Explore the Potential of CLIP for Training-Free Open Vocabulary Semantic Segmentation

Tong Shao, Zhuotao Tian✉, Hang Zhao, and Jingyong Su✉

Harbin Institute of Technology, Shenzhen, China
22S151082@stu.hit.edu.cn, tianzhuotao@hit.edu.cn,
200110431@stu.hit.edu.cn, sujingyong@hit.edu.cn

Abstract. CLIP, as a vision-language model, has significantly advanced Open-Vocabulary Semantic Segmentation (OVSS) with its zero-shot capabilities. Despite its success, its application to OVSS faces challenges due to its initial image-level alignment training, which affects its performance in tasks requiring detailed local context. Our study delves into the impact of CLIP's [CLS] token on patch feature correlations, revealing a dominance of "global" patches that hinders local feature discrimination. To overcome this, we propose CLIPtrase, a novel training-free semantic segmentation strategy that enhances local feature awareness through recalibrated self-correlation among patches. This approach demonstrates notable improvements in segmentation accuracy and the ability to maintain semantic coherence across objects. Experiments show that we are 22.3% ahead of CLIP on average on 9 segmentation benchmarks, outperforming existing state-of-the-art training-free methods. The code are made publicly available at https://github.com/leaves162/CLIPtrase

Keywords: CLIP · Training-free · Semantic Segmentation

1 Introduction

CLIP [29], as a vision-language foundation model, has gained significant popularity in recent years [15,19,23,31,45,46]. Its remarkable zero-shot generalization capability has played a crucial role in advancing Open-Vocabulary Semantic Segmentation (OVSS) [17,20,26,33,42,44]. It is often employed as an encoder, and, in order to uphold the zero-shot generalization capability for OVSS, researchers have focused on developing intricate decoder designs to accommodate the pixel-level perception [7,27,28,36,48].

However, incorporating complex decoders to process the features extracted from the fixed CLIP model overlooks the fact that CLIP was initially trained by image-level alignment. Consequently, the globally aligned image-text features may not be well-suited for semantic segmentation that primarily relies on dense features with strong local context discrimination capabilities.

✉: Corresponding Authors

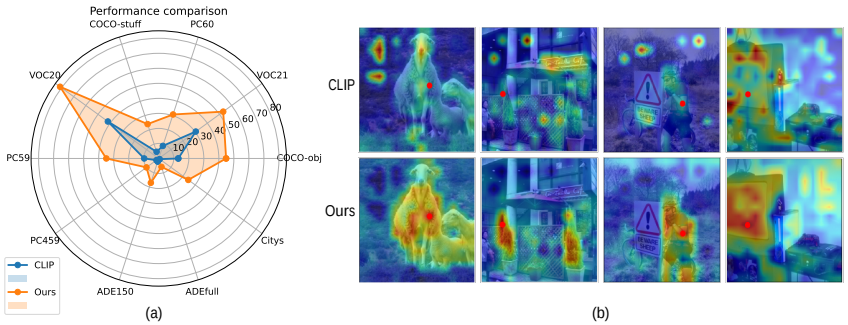

Fig. 1: Comparison between our method and CLIP. (a): Performance comparison on open-vocabulary semantic segmentation of our model and CLIP. (b): Comparison of randomly selected patch attention response heatmaps between our model and CLIP. The red dot in the picture is the selected patch position.

To delve deeper into the above issue, we conduct an analysis regarding the correlations within the CLIP patch features, as shown in Figure 2. We can observe that the [CLS] token in CLIP, which is utilized for image-text alignment, may disrupt the patch correlations. Specifically, in the deeper layers, the [CLS] token does not directly attend to the patches where the target objects in the image are located. Instead, several patches in the image gradually assume a global field of view, as displayed by the bright bars in the inter-patch attention maps of layer 9 and layer 11. These patches exhibit high attention towards all other patches, effectively aggregating crucial information from the entire image. Consequently, the [CLS] token primarily focuses on these "global" patches, while the attention weights assigned to other patches are nearly negligible.

The few "global" patches potentially facilitate the alignment between the entire image and text query by reducing the elements that the [CLS] token needs to concentrate on. However, it may negatively affect the local correlation. This is because the "global" patches have to gather information from all other regions to provide global cues for the [CLS] token, and disturb the correlation between local patches and the surrounding context consequently. As illustrated in the lower section of Figure 2, it can result in a decline in performance for semantic segmentation tasks. This observation raises a crucial question: *Can we enhance the utilization of CLIP features by reconstructing inter-patch correlations?*

To address this, we introduce a novel **TRA**ining-free semantic **SE**gmentation approach, called **CLIPtrase**, that involves calculating self-correlation to enhance the attention of features towards their neighboring patches, consequently improving local awareness. As illustrated in Figure 1(b), our enhanced CLIP enables clear discrimination between objects and backgrounds, while also revealing notable correlations among multiple objects of the same class. Subsequently, we employ a clustering and denoising process where all patches within a region compute similarity with text features and collaboratively determine the candidate classes for their respective region. As depicted in Figure 1, our method exhibits

significant improvements in both segmentation task performance and the visualization of patch semantic correlations, surpassing the initial CLIP model.

To summarize, our contributions are as follows:

- We conduct a new analysis of the limitations of CLIP in terms of semantic segmentation tasks and identify the issue brought by the "global patches".
- To alleviate the issue, we propose a simple yet effective training-free strategy to enhance the vanilla CLIP model, termed CLIPtrase.
- The enhanced CLIP can be directly applied to semantic segmentation tasks, and be combined with SAM [16] to complement precise classification boundaries of SAM with semantic generalization of CLIP, achieving state-of-the-art performance in training-free OVSS.

2 Preliminary

In this section, we will provide an introduction to the background of CLIP and the closely related OVSS models. More details regarding the related work are presented in the supplementary file.

