

# TFM<sup>2</sup>: Training-Free Mask Matching for Open-Vocabulary Semantic Segmentation

Yaoxin Zhuo<sup>1\*</sup>, Zachary Bessinger<sup>2</sup>, Lichen Wang<sup>2</sup>, Naji Khosravan<sup>2</sup>, Baoxin Li<sup>1</sup>, Sing Bing Kang<sup>2</sup>

<sup>1</sup> Arizona State University, <sup>2</sup> Zillow Group

{yzhuo6, baoxin.li}@asu.edu

{zacharybe, lichenw, najik, singbingk}@zillowgroup.com

## Abstract

The potential of Open-Vocabulary Semantic Segmentation (OVSS) in few-shot scenarios is not fully explored due to the complexity of extending few-shot concepts to semantic segmentation tasks. To address this challenge, we propose **Training-Free Mask Matching (TFM<sup>2</sup>)**, an efficient, mask-based adapter method that enhances OVSS models for the few-shot open vocabulary semantic segmentation task. **TFM<sup>2</sup>** is a key-value cache that explicitly designed for image masks. We introduce three modules to construct and refine the mask cache, subsequently enhancing the OVSS mask classification performance. Comprehensive experiments demonstrate that **TFM<sup>2</sup>** improves the performance of state-of-the-art OVSS methods by a margin of 1% to 5% across different settings. Moreover, **TFM<sup>2</sup>** is not limited to any specific methods or backbones. This work underscores the importance and potential of few-shot data in OVSS and presents a significant step toward leveraging this potential.

## 1. Introduction

Semantic segmentation is a fundamental computer vision task with many diverse applications, ranging from medical imaging and autonomous driving to augmented reality. It involves assigning a categorical label to every pixel. Traditional semantic segmentation methods belong to a close-set setting, which assumes a predetermined set of class categories consistent between the training and testing datasets. This assumption hinders the expansion of category numbers during the inference stage. Such limitations pose practical challenges where categories may include both seen and unseen elements. Seen elements are categories from training, while unseen elements are new categories emerging post-deployment. This limits applicability in practical environments. OVSS addresses this limitation by recognizing ar-

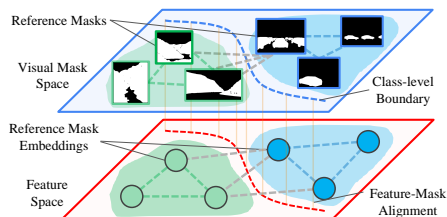


Figure 1. Concept of TFM<sup>2</sup>. It effectively aligns the mask similarity in visual space as well as the corresponding mask embeddings in feature space. A mask cache is built to generate classification logits, enhancing the final semantic segmentation prediction.

bitrary category regions through the development of vision-and-language pre-trained (VLP) models.

VLP models and their applications [2, 3, 8, 18, 30, 31, 44, 50, 51, 64, 69, 70, 79, 80] have been developed to generate robust cross-modal embeddings, demonstrating successful performance across various computer vision tasks. Inspired by their success, several OVSS methods have emerged, leveraging VLP models to overcome the constraints inherent in traditional semantic segmentation approaches. Most OVSS methods [57, 59, 65] utilize the text embeddings of VLP models to classify various mask proposals. By leveraging the power of text embeddings, OVSS methods can effectively segment images with arbitrary categories. Recent advancements use cache-based adapter networks to address resource constraints, reducing the need for full-network fine-tuning. Studies [49, 67, 68, 78] have shown that training-free key-value cache models effectively improve model performance with limited few-shot data. However, these models mainly use global image features. For local mask regions, a common approach is to crop the mask from the original image. Such a way inadvertently omits the comprehensive global context of the image.

Our key insight identifies a gap in OVSS research: the unexplored benefits of incorporating few-shot target domain data. Existing cache-based methods focus on global features, overlooking the need for region-based features in

\*Work was done while Yaoxin Zhuo was an intern at Zillow Group.

fine-grained segmentation. It is non-trivial to associate visual features with mask regions while ensuring high data and parameter efficiency. We propose a novel approach, **Training-Free Mask Matching (TFM<sup>2</sup>)**, which addresses these challenges by balancing task-specific performance with open-set generalization, a capacity that is critical for real-world applications. Unlike traditional closed-set segmentation, our OVSS approach recognizes arbitrary category masks, including unseen categories. Our method enhances mask classification performance during semantic segmentation inference, particularly in situations where only a limited number of annotated masks are available for the target categories. This reflects common real-world conditions where re-training or fine-tuning models is impractical due to insufficient target category data. By using limited data for accurate inference, TFM<sup>2</sup> bridges the gap between theoretical training and practical application, enhancing segmentation performance and flexibility.

As shown in Fig. 1, TFM<sup>2</sup> utilizes the few-shot masks to build up the training-free key-value mask cache, which is able to enhance the mask proposal classification performance of the trained OVSS model during inference. It is formed via three modules: a Dynamic Filter module, a Channel Reduction module, and a Feature Alignment module. Each of these contributes to the construction and refinement of the mask cache. TFM<sup>2</sup> is highly adaptable — it can easily adjust to newly added segmentation classes by updating cache key-value pairs for new regions, facilitating efficient and convenient continual model expansion. Our contributions are summarized as follows:

- We design a key-value mask cache based on limited few-shot data that improves upon open-vocabulary semantic segmentation metrics in a training-free manner.
- We employ three modules: Dynamic Filter, Channel Reduction, and Feature Alignment to further refine the mask cache, leading to enhanced mask cache.
- Comprehensive experiments demonstrate the strong generalization ability of our training-free TFM<sup>2</sup> on various models, backbones and datasets.

## 2. Related Work

### 2.1. Zero-Shot or Few-Shot Semantic Segmentation

Traditional semantic segmentation methods [1] classify each pixel in an image into a set category. MaskFormer [14] innovatively divides this task into mask generation and mask classification, showing competitive performance against traditional FCN-based methods [12, 43, 62]. Mask2Former [13] employs a mask-attention mechanism to focus on relevant image regions, with architecture based on DETR [9]. There are also some works [21, 25, 26, 29, 34–36, 61, 63, 66, 71, 77] that paid attention to the few-shot semantic segmentation task. In the common configurations

for zero/few-shot semantic segmentation, classes are partitioned into training and testing sets without overlapping. This setup is designed to evaluate the ability of a model to generalize to unseen classes during testing. While these methods have taken care of the masks between the training and testing phases, there remains a critical drawback: the images used for testing may have been seen by the model, which is unfair for evaluation.

