Split Matching for Inductive Zero-shot Semantic Segmentation

BMVC 2025 Submission # 43

Abstract

Zero-shot Semantic Segmentation (ZSS) targets the segmentation of unseen classes, *i.e.*, classes not annotated during training. While fine-tuned vision-language models show promise, they often overfit to seen classes due to the lack of supervision. Query-based methods offer strong potential by enabling object localization without explicit labels, but conventional approaches assume full supervision and thus tend to misclassify unseen classes as background in ZSS settings. To address this issue, we propose **Split Matching** (SM), a novel assignment strategy that decouples Hungarian matching into two components: one for seen classes in annotated regions and another for latent classes in unannotated regions (referred to as unseen candidates). Specifically, we split the queries into seen and candidate queries, enabling each to be optimized independently according to its available supervision. To discover unseen candidates, we cluster CLIP dense features to generate pseudo masks and extract region-level embeddings using CLS tokens. Matching is then conducted separately for the two groups based on both class and mask similarity. Additionally, we introduce a **Multi-scale Feature Enhancement** (MFE) module that refines decoder features through residual multi-scale aggregation, improving the model's ability to capture spatial details across resolutions. Besides, we also introduce a **Random Query** (RQ) strategy to further enhance the performance after training. Our method is the first to introduce decoupled Hungarian matching under the inductive ZSS setting, and achieves 0.8% and 1.1% higher hIoU on two ZSS benchmarks.

1 Introduction

Semantic segmentation [7, 8, 10, 11, 24, 36] serves as a fundamental task for computer vision. Existing approaches can be broadly classified into feature-based and query-based methods. Feature-based [7, 8, 24] methods treat semantic segmentation as a per-pixel classification problem, where dense features extracted from the backbone are directly finetuned for pixel-wise label prediction. In contrast, query-based segmentors [10, 11] employ a set of discrete, learnable vectors, referred to as queries, to jointly predict class labels and class-agnostic masks. These queries are passed through a transformer decoder to produce class scores and interact with dense backbone features to generate the corresponding masks.

However, achieving high performance in semantic segmentation typically requires large-scale datasets with pixel-level annotations [13], which are costly to obtain. To reduce annotation demands, Zero-shot Semantic Segmentation (ZSS) [2, 5, 6, 33] has emerged, aiming to segment unseen classes which are not annotated during training but must be segmented at test time, by transferring knowledge from seen classes, *i.e.*, classes with available training

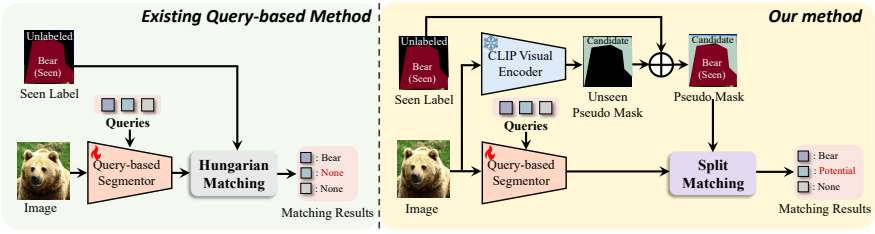

Figure 1: Comparisons between existing query-based segmentation models that fail to match the unseen candidates and the proposed split matching.

annotations. Recent advances in Vision-Language Models (VLMs) [26, 35] drives ZSS by enabling the transfer of vision-language alignment to segmentation tasks [14, 19, 38]. Existing methods typically adapt CLIP via adapters [21] or prompts [22] to handle seen classes, while relying on CLIP's generalization ability for unseen ones. However, such strategies typically rely on CLIP's zero-shot capability for unseen classes without additional training signals, which leads to overfitting to seen classes and poor generalization on unseen classes.

To mitigate overfitting to seen classes, we explore query-based segmentation as a more effective solution. Unlike feature-based methods that classify pixels independently, query-based approaches treat each object as a learnable query and perform mask-level classification, enabling better object-level reasoning and localization without explicit supervision. However, despite their ability to localize potential objects, existing query-based methods struggle in ZSS. Without annotations for unseen classes during training, queries for novel objects are often misclassified as seen classes or background. As shown on the left of Fig. 1, this matching imbalance biases the model toward seen classes and prevents it from learning useful representations for unseen ones, thus limiting the full potential of query-based segmentation in ZSS.