CLIP. Contrastive Language-Image Pre-training (CLIP [29]) is a multi-modal foundation model. It achieves impressive zero-shot generalization by engaging in "image-text" contrastive learning using a vast dataset of images. CLIP adopts a [CLS] token within the vision transformer, similar to the text classification setting, to represent the comprehensive features of the image. This enables the alignment between image and text, with the text that bears the closest resemblance to the features of the [CLS] token being considered as the predicted class of the image:

$$\boldsymbol{p} = \text{argmax}(\text{Sim}(\boldsymbol{t}, [CLS])), \tag{1}$$

where Sim calculates the similarity between visual [CLS] token and text embedding $\boldsymbol{t}$, to obtain the prediction $\boldsymbol{p}$.

CLIP-based Open Vocabulary Semantic Segmentation. Since CLIP is primarily trained for image-text alignment, it faces challenges in directly generating precise pixel-level predictions like segmentation [18, 32, 34, 35]. Despite some proposed approaches [5, 7, 12, 30, 38] that attempt to extract detailed cues from CLIP itself for segmentation, achieving satisfactory performance remains difficult. Consequently, some approaches in the literature alternatively leverage CLIP as an encoder, extracting semantic features that auxiliary trainable modules can utilize to produce segmentation results. They can be categorized into two directions as follows:

- Decoder-based: The essence of this idea is to design sophisticated decoders to refine CLIP features to adapt to semantic segmentation while retaining the generalization [2, 3, 8, 13, 24, 40, 47]. In SAN [41], Xu et al. propose a side network that generates masks concurrently with CLIP. These masks are subsequently employed as attention biases to capture the relevant features of each mask's corresponding [CLS] token, facilitating accurate mask classification.

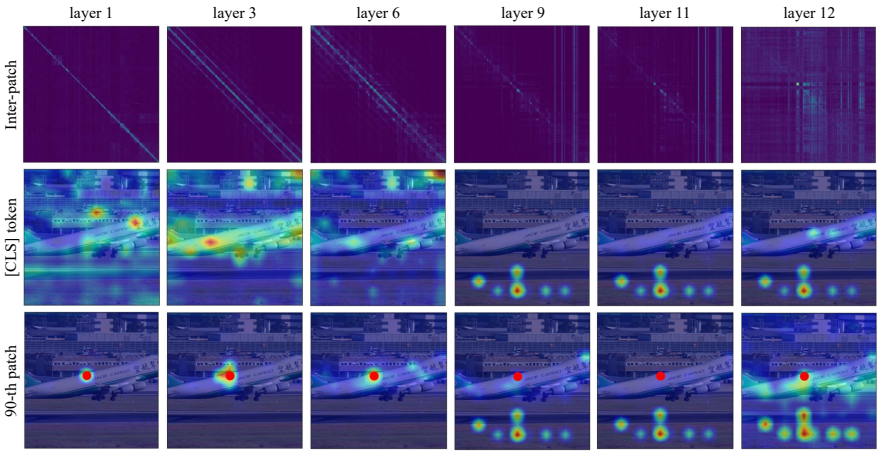

Fig. 2: Visualization of the "global" patch phenomenon in the attention map of different layers of ViT [9] in the CLIP visual branch. Inter-patch attention map is the attention weight map between all patch features, the size is 196*196. [CLS] token attention map is the attention weight matrix of [CLS] token on all patch features, interpolates from 14*14 to 224*224, and displays in the form of a heat map. 90-th patch attention map is the attention weight of a randomly selected patch, the red dot in the image is the selected patch position. Its display method is the same as the [CLS] token. More visualization are presented in supplementary file.

– Fine-tuning-based: Another direction considers directly fine-tuning CLIP for segmentation [14, 20, 22, 26]. For example, MAFT [14] uses the idea of attention bias to fine-tune the CLIP to adapt the mask classification. Additionally, OV-Seg [22] combines these two lines, it observes a domain shift problem when directly using CLIP to classify images masked by the proposals yielded by MaskFormer [4]. To tackle this issue, OV-Seg uses additional masked images to fine-tune the CLIP and make it as an encoder, so as to excel and effectively handle the specific context involving masked images.

3 Key Observations

Despite the decent results demonstrated by the two approaches in Section 2, they still possess their respective issues. For the decoder-based method, the CLIP's features are obtained via a frozen CLIP model without specific adjustments to adapt them for semantic segmentation. Instead, this issue is addressed through the addition of an external decoder. On the other hand, although fine-tuning allows for a certain degree of adaptation of CLIP's image-text aligned features, it carries the risk of overfitting to specific scenarios, e.g., the domain of the masked images used for fine-tuning, leading to a decline in performance. Therefore, we need to consider whether it is possible to optimize CLIP's features without fine-tuning, in order to unearth more information that can aid in semantic segmenta-

tion. To answer this question, we start by investigating the correlations between the [CLS] token and the patch tokens.

The "global patch". The [CLS] token, used for achieving image-text alignment alongside the text embedding, provides a holistic representation of the entire image. It captures the essence or the most prominent information regarding objects within the image. However, since the patch tokens do not directly interact with the text embedding, their dynamics are not apparent. To delve into the intricate behaviors of the [CLS] token and the patch tokens, we examine the attention maps across different layers, as depicted in Figure 2.

Contrary to our assumptions, the [CLS] token does not directly interact with patches where target objects are located. Instead, in deeper layers, certain patches emerge as proxies for a global field of view, herein termed "global" patches. They attract high attention weights across the board, visible as bright stripes in the inter-patch attention maps of layer 9 and layer 11. The attention map for the [CLS] token shows it mainly connects with these "global" patches, while attention to other areas is minimal. The causes of this phenomenon may be:

- The high number of patches complicates the learning process for the [CLS] token. Similar to how models tend to pick up on fewer, distinct and sparse features, we think the "global" patch acts similarly, by significantly reducing the number of patches the [CLS] token needs to focus on.
- Much of the visual information are unnecessary for [CLS] token to accomplish the feature alignment. Visual details usually have more to offer than the text category it needs to match, especially when considering backgrounds or parts related to other co-occurring objects. Therefore, the "global" patches capture the essential information from the image by summarizing the visual content, as proxies, to provide the necessary visual essence to the [CLS] token.