### 2.2. Open-Vocabulary Semantic Segmentation

Unlike traditional semantic segmentation, recent open-vocabulary segmentation works [4, 5, 11, 15, 15, 32, 38, 39, 41, 46, 52–54, 57–59, 72, 73] showed it can handle unseen categories, making it closer to real-world scenarios. The pioneering work [72] learns a joint embedding from visual and word features for concepts and images. SimSeg [59] proposes a two-stage framework to decouple the task into class-agnostic mask generation and mask category classification. SAN [57, 58] separates mask recognition from mask prediction using a side-adaptor network that learns from frozen CLIP features. FOSSIL [4] focuses on unsupervised settings by leveraging a text-conditioned diffusion model to generate visual embeddings, which significantly enhance retrieval inference performance. OVSeg [32] addresses CLIP’s limitations in classifying masked regions for semantic segmentation tasks by fine-tuning the CLIP with the COCO-Caption dataset. MaskCLIP [73] treating mask proposals as the attention mask in the CLIP for computational efficiency. SegCLIP [38] is a CLIP-based model that can be trained with annotation-free image-text pairs for weakly-supervised semantic segmentation. FreeDA [5] leveraged the diffusion model to strengthen fine-grained relationships between visual regions and semantic classes, further enhancing local-global similarities during semantic segmentation inference. However, they always require re-training extra parameters or require extra pre-trained models. Both of which incur relatively high computation costs. Our work builds on top of open-vocabulary segmentation models to use limited few-shot masks to boost performance on the target dataset in a training-free fashion.

### 2.3. Adapter for Few-Shot Learning.

Vision-and-Language Pre-trained (VLP) models provide the transferability of few-shot learning. Some methods rely on utilizing prompt learning to enhance the VLP performance. For example, CoOp [76] designed a set of learnable prompt tokens for the text encoder to improve the image classification performance. CoCoOp [75] proposed an extra network to generate image tokens for text features based on CoOp, which is targeted at generating input-conditional prompts. CLIP-Adapter [19] was proposed as an alternative to prompt-based approaches for few-shot image classification tasks. By fine-tuning extra layers with the de-

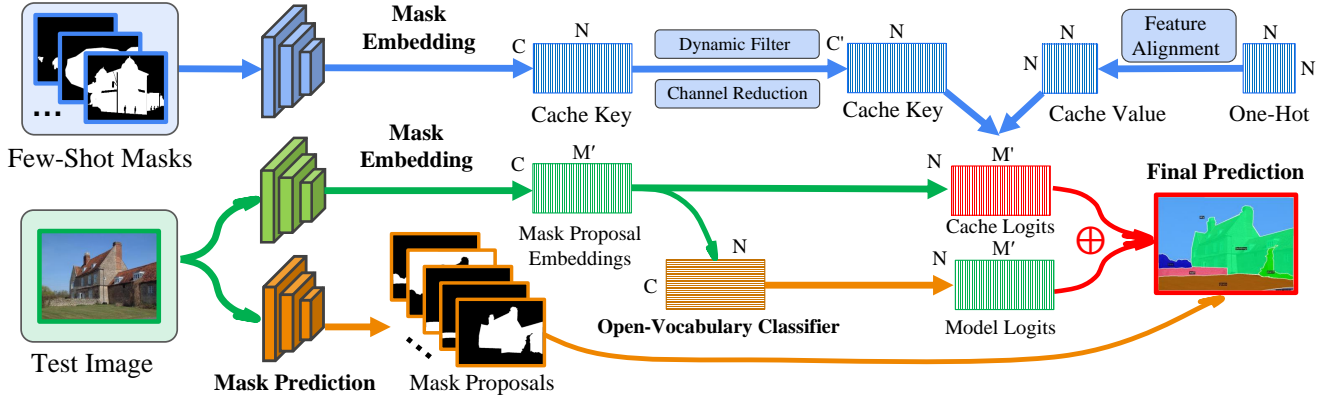


Figure 2. The framework of TFM<sup>2</sup>. It utilizes the few-shot masks to construct the Mask Cache. The Dynamic Filter and Channel Reduction modules refine the key of Mask Cache, while the Feature Alignment module refines the value of Mask Cache. During the inference stage, TFM<sup>2</sup> will provide the Cache logits for mask proposals. The original classification logits will be fused with Cache logits to further enhance the semantic segmentation results. In the figure, the blue arrows illustrate the process of building and refining the mask cache using the three modules. The green and orange arrows represent the inference steps of a frozen OVSS model given a test image. The red arrows depict the fusion of mask proposal logits between the OVSS classifier and TFM<sup>2</sup>, resulting in the final prediction with mask proposals.

signed residual connections, CLIP-Adapter achieves similar or even better performance on multiple few-shot image classification datasets. Inspired by the similarity-based retrieval ideas, Zhang *et al.* proposed the training-free adaptation method termed Tip-Adapter [68]. Tip-Adapter constructs the adapter by key-value cache model from few-shot training images while keeping the CLIP frozen.

To further incorporate diverse pre-training knowledge to assist few-shot image classification, CaFo [67] was proposed in the style of combining GPT3 [6], CLIP [44], DINO [10] and DALL-E [45]. It utilizes the GPT3 to generate text prompts, which are the input of DALL-E to generate pseudo images for each class. The generated images and real training images will be mixed to build up the cache model by using CLIP and DINO visual features. SuS-X [49] constructs the support set to infuse the visual information and consider the distances to further improve the prediction of the model. APE [78] is a prior refinement approach focusing on refining pre-trained CLIP visual features. It maintains the trilateral affinity relations among the testing image visual feature, text features, and training image visual features in the computational efficiency fashion.

However, these methods are designed only for the few-shot image classification scenario, which requires only the image-level features to build up the cache. How to achieve mask adapter for high-performance semantic segmentation is still not fully realized. Extending these adapters from entire image-level features to mask-level features is still under-explored. Unlike these Adapter works, we propose TFM<sup>2</sup> by exploring how to build up a key-value cache specifically for masks, resulting in a training-free and very versatile method to enhance the pre-trained OVSS models.

### 3. Method

We show the framework of TFM<sup>2</sup> in Fig. 2. The goal of few-shot OVSS is to utilize limited few-shot masks from the target dataset, further improving the performance of pre-trained OVSS methods.

#### 3.1. Mask Cache Construction

Generating the mask cache, which fully preserves the visual-textual knowledge of the target object, is the crucial step. To achieve this, feature-vector extraction in a compact format is essential. Current mainstream OVSS models [32, 57, 59] typically employ two decoupled branches to achieve semantic segmentation: one for mask prediction and one for mask classification. To this end, we design a specific way to extract mask features.

Given the  $K$ -shots  $N$ -classes reference masks (source from the training split in the target dataset), we can obtain their corresponding images denoted by  $\mathbf{I} \in \mathbb{R}^{3 \times W \times H}$ , where  $W$  represents the width of the image and  $H$  denotes the height. It is crucial to highlight that these images only offer the annotations of these provided  $K \cdot N$  mask regions. The pre-trained OVSS model can generate the binary mask proposals (by mask prediction branch) and corresponding mask embeddings (by mask classification branch) as follows:

$$\hat{\mathbf{M}}_{\mathbf{I}} = \text{MaskPrediction}(\mathbf{I}), \quad (1)$$

$$\hat{\mathbf{V}}_{\mathbf{I}} = \text{MaskEmbedding}(\mathbf{I}), \quad (2)$$

where the  $\hat{\mathbf{M}}_{\mathbf{I}} \in \{0, 1\}^{\hat{M} \times W \times H}$  represents the binary mask proposals, the  $\hat{M}$  is the number of predicted mask proposals. It can also be viewed as  $\hat{\mathbf{M}}_{\mathbf{I}} = [\hat{m}_1, \hat{m}_2, \dots, \hat{m}_{\hat{M}}]$ ,

where  $\hat{m} \in \{0, 1\}^{W \times H}$  represents every individual binary mask proposals. The  $\hat{\mathbf{V}}_{\mathbf{I}} \in \mathbb{R}^{\hat{M} \times C}$  are the  $C$ -dimensional  $L_2$  normalized embeddings of predicted mask proposals. It can also be viewed as  $\hat{\mathbf{V}}_{\mathbf{I}} = [\hat{v}_1, \hat{v}_2, \dots, \hat{v}_{\hat{M}}]$ , where  $\hat{v} \in \mathbb{R}^{1 \times C}$  represents mask proposal visual embedding.

