BTGNET: A BACKGROUND TEXT GUIDED MODEL WITH JOINT QUERY REFINEMENT FOR FEW-SHOT SEMANTIC SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing few-shot semantic segmentation (FSS) methods have extensively explored visual background information to suppress distracting regions, while multimodal approaches mainly rely on foreground-oriented text prompts. In contrast, the potential of background words as an explicit textual prior remains largely unexplored. We observe that category-related background words can provide complementary language cues for distinguishing target objects from visually confusing regions, while being easier to obtain and extend than image-specific background annotations. Based on this insight, we propose BTGNet, a Background Text Guided Network that explicitly incorporates background words into few-shot segmentation. Specifically, we employ a large language model to automatically generate task-adaptive, category-related background vocabularies and align them with visual representations through a frozen CLIP encoder to guide Grad-CAM-based query activation. To further improve the coarse activations produced by frozen CLIP, we propose a joint refinement strategy that combines fixed attention priors with learnable dynamic refinement. Extensive experiments on PASCAL-5^i and COCO-20^i demonstrate consistent improvements over state-of-the-art methods. Further analyses show that semantically relevant background words, rather than arbitrary additional text, are critical to performance gains, highlighting background words as a generalizable textual prior for multimodal FSS.

1 INTRODUCTION

With the rise of deep learning, semantic segmentation has achieved remarkable progress across domains such as autonomous drivingChen et al. (2024a); Krishna et al. (2024), medical imagingLing et al. (2024); Wang et al. (2024b), and industrial inspectionLi et al. (2024); Zhang et al. (2025). However, practical deployment is usually hindered by the scarcity of annotated data. Few-shot semantic segmentation (FSS) addresses this challenge by enabling accurate segmentation from only a few annotated samples, thereby improving generalization.Lang et al. (2022).

When faced with the recognition of unknown categories, most FSS methods mimic the human processing procedure by comparing the feature similarities between the query and the support image Lang et al. (2022); Peng et al. (2023) or mining pixel-level relationships Shi et al. (2022); Liu et al. (2022a). Although effective, these methods usually overlook the contribution of background context in aiding object recognition. Thus, recent works introduce background learning to filter noise and irrelevant base classes more effectively Huang et al. (2025); Liu et al. (2022b). However, most approaches remain limited to the visual modality and fail to exploit richer semantics available in other modalities. The emergence of multimodal models offers new opportunities Radford et al. (2021); Liu et al. (2024). Based on this, some studies have introduce text modalities to guide segmentation, but mainly focus on foreground descriptions, still neglecting the crucial role of background text.

In fact, when distinguishing new classes, background information can help human reasoning. For example, for the images with an eagle in the sky and a swan in a lake, background cues like 'sky' or 'lake' can help filter irrelevant regions and guide object localization. In addition, relying solely on foreground text can misidentify similar objects, as eagles are misidentified as swans. However, such confusion can be reduced through reasoning about background information, for example, eagles are

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.

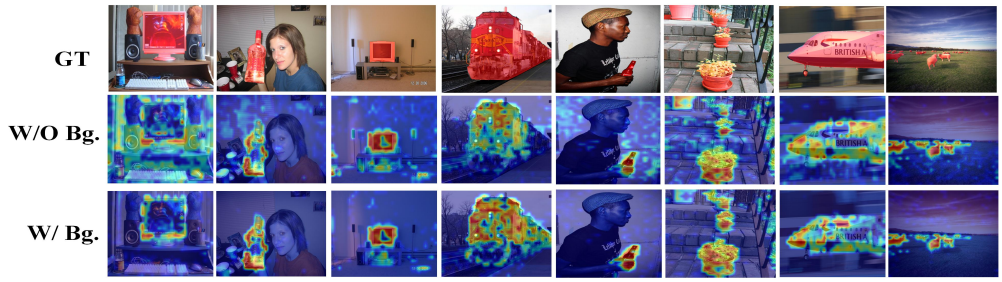

Figure 1: Comparisons of Grad-CAM activation maps with and without background word information. Here, w/o Bg. denotes without background words, w/ Bg. denotes with background words.

rarely seen in the water and swans generally do not appear in the sky. To validate the above claim, we generated activation maps with foreground and combined foreground-background prompts. Figure 1 illustrates the comparison of activation maps. It is obvious that the integration of background textual information can generate more accurate activation maps. It effectively filters out irrelevant regions and enhances the localization of target objects, highlighting the value of background text in FSS. Moreover, both foreground and background descriptions of the support image are available in typical few-shot settings. However, significant discrepancies usually exist between the backgrounds of support and query images, which prevents the direct transfer of background textual cues. Thus, an effective strategy is required to accurately infer and utilize category-specific background text priors.

To address these issues, we propose BTGNet, a novel FSS model that leverages textual background information to improve generalization. Specifically, BTGNet uses an LLM-assisted background text generation module to automatically construct task-adaptive, class-related background vocabularies without manual specification, which are aligned with visual features through CLIP and used to guide Grad-CAM Selvaraju et al. (2017) for query activation map generation. In addition, to mitigate limitations caused by frozen CLIP, we propose a joint optimization strategy by combining dynamic and fixed refinement. The dynamic branch enables learnable updates to the activation maps, while the fixed branch provides stable regularization. The main contributions of this work are as follows.

(1) We reveal that category-related background words provide complementary textual priors for foreground discrimination in FSS, and introduce an LLM-based approach to efficiently construct such background cues, substantially improving activation quality and segmentation performance.

(2) To address the limitations of static activation maps from frozen CLIP, we propose a hybrid dynamic-fixed refinement strategy that improves segmentation precision and robustness.

(3) Extensive experiments validate the effectiveness of BTGNet and reveal how background words and joint optimization affect model performance, offering insights for future framework design.

2 Related Work

2.1 Visual Background Correlation

Although most existing FSS methods primarily focus on foreground correlations, recent studies have demonstrated that visual background information can also help suppress noise and improve segmentation accuracy. BAM Lang et al. (2023) learns what not to segment by exploiting non-target base-class cues, while PFENet++ Luo et al. (2023) filters noisy contextual priors to improve foreground-background discrimination. BLPLNet Wang et al. (2025b) learns background prototypes from non-target regions to suppress noise. Similarly, NTRNetLiu et al. (2022b) uses background prototypes to remove background interference and enhance foreground segmentation. FBINetHuang et al. (2025) optimizes background prototypes using reversed foreground masks to improve accuracy.

2.2 Vision-Language Alignment

With advancements in multimodal models, incorporating textual features has emerged as a novel method. LLaFS++Zhu et al. (2025) leverages LLM with fine-grained instructions for segmentation

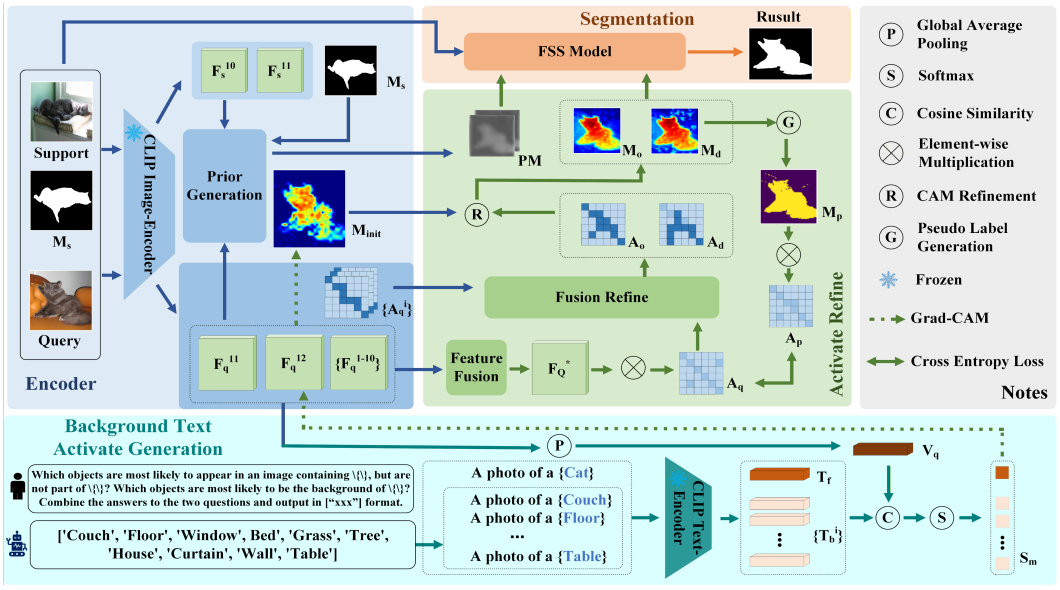

Figure 2: The overall BTGNet consists of four components: the Encoder uses the CLIP Image Encoder to extract features; the Background Text Activation Generation produces activation maps to optimize query features; the Activation Refinement further refines these activation maps; the Segmentation part leverages the refined maps to improve existing FSS methods.

tasks. MIANetYang et al. (2023) uses semantic word embeddings and instance details for precise segmentation. CLIPSegLüddecke & Ecker (2021) first introduces the CLIP model into FSS task. However, it primarily treats CLIP as a verification method. DPNetChen et al. (2025) incorporates CLIPLin et al. (2023) to generate combined language-image prototypes. PI-CLIPWang et al. (2024a) applies CLIP-guided prompts, but its fixed refinement method restricts performance.