To enable query-based method to segment unseen classes, we propose Split Matching (SM). Specifically, we divide the queries into two groups: ***seen queries***, which segment annotated seen classes, and ***candidate queries***, which target latent classes in unannotated regions (referred to as unseen candidates). We first apply multi-scale K-Means clustering [5] on CLIP dense features to generate pseudo masks that localize unseen candidates. The corresponding image patches are then cropped and fed into the frozen CLIP visual encoder to obtain CLS tokens, which are subsequently fused with semantic embeddings to form joint class embedding. We compute class similarity between the joint embeddings and query predictions, and measure mask similarity using the pseudo masks. Hungarian matching is then applied separately to the seen and candidate queries based on the combined similarities. To better support queries in capturing semantic cues at different scales, we introduce a Multi-scale Feature Enhancement (MFE) module. It enhances the visual feature with multi-scale context and applies spatial normalization, refining the transformer decoder's key and value through residual multi-scale aggregation. Moreover, we introduce a Random Query (RQ) strategy that injects a few newly initialized, untrained queries during inference, alongside the trained seen and candidate queries. This increases the density of queries in the feature space and helps discover unseen classes more effectively.

Different from existing methods that rely on dense features extracted from CLIP [5, 33] and struggle to optimize unannotated regions due to the absence of explicit supervision, our method introduces Split Matching (SM), which separates queries into seen and candidate groups, enabling targeted label assignment even in unannotated areas. This design effectively mitigates the common issue of misclassifying unseen objects as background, a key limitation

of prior methods. Moreover, unlike open-vocabulary segmentation approaches [32, 34] that perform Hungarian matching with fully annotated data, our method operates entirely under the zero-shot setting, without requiring any pixel-level labels for unseen classes. To the best of our knowledge, we are the first to explicitly separate seen and candidate queries via Hungarian matching under the inductive ZSS. To summarize, our contributions are:

- We propose Split Matching (SM), a novel query-based assignment framework for zero-shot segmentation. SM explicitly separates seen and unseen queries and matches them independently using pseudo masks derived from CLIP dense features.
- We introduce a Multi-scale Feature Enhancement (MFE) module, which enriches decoder features via residual multi-scale fusion. Additionally, we propose a Random Query (RQ) strategy that increases query density at inference time to uncover more latent objects.
- Our method achieves 0.8% and 1.1% higher hIoU on PASCAL VOC and COCO-Stuff benchmarks under the zero-shot setting.

2 Related Works

Semantic Segmentation assigns a class label to each pixel in an image. Traditional CNN-based methods [8, 9, 24] enhance per-pixel classification through dilated convolutions and multi-scale context aggregation but struggle to capture long-range dependencies. With the rise of Vision Transformers [15], encoder-based models [18, 30] have shown strong capabilities in modeling global context. Inspired by DETR [4], query-based frameworks such as MaskFormer [10] and Mask2Former [11] reformulate segmentation as set prediction using object queries and Hungarian matching. Although these models achieve impressive results under full supervision, they are not directly applicable to zero-shot segmentation (ZSS) due to their reliance on complete annotations and inherent bias toward seen classes. To address this, we propose Split Matching, the first method to explicitly separate seen and candidate queries under the inductive ZSS setting. By decoupling the matching process for annotated and unannotated regions, our approach enables targeted supervision and significantly improves generalization to unseen classes, even in the absence of pixel-level annotations.

Zero-shot and Open-Vocabulary Segmentation. Although semantic segmentation has made remarkable progress, it still relies heavily on large-scale pixel-level annotations, which are expensive and time-consuming to obtain. To alleviate this, two related directions have emerged: zero-shot segmentation (ZSS) [5, 6, 14, 19, 33, 38] and open-vocabulary segmentation (OVS) [27, 32, 34, 37]. Both paradigms aim to improve generalization by leveraging large-scale vision-language models such as CLIP [26], either by introducing lightweight adapters [21, 25] or designing task-specific visual prompts [22]. Despite this shared goal, their setups differ fundamentally. ZSS is defined under a partially labeled training regime, where only a subset of classes is annotated and the rest are marked as ignored. Evaluation is conducted on the same domain, including both seen and unseen classes. OVS, on the other hand, assumes access to fully labeled training data and evaluates on datasets with novel classes and distribution shifts. While OVS benefits from full supervision and can leverage existing segmentation architectures, ZSS faces the challenge of incomplete supervision. Unannotated regions corresponding to unseen classes are often treated as background during training, making it difficult for models to learn discriminative representations for unseen concepts. To tackle this issue, we propose **Split Matching (SM)**, a label assignment strategy tailored for query-based models in the zero-shot setting. SM dynamically aligns predicted queries with pseudo labels derived from external vision-language features, enabling the discovery of unseen objects even in the absence of ground-truth annotations.

