Training-Free Open-Vocabulary Semantic Segmentation with Context Pyramid Refinement

Jialei Chen[1], Qi Fan[2*], Zhenzhen Quan[3], Dongyue Li[1], Xu Zheng[4], Hiroshi Murase[1], Daisuke Deguchi[1]

[1]Nagoya University, Furo-cho, Chikusa-ku, Nagoya, Aichi, Japan.
[2]Nanjing University, Nanjing, Jinagsu, China.
[3]Shandong University, Qingdao, Shandong, China.
[4]The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, GuangDong, China.

*Corresponding author(s). E-mail(s): fanqi@nju.edu.cn;
Contributing authors: jialeichen2021@gmail.com; quanzhenzhen1991@163.com; nv.31n.2702@s.thers.ac.jp; xzheng287@connect.hkust-gz.edu.cn; murase@nagoya-u.jp; ddeguchi@nagoya-u.jp;

Abstract

Training-free open-vocabulary semantic segmentation aims to adapt vision-language models, such as CLIP, for segmenting novel classes without any finetuning. However, existing methods often suffer from fragmented segmentation due to limited contextual awareness and weak object coherence. Subsequently, semantically related regions and different parts of the same object are often misclassified to different classes. To address these challenges, we enforce intra-object coherence and hierarchically refine contextual features across CLIP's layers without any training. First, we introduce a context aggregation module that enhances both intra- and inter-layer contexts in CLIP features, thereby promoting consistent labeling of semantically related regions. Next, we propose an object-aware attention calibration module that integrates features from different regions of the same object to mitigate fragmentation. Finally, we inject object-level semantics into object features to ensure consistent category assignment across all parts. Our method refines CLIP features by leveraging hierarchical context and enforcing object-level coherence, leading to improved open-vocabulary segmentation performance without finetuning. Moreover, its plug-and-play design enables seamless integration into existing frameworks. Extensive experiments on eight benchmarks demonstrate the effectiveness of our approach.

Keywords: Context Pyramid Refinement, Object Coherence Refinement, Object Semantic Refinement, Open-Vocabulary Semantic Segmentation

1 Introduction

Semantic segmentation, an important task in computer vision, assigns a category label to each pixel in input images [1–6]. Traditional semantic segmentation [1–5] often lack flexibility, as they are limited to predefined classes and require retraining to incorporate new categories, which is typically computationally expensive. The emergence of Contrastive Language-Image Pre-training (CLIP) models [7], trained on large-scale datasets of image-text pairs, has demonstrated strong capabilities in semantic understanding, including tasks like image recognition [8–10]. Despite its strong vision-language alignment for image recognition, CLIP's generalization capability remains limited in open-vocabulary semantic segmentation.

This limitation motivates enhancing CLIP's spatial understanding to support dense prediction tasks. Accordingly, recent works [11–15] extend CLIP to open-vocabulary semantic segmentation, where input images are segmented based on textual class descriptions. Existing open-vocabulary semantic segmentation methods can be broadly categorized into training-based [16–19] and training-free approaches [11, 12, 20, 21]. Although training-based [14, 22–24] methods typically achieve strong performance, they often rely on substantial computational resources. Therefore, recent works have increasingly focused on training-free methods. Training-free methods [13, 25–28] generally aim to utilize CLIP's vision-language alignment for segmentation without modifying its parameters. One key research direction focuses on refining CLIP's localization properties. To this end, some methods [11–13, 26] modify CLIP's original attention maps to reaggregate dense features and improve segmentation quality, as illustrated in Fig. 1 (top). Despite these efforts, the resulting segmentation outputs often remain fragmented, leading to sub-optimal performance.

To investigate the underlying causes of fragmented predictions in training-free open-vocabulary segmentation, we conduct a qualitative analysis of representative methods, *e.g.*, ClearCLIP [13], SCLIP [11], CLIP-Surgery [30], and ProxyCLIP [12], by visualizing both their segmentation outputs and corresponding attention maps, as shown in Fig. 2. This analysis reveals that inaccurate segmentation results are not solely caused by poor localization. In fact, attention maps often correctly highlight discriminative regions, yet erroneous predictions still occur. As illustrated in Fig. 2, although the attention map successfully captures the baby's head, the hat is misclassified as "sheep." This failure indicates limited **contextual awareness**, where the model relies primarily on local visual cues and fails to associate the hat with the surrounding person context. Beyond local misclassification, Fig. 2 further exposes a second, more structural issue: weak **object-level coherence**. Even within a single object, different regions are assigned inconsistent labels, for example, parts of the same hat are incorrectly classified as "person." This suggests that, despite reasonable localization, the model lacks a mechanism to enforce semantic consistency across regions belonging to the same object. Importantly, these observations indicate that