3 Method

3.1 Overall Architecture

Figure 2 illustrates the overall structure of BTGNet, and the model consists of four main components: CLIP Prior Generation, Background Text Activation Generation, Activation Refinement, and Segmentation. Details of each component are described in the following subsections.

3.2 CLIP Prior Generation

To better exploit the information from vision and language, we utilize the CLIP image encoder due to its strong capability in capturing rich semantic representations aligned with textual concepts. Unlike conventional backbones pre-trained for classification tasks, the CLIP encoder inherently supports cross-modal alignment, making it suited for leveraging textual cues in segmentation.

The CLIP Prior Generation takes the support image (I_s), the query image (I_q), and the support label (M_s) as input. During feature encoding, the I_q and I_s filtered by M_s are fed into the CLIP image encoder to extract features and attention maps. The process is described in Eqs.1 and Eq.2.

$$\left\{\left(F_s^i, A_s^i\right)\right\}_{i=1}^{12} = \text{CLIP}_v(I_s \odot M_s) \quad (1) \qquad \left\{\left(F_q^i, A_q^i\right)\right\}_{i=1}^{12} = \text{CLIP}_v(I_q) \quad (2)$$

where $\odot$ denotes the element-wise product. $\text{CLIP}_v(\cdot)$ denotes the intermediate features and attention matrices extracted from the CLIP image encoder. F_s^i and F_q^i represent the feature outputs for the I_s and I_q, and A_s^i and A_q^i denote the attention matrices for the I_s and I_q in the i-th layer.

For the i-th layer of the CLIP image encoder, the input $x^i \in \mathbb{R}^{N \times D}$ is obtained by applying patch embedding to the input image. It is then linearly projected to produce the query (Q_h^i), key (K_h^i),

and value (V_h^i) for each attention head. Within each layer, the attention matrix A^i is obtained by averaging the multi-head attention matrices A_h^i across all heads, as shown in Eq.3 and Eq.4.

$$A^i = \frac{1}{M}\sum_{h=1}^{M} A_h^i \in \mathbb{R}^{N\times N} \quad (3) \qquad A_h^i = \text{softmax}\left(\frac{Q_h^i (K_h^i)^\top}{\sqrt{d_k^i}}\right) \in \mathbb{R}^{N\times N} \quad (4)$$

where M is the number of attention heads and d_k^i is the dimensionality of the vectors. The output F^i is obtained by concatenating the outputs from all multi-head results, then applying linear projection and reshaping operation ($\mathcal{R}(\cdot)$), as shown in Eq.5.

$$F^i = \mathcal{R}(\text{cat}(\{A_h^i V_h^i | h \in [1, M]\})) \in \mathbb{R}^{C\times H\times W} \quad (5)$$

In addition to CLIP Prior Generation, we follow Tian et al. (2020) to generate prior mask and choose the support and query features from the 10th and 11th layers in CLIP to calculate the prior mask PM, as these layers provide high-level semantic abstraction and can suppress background noise.

3.3 Background Text Activation Generation

Existing multimodal FSS methods typically rely on fixed text prompts centered on foreground categories, while overlooking background semantics that can help distinguish target regions. Manually defining category-specific background vocabularies is difficult to scale, as each foreground class may correspond to diverse contextual concepts and the vocabulary must be redesigned when transferring to new datasets or tasks. To address this, we introduce an LLM-based background word generator (LGBW), which automatically constructs category-adaptive background vocabularies from the foreground label space. Given a foreground category c, LGBW queries the LLM with: (1) What objects are most likely to appear in an image containing $\{c\}$ but not belonging to $\{c\}$? and (2) What background contextual objects commonly appears around $\{\text{c}\}$? The responses are parsed into word lists. LGBW is executed once per category and the generated vocabularies are then fixed, with base-class words generated before training and novel-class words generated only at evaluation (Appendix C).

For activation map generation, we adopt Grad-CAM Selvaraju et al. (2017) on the frozen CLIP image encoder to produce an initial activation map. We encode the foreground category and the generated background words with a fixed CLIP prompt template `'a photo of a {}'`, obtaining the foreground text feature T_f and background text features $\{T_b^i\}_{i=1}^B$ and concatenate them as

$$T = \text{concat}(T_f, \{T_b^i \mid i \in [1, B]\}), \quad (6)$$

and compute the similarity between the pooled last-layer visual feature V_q and all text features, followed by softmax normalization:

$$S_m = \text{softmax}(\text{Sim}(T, V_q)). \quad (7)$$

Here, B denotes the number of background texts, $\text{Sim}(\cdot)$ is cosine similarity, and V_q is the globally pooled final-layer query feature F_q^{12}. To mitigate conflicts among background texts, we define a one-vs-all indicator $y = [1, 0, \ldots, 0] \in \mathbb{R}^{B+1}$ (foreground marked as 1 and backgrounds as 0), and formulate the Grad-CAM objective as:

$$\text{S}_\text{c}(y, S_m) = \sum_{k=0}^{B} y_k \log S_m(k). \quad (8)$$

where $S_m(0)$ is the foreground score. Since S_c is monotonic with respect to the foreground confidence, we use ∇S_c to compute Grad-CAM saliency. Then we apply Grad-CAM by computing gradients of the target score S_c with respect to the last-layer features:

$$g_{i,j}^k = \frac{\partial S_c}{\partial (F_q^{12})_{i,j}^k}, \quad (9)$$

where $g_{i,j}^k$ denotes the gradient at spatial location (i, j) of the k-th channel. We then globally average the gradients to obtain the channel importance weights in Eq. 10, and compute the initial activation map by a weighted sum of feature maps followed by ReLU in Eq. 11.

$$\alpha_k = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W} g_{i,j}^k \quad (10) \qquad M_{\text{init}}(i, j) = \text{ReLU}\left(\sum_{k=1}^{K} \alpha_k (F_q^{12})_{i,j}^k\right) \quad (11)$$

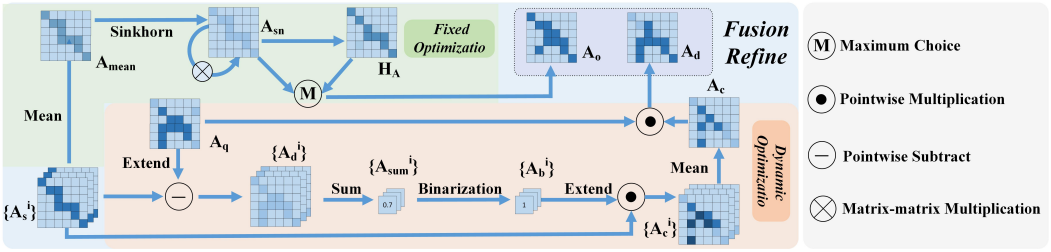

Figure 3: Joint Activation Refinement Strategy. The fixed path refines CLIP attention with averaging and Sinkhorn normalization, while the dynamic path builds adaptive matrices via attention selection.

Table 1: Comparison with other state-of-the-arts on PASCAL-5^i for 1-shot and 5-shot setting.

Methods	BackBone	1-shot					5-shot				
		Fold-0	Fold-1	Fold-2	Fold-3	Mean	Fold-0	Fold-1	Fold-2	Fold-3	Mean
PFENet (TPAMI'20)Tian et al. (2020)	ResNet-50	61.7	69.5	55.4	56.3	60.8	63.1	70.7	55.8	57.9	61.9
HSNet (ICCV'21)Min et al. (2021)	Swin/B	67.9	74.0	60.3	67.0	67.3	72.2	77.5	64.0	72.6	71.6
DCAMA (ECCV'22)Shi et al. (2022)	Swin/B	72.2	73.8	64.3	67.1	69.3	75.7	77.1	72.0	74.8	74.9
FPTrans (NIPS'22)Zhang et al. (2022)	DeiT/B	72.3	70.6	68.3	64.1	68.8	76.7	79.0	81.0	75.1	78.0
HDMNet (CVPR'23)Peng et al. (2023)	ResNet-50	71.0	75.4	68.9	62.1	69.4	71.3	76.2	71.3	68.5	71.8
DeFSS (ICCV'25)Qin et al. (2025)	ResNet-50	73.7	76.6	70.9	65.8	71.7	75.0	78.6	74.7	72.2	75.1
DpFSS (arXiv'25)	DINOv2	75.6	80.3	72.2	73.4	75.4	78.1	83.1	73.6	76.3	77.8
PGMA-Net (TMM'24)Chen et al. (2024b)	CLIP-ViT/B	74.0	81.9	66.8	73.7	74.1	74.5	82.2	67.2	74.4	74.6
PI-CLIP (CVPR'24)Wang et al. (2024a)	CLIP-RN50	76.4	83.5	74.7	72.8	76.8	76.7	83.8	75.2	73.2	77.2
VRP-SAM (CVPR'24)Sun et al. (2024)	SAM-RN50	74.5	77.3	69.5	65.8	71.8	76.3	76.8	69.5	63.1	71.4
FS-SAM2 (ICIAP'25)Forni et al. (2025)	SAM2-RN50	74.0	79.6	73.5	66.7	73.4	75.7	79.6	75.3	68.8	74.8
USD (TNNLS'25)Wang et al. (2025a)	SAM-CLIP-RN50	79.8	82.9	76.6	74.1	78.4	80.2	83.1	76.8	74.3	78.6
DSV-LFS (CVPR'25)Karimi & Poullis (2025)	LLaVA-SAM-RN50	71.7	82.0	71.2	75.0	75.0	72.0	82.0	71.3	75.5	75.2
FCP (AAAI'25)Park et al. (2025)	SAM-RN50	74.9	77.4	71.8	68.8	73.2	77.2	78.8	72.2	67.7	74.0
FDCMNet (ESWA'26)Chen et al. (2026)	CLIP-RN101	79.3	85.4	77.3	76.3	79.6	79.5	**85.9**	78.1	76.4	79.9
BTGNet-P (ours)	CLIP-RN50	78.7	85.0	76.3	77.5	79.4	79.0	85.2	77.1	78.1	79.9
BTGNet-H (ours)	CLIP-RN50	79.8	85.6	78.4	78.1	80.5	79.9	85.6	80.0	78.1	80.9
BTGNet-D (ours)	CLIP-ViT	**80.7**	**86.8**	**79.4**	**79.0**	**81.5**	**81.6**	**86.9**	**81.4**	**79.6**	**82.4**