Besides the mask prediction part, the OVSS models also utilize the text encoders from VLP models to do the mask classification. This is achieved by concatenating each class text embeddings  $w_n \in \mathbb{R}^{1 \times C}$  of all categories, represented as  $\mathbf{W}_{\text{classifier}} \in \mathbb{R}^{N \times C}$ , where  $N$  is the number of classes in the target dataset. These text embeddings are derived by integrating the class names into predefined sentence templates and fed into text encoders of the VLP model. Subsequently, the final category label logits are as follows:

$$\hat{\mathbf{L}}_{\mathbf{I}} = \hat{\mathbf{V}}_{\mathbf{I}} \times \mathbf{W}_{\text{classifier}}^{\top}, \quad (3)$$

where the  $\hat{\mathbf{L}}_{\mathbf{I}} \in \mathbb{R}^{\hat{M} \times N}$  is the classification logits for the predicted mask proposals. For the OVSS task, the final segmentation result of image  $\mathbf{I}$  could be achieved by:

$$\hat{\mathbf{S}}_{\mathbf{I}} = \hat{\mathbf{L}}_{\mathbf{I}}^{\top} \times \hat{\mathbf{M}}_{\mathbf{I}}, \quad (4)$$

where  $\hat{\mathbf{S}}_{\mathbf{I}} \in \mathbb{R}^{N \times W \times H}$  is the output in standard semantic segmentation format. It will be used to do softmax and then compared with ground truth to get the final performance. Based on these, we designed one way to extract the mask region-related visual features to further build up the mask cache. Given the set of few-shot reference masks,  $\mathbf{M}$ , we first employ the Intersection over Union (IoU) between the pre-trained model's mask proposals  $\hat{\mathbf{M}}_{\mathbf{I}}$  from Eq. (1) and the reference mask  $m \in \mathbf{M}$  within the image  $\mathbf{I}$ :

$$v_m = \hat{v}_i, \text{ where } i = \arg \max_i \left\{ \frac{m \cap \hat{m}_i}{m \cup \hat{m}_i} \right\}, \forall i \in [1, \hat{M}], \quad (5)$$

where we calculate the IoU between the reference mask  $m$  and every mask proposal  $\hat{m} \in \hat{\mathbf{M}}_{\mathbf{I}}$ . The feature of the mask proposal with the highest IoU will be selected as the reference mask feature. We average the visual features of all  $K$ -shot reference mask regions (not images) belonging to the same class to construct the class  $n$  mask cache key  $v_n = \frac{1}{K} \sum_{k=1}^K v_k$ . The  $v_n$  represents the averaged mask visual features of all  $K$  masks for a given class  $n$ , where  $n \in [1, 2, \dots, N]$ . We then concatenate the  $N$  class mask features together to form the key  $\mathbf{F}_{\text{train}} = [v_1, v_2, \dots, v_N]$ , where the  $\mathbf{F} \in \mathbb{R}^{N \times C}$  are the full set of keys of the mask cache model. We can also concatenate the one-hot label vectors as the set of mask cache values,  $\mathbf{Y}_{\text{train}} = \text{OneHotLabel}([\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_N])$ , where the  $\mathbf{Y}_{\text{train}} \in \{0, 1\}^{N \times N}$ . The cache classification logits can be obtained with key and value as:

$$\hat{\mathbf{L}}_{\mathbf{I}}^{\text{cache}} = \hat{\mathbf{V}}_{\mathbf{I}} \times \mathbf{F}_{\text{train}}^{\top} \times \mathbf{Y}_{\text{train}}, \quad (6)$$

where the  $\hat{\mathbf{L}}_{\mathbf{I}}^{\text{cache}}$  represents the mask proposal classification logits. The mask cache will measure the similarities between the mask proposals and the cache keys, then use the similarity scores to be multiplied by the cache values to generate the final classification predictions. It leverages the reference information from the reference masks to improve mask proposal classification accuracy.

### 3.2. Intra-Class Dynamic Filter

Adding more samples might slightly improve performance; however, it is not the primary goal of this paper, and we have limited data available. Additionally, increasing the number of samples could introduce more outliers and noise, potentially affecting the results. Some outlier mask visual features begin to manifest, influencing the key of the mask cache since we average all mask features to form the key.

We introduce a way to selectively exclude uninformative reference mask visual features, particularly in the large number of shots (like 16- and 32-shot) settings. In the  $K$ -shots  $N$ -classes setting, we have  $K$ -many  $L_2$ -normalized training mask visual features,  $v_1^n, v_2^n, \dots, v_K^n$  for the class  $n$ . For the  $k$ -th sample, we first calculate its averaged intra-class cosine similarities to other samples in the same class as  $s_{v_k^n}^{\text{intra}} = \frac{1}{K-1} \sum_{i=1, i \neq k}^K d_{\text{cos}}(v_k^n, v_i^n)$ . The  $d_{\text{cos}}$  is the calculation of cosine similarity between two mask visual features. Based on that, we can get the overall classes' average intra-class cosines similarity for all  $K \cdot N$  masks  $S_{\text{global}}^{\text{intra}} = \frac{1}{KN} \sum_{i=1}^K \sum_{j=1}^N s_{v_i^j}^{\text{intra}}$ . We can also calculate the class  $n$  intra-class average cosine similarity  $S_n^{\text{intra}} = \frac{1}{K} \sum_{i=1}^K s_{v_i^n}^{\text{intra}}$ . When building up the mask cache's keys for each class, we filter out the samples  $v_k^n$  if their intra-class similarity  $s_{v_k^n}^{\text{intra}}$  is lower than both  $S_{\text{global}}^{\text{intra}}$  and  $S_n^{\text{intra}}$ . After filtering out samples, the new class key  $v'_n$  would be:

$$v'_n = \frac{\sum_{k=1}^K v_k}{|\sum_{k=1}^K v_k|}, \text{ if } v_k \text{ is kept}, \quad (7)$$

where  $|\sum_{k=1}^K v_k|$  is the number of remaining mask visual features. Finally, the new cache key  $\mathbf{F}'_{\text{train}}$  would be:

$$\mathbf{F}'_{\text{train}} = [v'_1, v'_2, \dots, v'_N], \quad (8)$$

Each key consists of all mask visual features filtered by the designed standard based on intra-class similarity. In this manner, the mask cache aims to capture and store the representative visual features of each class in the keys.

### 3.3. Inter-Class Channel Reduction

The distribution of inter-class samples directly influences the classification boundaries of each category, which is especially crucial in our training-free framework. Success hinges on utilizing the VLP model, which incorporates joint embeddings of both visual and textual features. However,

the VLP joint embeddings often encompass both domain-irrelevant and redundant information due to the pre-training process on large-scale noisy data. To address this issue, we design an efficient way to select the most discriminative feature channels  $C'$  from the original VLP embeddings with channels  $C$  by the standards of minimal inter-class similarity and maximum inter-class variance.