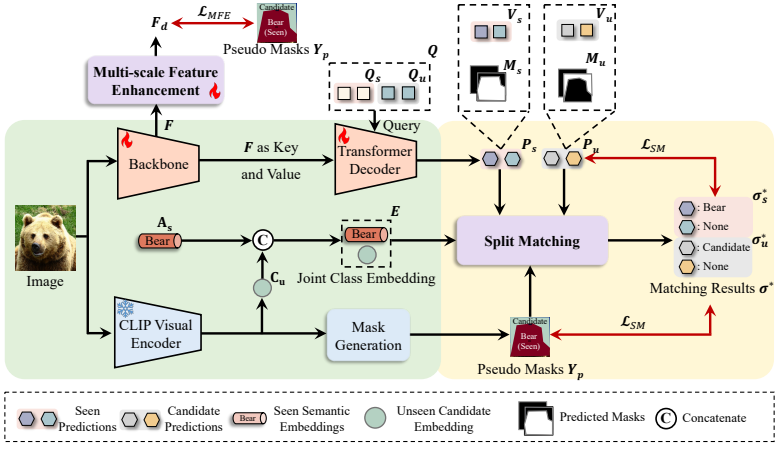

Figure 2: Training pipline overview of the proposed method. During training, input images are fed into a trainable backbone to extract dense features **F**, while a frozen CLIP encoder provides CLIP features. These are used to generate class embeddings $\mathbf{C}_u$ and pseudo masks $\mathbf{Y}_p$ for both seen and unseen classes. The dense features serve as keys and values in a transformer decoder, which interacts with queries **Q** to predict masks and query features. These are aligned with pseudo masks via the Split Matching. A Multi-scale Feature Enhancement module further refines **F** with auxiliary loss $\mathcal{L}_{MFE}$.

3 Methods

3.1 Preliminaries

Task Definition. ZSS aims to segment both seen and unseen classes without annotations for unseen classes during training. Formally, let $\mathcal{D} = \{\mathbf{X}^i, \mathbf{Y}_s^i\}_{i=1}^M$ represent a dataset of images $\mathbf{X}$ and their pixel-level annotations $\mathbf{Y}_s$ for seen classes, and M is the dataset size. Additionally, let $\mathbf{A} \in \mathbb{R}^{N \times C}$ denote the semantic (text) embeddings of all classes, divided into seen $\mathbf{A}_s \in \mathbb{R}^{N_s \times C}$ and unseen $\mathbf{A}_u \in \mathbb{R}^{N_u \times C}$, such that $(\mathbf{A}_s \cap \mathbf{A}_u = \varnothing)$ and $N_s + N_u = N$ where C indicates the channel number and N is the number of classes in the dataset. Our method applies the *Inductive settings*, where unseen embeddings $\mathbf{A}_u$ are inaccessible during training and evaluates model performance on seen and unseen classes during inference. Meanwhile, all training images are preserved during training, and regions corresponding to unseen classes are consistently labeled as "ignored", ensuring that no unseen information is used.

Method Overview. Our core idea is to mitigate seen-class bias from incomplete annotations by decoupling the optimization of seen and unannotated regions, allowing better discovery of unseen classes without harming seen-class performance. As shown in Fig.2, we divide the query space into **seen queries** $\mathbf{Q}_s$ and **candidate queries** $\mathbf{Q}_u$, responsible for segmenting annotated seen classes and unseen candidates, respectively. Given an input image, a trainable backbone extracts dense features, which interact with a set of randomly initialized queries through a transformer decoder. These queries are split into $\mathbf{Q}_s$ and $\mathbf{Q}_u$, with the latter guided by pseudo masks and class embeddings for unseen candidates $\mathbf{C}_u$ derived from a frozen CLIP encoder. We then propose **Split Matching** (Sec.3.2) to assign labels to both query types via similarity with model outputs, pseudo masks, and class embeddings. Additionally, a **Multi-scale Feature Enhancement** module (Sec. 3.3) is introduced to refine backbone features,

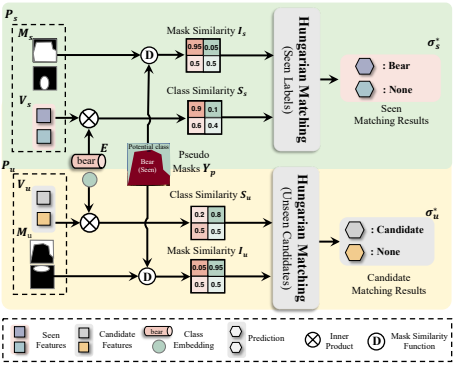

Figure 3: Overview of SM.

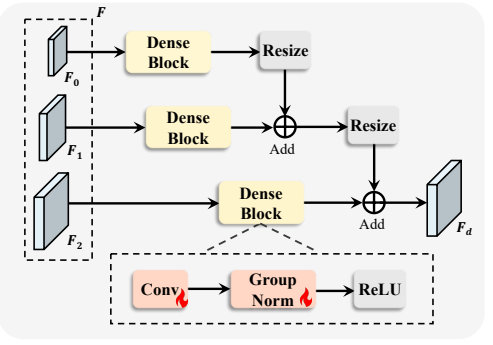

Figure 4: Overview of MFE.

and a **Random Query (RQ)** strategy (Sec. 3.4) is introduced during inference.