3.4 Activation Refinement

The initial Grad-CAM activation maps are usually coarse, limiting model ability to guide segmentation. To address this, we propose a joint refinement strategy with two complementary branches: a fixed refinement branch and a learnable dynamic refinement branch. The fixed branch leverages intermediate attention maps from the frozen CLIP encoder to suppress background noise and provide stable semantic constraints, while the dynamic branch learns adaptive feature patterns to enhance foreground responses and recover target details. The two matrices are jointly applied to refine the activation maps, improving foreground localization while reducing background interference, as shown in Figure 3. First, to generate the fixed refinement matrix A_o, we average the intermediate attention features A_s^i to obtain A_{mean}, and apply Sinkhorn normalization Sinkhorn (1964) to obtain the normalized matrix A_{sn}. Then, we compute the higher-order optimization matrix H_A by Eq.12.

$$H_A = A_{\text{sn}} A_{\text{sn}}^T \tag{12}$$

The final fixed refinement matrix A_o is obtained by taking the element-wise maximum between H_A and A_{sn}, which preserves high-saliency local attention regions and improves activation refinement. To overcome the limited diversity of fixed refinement, we further construct a dynamic refinement approach using intermediate CLIP visual features. Specifically, a learnable layer is introduced to extract pattern features from the intermediate feature set F_Q^*, as defined in Eq. 13.

$$F_Q^* = \text{conv}\left(\text{cat}\left(\{F_q^i | i \in [1, 12]\}\right)\right) \in \mathbb{R}^{C \times H \times W} \tag{13}$$

where $\text{conv}(\cdot)$ denotes the convolution. The dynamic refinement matrix A_q is generated by Eq.14.

$$A_q = \mathcal{R}(F_Q^*)\mathcal{R}(F_Q^*)^T \in \mathbb{R}^{HW \times HW} \tag{14}$$

A_q is further refined with the intermediate attention matrices A_s^i. We compute the distance between each A_s^i and A_q using the summed element-wise absolute difference, and use the average distance as a selection threshold. The attention matrices with distances below this threshold are selected and

Table 2: Comparison with other state-of-the-arts on COCO-20^i for 1-shot and 5-shot setting.

Methods	BackBone	1-shot					5-shot				
		Fold-0	Fold-1	Fold-2	Fold-3	Mean	Fold-0	Fold-1	Fold-2	Fold-3	Mean
PFENet (TPAMI'20)Tian et al. (2020)	ResNet-101	34.3	33.0	32.3	30.1	32.4	38.5	38.6	38.2	34.3	37.4
HSNet (ICCV'21)Min et al. (2021)	Swin/B	43.6	49.9	49.4	46.4	47.3	50.1	58.6	56.7	55.1	55.1
DCAMA (ECCV'22)Shi et al. (2022)	Swin/B	49.5	52.7	52.8	48.7	50.9	55.4	60.3	59.9	57.5	58.3
FPTrans (NIPS'22)Zhang et al. (2022)	DeiT/B	44.4	48.9	50.6	44.0	47.0	54.2	62.5	61.3	57.6	58.9
HDMNet (CVPR'23)Peng et al. (2023)	ResNet-50	43.8	55.3	51.6	49.4	50.0	50.6	61.6	55.7	56.0	56.0
DeFSS (ICCV'25)Qin et al. (2025)	ResNet-50	46.5	59.7	54.8	53.2	53.7	55.1	66.0	61.3	57.5	60.0
DpFSS (arXiv'25)	DINOv2	58.6	66.0	**62.3**	62.6	62.4	62.1	70.2	66.1	63.0	65.4
LLaFS (CVPR'24)Zhu et al. (2024)	Llama-QFormer	47.5	58.8	56.2	53.0	53.9	53.2	63.8	63.1	60.0	60.0
PI-CLIP (CVPR'24)Wang et al. (2024a)	CLIP-RN50	49.3	65.7	55.8	56.3	56.8	56.4	66.2	55.9	58.0	59.1
VRP-SAM (CVPR'24)Sun et al. (2024)	SAM-RN50	44.3	54.3	52.3	50.0	50.2	50.5	59.5	56.9	54.9	55.5
FS-SAM2 (ICIAP'25)Forni et al. (2025)	SAM2-RN50	49.3	57.6	58.3	56.1	55.3	55.1	60.6	58.7	59.3	58.4
USD (TNNLS'25)Wang et al. (2025a)	SAM-CLIP-RN50	52.6	63.5	58.4	57.0	57.9	53.8	64.0	59.8	57.2	58.7
FCP (AAAI'25)Park et al. (2025)	SAM-RN50	46.4	56.4	55.3	51.8	52.5	52.6	63.3	59.8	56.1	58.0
FDCMNet (ESWA'26)Chen et al. (2026)	CLIP-RN101	53.4	66.8	56.9	56.7	58.5	57.5	67.6	57.0	58.8	60.2
BTGNet-P (ours)	CLIP-RN50	55.5	68.3	60.4	61.1	61.3	57.7	70.7	65.3	62.9	64.2
BTGNet-H (ours)	CLIP-RN50	56.8	69.9	59.3	62.3	62.1	57.9	69.8	62.9	63.4	63.5
BTGNet-D (ours)	CLIP-ViT	**59.5**	**67.4**	62.1	**63.9**	**63.2**	**62.7**	**70.5**	**66.5**	**64.8**	**66.1**

averaged to obtain A_c, which captures stable and consistent attention patterns. The final dynamic refinement matrix A_d is then generated by element-wise multiplying A_q and A_c. Finally, we refine M_{init} by performing matrix multiplication with the dynamic refinement matrix A_d and the fixed refinement matrix A_o, obtaining the final refined activation maps M_o and M_d, respectively.

3.5 Segmentation

Our framework is plug-and-play with any encoder-decoder based FSS model. Given a FSS backbone with encoder $E(\cdot)$ and decoder $D(\cdot)$, first extract query features F_q and support features F_s from the query and support image. Meanwhile, our refinement branch produces three prior maps M_o, M_d, and P_M, concatenated the aligned priors with query feature:

$$\hat{F}_q = \text{Concat}(F_q, M_o, M_d, P_M), \quad F'_q = \phi(\hat{F}_q), \tag{15}$$

where $\phi(\cdot)$ is a lightweight 1×1 projection to keep the channel size unchanged. Finally, the segmentation result P is predicted by the original FSS decoder as $P = D(F'_q, F_s)$.

3.6 Design of Loss Function

The total loss of BTGNet comprises the segmentation loss L_s and dynamic refinement loss L_d. The segmentation loss L_s is used to guide the model to produce accurate predictions, defined as Eq.16.

$$L_s = CE(P, GT) \tag{16}$$

where CE denotes the cross-entropy loss and GT is the ground truth label.

To improve structural consistency and suppress noisy patterns in dynamic refinement, BTGNet introduces a dynamic refinement loss L_d. We first generate a pseudo-label map M_p by thresholding the prior map M_d with τ, and construct the affinity target $A_p = mm^\top$, where $m = \text{vec}(M_p)$. Each row of A_p and A_q is treated as a token-to-token distribution, and softmax is applied before computing the cross-entropy loss:

$$L_d = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\left[\text{softmax}(A_p)\right]_{ij}\log\left[\text{softmax}(A_q)\right]_{ij}, \tag{17}$$

where N denotes the number of tokens. The overall loss is defined as $L_{\text{all}} = \sigma L_d + \omega L_s$, where σ (default=0.1) and ω (default=1.0) are the weights of L_d and L_s, respectively.

4 Experiments and Results

4.1 Dataset and Implementation Details

We evaluate BTGNet on PASCAL-5^i Shaban et al. (2017) and COCO-20^i Nguyen & Todorovic (2019). Dataset and implementation include seeds and hyp-parameter are provided in Appendix A.

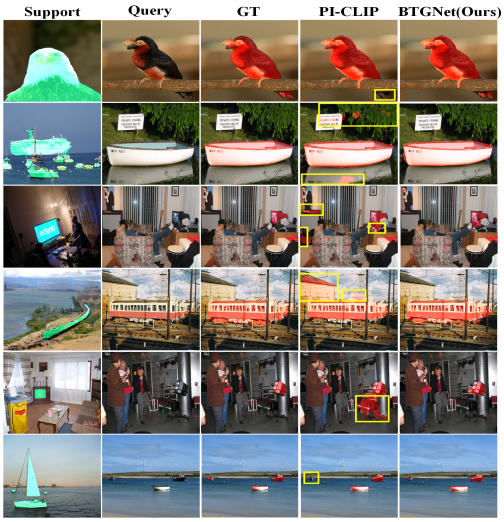

Figure 4: Segmentation results between BTGNet and PI-CLIP. BTGNet shows better noise suppression and boundary awareness.