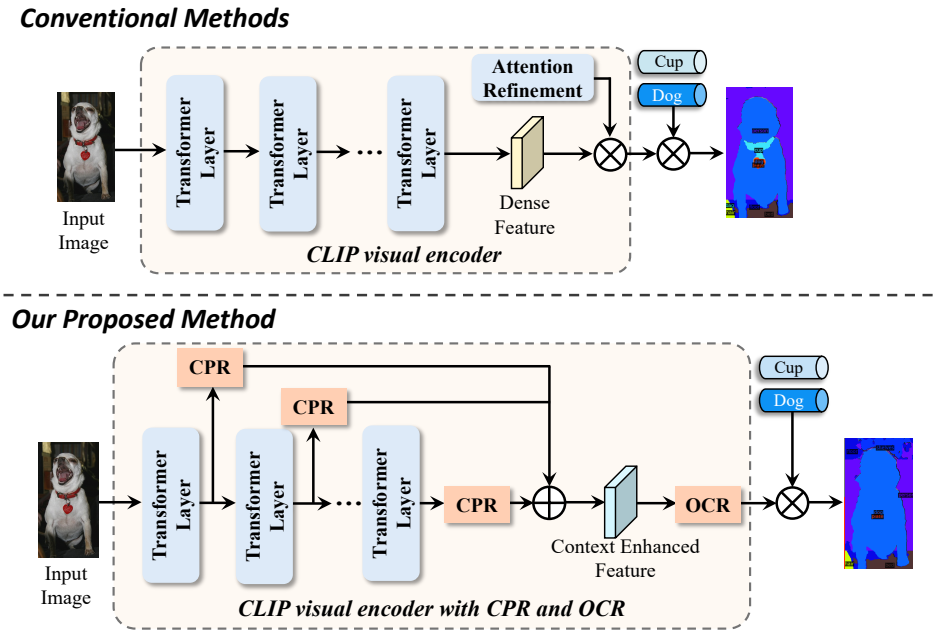

Fig. 1: Comparison of our method with conventional methods. **Top**: Conventional methods rely heavily on features from CLIP's final layer, resulting in limited contextual awareness. **Bottom**: Our method aggregates both intra-layer and inter-layer context of CLIP while enhancing object coherence for semantic segmentation. Our method can be seamlessly integrated into both weakly supervised training-based, *e.g.*, CLIPDINOiser [29], and training-free, *e.g.*, ProxyCLIP [12], frameworks.

refining attention maps alone is insufficient to resolve fragmented predictions. The reason instead lies in CLIP's dense visual representations, which lack explicit contextual reasoning and object-level semantic structure required for dense prediction tasks.

Motivated by this diagnosis, we move beyond treating CLIP as a black-box feature extractor and reorganizing its dense features. Specifically, we inject contextual refinement to make each feature interpretable relative to its surroundings, and enforce object-level coherence to ensure consistent semantics across regions of the same object, together mitigating fragmented predictions and enhancing structural consistency. To improve CLIP's limited contextual awareness, we propose the Context Pyramid Refinement (CPR) module, which aggregates intra-layer context within each CLIP layer and fuses context-enhanced features across layers to capture inter-layer context. By jointly leveraging local and hierarchical context, this module encourages semantically related regions to be assigned to the same category. To improve object coherence, we introduce Object Coherence Refinement (OCR). It calibrates attention maps using clustering centroids computed from dense features, which encourages more holistic feature extraction across entire objects. Moreover, we introduce Object Semantic Refinement (OSR), which further enhances feature consistency by injecting object-level CLS tokens into the dense features. This module enforces semantic