Now there are studies [6,39] that hold similar views to ours, which can support the credibility of our statement to a certain extent.

The effects brought to the local patches. For different local patches, as illustrated in Figure 2, it is evident that the attention weights of different local patches are predominantly influenced by the global patches. This observation reveals a lack of correlation between patches sharing the same semantics. As previously mentioned, the global patches exhibit higher attention weights towards all patches. Consequently, due to the softmax operation, the attention weights of individual local patches are primarily determined by their correlation with the global patches. This leads to a considerable suppression of the original semantic relevance between patches, rendering it nearly negligible.

Our thoughts. The learning process of the [CLS] token naturally gives rise to the prominence of the global patches. While the global patches contribute to the learning of the [CLS] token, they also significantly undermine the semantic correlation between patches. This loss of semantic correlation is particularly detrimental for dense feature tasks such as semantic segmentation. Consequently, this may be one of the primary reasons why CLIP alone is not inherently suitable for directly handling dense feature tasks.

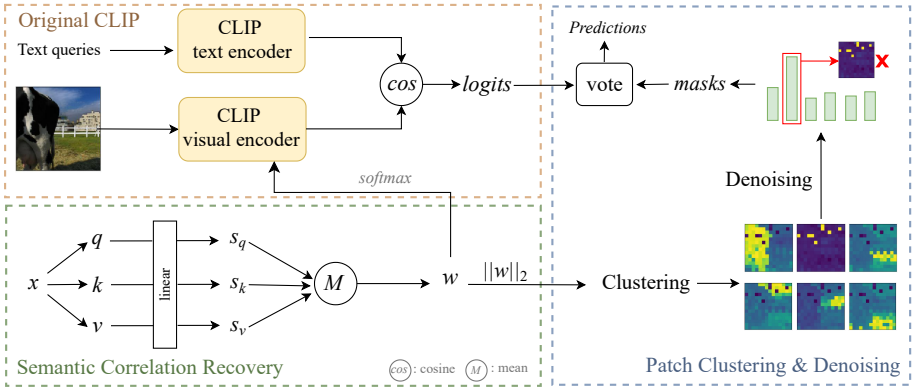

Fig. 3: Illustration of the key components for our CLIPtrase framework. The semantic correlation operation restores the semantic location between patches. While the restored w continues to forward to obtain CLIP visual features, we use clustering to obtain prototype attention weights of different categories and generate masks to improve classification results and refine object boundaries. All modules in the model are frozen to accomplish training-free setting.

4 Method

Through the analysis, we have identified the mechanisms by which the semantic correlation among patches is diminished within CLIP. This inspires us to restore the semantic correlation to enable CLIP's seamless application to semantic segmentation tasks. Considering that the primary cause of reduced semantic correlation stems from the dilution of original semantic attention, we focus on reinforcing the semantic alignment among patches, especially those with similar meanings or those in close proximity.

To this end, we propose CLIPtrase, as a strategy to enable the direct adaptation of CLIP for semantic segmentation tasks without additional training. CLIPtrase is composed of three core components: Semantic Correlation Recovery, Patch Clustering, and Denoising. Details are as follows.

4.1 Semantic Correlation Recovery

To preserve CLIP's generalization capabilities to the greatest extent, our semantic correlation restoration focuses on the final layer of the visual encoder.

We hypothesize that the input image is propagated through the shallow layers of the Vision Transformer (ViT) to yield the feature representation $\boldsymbol{x} \in \mathbb{R}^{B\times(HW+1)\times D}$, where B represents the batch size, H and W the number of patches segmented from the image's height and width respectively, and extra one is [CLS] token. D is the dimension of the intermediate features within the CLIP visual encoder.

The multi-head attention mechanism within the original CLIP model operates as follows: the input $\boldsymbol{x}$ is first transformed by the linear layer denoted as $\sigma(\boldsymbol{x})$. This transformed output is then divided into three distinct streams: queries ($\boldsymbol{q}$), keys ($\boldsymbol{k}$), and values ($\boldsymbol{v}$). Each stream undergoes processing by multiple attention heads, facilitating the model's capacity to learn diverse aspects of the data. Finally, the model applies attention weights across these streams, culminating in the aggregated output post-attention. This process is shown as follows:

$$\begin{aligned} \boldsymbol{q}_i = \sigma_{i,q}(\boldsymbol{x}),\ \boldsymbol{k}_i = \sigma_{i,k}(\boldsymbol{x}),\ \boldsymbol{v}_i = \sigma_{i,v}(\boldsymbol{x}),\ i \in [0, H-1] \\ \boldsymbol{z}_i = \text{Softmax}(\frac{\boldsymbol{q}_i^T \boldsymbol{k}_i}{\sqrt{d_k}})\boldsymbol{v}_i \\ \boldsymbol{x} = \text{Concat}(\boldsymbol{z}_0, ..., \boldsymbol{z}_{H-1}) \end{aligned} \tag{2}$$

where H is the number of heads, d_k denotes the number of dimensions of $\boldsymbol{k}$, $\boldsymbol{z}$ is the intermediate output, and the operations of other layers are omitted here.