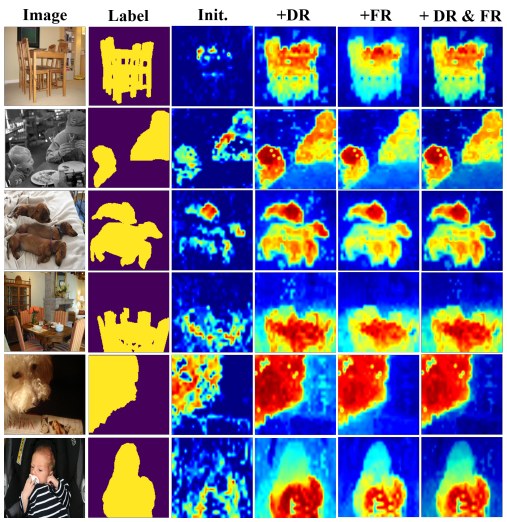

Figure 5: Comparison of different refinement strategies. Joint dynamic and fixed refinement achieves the best performance.

4.2 Comparisons with State-of-the-Arts

4.2.1 Quantitative Analysis

To validate the effectiveness of BTGNet, we compare it with state-of-the-art(SOTA) methods on PASCAL-5^i and COCO-20^i. Our BTGNet employs PFENet, HDMNet, DeFSS as the base FSS model (segmentation in BTGNet), named BTGNet-P,BTGNet-H and BTGNet-D, respectively. Table 1 presents the comparison result on the PASCAL-5^i under 1-shot and 5-shot settings, and our BTGNet outperforms all SOTA approaches. Specifically, under the 1-shot setting, BTGNet-P, BTGNet-H, and BTGNet-D improve the mean mIoU over their baselines by 18.6%, 11.1%, and 9.8%, respectively, demonstrating that our background-related word mechanism effectively enhances the model's discriminative ability. Notably, BTGNet-D achieves a mean mIoU of 81.5%, surpassing the current best multimodal FSS model FDCMNet by 1.9%. Under the 5-shot setting, BTGNet still demonstrate superior performance. Specifically, BTGNet-D achieves an mIoU of 82.4%, surpassing the existing SAM-based method USD and CLIP-based method FDCMNet by 3.8 and 2.5%, respectively.

Table 2 presents the 1-shot and 5-shot segmentation performance of BTGNet on the COCO-20^i. Even on the more challenging benchmark, BTGNet also achieves the best performance. Specifically, in the 1-shot and 5-shot setting, BTGNet-D achieves a mean mIoU of 63.2% and 66.1%, surpassing the current SOTA method DpFSS by 0.8% and 0.7%. In particular, even with the ResNet-50 backbone, BTGNet-H and BTGNet-P still outperform many methods based on more complex backbone such as ResNet-101, Swin-B, as well as multimodal approach, these suggest that the effectiveness of BTGNet may stem from the proposed mechanism rather than the complexity of the backbone.

4.2.2 Qualitative Analysis

To further illustrate the segmentation capability of BTGNet, we visualize and compare it with the SOTA CLIP-based method PI-CLIP in Figure 4. As shown in the first row, PI-CLIP performs poorly in scenarios involving object occlusion, such as failing to segment the tail region. In contrast, BTGNet successfully captures the tail area, demonstrating stronger cross-region structural perception. In the second row, PI-CLIP mistakenly segments the reflection of a boat on the water surface as part of the target object, whereas BTGNet effectively suppresses such background confusion. Moreover, in small-object scenarios, such as the example in the last row, PI-CLIP fails to detect the distant boat, while BTGNet accurately segments the target. Similar situations can be seen in other examples, where PI-CLIP usually missegments background regions. In contrast, BTGNet can effectively suppress background and focus on the target regions. These comparisons further validating the effectiveness of incorporating background textual information.

Table 3: Ablation experiments for each module of BTGNet-H on PASCAL-5^i. where Pa denotes learnable parameters, and LGBW denotes the LLM-generated category-relevant background words, DR represents dynamic refinement, and FR refers to fixed refinement.

LGBW	DR	FR	F-0	F-1	F-2	F-3	Mean	Pa(M)	FLOPs(G)
-	-	-	72.5	77.8	69.9	63.8	71.0	4.6	666.4
-	-	✓	74.3	81.6	72.7	70.9	74.7	4.6	666.4
✓	-	-	78.0	83.6	76.7	76.4	78.7	4.6	666.4
✓	-	✓	79.5	85.1	77.3	77.1	79.8	4.6	666.4
✓	✓	-	79.4	85.3	78.1	76.9	79.9	8.1	672.9
✓	✓	✓	**79.8**	**85.6**	**78.4**	**78.1**	**80.5**	8.1	672.9

Table 4: Efficiency comparison of different models on PASCAL-5^i. FPS is measured end-to-end on an NVIDIA RTX 4090 GPU.

Model	Pa(M)	FLOPs(G)	FPS	mIoU(%)
PI-CLIPWang et al. (2024a)	4.16	666.4	29.1	76.8
VRP-SAMSun et al. (2024)	1.6	889.0	13.3	71.9
USDWang et al. (2025a)	11.9	534.0	14.7	78.4
BTGNet-H	8.1	672.9	27.8	80.5

Table 5: Performance comparison of different text word generation methods on COCO-20^i.

Method	Fold-0	Fold-1	Fold-2	Fold-3	Mean
None	52.7	65.3	53.0	55.9	56.7
+ RBW	52.1	65.5	51.2	55.3	56.0
+ FBW-Pascal	55.0	67.5	56.8	58.8	59.5
+ FBW-COCO	56.1	68.3	57.7	60.9	60.8
+ LGBW (Ours)	**56.8**	**69.9**	**59.3**	**62.3**	**62.1**

4.3 Ablation Study

To evaluate the core components of BTGNet, we performed ablation studies on PASCAL-5^i to assess both performance gains and computational overhead. The results are reported in Table 3. It can be seen that incorporating LGBW consistently improves performance across all folds, yielding a 7.7% mIoU gain without extra trainable parameters or FLOPs. This indicates that category-relevant background words effectively suppress background noise and enhance query discriminability. In addition, different refinement strategies further improve segmentation accuracy by optimizing the initial activation maps. DR introduces a lightweight learnable module for dynamic refinement, while FR is parameter-free. Overall, BTGNet shows a favorable balance between accuracy and efficiency.

4.4 Comparison of Computational Efficiency

Although BTGNet achieves strong performance across different datasets, the additional components introduced may incur extra computational cost. To further examine the trade-off between accuracy and efficiency, we conduct an efficiency analysis and compare BTGNet with representative FSS methods. We report the number of trainable parameters, forward FLOPs, and inference speed. The results are summarized in Table4. As shown in Table 4, BTGNet delivers the highest mIoU while maintaining competitive inference efficiency with moderate parameters and FLOPs. Compared with mainstream multimodal methods, BTGNet-H outperforms PI-CLIP by +3.7% mIoU with similar FLOPs and minor reduction in FPS. Meanwhile, VRP-SAM and USD exhibit lower inference speed.

4.5 Further Analysis

4.5.1 Activation Refinement Strategy

Although ablation results have shown that both dynamic and fixed refinements improve performance, their specific effects on the model's decision process and how they lead to improved accuracy remain unclear. To investigate this, we visualize the refined CAMs under different strategies in Figure 5. It can be seen that dynamic refinement significantly strengthens foreground perception but intensifies background attention, whereas fixed refinement better suppresses background noise at the cost of weaker foreground activation. Our joint refinement balances these behaviors, achieving strong foreground focus with minimal background interference, validating its superiority.

4.5.2 Background Word Generation Comparison.

To investigate different background word strategies, we compare LGBW with no background words, randomly selected background words (RBW), and two Fixed Background Word (FBW) variants tailored to PASCAL and COCO categories, respectively. As shown in Table 5, RBW leads to a performance drop, indicating that irrelevant or semantically conflicting background concepts may interfere with foreground discrimination. In contrast, both FBW variants improve performance, con-

Table 6: Prompt templates robustness analysis of LGBW.

Prompt	Template
Prompt-A	What objects are most likely to appear in an image containing {c} but not belonging to {c}?
Prompt-B	What background contextual objects commonly appear around {c}?
Prompt-C	In scenes where {c} appears, what other objects commonly co-occur as context (excluding {c} itself)?
Prompt-D	List typical contextual objects that are usually seen around {c} (excluding {c}).

Table 7: different prompts robustness analysis of LGBW module using BTG-H.

Prompt	Stability↑		Mean mIoU(%)	
	Overlap	Acc	PASCAL-5^i	COCO-20^i
Prompt-A	-	0.99	80.5	62.1
Prompt-B	0.81	0.99	80.5	62.1
Prompt-C	0.78	0.98	80.2	61.9
Prompt-D	0.76	0.95	80.3	62.0

Table 8: Cross-LLM consistency analysis of LGBW module using BTG-H.