Although our method utilizes features from unannotated regions, it strictly adheres to the inductive zero-shot setting. At the beginning of training, all unannotated regions are uniformly treated as 'ignored' without introducing any class-specific supervision or bias. Unseen candidates are identified in a fully self-supervised manner, without any assumptions regarding the number, identity, or distribution of unseen classes. As a result, the model remains completely agnostic to unseen classes throughout the entire training process, ensuring that no unseen-related information is leaked.

3.2 Split Matching (SM)

Hungarian matching requires splitting ground-truth into class labels and corresponding class-agnostic masks, followed by assigning each ground-truth instance to a query based on similarity. However, in ZSS, unannotated regions lack ground-truth labels, making it impossible to directly apply Hungarian matching, which relies on full supervision. To overcome this, Split Matching (SM) generates pseudo masks for latent classes in unannotated areas and assigns them to candidate queries, while optimizing seen and candidate queries separately.

To generate the pseudo masks and their corresponding class embeddings for unseen candidates, we use the multi-scale K-means and mask fusion methods from CLIP-ZSS [5]. Specifically, given an image $\mathbf{X}$, we feed it into a frozen CLIP visual encoder to obtain the CLIP dense features $\mathbf{O}$. Then, we compute the seed for K-means methods with

$$\mathbf{G} = \left\{ \sum_{u=i}^{i+s-1} \sum_{v=j}^{j+s-1} \frac{\mathbf{O}[u,v]}{s^2} \,\middle|\, i \in I, j \in J \right\}, \tag{1}$$

where $I = \{0, [s/2], \ldots, [H-s]\}$ and $J = \{0, [s/2], \ldots, [W-s]\}$ are index sets, $[\cdot]$ denotes rounding, and $s \in S$ is the window size. After we obtain $\mathbf{G}$, we merge the masks corresponding to $\mathbf{G}$ by the mask fusion algorithm [5], which merge the semantically similar masks by the cosine similarity among cluster centroids, to obtain pseudo label $\mathbf{Y}_u \in [0,1]^{U \times H \times W}$ for unseen candidates where U indicates the number of unseen candidates. After obtaining $\mathbf{Y}_u$, we use $\mathbf{Y}_u$ to mask the input images $\mathbf{X}$ into $\mathbf{X}_m \in \mathbb{R}^{U \times 3 \times H \times W}$. Finally, we feed $\mathbf{X}_m$ into the frozen CLIP visual encoder for $\mathbf{C}_u \in \mathbb{R}^{U \times C}$ as class embeddings for unseen candidates.

After obtaining $\mathbf{Y}_u$ and $\mathbf{C}_u$, we apply split matching as illustrated in Fig. 3. Specifically, we first feed the randomly initialized queries $\mathbf{Q}$ into a transformer decoder to obtain the predictions for each query, denoted as $\mathbf{P} = \{\mathbf{P}_s, \mathbf{P}_u\}$. Here, we denote the predictions for

seen and candidate queries as $\mathbf{P}_s = \{\mathbf{V}_s, \mathbf{M}_s\}$ and $\mathbf{P}_u = \{\mathbf{V}_u, \mathbf{M}_u\}$, respectively. Specifically, $\mathbf{V}_s \in \mathbb{R}^{K_s \times C}$ and $\mathbf{M}_s \in \mathbb{R}^{K_s \times H \times W}$ denote the features after transformer decoder and predicted masks for the K_s seen queries. Similarly, $\mathbf{V}_u \in \mathbb{R}^{K_u \times C}$ and $\mathbf{M}_u \in \mathbb{R}^{K_u \times H \times W}$ correspond to the predictions for the K_u candidate queries. In both cases, the queries are projected into a C-dimensional semantic space. Similar to the conventional hungarian matching, SM also needs to compute the class similarity and mask similarity. For class similarity, we concatenate the seen class embeddings $\mathbf{A}_s$ (extracted from the CLIP textual encoder) and candidate-class embeddings $\mathbf{C}_u$ to form a joint class embedding $\mathbf{E} = \text{cat}(\mathbf{A}_s, \mathbf{C}_u) \in \mathbb{R}^{(N_s+U)\times C}$, where 'cat' denotes concatenation. Finally, we separately compute the similarity between the predictions and joint class embeddings for seen and candidate queries, respectively. $\mathbf{S}_s = Sigmoid(\mathbf{V}_s \cdot \mathbf{E}^\top), \mathbf{S}_u = Sigmoid(\mathbf{V}_u \cdot \mathbf{E}^\top)$. Then, for mask similarity, $\mathbf{Y}_s$ and $\mathbf{Y}_u$ are added to be the total pseudo label $\mathbf{Y}_p$ for the mask matching step. The mask similarities are then calculated by comparing these predictions with $\mathbf{Y}_p$, $\mathbf{I}_s = D(\mathbf{M}_s, \mathbf{Y}_p), \mathbf{I}_u = D(\mathbf{M}_u, \mathbf{Y}_u)$ where D indicates the function of calculating the similarity between predicted and pseudo masks, *e.g.*, BCE loss. Next, Hungarian matching is then performed independently for seen and candidate queries, ensuring each query group is only matched to its corresponding classes.