We set a binary flag set  $\mathbf{B} \in \{0, 1\}^C$ , where  $\mathbf{B}_c = 1$  ( $c = 1, 2, 3, \dots, C$ ) represents whether the  $c$ -th element in the feature vector will be kept. Notably, this  $\mathbf{B}$  targets the feature channels of the mask cache, which provides mask classification logits. Since VLP models provide strong joint-embedding feature space, the  $N$  category text embeddings can approximate the visual prototypes for these mask classes, which means we can view the  $N$  category text embeddings as the visual clustering centers for all  $K$  mask visual features. The optimization goal is to minimize the inter-class similarity of  $K \cdot N$  masks as follows:

$$\min_{\mathbf{B}} S^{\text{inter}} = \frac{\sum_{i=1}^N \sum_{j=1, j \neq i}^N d_{\cos}(w_i \circ \mathbf{B}, w_j \circ \mathbf{B})}{(N-1)^2}, \quad (9)$$

where the  $w_n \in \mathbb{R}^{1 \times C}$  is the text embedding of class  $n$ .  $w_i \circ \mathbf{B}$  means only keep the selected feature elements and  $\mathbf{B}\mathbf{B}^T = C'$ . Since the text features are also  $L_2$ -normalized, we can calculate their inter-class channel similarities by:

$$S^{\text{inter}} = \sum_{c=1}^C S_c^{\text{inter}} = \sum_{c=1}^C \frac{\sum_{i=1}^N \sum_{j=1, j \neq i}^N t_c^i t_c^j}{(N-1)^2}, \quad (10)$$

where the  $c$  is the index of selected feature channels with  $\mathbf{B}_c = 1$ ,  $t_c$  represents the element of the text feature  $t$  at channel  $c$ , the  $\frac{1}{N^2} \sum_{i=1}^N \sum_{j=1, j \neq i}^N t_c^i t_c^j$  represents the average inter-class similarities of the  $c$ -th channel. Solving the Eq. (9) optimization can be achieved by selecting the  $C'$  channels with the smallest inter-class similarities. We calculate the inter-class channel variance for the channel  $c$  as:

$$V_c^{\text{inter}} = \frac{1}{N} \sum_{i=1}^N (t_c^i - \bar{t}_c)^2, \quad (11)$$

where  $\bar{t}_c$  represents the average variance of the  $c$ -th channel in class text embedding  $t$ . This variance criterion is selecting the  $C'$  channels with the largest variances. Finally, we consider both inter-class similarities and inter-class variance by a balance factor  $\lambda_1$  into the channels selection:

$$J_c^{\text{inter}} = \lambda_1 S_c^{\text{inter}} - (1 - \lambda_1) V_c^{\text{inter}}, \quad (12)$$

We select the  $C'$  with the smallest  $J_c^{\text{inter}}$  for the final key of the mask cache. The mask proposal visual features and cache keys will be multiplied with  $\mathbf{B}$  first to reduce the channel dimension. The new mask cache classification logits for the mask proposals would be:

$$\hat{\mathbf{L}}_{\mathbf{I}}^{\text{cache}} = (\hat{\mathbf{V}}_{\mathbf{I}} \circ \mathbf{B}) \times (\mathbf{F}'_{\text{train}} \circ \mathbf{B})^{\top} \times \mathbf{Y}_{\text{train}}, \quad (13)$$

where the  $(\hat{\mathbf{V}}_{\mathbf{I}} \circ \mathbf{B})$  and  $(\mathbf{F}'_{\text{train}} \circ \mathbf{B})$  are reducing the dimensions of the mask proposal features and cache keys. The remaining feature channels focus on the most discriminative information, which is essential for accurately measuring the similarity between mask proposals and cache keys. This design can help the mask cache to improve the ability to measure the similarity between mask proposals and keys.

### 3.4. Cache Value Feature Alignment

While the mask cache effectively links mask visual embeddings and text embeddings through keys and values, the term  $(\mathbf{F}'_{\text{train}} \circ \mathbf{B})$  in Eq. (13) cannot accurately match the one-hot labels  $\mathbf{Y}_{\text{train}}$ . This discrepancy necessitates the subsequent multiplication of these terms with  $\mathbf{Y}_{\text{train}}$ . To regularize the feature space, we compute the Kullback-Leibler (KL) divergence for measuring the difference between the distribution of the visual-text embedding similarities and the one-hot labels. For evaluating the capacity of the mask cache keys, we calculate the KL divergence as follows:

$$D_{\text{KL}} = d_{\text{KL}}(\mathbf{Y}_{\text{train}}, \text{softmax}(\mathbf{F}'_{\text{train}} \circ \mathbf{B} \times \mathbf{W}_{\text{classifier}}^{\top})), \quad (14)$$

where the  $d_{\text{KL}}$  is the KL-divergence function and  $\mathbf{F}'_{\text{train}} \circ \mathbf{B} \times \mathbf{W}_{\text{classifier}}^{\top}$  is measuring the similarity between visual embedding and text embedding for each class. We use the KL-divergence to measure the distribution gap between one hot label and cache keys. Then we can further refine the mask cache values by the KL-divergence score  $D_{\text{KL}}$ :

$$\mathbf{Y}'_{\text{train}} = \mathbf{Y}_{\text{train}} e^{(\lambda_2 D_{\text{KL}})}, \quad (15)$$

where  $\lambda_2$  is the smoothing factor.  $e^{(\lambda_2 D_{\text{KL}})}$  can be viewed as a soft score for the mask cache value, indicating the information gap between the mask cache's keys and values.

### 3.5. Mask Proposal Classification Logits Fusion

After applying the above three modules to refine the keys and values of the mask cache, the final adaption of TFM<sup>2</sup> on the trained OVSS model can be achieved through a weighted average of the original mask proposal classification logits and the logits emanating from the cache model. During the inference stage, the original mask proposal classification logits  $\hat{\mathbf{L}}_{\text{origin}}$  can be achieved by Eq. (3). The mask proposal visual features  $\hat{\mathbf{V}}_{\mathbf{I}}$  can serve as the queries for retrieval within TFM<sup>2</sup>, thus obtaining the mask cache logits by Eq. (8), Eq. (13) and Eq. (15):

$$\hat{\mathbf{L}}_{\mathbf{I}}^{\text{cache}} = (\hat{\mathbf{V}}_{\mathbf{I}} \circ \mathbf{B}) \times (\mathbf{F}'_{\text{train}} \circ \mathbf{B})^{\top} \times \mathbf{Y}'_{\text{train}} \quad (16)$$

The new mask classification logits can be obtained by:

$$\hat{\mathbf{L}}_{\text{final}} = \lambda_3 \hat{\mathbf{L}}_{\text{origin}} + (1 - \lambda_3) \hat{\mathbf{L}}_{\text{cache}} \quad (17)$$

where  $\lambda_3$  is the balance factor that harmonizes the original mask proposal classification logits and the mask cache classification logits. For the mask classification logits fusion,