LLM	PASCAL-5^i		COCO-20^i	
	1-Shot	5-Shot	1-Shot	5-Shot
GPT 5.6-sol	**80.5**	**80.9**	**62.1**	**63.5**
DeepSeek V3.2	80.0	80.4	61.8	63.2
Kimi K3	80.2	80.7	62.0	63.3
Gemini 3.8 Flash	80.3	80.8	61.9	63.4
Claude Sonnet 5	80.4	80.7	62.0	63.4

firming that meaningful background information can effectively enhance the model's discriminative ability. Further comparison between the two FBW variants on COCO-20^i shows that, FBW-COCO achieves higher mIoU than FBW-Pascal, suggesting that background words better aligned with the target provide more effective semantic cues for discrimination. In comparison, our LGBW leverages LLM semantic reasoning to automatically generate category-adaptive, contextually relevant background concepts without dataset-specific manual design, achieving the best performance. These results demonstrate the effectiveness of background words and highlight the efficiency and generalizability of LLM-based generation, which eliminates tedious task-specific manual design.

4.5.3 Prompt Robustness Analysis.

Since LGBW generates background words through prompts, its outputs may vary with different prompts. To assess its robustness, we evaluate LGBW from two perspectives: (i) output stability under different prompts and (ii) semantic correctness of the generated words via human judgment. Specifically, we test several prompt templates, as summarized in Table 6, and report the results in Table 7. It can be seen that LGBW remains highly stable across different prompts, the generated background words are consistently semantically valid and category-relevant. Moreover, the segmentation performance is nearly unaffected by prompt variations, with mIoU changes within 0.3 on PASCAL-5^i and 0.2 on COCO-20^i. These results show that LGBW is robust to different prompts and can reliably generate category-adaptive background word.

4.5.4 Cross-LLM Consistency Analysis.

Building on the prompt robustness analysis, it is further necessary to examine whether the effectiveness of LGBW depends on the choice of a specific large language model. Therefore, we evaluate different LLMs under the same setting. The results are reported in Table 8. It can be seen that although different LLMs may produce slightly different background word lists under the fixed prompt, the resulting downstream segmentation performance remains overall consistent. GPT5.6 achieves the best results, while the gaps between other LLMs are relatively small. These results show that BTGNet does not rely on a specific LLM and exhibits strong compatibility. Moreover, compared with manually defined background words, LGBW can adapt more efficiently to different tasks.

5 Conclusion

In this paper, we reveal an overlooked insight in multimodal FSS that category-related background words can provide effective complementary textual priors for foreground discrimination. Based on this insight, we propose BTGNet, which incorporates LLM-generated background words into CLIP-guided activation generation and further refines the activations through a joint dynamic-fixed strategy. Experiments on PASCAL-5^i and COCO-20^i demonstrate that background words can substantially improve few-shot segmentation, highlighting their potential as a generalizable textual prior.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing, manuscript organization, and improving the clarity of technical descriptions. They were not used to generate experimental data or replace model training, evaluation, or empirical analysis. All AI-assisted content was carefully reviewed and verified by the authors. The authors take full responsibility for the final content of the paper, including all text, results, and conclusions.

ETHICS STATEMENT

This work focuses on few-shot semantic segmentation and does not involve the collection of private data, human subjects, or personally identifiable information. All experiments are conducted on publicly available benchmark datasets following their established evaluation protocols. Large language models are used only to generate category-related background words for visual representation learning, without processing private or sensitive user information. The proposed method is intended for general-purpose visual recognition research and does not target harmful or privacy-invasive applications.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the BTGNet architecture, LLM-based background text generation, joint activation refinement strategy, loss functions, and training settings in the main paper and supplementary materials. We further report the prompt templates, generated background words, hyperparameter settings, and additional implementation details required to reproduce our experiments. The core implementation code and model configurations are also provided in the supplementary materials to facilitate reproduction of the reported results.

REFERENCES

Bike Chen, Wei Peng, Xiaofeng Cao, and Juha Röning. Hyperbolic uncertainty aware semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 25(2):1275–1290, 2024a. doi: 10.1109/TITS.2023.3312290.

Shangjing Chen, Feng Xu, Xin Lyu, Dafa Wang, and Xin Li. Clip-driven feature disambiguation and cross-modal synergy for few-shot semantic segmentation. *Expert Systems with Applications*, 306:130892, 2026.

Shuai Chen, Fanman Meng, Runtong Zhang, Heqian Qiu, Hongliang Li, Qingbo Wu, and Linfeng Xu. Visual and textual prior guided mask assemble for few-shot segmentation and beyond. *IEEE Transactions on Multimedia*, 26:7197–7209, 2024b. doi: 10.1109/TMM.2024.3361181.

Xiaoming Chen, Zhangyan Zhao, Jingjing Cao, Yuhang Zou, and Haipeng Liu. Dpnet: A dual prototype few-shot semantic segmentation network for crack detection. *Knowledge-Based Systems*, pp. 113733, 2025.

Bernardo Forni, Gabriele Lombardi, Federico Pozzi, and Mirco Planamente. Fs-sam2: Adapting segment anything model 2 for few-shot semantic segmentation via low-rank adaptation. In *International Conference on Image Analysis and Processing*, pp. 67–78. Springer, 2025.

Zhifu Huang, Ziwei Chen, and Yu Liu. Fbinet: Few-shot semantic segmentation with foreground and background iteration. *IEEE Transactions on Instrumentation and Measurement*, 2025.

Amin Karimi and Charalambos Poullis. Dsv-lfs: Unifying llm-driven semantic cues with visual features for robust few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

N. Bala Krishna, K. Lokesh, Dudekula Azmeer Vali, Kanna Mahathi, Koramatla Siva Kumar, and Gudisirayula Padmini. Machine vision-driven semantic segmentation for autonomous navigation. In *2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, pp. 1775–1780, 2024. doi: 10.1109/ICAAIC60222.2024.10574895.

Chunbo Lang, Gong Cheng, Binfei Tu, and Junwei Han. Learning what not to segment: A new perspective on few-shot segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8057–8067, 2022.

Chunbo Lang, Gong Cheng, Binfei Tu, Chao Li, and Junwei Han. Base and meta: A new perspective on few-shot segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45 (9):10669–10686, 2023.

Mingxu Li, Bo Peng, and Donghai Zhai. Latent space segmentation model for visual surface defect inspection. *IEEE Transactions on Instrumentation and Measurement*, 73:1–11, 2024. doi: 10.1109/TIM.2024.3446650.

Yuqi Lin, Minghao Chen, Wenxiao Wang, Boxi Wu, Ke Li, Binbin Lin, Haifeng Liu, and Xiaofei He. Clip is also an efficient segmenter: A text-driven approach for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15305–15314, 2023.

Yating Ling, Yuling Wang, Wenli Dai, Jie Yu, Ping Liang, and Dexing Kong. Mtanet: Multi-task attention network for automatic medical image segmentation and classification. *IEEE Transactions on Medical Imaging*, 43(2):674–685, 2024. doi: 10.1109/TMI.2023.3317088.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024.

Weide Liu, Chi Zhang, Henghui Ding, Tzu-Yi Hung, and Guosheng Lin. Few-shot segmentation with optimal transport matching and message flow. *IEEE Transactions on Multimedia*, 25:5130–5141, 2022a.

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, 2022b.

Timo Lüddecke and Alexander S. Ecker. Prompt-based multi-modal image segmentation. *CoRR*, abs/2112.10003, 2021. URL `https://arxiv.org/abs/2112.10003`.

Xiaoliu Luo, Zhuotao Tian, Taiping Zhang, Bei Yu, Yuan Yan Tang, and Jiaya Jia. Pfenet++: Boosting few-shot semantic segmentation with the noise-filtered context-aware prior mask. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(2):1273–1289, 2023.

Xiaoliu Luo, Ting Xie, Weisen Qin, Zhao Duan, Jin Tan, and Taiping Zhang. Combining hierarchical sparse representation with adaptive prompt for few-shot segmentation. *Expert Systems with Applications*, 260:125377, 2025.

Juhong Min, Dahyun Kang, and Minsu Cho. Hypercorrelation squeeze for few-shot segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6941–6952, 2021.

Khoi Nguyen and Sinisa Todorovic. Feature weighting and boosting for few-shot segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 622–631, 2019.

Suho Park, SuBeen Lee, Hyun Seok Seong, Jaejoon Yoo, and Jae-Pil Heo. Foreground-covering prototype generation and matching for sam-aided few-shot segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 6425–6433, 2025.

Bohao Peng, Zhuotao Tian, Xiaoyang Wu, Chengyao Wang, Shu Liu, Jingyong Su, and Jiaya Jia. Hierarchical dense correlation distillation for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23641–23651, 2023.

Zishu Qin, Junhao Xu, and Weifeng Ge. Defss: Image-to-mask denoising learning for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22232–22240, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*, 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.

Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879, 1964.

Yanpeng Sun, Jiahui Chen, Shan Zhang, Xinyu Zhang, Qiang Chen, Gang Zhang, Errui Ding, Jingdong Wang, and Zechao Li. Vrp-sam: Sam with visual reference prompt. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23565–23574, 2024.

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.

Jin Wang, Bingfeng Zhang, Jian Pang, Honglong Chen, and Weifeng Liu. Rethinking prior information generation with clip for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3941–3951, 2024a.

Jin Wang, Bingfeng Zhang, Jian Pang, Weifeng Liu, Baodi Liu, and Honglong Chen. Unbiased semantic decoding with vision foundation models for few-shot segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 2025a.