$$\sigma_s^* = \arg\min \sum_{i=0}^{K_s-1} \mathcal{L}_{match}(\mathbf{S}_s^{\sigma(i)}, \mathbf{M}_s^{\sigma(i)}, \mathbf{E}_i, \mathbf{Y}_s^i), \quad \sigma_u^* = \arg\min \sum_{i=0}^{K_u-1} \mathcal{L}_{match}(\mathbf{S}_u^{\sigma(i)}, \mathbf{M}_u^{\sigma(i)}, \mathbf{E}_i, \mathbf{Y}_u^i) \tag{2}$$

where $\mathcal{L}_{match}(\mathbf{S}^\sigma, \mathbf{M}^\sigma, \mathbf{E}, \mathbf{Y}) = \mathcal{L}_{cls}(\mathbf{S}^\sigma, \mathbf{E}) + \mathcal{L}_{mask}(\mathbf{I}^\sigma, \mathbf{Y})$, consisting of $\mathcal{L}_{cls}$ and $\mathcal{L}_{mask}$. The classification loss $\mathcal{L}_{cls}$ implemented using focal loss, while the mask loss $\mathcal{L}_{mask}$ is computed using the DICE loss. More details of Hungarian matching is shown in **Supplementary Materials**. After computing σ_s^* and σ_u^*, we concatenate them into a unified assignment σ^*. σ^* is then used to optimize the matching loss $\mathcal{L}_{match}^*$ across all queries,

$$\mathcal{L}_{match}^*(\mathbf{S}, \mathbf{M}, \mathbf{E}, \mathbf{Y}) = \mathcal{L}_{cls}(\mathbf{S}^{\sigma^*}, \mathbf{E}) + \mathcal{L}_{mask}(\mathbf{I}^{\sigma^*}, \mathbf{Y}_p), \tag{3}$$

where $\mathbf{S}^{\sigma^*}$ and $\mathbf{I}^{\sigma^*}$ are the classification scores and mask similarities after matching with the optimal assignment σ^*. To enhance the discriminability of candidate queries, we introduce a cosine similarity loss: $\mathcal{L}_{cos} = 1 - \cos(\mathbf{V}_u', \mathbf{C}_u)$, where $\mathbf{V}_u'$ is the candidate queries which are assigned with non-igored classes under σ_u^*, and $\cos(a, b)$ denotes cosine similarity. The final loss for split matching is $\mathcal{L}_{SM} = \mathcal{L}_{match}^* + \mathcal{L}_{cos}$.

3.3 Multi-scale Feature Enhancement

Although SM facilitates the adaptation of query-based approaches to zero-shot segmentation, the key and value features in transformer decoders—responsible for associating queries with relevant image regions—remain suboptimal. Due to the lack of further refinement, the matching performance suffers, ultimately constraining the full potential of the model. To tackle this issue, we propose an Multi-scale Feature Enhancement (MFE) module, as illustrated in Fig. 4, designed to assist in identifying the relevant regions by effectively combining multi-scale features. The MFE leverages hierarchical features extracted by the pixel decoder to provide a comprehensive representation, capturing both fine-grained and global contextual information. The multi-scale outputs of the pixel decoder, $\mathbf{F} = \{\mathbf{F}_0, \mathbf{F}_1, \mathbf{F}_2\}$, represent features at finer resolutions, where $\mathbf{F}_0$ has the coarsest resolution and $\mathbf{F}_2$ the finest where $\mathbf{F}_i \in \mathbb{R}^{C \times (H/r^{2-i}) \times (W/r^{2-i})}$, with r denoting the scale factor.