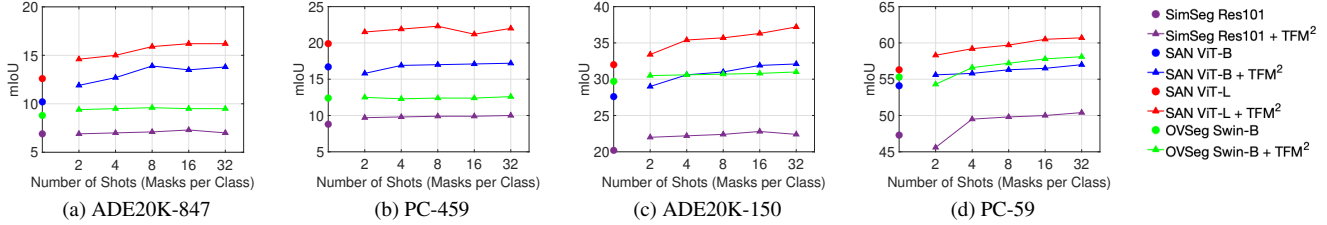


Figure 3. The mIoU of TFM<sup>2</sup> with varying number of shots and recent SOTA OVSS methods on four datasets. In almost all cases, TFM<sup>2</sup> improves the performance of multiple OVSS methods without additional training.

Method	Pre-Trained Dataset	Ensemble	ADE-847	PC-459	ADE-150	PC-59
SimSeg (ECCV 2022) [59]	COCO-Stuff	Yes	6.8	8.8	20.2	47.3
OVSeg (CVPR 2023) [32]	COCO-Stuff	Yes	9.0	12.4	29.7	55.3
FC-CLIP (NeurIPS 2023) [65]	COCO-Panoptic	Yes	14.8	18.2	34.1	58.4
ALIGN (ICML 2021) [27]	-	No	4.8	5.8	12.9	22.4
GroupViT (CVPR 2022) [55]	GCC [47] + YFCC [48]	No	4.3	4.9	10.6	25.9
Kunyang <i>et al.</i> (ICCV 2023) [23]	COCO-Panoptic	No	3.5	7.1	18.8	45.2
OpenSeg (ECCV 2022) [20]	COCO-Panoptic + COCO-Caption	No	6.8	11.2	24.8	45.9
MaskCLIP (ICML 2023) [73]	COCO-Panoptic	No	8.2	10.0	23.7	45.9
SAN (CVPR 2023) [57] (ViT-B)	COCO-Stuff	No	10.2	16.7	27.6	54.1
SAN (CVPR 2023) [57] (ViT-L)	COCO-Stuff	No	12.6	19.9	32.0	56.3
ODISE (CVPR 2023) [56]	COCO-Panoptic	No	11.1	14.5	29.9	57.3
DeOp (ICCV 2023) [22]	COCO-Panoptic	No	7.1	9.4	22.9	48.8
MasQCLIP (ICCV 2023) [60]	COCO-Panoptic	No	10.7	18.2	30.4	57.8
SimSeg (ResNet101)	COCO-Stuff	Yes	6.8	8.8	20.2	47.3
SimSeg (ResNet101) + TFM <sup>2</sup>	COCO-Stuff	Yes	7.0(+0.2)	9.9(+1.1)	22.4(+2.2)	50.4(+3.1)
OVSeg (Swin-B)	COCO-Stuff	Yes	9.0	12.4	29.7	55.3
OVSeg (Swin-B) + TFM <sup>2</sup>	COCO-Stuff	Yes	9.5(+0.5)	12.6(+0.2)	31.0(+1.3)	58.1(+2.8)
SAN (ViT-B)	COCO-Stuff	No	10.2	16.7	27.6	54.1
SAN (ViT-B) + TFM <sup>2</sup>	COCO-Stuff	No	13.8(+3.6)	17.2(+0.5)	32.1(+4.5)	57.0(+2.9)
SAN (ViT-L)	COCO-Stuff	No	12.6	19.9	32.0	56.3
SAN (ViT-L) + TFM <sup>2</sup>	COCO-Stuff	No	16.2(+3.6)	22.0(+2.1)	37.2(+5.2)	60.7(+4.4)

Table 1. The mIoU comparison results of applying 32-shot TFM<sup>2</sup> on multiple OVSS models with current mainstream OVSS methods.

an averaged result is computed and re-scaled to align with the range of the original logits. Analogous to the Eq. (4), the TFM<sup>2</sup> refines semantic segmentation result as:

$$\hat{\mathbf{S}}_{\text{final}} = \hat{\mathbf{L}}_{\text{final}}^{\top} \times \hat{\mathbf{M}}_{\mathbf{I}} \quad (18)$$

In summary, we propose TFM<sup>2</sup>, which starts from basic Mask Cache and is refined by three modules. TFM<sup>2</sup> provides the reference mask classification logits by its refined Cache. The original model mask classification logits  $\hat{\mathbf{L}}_{\text{origin}}$  will be fused with TFM<sup>2</sup> mask classification logits  $\hat{\mathbf{L}}_{\text{cache}}$  by Eq. (17). The final semantic segmentation will benefit from the enhanced mask classification result by Eq. (18).

## 4. Evaluation

### 4.1. Dataset and Experimental Settings

We evaluate TFM<sup>2</sup> on four datasets frequently employed in open-vocabulary semantic segmentation research [57–

59]. We use the ADE20k dataset [74], a popular choice for scene classification tasks, in two variants: one with 150 classes (ADE20k-150) and the other with 847 classes (ADE20k-847). Additionally, we utilize the Pascal Context dataset, specifically its PC-59 and PC-459 versions [40], which expand the Pascal VOC 2010 dataset by adding an extra 59 and 459 classes, respectively. Please note that we choose not to include Pascal VOC [17] in our evaluation due to its high label context similarity [57, 58] with the COCO-Stuff dataset, making it not ideal for assessing the effectiveness of open-vocabulary semantic segmentation models.

The baseline OVSS models are trained on either COCO Stuff [7] or COCO Panoptic [33] datasets, following standard practice [13, 32, 57–59]. We set the  $N = 847, 459, 150,$  and  $59,$  which are equal to the number of mask classes for four datasets. For the few-shot OVSS scenarios, we provide TFM<sup>2</sup> with  $K = 2, 4, 8, 16,$  and  $32$  shot masks derived from the training splits of the four datasets. We set the  $C' = \frac{1}{2}C.$

TFM<sup>2</sup> will be applied to a range of OVSS methods with various backbones. We employ the mean of class-wise intersection over union (**mIoU**) as our performance metric of choice to evaluate the semantic segmentation accuracy.

## 4.2. Implementation Details

We select SAN [57] with two Vision-Transformer backbones ViT-B and ViT-L [16]. We also select SimSeg [59] with a ResNet101 [24] backbone, and OVSeg [32] with a Swin-Transformer [37] backbone Swin-B. We employ the officially released pre-trained models for each baseline. We carry out all experiments using the TFM<sup>2</sup>, implemented in PyTorch [42] on a single NVIDIA V100 GPU, with  $\lambda_1 = 0.7$ ,  $\lambda_2 = 0.1$ , and  $\lambda_3 = 0.5$ . For the ADE20k-847 and PC-459 datasets, we retain all reference masks (without applying the Dynamic Filter module) if the shot number exceeds the mask number. Because some classes only have one mask, or their mask numbers are less than the shot number. The codes will be released upon the paper’s decision.