Jinting Wang, Yujiao Tang, Yang Xiao, Joey Tianyi Zhou, Zhiwen Fang, and Feng Yang. Grenet: Gradually recurrent network with curriculum learning for 2-d medical image segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 35(7):10018–10032, 2024b. doi: 10.1109/TNNLS.2023.3238381.

Yicong Wang, Rong Huang, Shubo Zhou, Xueqin Jiang, and Zhijun Fang. Learning prototypes from background and latent objects for few-shot semantic segmentation. *Knowledge-Based Systems*, 314:113218, 2025b.

Qianxiong Xu, Wenting Zhao, Guosheng Lin, and Cheng Long. Self-calibrated cross attention network for few-shot segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 655–665, 2023.

Yong Yang, Qiong Chen, Yuan Feng, and Tianlin Huang. Mianet: Aggregating unbiased instance and general information for few-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7131–7140, 2023.

Jian-Wei Zhang, Yifan Sun, Yi Yang, and Wei Chen. Feature-proxy transformer for few-shot segmentation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 6575–6588. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/2ae33575c3374050654ae7802326c81d-Paper-Conference.pdf.

Zilong Zhang, Chang Niu, Zhibin Zhao, Xingwu Zhang, and Xuefeng Chen. Small object few-shot segmentation for vision-based industrial inspection. *IEEE Transactions on Industrial Informatics*, 21(6):4388–4399, 2025. doi: 10.1109/TII.2025.3538070.

Lanyun Zhu, Tianrun Chen, Deyi Ji, Jieping Ye, and Jun Liu. Llafs: When large language models meet few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3065–3075, 2024.

Lanyun Zhu, Tianrun Chen, Deyi Ji, Peng Xu, Jieping Ye, and Jun Liu. Llafs++: Few-shot image segmentation with large language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

A Implementation Details

Basic Environment Settings

All experiments were conducted on a computer with Ubuntu 20.04.4 LTS operating system also having an Intel Xeon Gold 6430 CPU, 64G of RAM and an NVIDIA RTX 4090 GPU. The equipped software runtime environment was also set up with Pycharm2024, python 3.8.20, PyTorch 2.1.1, CUDA 12.4 and cuDNN 9.5.0.

Model Hyperparameters Settings

To enhance the reproducibility of our work, we list the core hyperparameter configurations used during BTGNet training, as shown in Table 9. The key settings include: a learning rate of 0.0001, batch size of 16, total training epochs set to 200, and the optimizer selected as SGD with a weight decay of 0.01 and momentum of 0.9. Additionally, auxiliary losses are balanced with weights of 1.0, and ViT-B/16 and ResNet-50 was used as the visual backbone of the CLIP encoder, with its parameters frozen during training. AAll experiments are repeated with three different random seeds, and we report the average results across the three runs to ensure reproducibility.

Table 9: Main hyperparameter settings used in the experiments.

Category	Parameter	Value
Augmentation	Train_h	473
	Train_w	473
	Val_size	473
	Scale_min	0.9
	Scale_max	1.1
	Rotate_min	-10
	Rotate_max	10
	Ignore_label	255
	Padding_label	255
Optimizer	Batch_size	16
	Base_lr	0.0001
	Epochs	200
	Weight_decay	0.01
	Momentum	0.9
	Warmup	False
	Stop_interval	80
	Power	0.9
Others	Workers	8
	Aux_weight1	0.1
	Aux_weight2	1.0
	Random_seed	7/86/321
	CLIP_weight	ViT-B-16/ResNet-50

Semantic information protocol.

Following multimodal FSS settings that leverage vision-language models, BTGNet assumes that the semantic name of the target category is available for each few-shot episode. Importantly, the proposed LGBW does not access query images, query annotations, test-set statistics, or manually designed test-specific background descriptions. During training, background vocabularies are generated only for base categories. For novel categories, background words are generated from the category name only after model training is completed, with all model parameters fixed. Therefore, the novel-category background vocabulary serves as an external semantic prior rather than test-set supervision.

B LGBW Protocol Details

LLM-guided Background Word Generation (LGBW) constructs a per-class background vocabulary that complements the foreground category in activation generation. To ensure reproducibility and avoid repeated LLM inference, LGBW is executed once for each category and its output is cached.

Table 10: FB-mIoU(%) Comparison on PASCAL-5i

Method	Backbone	1-shot	5-shot
BAMLang et al. (2022)	RN101	80.2	84.1
NTRNetLiu et al. (2022b)	RN50	75.3	78.2
SCCANXu et al. (2023)	RN101	78.5	82.1
FBINetHuang et al. (2025)	RN101	77.2	80.7
HSRapLuo et al. (2025)	RN101	79.3	83.6
PI-CLIPWang et al. (2024a)	CLIP-RN50	87.6	–
BTGNet-P	CLIP-RN50	**87.9**	**88.4**
BTGNet-H	CLIP-RN50	**88.7**	**89.0**
BTGNet-D	CLIP-ViT	**91.3**	**92.0**

For base categories, vocabulary generation is performed before training. For novel categories, generation is performed only at evaluation time after the trained model has been frozen. In both cases, LGBW receives only the category name and does not access support/query images, segmentation masks, or dataset-specific image statistics.

For each category name/description c, we query LLM with the two prompts in Sec. 3.3 and require the output to be a bracketed noun list `['word1','word2',...']`. We use deterministic decoding to reduce variance (temperature$=0$, top-$p=1.0$, max_tokens$=128$). We request K candidates per prompt (default $K=20$), merge the two lists, then normalize (e.g., lowercasing letters) and deduplicate.

C Background Word Generation and Analysis

PASCAL-5^i Dataset

To investigate how background-aware representations contribute to semantic reasoning, we leverage the LLM to generate background words for each object category in the PASCAL-5^i dataset. As shown in Table 11, the generated background words are semantically relevant to the foreground class and reflect frequent spatial co-occurrences observed in natural scenes. For example, the category 'train' is often associated with contextual elements like 'railway', 'platform', 'bridge', and 'tunnel', indicating typical environments in which trains are found. Similarly, 'sheep' co-occurs with 'grass', 'barn', 'valley', and 'mountain', aligning with pastoral landscapes. Unlike manually predefined background word lists used in prior works, which are usually limited to generic scene terms like "sky," "grass," or "building", the background concepts generated by LLM are automatically inferred through prompt-based querying conditioned on each specific foreground category. This results in richer, more diverse, and highly category-specific background descriptions. Such dynamically generated context captures fine-grained co-occurrence patterns and reflects a deeper semantic association between objects and their environments.

MS-COCO-20^i Dataset

Building on our analysis of PASCAL-5^i, we further evaluated the generalizability of our prompting strategy on the more challenging MS-COCO-20^i dataset, which includes 80 object categories with significantly more diverse and cluttered scene contexts. Using the same LLM-based template, we generate contextual background words for each class. The results are presented in Table 12. Despite the increased complexity of MS-COCO-20^i, our approach remains robust. For instance, 'surfboard' is associated with 'wave', 'ocean', 'beach', and 'sun', while 'keyboard' is associated with elements like 'mouse', 'desk', 'monitor', and 'cable.'

This results indicates that our designed prompt template does not rely on dataset-specific tuning or handcrafted heuristics. Instead, it robustly adapts to varying foreground contexts and produces background descriptions that are both discriminative and transferable. Such adaptability is critical in few-shot settings, where background complexity often leads to confusion. The ability to generalize

Table 11: Background Words Produced by Our Proposed LLM-Based Generation Method for PASCAL-5^i Foreground Categories

Object	Background Words
Aeroplane	[cloud, sky, runway, hangar, airport, person, mountain, tree, control tower, sun]
Bicycle	[person, road, tree, car, building, streetlight, grass, sidewalk, sky, sign]
Bird	[tree, sky, branch, grass, nest, leaf, water, flower, feeder, cloud]
Boat	[water, sea, sky, dock, person, wave, beach, fish, cloud]
Bottle	[table, cap, shelf, counter, kitchen, hand, wall]
Bus	[road, person, car, street, tree, building, traffic light, sign, sky, sidewalk]
Car	[road, person, tree, building, street, traffic light, sign, sky, sidewalk, parking lot, bus]
Cat	[couch, window, floor, bed, grass, tree, house, curtain, table, wall]
Chair	[table, person, floor, desk, room, wall, cushion, window, carpet, lamp]
Cow	[grass, field, farm, fence, tree, barn, sky, hay, person, cloud]
Diningtable	[chair, plate, food, cup, person, room, window, wall, utensil]
Dog	[grass, leash, person, yard, tree, house, park, collar, ball, fence]
Horse	[grass, field, barn, fence, person, saddle, tree, stable, sky, hay]
Motorbike	[road, person, helmet, tree, building, street, traffic light, sign, sky, sidewalk]
Person	[street, building, tree, sky, road, car, grass, chair, table]
Pottedplant	[window, table, wall, curtain, floor, shelf, chair, couch, desk, lamp, picture, vase, books, cushion, rug, plantstand, indoor, outdoor, room, house]
Sheep	[grass, field, fence, tree, hill, sky, barn, farm, shepherd, dog, cloud, mountain, valley, road, path, stone, water, stream, flower, bush]
Sofa	[cushion, pillow, table, lamp, rug, curtain, wall, picture, window, floor, plant, bookshelf, television, coffee table, blanket, vase, painting, ceiling, light, room]
Train	[railway, station, platform, track, bridge, tunnel, signal, tree, mountain, sky, field, city, building, passenger, luggage, bench, light, sign, road, river]
TVmonitor	[cabinet, stand, table, wall, console, remote, shelf, decoration, clock, plant, curtain, window, furniture, cable, game console, DVD player, sound system, room]

across datasets confirms the practical utility of our method and its potential as a plug-and-play module for multi-domain segmentation tasks.