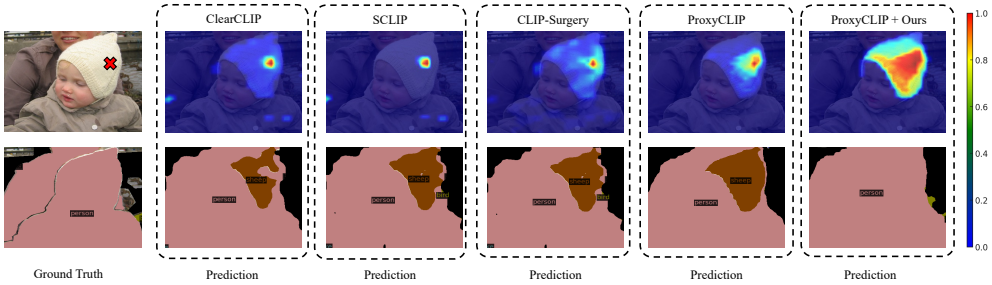

Fig. 2: Visualizations of attention maps and segmentation results from different methods. From left to right, the visualized methods include: query-query (Q-Q) attention from ClearCLIP [13], key-key (K-K) attention from SCLIP, value-value (V-V) attention from CLIP-surgery [30], and proxy attention from ProxyCLIP [12]. The red cross indicates the origin of each attention map. The shown attention maps are averaged across all attention heads. The figure illustrates that fragmented segmentation arises from both insufficient contextual awareness and object coherence.

consistency within each object. By improving contextual awareness and object-level coherence, our method enables CLIP to more accurately group semantically related regions and segment complete object instances, as illustrated in the bottom of Fig. 1.

Compared with training-based segmentation approaches, existing models such as FCN [31], DeepLabV3+[32], PSPNet[6], Swin Transformer [33], and Mask2Former [34] are not directly comparable to our setting, as they rely on supervised training to adapt multi-level features for segmentation. Moreover, the primary objectives of these methods differ from ours. Specifically, FCN [31], DeepLabV3+[32], PSPNet[6], and FPN [35] are mainly designed to address the limited receptive field in CNNs through multi-scale feature fusion, using direct addition, dilated convolutions, and pyramid pooling, respectively. Swin Transformer incorporates shifted window attention to balance efficiency and accuracy at the backbone level, while Mask2Former employs masked attention to improve query–object alignment. Deformable Attention [36], on the other hand, accelerates attention computation via sparse and content-adaptive sampling. Unlike prior multi-scale fusion approaches such as FPN [35] and Deformable Attention [36], which are typically designed for CNN-based architectures or require training to learn adaptive offsets. Unlike training-based methods that rely on parameter updates to adapt model representations, our method instead targets the inherent limitations of CLIP-based segmentation, namely limited contextual awareness, by introducing a hierarchical context refinement module that progressively aggregates multi-scale features through sliding windows across CLIP layers, enabling semantically coherent segmentation without any training. Existing training-free methods primarily focus on enhancing CLIP's localization capability [11–13, 21, 26], while paying limited attention to the contextual awareness of dense features. In contrast to prior approaches, our Context Pyramid Refinement (CPR) aggregates multi-level contextual information across CLIP layers in a hierarchical manner, promoting consistent labeling of semantically related regions. Unlike Swin's [33] fixed and shifted windows designed to reduce attention cost via resolution downsampling, CPR uses sliding windows with increasing sizes across layers to hierarchically aggregate contextual

cues while preserving feature resolution, enabling explicit multi-scale context modeling. Unlike GEM [26], which typically relies on implicit grouping through self-self attention, OCR explicitly clusters features into object-level prototypes and calibrates attention maps with the resulting centroids to enhance object-level awareness and encourage coherent feature aggregation. OSR enhances feature consistency by leveraging region-aware CLS tokens, effectively addressing limitations of prior methods, such as reliance on single tokens [37] and unstable token extraction processes [27]. Our approach is plug-and-play and can be seamlessly integrated into both training-based and training-free frameworks. Extensive experiments demonstrate that our method achieves state-of-the-art performance on eight open-vocabulary semantic segmentation benchmarks. To summarize, the main contributions of this work are as follows:

- We propose Context Pyramid Refinement (CPR), a novel context aggregation module that integrates both intra-layer and inter-layer contextual information to enhance CLIP for open-vocabulary semantic segmentation.

- We introduce Object Coherence Refinement (OCR) and Object Semantic Refinement (OSR) to enhance object-level coherence via attention calibration and to enforce semantic consistency within each object.

