenge. Int J Comput Vis. 2010;88(2):303–38. doi:10.1007/s11263-009-0275-4.
44. Hariharan B, Arbeláez P, Girshick R, Malik J. Simultaneous detection and segmentation. In: Computer Vision—ECCV 2014. Cham: Springer International Publishing; 2014. p. 297–312.
45. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft coco: common objects in context. In: Computer Vision—ECCV 2014. Cham: Springer International Publishing; 2014. p. 740–55.
46. Nguyen K, Todorovic S. Feature weighting and boosting for few-shot segmentation. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV); 2019 Oct 27–Nov 2; Seoul, Republic of Korea: IEEE. p. 622–31.
47. Liu J, Bao Y, Xie GS, Xiong H, Sonke JJ, Gavves E. Dynamic prototype convolution network for few-shot semantic segmentation. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022 Jun 18–24; New Orleans, LA, USA: IEEE; 2022. p. 11543–52.
48. Rakelly K, Shelhamer E, Darrell T, Efros A, Levine S. Conditional networks for few-shot semantic segmentation. In: International Conference on Learning Representations; 2018.
49. Zhang C, Lin G, Liu F, Yao R, Shen C. CaNet: class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019 Jun 15–20; Long Beach, CA, USA: IEEE; 2019. p. 5212–21.

50. Zhang C, Lin G, Liu F, Guo J, Wu Q, Yao R. Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV); 2019 Oct 27–Nov 2; Seoul, Republic of Korea: IEEE; 2019. p. 9586–94.
51. Bao X, Qin J, Sun S, Wang X, Zheng Y. Relevant intrinsic feature enhancement network for few-shot semantic segmentation. Proc AAAI Conf Artif Intell. 2024;38(2):765–73. doi:10.1609/aaai.v38i2.27834.
52. Liu Y, Liu N, Wu Y, Cholakkal H, Anwer RM, Yao X, et al. NTRENet++: unleashing the power of non-target knowledge for few-shot semantic segmentation. IEEE Trans Circuits Syst Video Technol. 2024.
53. Lang C, Cheng G, Tu B, Han J. Few-shot segmentation via divide-and-conquer proxies. Int J Comput Vis. 2024;132(1):261–83. doi:10.1007/s11263-023-01886-8.
54. Wang Y, Sun R, Zhang Z, Zhang T. Adaptive agent transformer for few-shot segmentation. In: European Conference on Computer Vision; 2022; Springer. p. 36–52.
55. Lang C, Cheng G, Tu B, Han J. Learning what not to segment: a new perspective on few-shot segmentation. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022 Jun 18–24; New Orleans, LA, USA: IEEE; 2022. p. 8047–57.
56. Shi G, Zhu W, Wu Y, Zhao D, Zheng K, Lu T. Few-shot semantic segmentation via perceptual attention and spatial control. In: Proceedings of the 32nd ACM International Conference on Multimedia; 2024. p. 5374–83.
57. Fan Q, Pei W, Tai YW, Tang CK. Self-support few-shot semantic segmentation. In: European Conference on Computer Vision; 2022; Cham: Springer Nature Switzerland. p. 701–19.
58. Liu Y, Liu N, Yao X, Han J. Intermediate prototype mining transformer for few-shot semantic segmentation. Adv Neural Inform Process Syst. 2022;35:38020–31.