## 4.3. Performance Analysis

Fig. 3 illustrates the performance of various OVSS methods with TFM<sup>2</sup> across four datasets. Our TFM<sup>2</sup> consistently improves mIoU with few-shot masks across backbones. Performance on ADE20K-847 and PC-459 is impacted by ensembling complexity and dataset characteristics. SimSeg and OVSeg achieve their best results by fusing logits from trained models with those from frozen Vision-and-Language Pre-trained (VLP) models like CLIP. Adding TFM<sup>2</sup> introduces additional complexity, and for simplicity, we used an averaged fusion. More advanced strategies might improve results, but our focus was to demonstrate the viability of TFM<sup>2</sup>. ADE20K-847 and PC-459 have more classes but limited images per class, affecting reference mask availability. This limited reference mask availability, especially in larger-shot settings, affected its performance. Despite using all available reference masks, some classes still lacked sufficient data. Lastly, TFM<sup>2</sup> performs better on SAN than on SimSeg and OVSeg, as the latter already use ensembling with fine-tuned and frozen CLIP predictions. Tuning fusion weights for these varied logits across datasets is complex, so we kept the fusion simple by applying TFM<sup>2</sup> only to model logits.

We also compare 32-shot TFM<sup>2</sup> with multiple OVSS methods in Tab. 1, with additional shot number results in the supplementary material. TFM<sup>2</sup> improves performance as more data per class becomes available, as demonstrated with 16-shot and 32-shot settings. Even in more limited settings, such as 2-shot and 4-shot, TFM<sup>2</sup> consistently outperforms other methods across multiple backbones and datasets. Notably, SAN with TFM<sup>2</sup> achieves significant gains in a training-free fashion.

TFM<sup>2</sup> could be improved with fine-tuning. Fine-tuning

Module Name				Shot Number				
Mask Cache	Dynamic Filter	Channel Reduction	Feature Alignment	2	4	8	16	32
✓				27.7	28.3	29.4	29.7	29.9
✓	✓			27.7	28.3	29.4	30.4	30.6
✓	✓	✓		28.5	29.9	30.5	31.4	31.5
✓	✓	✓	✓	<b>29.0</b>	<b>30.6</b>	<b>31.0</b>	<b>31.9</b>	<b>32.1</b>

Table 2. The quantitative ablation table for the four versions: “Mask Cache”, “Mask Cache + Dynamic Filter”, “Mask Cache + Dynamic Filter”, and TFM<sup>2</sup> with SAN(ViT-B) on ADE-150.

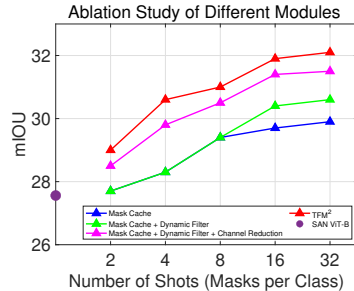


Figure 4. Ablation figure of different modules with SAN(ViT-B) on ADE-150 dataset. We vary the number of  $K$ -shots and visualize the trends of different combinations.

the last layer improved mIoU by 0.7 on ADE20k-150 (SAN with ViT-L). Using few-shot samples for fine-tuning contradicts the training-free principle, so fine-tuned results are excluded. Consequently, we have chosen not to include fine-tuned experimental results. All presented results adhere to a training-free methodology. Incorporating TFM<sup>2</sup> does not impact inference speed, as it only involves lightweight operations. Remarkably, the speeds remain unchanged from the original methods, as TFM<sup>2</sup> performs only two lightweight operations during inference: (1) calculating similarity with the cache keys and (2) multiplying scores by the cache values, both of which are linear time operations. Mask Cache construction is a one-time setup, offering reusable benefits with minimal overhead. We view this as an upfront investment, given the reusability and benefits it offers.

## 4.4. Ablation Study

We conduct an ablation study on TFM<sup>2</sup> to analyze the impacts of the designed key-value mask cache and three modules. We select SAN with a ViT-B backbone as the OVSS baseline and consider four versions of TFM<sup>2</sup>. The first version, termed “Mask Cache”, integrates the mask cache logits and the original model logits as described in Sec. 3.1. The second version, “Mask Cache + Dynamic Filter”, includes the Mask Cache with an optimized key using Dynamic Filter, as detailed in Sec. 3.2. The third version, “Mask Cache + Dynamic Filter + Channel Reduction”, comprises the Mask Cache with Dynamic Filter and Channel Reduction keys, as in Sec. 3.3. The final version using all modules is TFM<sup>2</sup>, as defined in Sec. 3.4. All versions utilize Eq. (17) to fuse mask proposal classification logits

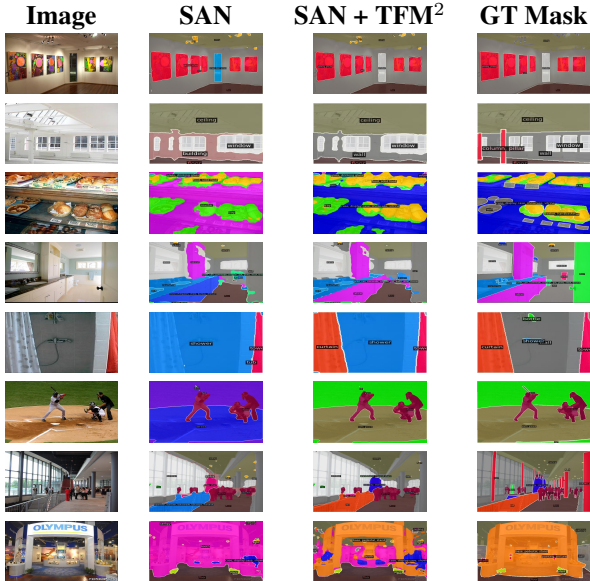


Figure 5. Qualitative examples showing TFM<sup>2</sup>'s role in improving mask proposal classification on ADE20k-150. The second column shows SAN inference without TFM<sup>2</sup>. We see that SAN + TFM<sup>2</sup> (third column) can improve semantic segmentation when compared with the ground truth (fourth column). Please note that the color palette is the same for all mask classes.

with original logits and subsequently generate the final semantic segmentation results, as outlined in Eq. (4).

As depicted in Tab. 2 and Fig. 4, the Mask Cache can enhance the performance of the OVSS method for  $K = 2$  to 32 shots. In the settings of larger shots, the performance can be further improved by Dynamic Filter. After applying Channel Reduction, the third version consistently outperforms the second one. Ultimately, with the aid of designed Feature Alignment, TFM<sup>2</sup> achieves the best results.

#### 4.5. Qualitative Results

In Fig. 5, we show several mask predictions from the SAN and SAN with TFM<sup>2</sup> on the ADE20K-150 dataset. These figures suggest that TFM<sup>2</sup> may assist SAN in correctly classifying some mask proposals, which further enhances the semantic segmentation performance. Each row contains the original image, SAN output, SAN with TFM<sup>2</sup>, and the ground truth. If we look at the fourth row, incorporating TFM<sup>2</sup> enables the segmentation of the shower and the shower curtain, a task that SAN alone could not do. Similarly, in the last row, we see a real-world scene of an expo booth that SAN has predominantly segmented as a single object. Including TFM<sup>2</sup> during inference time allows for segmenting fine-grained, smaller objects, potentially making it more applicable to real-world scenarios.