To enhance the attention of each patch towards itself and other patches within the same semantic region, we adopt self-correlation, denoted by $\gamma(\cdot)$. For this purpose, Cosine similarity is used to ascertain the semantic correlation across each branch of $\boldsymbol{q}$, $\boldsymbol{k}$, and $\boldsymbol{v}$ along the feature dimension. Subsequently, these correlations are averaged to procure a more consistent semantic correlation matrix $\boldsymbol{w} \in \mathbb{R}^{B\times(HW+1)\times(HW+1)}$:

$$\begin{aligned} \boldsymbol{q}_i = \sigma_{i,q}(\boldsymbol{x}),\ \boldsymbol{k}_i = \sigma_{i,k}(\boldsymbol{x}),\ \boldsymbol{v}_i = \sigma_{i,v}(\boldsymbol{x}),\ i \in [0, H-1] \\ \boldsymbol{w}_{i,q} = \gamma(\boldsymbol{q}_i, \boldsymbol{q}_i),\ \boldsymbol{w}_{i,k} = \gamma(\boldsymbol{k}_i, \boldsymbol{k}_i), \boldsymbol{w}_{i,v} = \gamma(\boldsymbol{v}_i, \boldsymbol{v}_i) \\ \boldsymbol{w} = \frac{1}{3H}\sum_{i=0}^{H-1}(\boldsymbol{w}_{i,q} + \boldsymbol{w}_{i,k} + \boldsymbol{w}_{i,v}) \end{aligned} \tag{3}$$

Besides, it is worth noting that the rationale behind adopting Cosine similarity for assessing correlations lies in that weights formed by Cosine similarity can forcibly constrain maximum attention to itself, thereby enhancing self-correlation. Comparing with the inner product, its bounded value range ensures that the resulting matrix is more amenable to subsequent clustering operation, providing a structured and constrained space for efficiently grouping semantically related patches, which shown in Section 5.3.

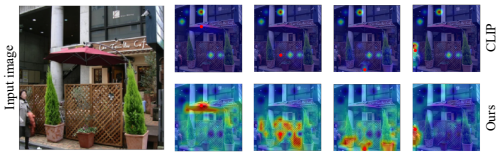

Fig. 4: CLIP attention map before and after semantic correlation recovery. The red dot indicates the selected patch position. Our method significantly restores the correlation between adjacent or semantically similar patches.

Compared with the original CLIP, the attention weights are no longer concentrate on the global patches. As shown in the Figure 4, we can restore the

high correlation between patches with similar semantics or in close proximity, thereby achieving the purpose of recovery.

4.2 Patch Clustering

Through the proposed recovery process, semantic correlation is enhanced, making patch features exhibit increased similarity with those sharing the same semantics. This enhancement may also improve the alignment between the visual patch and text features, as evidenced by the later experiments.

Consequently, we take a step further by directly clustering based on enhanced semantic correlation to derive masks for individual objects, resulting in a new training-free pipeline for open-vocabulary semantic segmentation. Within each clustered region, patches can calculate their similarity with text features and collectively identify the candidate classes pertinent to their respective region.

Clustering based on density. In our method, we apply the density-based clustering technique DBSCAN [10] to the attention weights, denoted as the function DBSCAN$(\cdot)$. The essence of density clustering lies in determining categories based on the proximity of samples. DBSCAN asserts that points within a neighborhood are considered density-reachable, and a class is defined as the largest collection of density-reachable points.

Specifically, we perform L2 normalization upon the attention weight $\boldsymbol{w}/||\boldsymbol{w}||_2$ for DBSCAN clustering. It is important to mention that the attention weight $\boldsymbol{w}$ does not include the [CLS] token, resulting in a shape of $\boldsymbol{w} \in \mathbb{R}^{HW \times HW}$. For clarity, we omit the batch size in subsequent expressions. Once we have obtained the clustering result $\boldsymbol{c}$ with N clustering results that assigns cluster IDs for different patches, we can obtain the prototype $\boldsymbol{p}_k$ of the k-th class as follows:

$$\boldsymbol{c} = \text{DBSCAN}(\boldsymbol{w}/||\boldsymbol{w}||_2) \in \mathbb{R}^{HW} \tag{4}$$

$$\boldsymbol{p}_k = \frac{1}{N_k} \sum_{i=1, \boldsymbol{c}_i = k}^{HW} \boldsymbol{w}_i, \tag{5}$$

The prototype of each class is denoted as $\boldsymbol{p}_k$, and N_k represents the number of samples in the k-th class. Consequently, the resulting clustering attention weight prototypes can be represented as $\boldsymbol{p} \in \mathbb{R}^{N \times H \times W}$, where N is the number of clustering. The attention weight matrix obtained by using Cosine similarity is not sensitive to the parameters of DBSCAN during clustering, which we will mention in subsequent Section 5.3.

The semantic segmentation results. After obtaining the prototype attention weights, we proceed to yield the semantic segmentation results $\boldsymbol{m}$ for different classes in an adaptive manner by comparing their clustering results:

$$\boldsymbol{m} = \text{argmax}(\boldsymbol{p}). \tag{6}$$

This allows the attention weights to dynamically determine the category of the object edges. By considering the logits of all pixels within each mask, a collaborative decision is made to determine the class of the corresponding region.

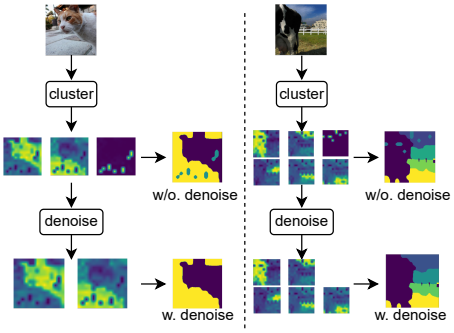

Fig. 5: Examples of clustering and denoising process. It can be clearly seen that there is noise caused by global patch in the results without denoising.

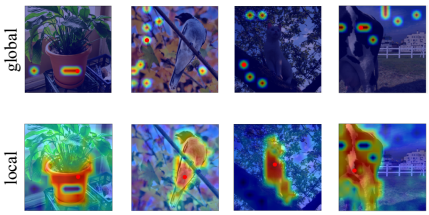

Fig. 6: Comparison of attention maps between local and global patches. The red dot indicates the selected patch position. Global patches have high weights between each other but they have no semantic relevance at all.

Specifically, we use the prediction results of CLIP to vote within each mask to obtain the final pixel-level prediction results. As illustrated in Eq. (1), we use the enhanced CLIP visual encoder to obtain patch features $x \in \mathbb{R}^{HW \times D}$, and calculate the logits ($\mathbb{R}^{C \times H \times W}$) with the text features $t \in \mathbb{R}^{C \times D}$, where C is the number of text classes. We omit the interpolation operation from patch to original image. Finally, the patch logits within each mask collaboratively vote for their candidate class for their respective region.