D Foreground Background Segmentation Comparison

To further validate the effectiveness and generalization ability of BTGNet, we adopt FB-mIoU as an additional evaluation metric in the PASCAL-5^i dataset. Unlike standard mIoU, which evaluates overall pixel-wise overlap, FB-mIoU separately computes the Intersection over Union (IoU) for foreground and background regions and then averages them. By emphasizing the ability of the model to distinguish foreground from background, FB-mIoU offers deeper insight into segmentation quality in sparse supervision scenarios. FB-mIoU is defined as Eq.18.

$$\text{FB-mIoU} = \frac{1}{2}\left(\text{IoU}_{fg} + \text{IoU}_{bg}\right) \tag{18}$$

where each IoU is computed using the standard pixel-level overlap, defined as Eq.19.

$$\text{IoU} = \frac{TP}{TP + FP + FN} \tag{19}$$

Table 10 presents the segmentation result under 1-shot and 5-shot settings. BTGNet-P, BTGNet-H, and BTGNet-D achieve FB-mIoU scores of 87.9/88.4, 88.7/89.0, and 91.3/92.0 under the 1-shot and 5-shot settings, respectively, consistently outperforming all approaches. Compared with classical baselines such as FBINet and HSRap, BTGNet consistently outperforms representative visual-background FSS methods such as FBINet and HSRap under both 1-shot and 5-shot settings. Even when compared with the latest state-of-the-art method PI-CLIP, BTGNet still achieves superior performance. Specifically, BTGNet-H outperforms PI-CLIP by 1.1% in the 1-shot setting. These results highlight the strong generalization ability of BTGNet across different few-shot scenarios.

E Auxiliary Loss Analysis

Ablation studies demonstrate that dynamic refinement alone yields suboptimal performance, but combined with dynamic refinement loss L_d, it significantly improves results. Given the loss strength

Table 12: Background Words Produced by Our Proposed LLM-Based Generation Method for MS-COCO-20^i Foreground Categories

Object	Background Words
person	[street, building, tree, sky, road, car, grass, chair, table]
bicycle	[person, road, tree, car, building, streetlight, grass, sidewalk, sky, sign]
car	[road, person, tree, building, street, traffic light, sign, sky, sidewalk, parking lot, bus]
motorbike	[road, person, helmet, tree, building, street, traffic light, sign, sky, sidewalk]
aeroplane	[cloud, sky, runway, hangar, airport, person, mountain, tree, control tower, sun]
bus	[road, person, car, street, tree, building, traffic light, sign, sky, sidewalk]
train	[railway, station, platform, track, bridge, tunnel, signal, tree, mountain, sky, field, city, building, passenger, luggage, bench, light, sign, road, river]
truck	[person, traffic light, road, car, bus, stop sign, parking meter]
boat	[water, sea, sky, dock, person, wave, beach, fish, cloud]
traffic light	[car, bus, truck, road, pole, street sign, person]
fire hydrant	[street, sidewalk, dog, car, person, grass, road]
stop sign	[road, car, bus, truck, pole, traffic light, person]
parking meter	[car, street, sidewalk, person, bench, road, truck]
bench	[person, park, tree, grass, bird, dog, path]
bird	[tree, sky, branch, grass, nest, leaf, water, flower, feeder, cloud]
cat	[couch, window, floor, bed, grass, tree, house, curtain, table, wall]
dog	[grass, leash, person, yard, tree, house, park, collar, ball, fence]
horse	[grass, field, barn, fence, person, saddle, tree, stable, sky, hay]
sheep	[grass, field, fence, tree, hill, sky, barn, farm, shepherd, dog, cloud, mountain, valley, road, path, stone, water, stream, flower, bush]
cow	[grass, field, farm, fence, tree, barn, sky, hay, person, cloud]
elephant	[savanna, tree, water, zoo, person, grass, fence]
bear	[forest, tree, river, person, rocks, grass, cave]
zebra	[savanna, grass, tree, zoo, person, water, fence]
giraffe	[savanna, tree, zoo, person, grass, fence, sky]
backpack	[person, book, laptop, chair, school, desk, bus]
umbrella	[rain, person, street, bench, bag, coat, puddle]
handbag	[person, dress, chair, store, table, mirror, shoes]
tie	[person, suit, shirt, office, desk, chair, meeting]
suitcase	[person, airport, car, bus, train, hotel, elevator]
frisbee	[person, park, grass, dog, tree, bench, sky]
skis	[snow, person, mountain, ski poles, goggles, jacket, gloves]
snowboard	[snow, person, mountain, goggles, jacket, gloves, boots]
sports ball	[person, field, grass, court, shoes, net, bench]
kite	[sky, wind, person, park, grass, string, tree]
baseball bat	[person, baseball bat, field, cap, uniform, grass, bench]
baseball glove	[person, baseball glove, field, cap, uniform, grass, bench]
skateboard	[person, ramp, street, shoes, helmet, park, concrete]
surfboard	[wave, ocean, person, wetsuit, beach, sun, sand]
tennis racket	[person, tennis ball, court, net, shoes, uniform, bench]
bottle	[table, cap, shelf, counter, kitchen, hand, wall]
wine glass	[table, person, bottle, restaurant, diningtable, chair, meal]
cup	[table, person, saucer, coffee, kitchen, diningtable, spoon]
fork	[plate, fork, spoon, diningtable, person, food, cutting board]
knife	[plate, knife, spoon, diningtable, person, food, napkin]
spoon	[spoon, table, person, soup, cereal, diningtable, kitchen]
bowl	[bowl, fork, knife, diningtable, person, soup, cereal]

primarily governed by weighting σ, selecting an appropriate value is paramount. We thus evaluate σ's impact on performance to determine the optimal setting. As shown in Table 13, $\sigma = 0.1$ achieves peak performance, while higher weights cause progressive deterioration. This occurs since excessive σ over-prioritizes dynamic activation map optimization, neglecting core segmentation tasks and ultimately degrading performance.

Table 12 (Continued): Background Words Produced by Our Proposed LLM-Based Generation Method for MS-COCO-20^i Foreground Categories

Object	Background Words
banana	[fruit bowl, table, person, kitchen, hand, plate, other fruits]
apple	[fruit bowl, table, person, kitchen, hand, plate, other fruits]
sandwich	[plate, table, person, kitchen, hand, napkin, cup]
orange	[fruit bowl, table, person, kitchen, hand, plate, other fruits]
broccoli	[plate, table, person, kitchen, fork, knife, other vegetables]
carrot	[plate, table, person, kitchen, fork, knife, other vegetables]
hot dog	[plate, table, person, ketchup, mustard, bun, picnic]
pizza	[box, plate, table, person, oven, cheese, diningtable]
donut	[plate, coffee, person, bakery, napkin, table, cup]
cake	[plate, candles, person, celebration, table, knife, diningtable]
chair	[table, person, floor, desk, room, wall, cushion, window, carpet, lamp]
sofa	[cushion, pillow, table, lamp, rug, curtain, wall, picture, window, floor, plant, bookshelf, television, coffee table, blanket, vase, painting, ceiling, light, room]
pottedplant	[window, table, wall, curtain, floor, shelf, chair, couch, desk, lamp, picture, vase, books, cushion, rug, plantstand, indoor, outdoor, room, house]
bed	[person, pillow, blanket, nightstand, lamp, bedroom, curtains]
diningtable	[chair, plate, food, cup, person, room, window, wall, utensil]
toilet	[bathroom, sink, towel, person, mirror, shower, soap]
tvmonitor	[cabinet, stand, table, wall, console, remote, shelf, decoration, clock, plant, curtain, window, furniture, cable, game console, DVD player, sound system, room]
laptop	[desk, person, mouse, keyboard, chair, coffee cup, notebook]
mouse	[computer, desk, keyboard, person, monitor, mouse pad, cable]
remote	[tvmonitor, sofa, person, coffee table, living room, cushion, batteries]
keyboard	[computer, desk, mouse, monitor, person, chair, cable]
cell phone	[person, hand, table, charger, bag, earphones, coffee cup]
microwave	[kitchen, mitt, baking tray, person, microwave, counter, food]
oven	[kitchen, counter, plate, food, person, oven, refrigerator]
toaster	[kitchen, counter, bread, plate, person, knife, butter]
sink	[faucet, kitchen, dishes, soap, person, counter, sponge]
refrigerator	[kitchen, food, magnet, person, oven, counter, milk]
book	[person, glasses, table, lamp, chair, bookshelf, coffee cup]
clock	[wall, room, person, desk, calendar, lamp, table]
vase	[flowers, table, water, person, window, room, curtains]
scissors	[paper, desk, person, craft supplies, tape, table, envelope]
teddy bear	[child, bed, room, toy box, person, blanket, carpet]
hair drier	[bathroom, person, mirror, sink, outlet, comb, towel]
toothbrush	[sink, bathroom, toothpaste, person, mirror, cup, faucet]

Table 13: Parametric analysis of auxiliary loss weights σ using mIoU(%) on PASCAL-5^i.