## 5. Discussion

The primary focus of our research is on the mask proposal classification aspect. This is a critical component in semantic segmentation, as it involves identifying and classifying regions within an image that correspond to different objects. For the mask creation, a potential solution that does not require additional training is using Segment Anything [28]. This tool can be employed to refine the masks, a process that can significantly enhance the segmentation results. By refining the masks, we can achieve more accurate and precise segmentation. We conducted additional experiments to analyze the robustness of TFM<sup>2</sup> and found that its performance is highly sensitive to the quality of the few-shot masks when the number of shots is small ( $k = 2$ ). If the provided masks are not representative, they can negatively impact the constructed cache. However, as  $k$  increases, the performance stabilizes even with random sampling.

The performance of TFM<sup>2</sup> could potentially be improved by introducing trainable versions. The current version of TFM<sup>2</sup> is effective but operates on a fixed set of parameters from the trained OVSS models. By making these parameters trainable, we can allow the method to adapt to the specific characteristics of the data. This adaptability could improve performance, as the method would be better equipped to handle the unique challenges presented by different datasets. However, introducing trainable parameters introduces additional complexity. A different approach may be necessary for datasets with a highly skewed distribution, such as PC-459. These datasets present unique challenges, as the uneven data distribution can make it difficult for traditional methods to perform effectively.

## 6. Conclusion

In this paper, we proposed TFM<sup>2</sup> for enhancing the OVSS models with few-shot masks. To achieve that, we first propose a way to build up the mask cache, which stores the representative mask visual features as the key of the mask cache. Based on that, we also employ three modules: Dynamic Filter, Channel Reduction, and Feature Alignment to further refine the key and value of cache as the final adapter. Comprehensive experiments and results show that TFM<sup>2</sup> can surpass the performances of the original OVSS with only few-shot masks. Besides that, TFM<sup>2</sup> is not limited to any specific methods or backbones, which demonstrates the general ability to be applied to different OVSS methods. It is close to the real-world application scenario that the trained OVSS model is required to segment new classes with limited reference samples. We hope this study builds up new baselines for the few-shot open-vocabulary semantic segmentation task and it can inspire future research on improving the OVSS methods with few-shot data in an efficient way.



## References

- [1] Awesome-few-shot-semantic-segmentation. <https://github.com/WingkeungM/Awesome-Few-Shot-Semantic-Segmentation>. 2
- [2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *NeurIPS*, 35:23716–23736, 2022. 1
- [3] Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. Effective conditioned and composed image retrieval combining clip-based features. In *CVPR*, pages 21466–21474, 2022. 1
- [4] Luca Barsellotti, Roberto Amoroso, Lorenzo Baraldi, and Rita Cucchiara. Fossil: Free open-vocabulary semantic segmentation through synthetic references retrieval. In *WACV*, pages 1464–1473, 2024. 2
- [5] Luca Barsellotti, Roberto Amoroso, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Training-free open-vocabulary segmentation with offline diffusion-augmented prototype generation. In *CVPR*, pages 3689–3698, 2024. 2
- [6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 33:1877–1901, 2020. 3
- [7] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, pages 1209–1218, 2018. 6
- [8] Davide Caffagni, Federico Cocchi, Luca Barsellotti, Nicholas Moratelli, Sara Sarto, Lorenzo Baraldi, Marcella Cornia, and Rita Cucchiara. The (r) evolution of multi-modal large language models: A survey. *arXiv preprint arXiv:2402.12451*, 2024. 1
- [9] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020. 2
- [10] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021. 3
- [11] Junbum Cha, Jonghwan Mun, and Byungseok Roh. Learning to generate text-grounded mask for open-world semantic segmentation from only image-text pairs. In *CVPR*, pages 11165–11174, 2023. 2
- [12] 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. *PAMI*, 40(4):834–848, 2017. 2
- [13] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, pages 1290–1299, 2022. 2, 6
- [14] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Pixel classification is not all you need for semantic segmentation. *NeurIPS*, 34:17864–17875, 2021. 2
- [15] Seokju Cho, Heeseong Shin, Sunghwan Hong, Anurag Arnab, Paul Hongsuck Seo, and Seungryong Kim. Cat-seg: Cost aggregation for open-vocabulary semantic segmentation. In *CVPR*, pages 4113–4123, June 2024. 2
- [16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 7
- [17] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. *Pattern Anal. Stat. Model. Comput. Learn., Tech. Rep.*, 2007(1-45):5, 2012. 6
- [18] Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. Clip2video: Mastering video-text retrieval via image clip. *arXiv preprint arXiv:2106.11097*, 2021. 1
- [19] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *IJCV*, pages 1–15, 2023. 2
- [20] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *ECCV*, pages 540–557. Springer, 2022. 6
- [21] Sina Hajimiri, Malik Boudiaf, Ismail Ben Ayed, and Jose Dolz. A strong baseline for generalized few-shot semantic segmentation. In *CVPR*, pages 11269–11278, 2023. 2
- [22] Cong Han, Yujie Zhong, Dengjie Li, Kai Han, and Lin Ma. Open-vocabulary semantic segmentation with decoupled one-pass network. In *ICCV*, pages 1086–1096, October 2023. 6
- [23] Kunyang Han, Yong Liu, Jun Hao Liew, Henghui Ding, Jiajun Liu, Yitong Wang, Yansong Tang, Yujie Yang, Jiashi Feng, Yao Zhao, and Yunchao Wei. Global knowledge calibration for fast open-vocabulary segmentation. In *ICCV*, pages 797–807, October 2023. 6
- [24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 7
- [25] Weizhao He, Yang Zhang, Wei Zhuo, Linlin Shen, Jiaqi Yang, Songhe Deng, and Liang Sun. Apseg: Auto-prompt network for cross-domain few-shot semantic segmentation. In *CVPR*, pages 23762–23772, June 2024. 2
- [26] Kai Huang, Feige Wang, Ye Xi, and Yutao Gao. Prototypical kernel learning and open-set foreground perception for generalized few-shot semantic segmentation. In *ICCV*, pages 19256–19265, October 2023. 2
- [27] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, pages 4904–4916. PMLR, 2021. 6
- [28] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *ICCV*, pages 4015–4026, October 2023. 8