4.3 Denoising

Though DBSCAN can help remove certain noisy elements [10], the redundant clusters are still inevitable in the outcome, as illustrated in Figure 5, leading to inferior performance.

The evils in global patches. To investigate the cause of the noisy clusters, we re-examine the semantic correlation obtained through the recovery process discussed in Section 4.1. We observe that the noisy clusters are primarily formed by the global patches. From Figure 6, the reason why global patches tend to form noise clusters is due to their intrinsic cohesiveness and the high degree of similarity they share among themselves, which is not as pronounced when compared to correlation with other local image patches.

Identify and discard the noisy clusters. Based on the analysis above, we are surprised to find that for global patches after semantic correlation recovery, their attention weights with respect to other global patches are even greater than the weights with respect to themselves. On one hand, it reflects the significant disruptive effect of global patches on the correlation between local patches. On the other hand, it provides us with insights on how to identify global patches:

Specifically, we firstly extract the self-patch weights $\boldsymbol{w}_{i,i}$ from attention maps, which represent the weights between patch $\boldsymbol{i}$ and patch $\boldsymbol{i}$. Afterwards, we subtract the self-patch weights from the attention map $\boldsymbol{w}_i$ corresponding to patch $\boldsymbol{i}$. If

$\boldsymbol{w}_{i,j} > 0$, it indicates that the correlation between patch $\boldsymbol{i}$ and patch $\boldsymbol{i}$ is higher than the correlation between patch $\boldsymbol{i}$ and itself. This is clearly unreasonable, suggesting that patch $\boldsymbol{j}$ may be a global patch. Additionally, to eliminate the interference of other global patches and extreme outliers in the computation, we further calculate the average weight on patch $\boldsymbol{i}$:

$$\begin{gathered} \boldsymbol{w}_i^* = \frac{1}{HW} \sum_{j=0}^{HW-1} \boldsymbol{w}_{i,j} - \boldsymbol{w}_{i,i}, i \in [0, HW-1] \\ \boldsymbol{w}^* = \{w_i^* \leq 0, i \in [0, HW-1]\} \end{gathered} \tag{7}$$

patches of which correlation between themselves are greater than those with other patches are retained, and the patches that are excluded are considered as global patches. The essence of this implies that even after semantic correlation recovery, if the attention still cannot shift from global patches to the patches themselves or surrounding regions, they will become noise in the clustering.

We utilize the refined weights $\boldsymbol{w}^*$, to perform clustering again and obtain the denoised clustering results $\boldsymbol{p}^* \in \mathbb{R}^{(N-1)\times H\times W}$, which means exclude the noise cluster, just like Figure 5. Next we need to determine the masks of each object through attention weights just like Eq. (6) :

$$\boldsymbol{m}^* = \text{argmax}(\boldsymbol{p}^*, dim = 0) \tag{8}$$

By using the argmax function, the magnitude of clustering results determines the allocation of masks on object boundaries. The subsequent prediction part is the same as described in Section 4.2.

4.4 Integrating With Existing Works

Our method can greatly improve the semantic location capability of CLIP, which can naturally be combined with SAM [16]. The semantic location of CLIP can be used as point prompts to obtain more accurate masks from SAM compared to the Patch Clustering module. SAM can also utilize our CLIPtrase to introduce semantic information to achieve semantic segmentation.

It is worth noting that although SAM can replace our mask generating module, the denoising module is still necessary and is performed before SAM. This is because point prompts completely determine the segmentation area of SAM, its cost affected by noise is greater than our method based on attention weights.

In addition, our method can also be combined with other tasks, we expand this part in supplementary materials.

5 Experiment

5.1 Implementation Details

Experimental setting. All our experiments are based on ViT-B/16. In order to ensure that the experiments are consistent with CLIP, image preprocessing

only includes resize and normalize. In the analysis stages such as reason analysis and ablation experiments, we uniformly use the image size of 224*224. In the main experiment, we appropriately enlarge the size to 336*336 to improve performance. For the hyper-parameter settings in DBSCAN clustering, we empirically set the neighborhood distance threshold ϵ to 0.7 and the neighborhood sample number threshold min_sample to 3, which are consistent for all datasets.

Datasets. We evaluate our method on 9 segmentation datasets, including COCO-stuff [1] with 171 categories, PASCAL VOC2012 [11] with 20 categories, referred to as VOC20, PASCAL Context [25], one with 59 categories, referred to as PC59, and another with complete 459 categories, referred to as PC459, ADE20K [43], one is category 150, referred to as ADE150, and the other is 847 categories, referred to as ADEfull. In addition, in order to further improve the evaluation, we further add background class prediction on COCO (take the first 80 categories), VOC20, PC59: COCO-obj, VOC21, PC60.

Baselines. CLIPtrase is a training-free approach, with the original CLIP as its direct baseline. Additionally, we consider related works such as CLIP [29], CLIP-Surgery [21], and SCLIP [37], which extensively analyze dense feature tasks. However, since SCLIP incorporates a training module, we reproduce it without training to assess the impact of a training-free approach. Furthermore, to evaluate the efficacy of our model, we select representative models requiring training in open vocabulary semantic segmentation, including GroupViT [40], MaskCLIP [44], TCL [2],CAT-Seg [5], DeOP [13], OVSeg [22], SAN [41].

5.2 Results

Table 1 shows the comparison effect of our method with baselines. Compared with the original CLIP, our method greatly improves the potential of CLIP on semantic segmentation task. Originally, CLIP is unable to complete the semantic segmentation task, in which its performance on 6 datasets is less than 10%. Our method can greatly improve this dilemma.

In addition to evaluation on common benchmarks such as VOC20, we test on datasets with extremely many classes such as ADEfull and cases including background, to observe whether our method can improve semantic location on patches while ensuring category generalization as much as possible. Experiments prove that our method is effective. Our method is even close to the current SOTA training model on the ADEfull dataset (10.11% compared with 12.6%), which is hard even for training models.