σ	Backbone	Fold0	Fold1	Fold2	Fold3	Mean
0.1		**79.8**	85.6	**78.4**	**78.1**	**80.5**
0.3		79.7	85.5	**78.4**	77.5	80.3
0.5	ResNet-50	79.6	85.4	76.8	77.9	79.9
0.7		79.7	**85.7**	77.1	77.0	79.9
0.9		79.6	85.4	76.8	77.5	79.8

F Generalization analysis across backbones and Vision-Language Models

To further investigate whether the performance gains of BTGNet depend on a particular visual backbone or pretrained foundation representation, we conduct additional controlled experiments on several representative FSS frameworks with diverse architectures and pretrained models. Specifically, we select USD, LLaFS, FDCMNet, and Matcher, which cover ResNet-50, ResNet-101, and DINOv2-based visual backbones, as well as CLIP-based, SAM-CLIP-based, LLM-assisted, and SAM-DINOv2-based representation paradigms. For each baseline, we preserve its original segmentation architecture and pretrained visual representation, while incorporating the proposed background-text activation generation and joint refinement strategy. Therefore, each paired comparison is conducted under the same underlying FSS framework and pretrained representation, al-

Table 14: Generalization of the proposed background-text guidance and joint refinement strategy across different visual backbones and foundation models. All paired variants use the same underlying FSS framework and pretrained representation.

Base Model	Foundation Model	Backbone	Variant	PASCAL-5^i		COCO-20^i	
				1-shot	5-shot	1-shot	5-shot
USD	SAM+CLIP	RN50	Original	78.4	78.6	57.9	58.7
			w/ Ours	**82.0**	**82.7**	**61.8**	**62.5**
LLaFS	LLaMA	Q-Former	Original	73.5	75.6	53.9	60.0
			w/ Ours	78.3	80.2	**58.1**	**62.5**
FDCMNet	CLIP	RN101	Original	79.6	79.9	58.5	60.2
			w/ Ours	**80.8**	**81.2**	**61.3**	**63.0**
Matcher	SAM	DINOv2	Original	68.1	74.0	52.7	60.7
			w/ Ours	**74.7**	**79.2**	**58.0**	**65.9**

Table 15: Robustness under counterfactual background context shifts on PASCAL-5^i. Δ denotes the performance degradation from the original context.

Method	Natural	Random BG	Atypical BG	$\Delta_{Ra} \downarrow$	$\Delta_{At} \downarrow$
HDMNet	69.4	68.2	66.0	1.2	3.4
BTGNet-H w/o LGBW	74.7	74.3	73.9	**0.4**	**0.8**
BTGNet-H	**80.5**	**80.0**	**79.6**	0.5	0.9

lowing the additional performance gain to more directly reflect the effectiveness of the proposed background-aware semantic modeling rather than differences in backbone capacity or foundation-model pretraining.

As shown in Table 14, the proposed strategy consistently improves a diverse set of FSS frameworks built upon different visual backbones and pretrained multimodal representations. The gains are observed across CLIP-based CNN architectures, SAM-CLIP-based models, LLM-assisted multimodal frameworks, and DINOv2-based representations. This consistent improvement suggests that the proposed background-text guidance and joint refinement strategy is complementary to different pretrained representation paradigms, rather than being tied to a particular backbone or vision-language architecture.

Robustness to Counterfactual Background Contexts

Since BTGNet exploits category-related background words as complementary textual priors, a natural question is whether the model overly relies on typical object-context co-occurrences. To investigate this issue, we construct a controlled counterfactual background evaluation based on PASCAL-5^i. Specifically, given a query image and its ground-truth foreground mask, we preserve the foreground object unchanged while replacing only the background with that from another image containing no instance of the target category. The support images and foreground objects remain identical to the original evaluation setting. We consider three evaluation conditions: the original natural context, randomly sampled backgrounds that do not contain the target category, and semantically conflicting backgrounds that are inconsistent with the typical contexts associated with the target category. For each query object, multiple counterfactual backgrounds are sampled to reduce the influence of individual background images. We evaluate the base FSS model, BTGNet without background-text guidance, and the full BTGNet under the same setting.

As shown in Table 15, changing the background context noticeably degrades the performance of the original HDMNet, with the mIoU dropping by 1.2 and 3.4 points under random and atypical backgrounds, respectively. In contrast, both BTGNet variants remain substantially more stable, exhibiting less than 1.0 point degradation even under atypical contexts. More importantly, the performance gain brought by LGBW remains nearly unchanged across different background conditions. Specifically, LGBW improves BTGNet-H by 5.8, 5.7, and 5.7 mIoU under natural, random, and atypical contexts, respectively. This consistent improvement suggests that the proposed background textual guidance does not simply exploit the visual co-occurrence between foreground categories and their typical scene contexts. Instead, category-related background concepts provide comple-

Table 16: Robustness comparison of BTGNet-H and BTGNet-D across different GPUs (%).

Dataset	Model	GPU	Fold0	Fold1	Fold2	Fold3	Mean
PASCAL-5^i	BTGNet-H	RTX A6000	79.7	85.6	78.3	77.9	80.4
		RTX 4090	79.8	85.6	78.4	78.1	80.5
		RTX 2080Ti	79.8	85.7	78.2	77.7	80.4
	BTGNet-D	RTX A6000	80.6	86.9	79.4	79.0	81.5
		RTX 4090	80.7	86.8	79.4	79.0	81.5
		RTX 2080Ti	80.7	86.6	79.5	79.1	81.5
COCO-20^i	BTGNet-H	RTX A6000	52.7	64.8	56.2	58.3	58.0
		RTX 4090	52.8	64.9	56.3	58.3	58.1
		RTX 2080Ti	52.8	64.8	56.4	58.3	58.1
	BTGNet-D	RTX A6000	59.4	67.4	62.0	63.8	63.1
		RTX 4090	59.5	67.4	62.1	63.9	63.2
		RTX 2080Ti	59.6	67.2	62.2	63.6	63.1

mentary semantic cues for foreground discrimination that remain effective even when the actual query background deviates from typical object-context associations.

G Hardware Robustness Analysis

To validate the robustness and reproducibility of our method across varying hardware conditions, we conducted a comparative evaluation on three different GPU platforms:NVIDIA RTX A6000, NVIDIA RTX 2080TI, and RTX 4090. As shown in Table 16, experiments were performed on the PASCAL-5^i dataset under identical parameter settings and data conditions. The results demonstrate minimal variation across all folds, with a mean mIoU of 80.5% on RTX 4090 and 80.4% on RTX 2080TI and RTX A6000.

This negligible performance gap suggests that our method is largely robust to differences in computational hardware. Although the three GPUs vary significantly in processing power and architecture, the segmentation performance remains consistent. This validates the reproducibility and stability of our experimental findings across heterogeneous computing environments.

H Recommendations for the Application of the Proposed BTGNet

Based on the above extensive analysis and experiments for different tasks, we recommend the following points when applying the proposed BTGNet.

- **Standard model and dataset;** When using the same backbone and dataset configurations as those employed in this paper, it is recommended to directly adopt the hyperparameter settings described in SectionA. This ensures reproducibility and avoids unnecessary tuning, thereby minimizing finetuning overhead.
- **Novel model or Novel dataset;** For new base-model or datasets, it is essential to first analyze the object size distribution and scene complexity. If the dataset predominantly features large objects or coarse-grained scenes, consider increasing the weight of global context and using coarser activation granularity. Conversely, for small or cluttered objects, emphasize finer activation resolution and increase the use of dynamic refinement with a higher auxiliary loss weight. Additionally, ensure that the LLM-based prompt templates remain semantically compatible with the domain in question, modification may be needed in domain-specific settings such as medical or industrial imagery.

I Limitations and Future Learning

Limitations; While BTGNet demonstrates significant performance gains in few-shot segmentation through the incorporation of background textual cues and joint activation refinement, several limitations remain. One notable challenge lies in the applicability of the LLM-based background word

generation in domain-specific settings. In fields such as medical imaging or industrial defect inspection, the generated background words tend to be overly generic or semantically mismatched due to the LLM's limited exposure to specialized visual-language associations. Moreover, although LGBW automatically infers background words according to the given foreground category and prompt instructions when the training pipeline is initialized. This design makes LGBW dynamically and adaptive at the task and category level, but it does not yet update the background cues according to image-specific context, evolving scene conditions, or changing task objectives. Therefore, future work could explore instance-aware or feedback-driven background text generation, enabling the textual cues to be further adjusted according to query images, domain knowledge, or downstream segmentation responses. Another limitation lies in the computational overhead introduced by the joint refinement strategy. While combining fixed and dynamic refinement improves segmentation quality, it also increases training and inference time. Lightweight alternatives or pruning strategies could be explored to reduce this burden without sacrificing performance.

Future Learning; For future work, we plan to explore query object-driven and context-adaptive prompting mechanisms that dynamically refine or filter background words based on scene context or user guidance, while minimizing computational overhead. In particular, we aim to improve current background-word generation paradigm more intelligence, where the generation objective is automatically inferred from query object and can updated online, without requiring manual prior specification. Meanwhile, we will investigate lightweight alternatives to large-scale LLMs by leveraging compact domain-adapted language models, prompt retrieval modules, or cached semantic memory to generate background cues more efficiently. Additionally, we will design lightweight refinement mechanisms to reduce computational cost and make BTGNet more suitable for real-time or edge deployment.