The fusion process begins by refining $\mathbf{F}_0$, the lowest-resolution feature map, through a dense block consisting of a convolutional layer, group normalization [28], and a ReLU activation function. This block enhances local spatial relationships and prepares the feature map for integration with features at finer scales. The refined $\mathbf{F}_0$ is then resized to align with the spatial dimensions of $\mathbf{F}_1$, which undergoes a similar refinement process. The refined $\mathbf{F}_0$ and $\mathbf{F}_1$ are fused via element-wise addition, combining detailed lowest-resolution features with mid-level semantic information. This combined representation is resized again to match the spatial dimensions of $\mathbf{F}_2$, the finest feature map in the hierarchy. Simultaneously, $\mathbf{F}_2$ is processed through its own dense block to extract refined features. The previously fused representation of $\mathbf{F}_0$ and $\mathbf{F}_1$ is then combined with the refined $\mathbf{F}_2$ to produce the final unified feature map, $\mathbf{F}_d$. This hierarchical fusion ensures that fine-grained details from the highest-resolution features are effectively integrated with broader contextual cues. The final output, $\mathbf{F}_d$, leverages complementary information across scales to provide a rich, unified representation that allows queries to accurately identify relevant regions in the image. After obtaining $\mathbf{F}_d$, we optimize it through $\mathcal{L}_{MFE} = \mathcal{L}_{ce}(\mathbf{F}_d, \mathbf{Y}_p) + \mathcal{L}_{focal}(\mathbf{F}_d, \mathbf{Y}_p)$ where $\mathcal{L}_{ce}$ and $\mathcal{L}_{focal}$ are cross entropy and focal loss [23].

3.4 Training and Inference

The total loss is $\mathcal{L} = \mathcal{L}_{SM} + \mathcal{L}_{MFE}$. During inference, we exclude the MFE module and utilize only the backbone and transformer decoder and propose a Random Query (RQ) strategy. Specifically, we feed the trained seen and candidate queries, along with new randomly initialized queries $\mathbf{Q}_r$, into the Transformer decoder. These queries collectively interact with the visual features to generate segmentation masks, where the $\mathbf{Q}_r$ serve to enrich query diversity and improve coverage of unannotated regions. Unlike the trained queries, $\mathbf{Q}_r$ are not supervised during training and are introduced only at inference time to probe unannotated or ambiguous regions in the image. By increasing the density and diversity of queries in the feature space, $\mathbf{Q}_r$ enhance the model's ability to explore underrepresented regions that might correspond to unseen or latent objects. Importantly, $\mathbf{Q}_r$ act as complementary probes that are not biased by learned semantic categories, enabling the model to capture alternative activation patterns and recover instances that trained queries might overlook. More detailed inference process and the role of $\mathbf{Q}_r$ are illustrated in the ***Supplementary Materials***.

4 Experiments

Dataset. To assess the effectiveness of our proposed method, we conduct experiments on the widely-used benchmark COCO-Stuff [3] and PASCAL VOC [16], focusing on the task of zero-shot semantic segmentation (ZSS). We adopt the same seen and unseen class splits as in prior works [14, 37, 38]. Specifically, COCO-Stuff consists of a total of 171 classes with 156 seen and 15 unseen classes according to the standard protocol. The dataset includes 118,287 images for training and 5,000 images for testing. PASCAL VOC contains 10,582 images for training and 1,449 images for validation, including 15 seen and 5 unseen classes.
Implementation Details. The CLIP model applied in our method is based on the ViT-B/16 model, and the channel of the output text features is 512. All the experiments are conducted on 8 V100 GPUs, and the batch size is set to 16 for both datasets. The iterations are set to 20K and 80K for PASCAL VOC and COCO-Stuff. $\mathcal{L}_{cls}$ in $\mathcal{L}_{match}$ is focal loss [23] and the $\mathcal{L}_{mask}$ is a combination of IoU loss and DICE loss [11] in $\mathcal{L}_{match}$. We choose Mask2Former

Models	Backbone	PASCAL VOC			COCO-Stuff		
		hIoU	**sIoU**	**uIoU**	**hIoU**	**sIoU**	**uIoU**
SPNet [35]	ResNet101 [11]	26.1	78.0	15.6	14.0	35.2	8.7
ZS3 [1]		28.7	77.3	17.7	15.0	34.7	9.5
CaGNet [20]		39.7	78.4	26.6	18.2	33.5	12.2
SIGN [29]		41.7	75.4	28.9	20.9	32.3	15.5
Joint [3]		45.9	77.7	32.5	-	-	-
ZegFormer [10]		73.3	86.4	63.6	34.8	36.6	33.2
Zzseg [6]	ViT-B [15]	77.5	83.5	72.5	37.8	39.3	36.3
DeOP [14]		80.8	88.2	74.6	38.2	38.0	38.4
ZegCLIP [38]		84.3	91.9	77.8	40.8	40.2	41.4
OTSeg [33]		84.5	**92.1**	78.1	41.4	41.4	41.4
Ours	ResNet101 [11]	**85.3**	87.7	**83.1**	**42.5**	**42.6**	**42.4**

Table 1: Comparison with others. **Bold** and underline indicates the best and the second-best.

Method	hIoU	sIoU	uIoU
Baseline	24.6	31.8	20.0
+ SM	33.3	36.4	30.8
+ SM + MFE	36.3	36.8	35.8
+ SM + MFE + RQ	**36.6**	**36.8**	**36.4**

Table 2: Ablation on the proposed module.