Our method can greatly improve the potential of CLIP to be applied to semantic segmentation task and realize training-free open vocabulary semantic segmentation. Compared to models that require training, our method can surpass some of the baselines, but there is still a certain gap with the SOTA model like SAN. However, the biggest advantage of our model is that it can achieve training-free segmentation without training on additional datasets.

Our performance further proves that the [CLS] token in CLIP greatly damages the semantic relevance of patches. [CLS] token only learns category difference from global patches, ignoring the semantic location between local patches.

Table 1: Evaluation results (mIoU, %) of our method and the baseline models on ten semantic segmentation benchmarks. The best results on each dataset in the comparison of methods that *do not require training* are bolded. For more metrics (e.g., pAcc, mAcc, and fwIoU), please check the supplementary file.

	train	w. background			w/o. background						Avg.
		CO-obj	VOC21	PC60	CO-stuff	VOC20	PC59	PC459	ADE150	ADEfull	
GroupViT [40]	✓	27.5	52.3	18.7	15.3	79.7	23.4	-	10.4	-	-
MaskCLIP [44]	✓	20.6	43.4	23.2	16.7	74.9	26.4	-	11.9	-	-
TCL [2]	✓	30.4	51.2	24.3	19.6	77.5	30.3	-	14.9	-	-
CAT-Seg [5]	✓	-	78.3	-	-	93.7	57.5	16.6	27.2	8.4	-
DeOP [13]	✓	-	-	-	-	91.7	48.8	9.4	22.9	7.1	-
OVSeg [22]	✓	-	-	-	-	92.6	53.3	11	24.8	7.1	-
SAN [41]	✓	-	-	-	-	94	53.8	12.6	27.5	10.1	-
CLIP [29]	✗	12.63	17.31	8.91	4.70	41.06	9.82	1.88	2.30	0.79	11.04
CLIP-Surgery [21]	✗	-	-	-	21.9	-	29.3	-	-	-	-
SCLIP [37]	✗	40.43	51.79	29.00	21.61	79.12	32.59	8.97	14.76	5.28	31.50
CLIPtrase (Ours)	✗	**44.84**	**53.04**	**30.79**	**24.06**	**81.20**	**34.92**	**9.95**	**17.04**	**5.89**	**33.53**

The semantic correlation calculation restores it as much as possible, which is one of the reasons why our method is effective. We visualize the segmentation predictions in the supplementary file.

5.3 Ablation Study

We uniformly use the image size of 224*224 which is same as CLIP and perform on four common benchmarks COCO-stuff, VOC20, PC59, and ADE150.

Module effectiveness. Our method consists of three modules: semantic correlation recovery (SCR), patch clustering (PC) and denoising (D). As shown in Table 2, we gradually add each module to show the contribution of them to the overall performance. In addition, we analyze the time overhead by flops.

From the results, we can see that the most contribution of our method is semantic correlation recovery, which effectively alleviates the problem of global patches and greatly enhances the semantic correlation between patches. The semantic correlation clustering mainly uses the value of the attention weights of the clustered objects to adaptively obtain the masks of the corresponding objects, making the pixel classification consistency on the objects stronger and avoiding fragmented prediction results.

In addition, the time overhead of SCR and D has almost no increase, it mainly consumes time during clustering. It should be noted that this time overhead is obtained by running single-threaded DBSCAN on the CPU. If parallelization or GPU acceleration is performed, the overhead will be further reduced.

Semantic relevance calculation. In Section 4.1, we mentioned that the semantic similarity matrix obtained by Cosine similarity has a better effect on subsequent clustering due to its determined value range. Here we compare the performance of clustering using semantic similarity matrices obtained by inner product (multi) and Cosine (cos). From Table 2, it can be clearly seen that the

Table 2: Ablation results (mIoU, %) of common four datasets. The best results on average in the comparison are bolded. When comparing different methods of a specific module, other modules are kept unchanged.

	Ablation				Correlation		Cluster		Image size		Layers			
Module	CLIP	+SCR	+PC	+D	Multi	COS	f	p	224	336	9	10	11	12
COCO-stuff	4.70	21.82	22.46	22.84	12.64	22.84	20.44	22.84	22.84	24.06	0.47	1.50	7.75	22.84
VOC20	41.06	77.58	80.43	80.95	63.55	80.95	74.58	80.95	80.95	81.20	13.65	11.83	40.83	80.95
PC59	9.82	32.55	33.21	33.83	19.94	33.83	32.42	33.83	33.83	34.92	1.75	4.94	15.78	33.83
ADE150	2.30	15.79	16.00	16.35	7.49	16.35	15.01	16.35	16.35	17.04	0.41	1.74	4.08	16.35
Avg.	14.47	36.94	38.02	**38.50**	25.91	**38.50**	35.61	**38.50**	38.50	**39.31**	4.07	5.00	17.11	**38.50**
Flops	x1	x1.2	x1.9	x2.0	-	-	-	-	-	-	-	-	-	-

semantic similarity matrix obtained by cos in the fixed value range works very well. More results can be referred in supplementary materials.

Cluster object. In addition to clustering attention weights, we try to cluster features with enhanced self-correlation, which represented by $\boldsymbol{f}$, and the results are shown in the Table 2. Since the feature dimension is higher, clustering is more difficult, so its results are lower than the attention weight prototype $\boldsymbol{p}$ clustering.

Hyper-parameters. They mainly contains: the image size, the clustering parameters, and select which layer to restore semantic correlation.

For the image size, we compare the performance on 224 and 336. As the size increases, the model performance improves to a certain extent, but the number of patches also increases, thereby increasing the time overhead, which requires a trade-off. For better results, we select 336 resolution in our main results. During the analysis and discussion stage, we use 224 resolution to improve efficiency.