- Our method is training-free, integrates seamlessly with existing segmentation frameworks, and achieves state-of-the-art performance across multiple benchmarks.

2 Related Works

2.1 Vision-Language Pretraining

Traditional pretraining approaches [38–40] first train backbone models on large-scale image classification datasets such as ImageNet [41] to learn transferable features, which are then fine-tuned for downstream tasks including semantic segmentation [31, 32, 42, 43]. While this paradigm achieves strong performance in closed-set scenarios, the classes contained in the pretraining dataset is inherently limited [44–46]. As a result, when deployed in open-world scenarios with unlimited categories, such models often fail to recognize novel classes without additional retraining or fine-tuning.

To overcome this limitation, recent research has increasing focused on vision–language pretraining [7, 9, 47, 48], where the backbones are pretrained on massive image–text pairs and excel at open-vocabulary and zero-shot recognition. CLIP [7], as a representative example, has demonstrated strong performance in image-level tasks such as classification [7, 8, 49] and retrieval [50–52]. Despite their success, such models often struggle with pixel-level tasks like semantic segmentation, owing to their limited fine-grained localization capability. Although fine-tuning such models for segmentation is a potential solution [53–55], it demands substantial computational resources and large-scale, carefully curated datasets, which are time-consuming to obtain. To address this limitation, we investigate a training-free paradigm that enables pretrained CLIP [7] for open-vocabulary semantic segmentation.

2.2 Open-Vocabulary Semantic Segmentation

Traditional semantic segmentation typically represents each class using learnable embeddings which are optimized during training and fixed to a predefined set of categories [31–34, 56–59]. As a result, recognizing new classes often requires retraining the model, limiting the model's ability to adapt to open-world scenarios. To address this limitation, recent approaches exploit CLIP [7] for its strong vision–language alignment, which allows open-vocabulary segmentation without the need for retraining on novel categories. Existing approaches can be broadly categorized into training-based [14, 16–19, 22, 29, 37, 60, 61] and training-free methods [11–13, 20, 25, 26, 28]. Training-based methods often explore CLIP's segmentation capability by fine-tuning it on datasets with high-quality pixel-level annotations [14, 17, 22, 37, 60, 61]. Some studies also adopt text-based training [62–64], leveraging Internet-scale image–text pairs to fine-tune CLIP on a massive scale. However, both types of training-based approaches are typically time- and resource-intensive.

Recently, researchers have increasingly focused on training-free open-vocabulary semantic segmentation methods [11–13, 25, 26], which aim to leverage pretrained vision–language models for segmentation without any training [25, 26, 30, 65]. To achieve this, some approaches refine CLIP's [7] attention maps to improve spatial localization. For example, SCLIP [11], ClearCLIP [13], and MaskCLIP [25] replace CLIP's inherent attention maps with customized alternatives, enabling re-aggregation of CLIP's final-layer features to improve pixel-level predictions. There are also methods that adopt region-based inference strategies. CLIP-DIY [66] adopts a dense sliding-window strategy, dividing the image into overlapping patches at multiple scales, classifying each patch independently with CLIP, and aggregating the results into a dense segmentation map. ConceptFusion [65] incorporates external region proposals from SAM [67] or Mask2Former [34] to region-wisely segment the input image. While effective, these methods are typically constrained by their reliance on CLIP's final-layer features which provide limited contextual cues. As a result, segmentation often becomes fragmented, with semantically related regions assigned to different categories and different parts of the same object receiving inconsistent labels. Compared with other open-vocabulary semantic segmentation methods that also leverage mask clustering, our approach is designed to operate in a fully training-free manner. Specifically, [68] generates pseudo labels through clustering to identify potential classes and further employs learnable cluster centers as classifiers, and [69] aligns queries associated with the same category across images to encourage prediction consistency. While effective in their respective settings, both methods typically rely on additional training stages, whereas our method functions without any training or fine-tuning. CLIPtrase [70] aims to alleviate the limited utilization of dense visual tokens in CLIP by clustering attention maps from the final visual layer and averaging the logits within each region to obtain segmentation results. Although training-free, it primarily averages logits via attention-based spatial grouping, merely reorganizing existing predictions without modifying the underlying dense features. In contrast, our approach directly applies clustering to the dense features to extract both local CLS

tokens and object-level centroids. These features are jointly used to enrich CLIP's features with object-level coherence, moving beyond simple feature reorganization and enabling more consistent segmentation.