Method	hIoU	sIoU	uIoU
F	33.8	35.8	32.1
F + MLP	30.4	36.5	26.1
$\mathbf{C}_u$	**36.6**	**36.8**	**36.4**

Table 3: Ablation on the candidate class embedding.

Structure of MFE	hIoU	sIoU	uIoU
No Norm	35.8	36.2	35.3
BN	36.0	36.5	35.6
GN	**36.6**	**36.8**	**36.4**

Table 4: Ablation on the structure of MFE.

[11] with ResNet101 as the backbone with all other hyperparameters unchanged. 50 unseen and 50 random queries are added during inference. We apply the harmonic mean IoU (hIoU) following previous works [38] where $hIoU = \frac{2 \cdot sIoU \cdot uIoU}{sIoU + uIoU}$ as the metric where $sIoU$ and $uIoU$ indicate the mIoU (mean intersection over union) of the seen classes and unseen classes, respectively. More details are in the ***Supplementary Materials***.

4.1 Comparison with State-of-the-art methods

Table 1 shows that our method achieves state-of-the-art performance under the ResNet101 backbone, outperforming even transformer-based methods in terms of overall hIoU and uIoU. Specifically, we obtain the highest uIoU on PASCAL VOC (83.1%), which is a significant margin over ZegCLIP (77.8%) and OT-Seg (78.1%), demonstrating stronger generalization to unseen classes. Importantly, our method also achieves the best hIoU on both datasets, indicating more balanced segmentation. Although our sIoU on VOC is slightly lower than transformer-based counterparts, this reflects our model's ability to mitigate seen-class bias.

4.2 Ablation Studies

To evaluate the effectiveness of our method, we conduct ablation studies on COCO-Stuff for 40K iterations using ResNet-50 as the backbone, with all hyperparameters unchanged. Additional experiments are provided in the ***Supplementary Materials*** due to space limitations.

Ablations on Proposed Modules. Table 2 summarizes the contributions of each proposed module. The baseline achieves suboptimal performance, with lower unseen IoU (uIoU) indicating limited generalization. Adding the Split Matching (SM) module significantly improves the model's ability to capture unseen classes, as reflected in higher uIoU. The Multi-scale Feature Enhancement (MFE) further boosts the model's performance by enhancing the interaction between queries and features. Finally, the inclusion of random queries leads to the best results across all metrics, demonstrating our contribution.

Ablations on Unseen Class Embedding. Table 3 presents an ablation study on the design of unseen class embeddings $\mathbf{C}_u$. Using raw dense features **F** from the backbone provides

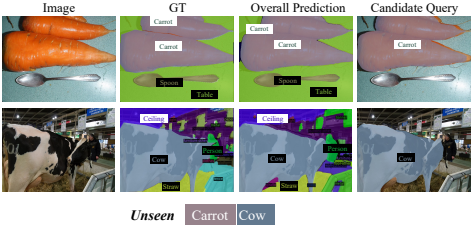

Figure 5: Candidate query predictions visualization, with each row displaying images, GT, overall and candidate query predictions.

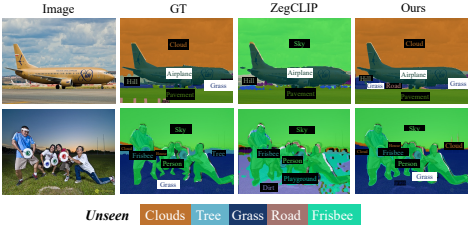

Figure 6: Visualization on predictions where each line shows the image, ground truth, ZegCLIP's prediction, and ours.

a reasonable baseline (uIoU: 32.1), while adding an MLP projection degrades performance, likely due to semantic distortion. In contrast, using CLS tokens from pseudo-masked regions yields the best results (uIoU: 36.4), suggesting that region-level CLS tokens better preserve semantic alignment. This also enables direct concatenation with seen-class text embeddings, maintaining cross-modal consistency without modality mismatch.

Ablations on structure of MFE. Table 4 presents the ablation study on the structure of the MFE module by comparing different normalization strategies: No Normalization (No Norm), Batch Normalization (BN), and Group Normalization (GN). Among the three, GN achieves the best performance across all metrics, yielding the highest hIoU (36.6), sIoU (36.8), and uIoU (36.4). These results demonstrate that incorporating Group Normalization into MFE significantly improves both seen and unseen segmentation performance.

4.3 Qualitative Analysis

Due to the space limitations, more results are in ***Supplementary Materials***.

Visualization of candidate queries. Fig. 5 visualizes the predictions of candidate queries. Notably, these queries successfully activate on previously unannotated regions, enabling the model to localize unseen classes such as *carrot* and *cow*. This demonstrates that candidate queries can effectively discover latent classes and assign semantically correct class labels, even without explicit supervision. **Prediction Visualzation.** Each row in Fig. 6 shows the input image, ground truth, ZegCLIP's prediction, and ours. Our method successfully segments unseen classes such as "clouds", "bushes", and "playingfield", which are missed or mislabeled by ZegCLIP. Notably, in both examples, the unseen class "clouds" is correctly identified by our model, demonstrating better generalization to unseen concepts.