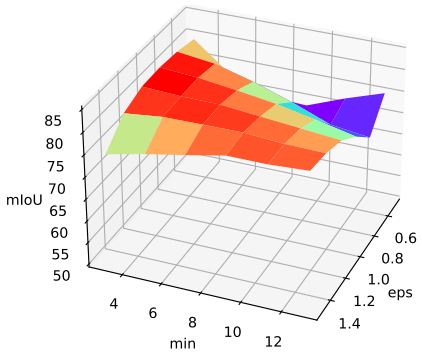

Fig. 7: Sensitivity of clustering parameter space of VOC20. Where eps represents ϵ and min is min_sample. Different colors represent different performance intervals. Except for some marginal extreme hyper-parameter values, the performance of most of the parameter space is above 75, and the best one is 81.01.

For DBSCAN clustering, its main hyper-parameters are neighborhood distance threshold $\boldsymbol{\epsilon}$ and the neighborhood sample number threshold min_sample. Thanks to the deterministic value range of Cosine similarity, the clustering $||w||_2$ jitter is small, allowing us to use a fixed set of parameters to deal with all datasets. In order to illustrate the sensitivity of our method to parameter selec-

Table 3: Evaluation results (mIoU, %) of our method combined with SAM. We replace the patch clustering module with SAM and advance the denoising module to weaken the impact of noise on SAM. The best results on each dataset are bolded.

	w. background			w/o. background						Avg.
	CO-obj	VOC21	PC60	CO-stuff	VOC20	PC59	PC459	ADE150	ADEfull	
CLIP	12.63	17.31	8.91	4.70	41.06	9.82	1.88	2.30	0.79	10.02
SCR+PC+D	**44.84**	53.04	30.79	24.06	81.20	34.92	9.95	17.04	5.89	33.54
SCR+D+SAM	44.23	**57.06**	**31.97**	**24.75**	**82.25**	**36.44**	**10.58**	**17.22**	**5.97**	**34.49**

tion, we use different permutations and combinations of ϵ and min_sample in clustering to observe fluctuations in the results of VOC20. Except for extreme values, our method can maintain stability in most parameter spaces.

As for which layer should be used to restore semantic relevance, our aim is to restore semantic correlation without destroying the CLIP modal alignment as much as possible. We try and compare the deep layers of the model in Table 2 and finally find that the last layer has the best effect.

Combined with SAM. Our method can also be combined with SAM [16] to provide semantic location. The high-resolution boundaries of SAM can further improve CLIP performance on segmentation tasks.

Here we use SAM to replace our mask generating module. We utilize the clustered patch tags to directly interpolate the patch positions of different categories back to the original image as point prompts, and obtain more accurate masks for each object through SAM. The prompts obtained by different masks serve as background points for each other. Essentially, SAM, like the clustering module, provides refined object boundaries for our method. As shown in the Table 3, its refined mask boundaries can further improve the performance of our method.

6 Concluding Remarks

Summary. This paper presents an analysis of the limitations that prevent CLIP from effectively performing dense feature tasks. We observe that the global patches disrupt the semantic correlation between image patches, and propose a method to address this issue. By adopting this method, CLIP can be directly applied to segmentation tasks. Extensive experiments demonstrate that our training-free open-vocabulary semantic segmentation approach can yield considerable improvements to other training-free counterparts.

Limitations. Our research delves into understanding and enhancing the capabilities of CLIP for segmentation without training. However, it's important to acknowledge that despite these improvements, there remains a discernible gap compared to the performance of state-of-the-art trainable models. Moreover, extending our analysis and enhancement techniques to tasks requiring denser feature extraction, like object detection and panoptic segmentation, may be a promising future direction.

Acknowledgements

This work was supported by National Natural Science Foundation of China (grant No. 62376068, grant No. 62350710797), by Guangdong Basic and Applied Basic Research Foundation (grant No. 2023B1515120065), by Guangdong S&T programme (grant No. 2023A0505050109), by Shenzhen Science and Technology Innovation Program (grant No. JCYJ20220818102414031).

References

1. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 1209–1218 (2018)
2. Cha, J., Mun, J., Roh, B.: Learning to generate text-grounded mask for open-world semantic segmentation from only image-text pairs. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 11165–11174 (2023)
3. Chen, J., Zhu, D., Qian, G., Ghanem, B., Yan, Z., Zhu, C., Xiao, F., Elhoseiny, M., Culatana, S.C.: Exploring open-vocabulary semantic segmentation without human labels. arXiv preprint arXiv:2306.00450 (2023)
4. Cheng, B., Schwing, A., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. Adv. Neural Inform. Process. Syst. **34**, 17864–17875 (2021)
5. Cho, S., Shin, H., Hong, S., An, S., Lee, S., Arnab, A., Seo, P.H., Kim, S.: Cat-seg: Cost aggregation for open-vocabulary semantic segmentation. arXiv preprint arXiv:2303.11797 (2023)
6. Darcet, T., Oquab, M., Mairal, J., Bojanowski, P.: Vision transformers need registers. In: Int. Conf. Learn. Represent. (2024)
7. Ding, J., Xue, N., Xia, G.S., Dai, D.: Decoupling zero-shot semantic segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 11583–11592 (2022)
8. Ding, Z., Wang, J., Tu, Z.: Open-vocabulary panoptic segmentation with maskclip. arXiv preprint arXiv:2208.08984 (2022)
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
10. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. p. 226–231. KDD'96, AAAI Press (1996)
11. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. Int. J. Comput. Vis. **111**, 98–136 (2015)
12. Guo, J., Wang, Q., Gao, Y., Jiang, X., Lin, S., Zhang, B.: Mvp-seg: Multi-view prompt learning for open-vocabulary semantic segmentation. In: Pattern Recognition and Computer Vision: 6th Chinese Conference. p. 158–171. Springer-Verlag (2023)
13. Han, C., Zhong, Y., Li, D., Han, K., Ma, L.: Open-vocabulary semantic segmentation with decoupled one-pass network. In: Int. Conf. Comput. Vis. pp. 1086–1096 (2023)
14. Jiao, S., Wei, Y., Wang, Y., Zhao, Y., Shi, H.: Learning mask-aware clip representations for zero-shot segmentation. Adv. Neural Inform. Process. Syst. **36** (2024)