- [29] Weicheng Kuo, Anelia Angelova, Jitendra Malik, and Tsung-Yi Lin. Shapemask: Learning to segment novel objects by refining shape priors. In *CVPR*, pages 9207–9216, 2019. [2](#)
- [30] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. [1](#)
- [31] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, pages 12888–12900. PMLR, 2022. [1](#)
- [32] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *CVPR*, pages 7061–7070, 2023. [2](#), [3](#), [6](#), [7](#)
- [33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. [6](#)
- [34] Jie Liu, Yanqi Bao, Guo-Sen Xie, Huan Xiong, Jan-Jakob Sonke, and Efstratios Gavves. Dynamic prototype convolution network for few-shot semantic segmentation. In *CVPR*, pages 11553–11562, June 2022. [2](#)
- [35] Sun-Ao Liu, Yiheng Zhang, Zhaofan Qiu, Hongtao Xie, Yongdong Zhang, and Ting Yao. Learning orthogonal prototypes for generalized few-shot semantic segmentation. In *CVPR*, pages 11319–11328, 2023. [2](#)
- [36] Xinyu Liu, Beiwen Tian, Zhen Wang, Rui Wang, Kehua Sheng, Bo Zhang, Hao Zhao, and Guyue Zhou. Delving into shape-aware zero-shot semantic segmentation. In *CVPR*, pages 2999–3009, 2023. [2](#)
- [37] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. [7](#)
- [38] Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li. Segclip: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. In *ICML*, pages 23033–23044. PMLR, 2023. [2](#)
- [39] Jiayun Luo, Siddhesh Khandelwal, Leonid Sigal, and Boyang Li. Emergent open-vocabulary semantic segmentation from off-the-shelf vision-language models. In *CVPR*, pages 4029–4040, June 2024. [2](#)
- [40] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, pages 891–898, 2014. [6](#)
- [41] Zhenliang Ni, Xinghao Chen, Yingjie Zhai, Yehui Tang, and Yunhe Wang. Context-guided spatial feature reconstruction for efficient semantic segmentation. *arXiv preprint arXiv:2405.06228*, 2024. [2](#)
- [42] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019. [7](#)
- [43] Siyuan Qiao, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation. In *CVPR*, pages 3997–4008, 2021. [2](#)
- [44] 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*, pages 8748–8763. PMLR, 2021. [1](#), [3](#)
- [45] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, pages 8821–8831. PMLR, 2021. [3](#)
- [46] Xiangheng Shan, Dongyue Wu, Guilin Zhu, Yuanjie Shao, Nong Sang, and Changxin Gao. Open-vocabulary semantic segmentation with image embedding balancing. In *CVPR*, pages 28412–28421, June 2024. [2](#)
- [47] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018. [6](#)
- [48] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. [6](#)
- [49] Vishaal Udandarao, Ankush Gupta, and Samuel Albanie. Sus-x: Training-free name-only transfer of vision-language models. In *ICCV*, pages 2725–2736, 2023. [1](#), [3](#)
- [50] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *ICML*, pages 23318–23340. PMLR, 2022. [1](#)
- [51] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. Image as a foreign language: BEiT pretraining for vision and vision-language tasks. In *CVPR*, 2023. [1](#)
- [52] 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. *arXiv preprint arXiv:2310.01403*, 2023. [2](#)
- [53] Bin Xie, Jiale Cao, Jin Xie, Fahad Shahbaz Khan, and Yanwei Pang. Sed: A simple encoder-decoder for open-vocabulary semantic segmentation. In *CVPR*, pages 3426–3436, June 2024. [2](#)
- [54] Bin Xie, Jiale Cao, Jin Xie, Fahad Shahbaz Khan, and Yanwei Pang. Sed: A simple encoder-decoder for open-vocabulary semantic segmentation. In *CVPR*, 2024. [2](#)
- [55] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *CVPR*, pages 18134–18144, 2022. [6](#)

- [56] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *CVPR*, pages 2955–2966, 2023. [6](#)
- [57] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. San: Side adapter network for open-vocabulary semantic segmentation. *PAMI*, 2023. [1](#), [2](#), [3](#), [6](#), [7](#)
- [58] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *CVPR*, pages 2945–2954, 2023. [2](#), [6](#)
- [59] 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*, pages 736–753. Springer, 2022. [1](#), [2](#), [3](#), [6](#), [7](#)
- [60] Xin Xu, Tianyi Xiong, Zheng Ding, and Zhuowen Tu. Masq-clip for open-vocabulary universal image segmentation. In *ICCV*, pages 887–898, October 2023. [6](#)
- [61] Yong Yang, Qiong Chen, Yuan Feng, and Tianlin Huang. Mianet: Aggregating unbiased instance and general information for few-shot semantic segmentation. In *CVPR*, pages 7131–7140, 2023. [2](#)
- [62] Minghao Yin, Zhuliang Yao, Yue Cao, Xiu Li, Zheng Zhang, Stephen Lin, and Han Hu. Disentangled non-local neural networks. In *ECCV*, pages 191–207. Springer, 2020. [2](#)
- [63] Xin You, Junjun He, Jie Yang, and Yun Gu. Learning with explicit shape priors for medical image segmentation. *arXiv preprint arXiv:2303.17967*, 2023. [2](#)
- [64] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. [1](#)
- [65] 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. [1](#), [6](#)
- [66] Gengwei Zhang, Shant Navasardyan, Ling Chen, Yao Zhao, Yunchao Wei, Humphrey Shi, et al. Mask matching transformer for few-shot segmentation. *NeurIPS*, 35:823–836, 2022. [2](#)
- [67] Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Yu Qiao, Peng Gao, and Hongsheng Li. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In *CVPR*, pages 15211–15222, 2023. [1](#), [3](#)
- [68] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *ECCV*, pages 493–510. Springer, 2022. [1](#), [3](#)
- [69] Tianyi Zhang, Kishore Kasichainula, Yaoxin Zhuo, Baoxin Li, Jae-Sun Seo, and Yu Cao. Patch-based selection and refinement for early object detection. In *WACV*, pages 729–738, 2024. [1](#)
- [70] Tianyi Zhang, Kishore Kasichainula, Yaoxin Zhuo, Baoxin Li, Jae-Sun Seo, and Yu Cao. Transformer-based selective super-resolution for efficient image refinement. In *AAAI*, volume 38, pages 7305–7313, 2024. [1](#)
- [71] Yi Zhang, Meng-Hao Guo, Miao Wang, and Shi-Min Hu. Exploring regional clues in clip for zero-shot semantic segmentation. In *CVPR*, pages 3270–3280, June 2024. [2](#)
- [72] Hang Zhao, Xavier Puig, Bolei Zhou, Sanja Fidler, and Antonio Torralba. Open vocabulary scene parsing. In *ICCV*, 2017. [2](#)
- [73] Zhuowen Tu Zheng Ding, Jieke Wang. Open-vocabulary universal image segmentation with maskclip. In *ICML*, 2023. [2](#), [6](#)
- [74] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, pages 633–641, 2017. [6](#)
- [75] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, pages 16816–16825, 2022. [2](#)
- [76] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 130(9):2337–2348, 2022. [2](#)
- [77] Ziqin Zhou, Hai-Ming Xu, Yangyang Shu, and Lingqiao Liu. Unlocking the potential of pre-trained vision transformers for few-shot semantic segmentation through relationship descriptors. In *CVPR*, pages 3817–3827, June 2024. [2](#)
- [78] Xiangyang Zhu, Renrui Zhang, Bowei He, Aojun Zhou, Dong Wang, Bin Zhao, and Peng Gao. Not all features matter: Enhancing few-shot clip with adaptive prior refinement. In *ICCV*, pages 2605–2615, October 2023. [1](#), [3](#)
- [79] Yaoxin Zhuo and Baoxin Li. Felga: Unsupervised fragment embedding for fine-grained cross-modal association. In *WACV*, pages 5635–5645, 2024. [1](#)
- [80] Yaoxin Zhuo, Yikang Li, Jenhao Hsiao, Chiuman Ho, and Baoxin Li. Clip4hashing: unsupervised deep hashing for cross-modal video-text retrieval. In *ICMR*, pages 158–166, 2022. [1](#)