5 Conclusion

In this paper, we propose **Split Matching** (SM), a novel decoupled assignment strategy tailored for query-based models in ZSS. By separating queries into seen and candidate groups and optimizing them with respect to annotated and unannotated regions, SM effectively mitigates the seen-class bias caused by incomplete supervision. To further facilitate the discovery of unseen classes, we leverage CLIP-derived pseudo masks and region-level embeddings, and introduce a **Multi-scale Feature Enhancement** (MFE) module to refine spatial representations. Additionally, we incorporate a **Random Query** (RQ) strategy during inference to improve query diversity and coverage of unannotated regions. Extensive experiments on standard ZSS benchmarks demonstrate that our approach achieves state-of-the-art results.

References

[1] Donghyeon Baek, Youngmin Oh, and Bumsub Ham. Exploiting a joint embedding space for generalized zero-shot semantic segmentation. In *ICCV*, 2021.

[2] Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. In *NeurIPS*, 2019.

[3] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, 2018.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[5] Jialei Chen, Daisuke Deguchi, Chenkai Zhang, Xu Zheng, and Hiroshi Murase. Clip is also a good teacher: A new learning framework for inductive zero-shot semantic segmentation. *arXiv preprint arXiv:2310.02296*, 2023.

[6] Jialei Chen, Daisuke Deguchi, Chenkai Zhang, and Hiroshi Murase. Generalizable semantic vision query generation for zero-shot panoptic and semantic segmentation. *arXiv preprint arXiv:2402.13697*, 2024.

[7] Jialei Chen, Daisuke Deguchi, Chenkai Zhang, Xu Zheng, and Hiroshi Murase. Frozen is better than learning: A new design of prototype-based classifier for semantic segmentation. *PR*, 2024.

[8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2017.

[9] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[10] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021.

[11] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022.

[12] Jiaxin Cheng, Soumyaroop Nandi, Prem Natarajan, and Wael Abd-Almageed. Sign: Spatial-information incorporated generative network for generalized zero-shot semantic segmentation. In *ICCV*, 2021.

[13] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[14] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *CVPR*, 2022.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[16] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 2015.

[17] Zhangxuan Gu, Siyuan Zhou, Li Niu, Zihan Zhao, and Liqing Zhang. Context-aware feature generation for zero-shot semantic segmentation. In *ACM MM*, 2020.

[18] Meng-Hao Guo, Cheng-Ze Lu, Qibin Hou, Zhengning Liu, Ming-Ming Cheng, and Shi-Min Hu. Segnext: Rethinking convolutional attention design for semantic segmentation. In *NeurIPS*, 2022.

[19] Cong Han, Yujie Zhong, Dengjie Li, Kai Han, and Lin Ma. Zero-shot semantic segmentation with decoupled one-pass network. *arXiv preprint arXiv:2304.01198*, 2023.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[21] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *ICML*, 2019.

[22] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, 2022.

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[24] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[25] Zhenzhen Quan, Jialei Chen, Daisuke Deguchi, Jie Sun, Chenkai Zhang, Yujun Li, and Hiroshi Murase. Semantic matters: A constrained approach for zero-shot video action recognition. *Pattern Recognition*, page 111402, 2025.

[26] 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 *ICML*, 2021.

[27] Size Wu, Wenwei Zhang, Lumin Xu, Sheng Jin, Xiangtai Li, Wentao Liu, and Chen Change Loy. CLIPSelf: Vision transformer distills itself for open-vocabulary dense prediction. In *ICLR*, 2024.

[28] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018.

[29] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero-and few-label semantic segmentation. In *CVPR*, 2019.

[30] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *NeurIPS*, 2021.

[31] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In *ECCV*, 2022.

[32] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *CVPR*, 2023.

[33] Jong Chul Ye, Yujin Oh, et al. Otseg: Multi-prompt sinkhorn attention for zero-shot semantic segmentation. In *ECCV*, 2024.

[34] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional CLIP. In *NeurIPS*, 2023.

[35] Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, et al. Recognize anything: A strong image tagging model. *arXiv preprint arXiv:2306.03514*, 2023.

[36] Xu Zheng, Yunhao Luo, Pengyuan Zhou, and Lin Wang. Distilling efficient vision transformers from cnns for semantic segmentation. *Pattern Recognition*, 158:111029, 2025.

[37] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *ECCV*, 2022.

[38] Ziqin Zhou, Yinjie Lei, Bowen Zhang, Lingqiao Liu, and Yifan Liu. Zegclip: Towards adapting clip for zero-shot semantic segmentation. In *CVPR*, 2023.