15. Khattak, M.U., Rasheed, H., Maaz, M., Khan, S., Khan, F.S.: Maple: Multi-modal prompt learning. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 19113–19122 (2023)
16. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
17. Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., Liu, S., Jia, J.: Lisa: Reasoning segmentation via large language model. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 9579–9589 (2024)
18. Lai, X., Tian, Z., Xu, X., Chen, Y., Liu, S., Zhao, H., Wang, L., Jia, J.: Decouplenet: Decoupled network for domain adaptive semantic segmentation. In: Eur. Conf. Comput. Vis. pp. 369–387. Springer (2022)
19. Lee, D., Song, S., Suh, J., Choi, J., Lee, S., Kim, H.J.: Read-only prompt optimization for vision-language few-shot learning. In: Int. Conf. Comput. Vis. pp. 1401–1411 (2023)
20. Li, B., Weinberger, K.Q., Belongie, S., Koltun, V., Ranftl, R.: Language-driven semantic segmentation. In: Int. Conf. Learn. Represent. (2022)
21. Li, Y., Wang, H., Duan, Y., Li, X.: Clip surgery for better explainability with enhancement in open-vocabulary tasks. arXiv preprint arXiv:2304.05653 (2023)
22. Liang, F., Wu, B., Dai, X., Li, K., Zhao, Y., Zhang, H., Zhang, P., Vajda, P., Marculescu, D.: Open-vocabulary semantic segmentation with mask-adapted clip. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 7061–7070 (2023)
23. Lin, Y., Chen, M., Wang, W., Wu, B., Li, K., Lin, B., Liu, H., He, X.: Clip is also an efficient segmenter: A text-driven approach for weakly supervised semantic segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 15305–15314 (2023)
24. Liu, Y., Bai, S., Li, G., Wang, Y., Tang, Y.: Open-vocabulary segmentation with semantic-assisted calibration. arXiv preprint arXiv:2312.04089 (2023)
25. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 891–898 (2014)
26. Mukhoti, J., Lin, T.Y., Poursaeed, O., Wang, R., Shah, A., Torr, P.H., Lim, S.N.: Open vocabulary semantic segmentation with patch aligned contrastive learning. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 19413–19423 (2023)
27. Peng, B., Tian, Z., Wu, X., Wang, C., Liu, S., Su, J., Jia, J.: Hierarchical dense correlation distillation for few-shot segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 23641–23651 (2023)
28. Qin, J., Wu, J., Yan, P., et al.: Freeseg: Unified, universal and open-vocabulary image segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 19446–19455 (2023)
29. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: Int. Conf. Mach. Learn. pp. 8748–8763. PMLR (2021)
30. Rao, Y., Zhao, W., Chen, G., Tang, Y., Zhu, Z., Huang, G., Zhou, J., Lu, J.: Denseclip: Language-guided dense prediction with context-aware prompting. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 18082–18091 (2022)
31. Shu, Y., Guo, X., Wu, J., Wang, X., Wang, J., Long, M.: Clipood: Generalizing clip to out-of-distributions. arXiv preprint arXiv:2302.00864 (2023)
32. Tian, Z., Cui, J., Jiang, L., Qi, X., Lai, X., Chen, Y., Liu, S., Jia, J.: Learning context-aware classifier for semantic segmentation. In: AAAI. vol. 37, pp. 2438–2446 (2023)

33. Tian, Z., Lai, X., Jiang, L., Liu, S., Shu, M., Zhao, H., Jia, J.: Generalized few-shot semantic segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 11553–11562 (2022)
34. Tian, Z., Lai, X., Jiang, L., Liu, S., Shu, M., Zhao, H., Jia, J.: Generalized few-shot semantic segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 11563–11572 (2022)
35. Tian, Z., Zhao, H., Shu, M., Yang, Z., Li, R., Jia, J.: Prior guided feature enrichment network for few-shot segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **44**(2), 1050–1065 (2020)
36. Tian, Z., Zhao, H., Shu, M., Yang, Z., Li, R., Jia, J.: Prior guided feature enrichment network for few-shot segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **44**(2), 1050–1065 (2022)
37. Wang, F., Mei, J., Yuille, A.: Sclip: Rethinking self-attention for dense vision-language inference. arXiv preprint arXiv:2312.01597 (2023)
38. Wu, L., Zhang, W., Jiang, T., Yang, W., Jin, X., Zeng, W.: [cls] token is all you need for zero-shot semantic segmentation. arXiv preprint arXiv:2304.06212 (2023)
39. Xiao, G., Tian, Y., Chen, B., Han, S., Lewis, M.: Efficient streaming language models with attention sinks. In: Int. Conf. Learn. Represent. (2024)
40. Xu, J., De Mello, S., Liu, S., Byeon, W., Breuel, T., Kautz, J., Wang, X.: Groupvit: Semantic segmentation emerges from text supervision. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 18134–18144 (2022)
41. Xu, M., Zhang, Z., Wei, F., Hu, H., Bai, X.: Side adapter network for open-vocabulary semantic segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 2945–2954 (2023)
42. Xu, M., Zhang, Z., Wei, F., Lin, Y., Cao, Y., Hu, H., Bai, X.: A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In: Eur. Conf. Comput. Vis. pp. 736–753. Springer (2022)
43. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. Int. J. Comput. Vis. **127**, 302–321 (2019)
44. Zhou, C., Loy, C.C., Dai, B.: Extract free dense labels from clip. In: Eur. Conf. Comput. Vis. pp. 696–712. Springer (2022)
45. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 16816–16825 (2022)
46. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. Int. J. Comput. Vis. **130**(9), 2337–2348 (2022)
47. Zhou, Q., Liu, Y., Yu, C., Li, J., Wang, Z., Wang, F.: LMSeg: Language-guided multi-dataset segmentation. In: Int. Conf. Learn. Represent. (2023)
48. Zhou, Z., Lei, Y., Zhang, B., Liu, L., Liu, Y.: Zegclip: Towards adapting clip for zero-shot semantic segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 11175–11185 (2023)