In contrast, we introduce a training-free framework that effectively refines CLIP features for open-vocabulary segmentation. First, we enhance both intra- and inter-layer contexts in dense features to promote the grouping of semantically related regions. Next, we calibrate the attention maps using feature clustering centroids as reference points for attention computation, thereby enhancing the aggregation of features from disparate regions of the same object. Finally, we incorporate object-level semantics to enforce consistent category assignments across the entire object. These modules jointly improve CLIP's contextual awareness and object coherence, achieving better segmentation performance without any training.

3 Methods

3.1 Preliminary and Overview

Preliminary. CLIP can be instantiated with various backbones among which Vision Transformers (ViTs) [44] are often adopted in practice [11, 12, 25]. A typical ViT-based CLIP consists of N transformer layers. Let $\mathbf{F}^i \in \mathbb{R}^{(L+1)\times C}$ denote the input feature at the i-th transformer layer, where $L+1$ is the sequence length, including an additional CLS token that typically aggregates global information of the input image. The self-attention mechanism computes the query, key, and value features as: $\mathbf{q} = \text{Embed}_q(\text{LN}_1(\mathbf{F}^i))$, $\mathbf{k} = \text{Embed}_k(\text{LN}_1(\mathbf{F}^i))$, $\mathbf{v} = \text{Embed}_v(\text{LN}_1(\mathbf{F}^i))$, where $\mathbf{q}$, $\mathbf{k}$, and $\mathbf{v}$ denote the query, key, and value features, respectively. LN_1 is a layer normalization operation, and Embed_q, Embed_k, and Embed_v are the corresponding linear projections. Then, $\mathbf{q}$, $\mathbf{k}$, and $\mathbf{v}$ are processed by:

$$\mathbf{y} = \mathbf{F}^i + \text{Embed}_y(Attn^\top \cdot \mathbf{v}), \quad \mathbf{F}^{i+1} = \mathbf{y} + \text{Proj}(\text{LN}_2(\mathbf{y})), \tag{1}$$

where Embed_y and Proj are two distinct projection layers, LN_2 denotes a second layer normalization, and $Attn$ represents the attention weights from the self-attention [71]. When an image is processed by CLIP's visual encoder, the final CLS token is extracted and compared to text embeddings with cosine similarity. The class with the highest score is selected as the prediction, enabling open-vocabulary recognition.

Method Overview. An overview of our method is illustrated in Fig. 3. The input image is first encoded by a CNN into patch embeddings, which are then passed through the first K layers of the CLIP visual encoder to produce the intermediate feature $\mathbf{F}^K$. From layers $K+1$ to $N-1$, each intermediate feature $\mathbf{F}^i$ is simultaneously fed into both the subsequent transformer layer and the proposed Context Pyramid Refinement (CPR). This module refines CLIP features by aggregating hierarchical local context via pairwise feature affinities. The context-refined features $\mathbf{F}_a^i$ from all layers are aggregated via element-wise addition to produce the context-aware feature $\mathbf{F}_a$. To further enhance object-level coherence, we propose the Object Coherence Refinement (OCR). This module first clusters the features for attention map computation to discover potential object regions and then refines these features by integrating

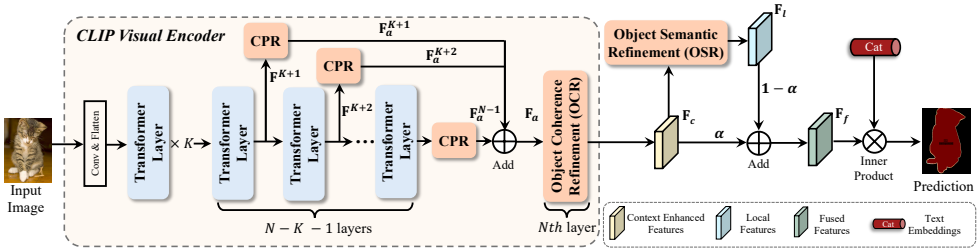

Fig. 3: Overview of our method. An input image is first processed by a convolutional layer and the first K transformer layers of CLIP. From layer $K{+}1$ to N, each intermediate feature $\mathbf{F}^i$ is fed to both the next transformer layer and the Context Pyramid Refinement (CPR), injecting multi-scale context for $\mathbf{F}_a^i$. The refined features $\mathbf{F}_a^i$ are aggregated and enhanced by the OCR into $\mathbf{F}_c$ for object-level coherence. Meanwhile, the OSR assigns region-specific CLS tokens to produce region-aware features $\mathbf{F}_l$. The final output is obtained by linearly fusing $\mathbf{F}_c$ and $\mathbf{F}_l$ with a weight α.

their corresponding cluster centroids. The original query-key attention of CLIP is subsequently replaced by a similarity map computed from these centroid-enhanced features, promoting object-centric aggregation and yielding coherence-aware features $\mathbf{F}_c$. Building on OCR, we introduce Object Semantic Refinement (OSR) to further improve object coherence by enforcing semantic consistency within each object. This module begins by clustering $\mathbf{F}_c$ to obtain non-overlapping regions, which are used to mask the original image and generate region-specific sub-images. Each sub-image is independently fed into the CLIP visual encoder to extract a region-level CLS token. The resulting tokens are injected back into their corresponding regions to produce the semantically enriched feature $\mathbf{F}_l$. Finally, the context- and object- aware feature $\mathbf{F}_c$ and $\mathbf{F}_l$ are linearly combined to produce the final output $\mathbf{F}_f$, which is compared with text embeddings for open-vocabulary segmentation without training. Detailed descriptions of the proposed method are provided in Sec. 3.2, Sec. 3.3, and Sec. 3.4.

3.2 Context Pyramid Refinement (CPR)

Existing methods have explored adapting CLIP for open-vocabulary semantic segmentation in a training-free manner. For example, GEM [26] enhances intermediate-layer features by modifying attention mechanisms to improve spatial coherence and fine-grained semantics. ProxyCLIP [12] and ClearCLIP [13] enhance final-layer attention maps of CLIP to improve object localization. CLIP-DINOiser [29] further introduces a lightweight distillation framework that transfers object-level coherence from DINO [72] to CLIP using a few trainable layers. Despite their promising results, these methods struggle to capture the hierarchical context necessary for semantic segmentation.

To address this limitation, we propose Context Pyramid Refinement (CPR) that enhances contextual awareness through intra-layer and inter-layer context aggregation. Specifically, we begin with an input image $\mathbf{X} \in \mathbb{R}^{H\times W\times 3}$ and extract patch embeddings using convolutional layers. The resulting embeddings are reshaped and processed by the first K layers of the transformer encoder. This process produces an

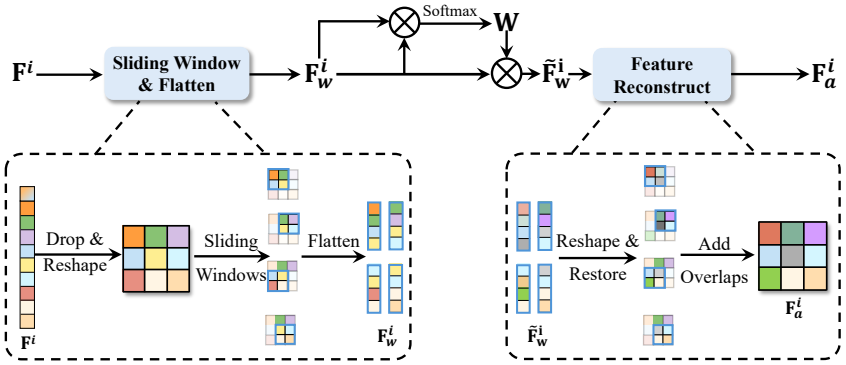

Fig. 4: Overview of Context Pyramid Refinement (CPR) within one layer. The sequenced features $\mathbf{F}^i$ are first partitioned into multiple regions via sliding windows, where the intra-layer context is refined through pairwise feature affinities, producing $\tilde{\mathbf{F}}_w^i$. The refined features are then reconstructed into $\mathbf{F}_a^i$ for segmentation.

intermediate feature map $\mathbf{F}^K \in \mathbb{R}^{(L+1)\times C}$. From the $(K{+}1)$th layer onward, each feature $\mathbf{F}^{K+1}$ is processed by the remaining transformer layers, with CPR applied at each stage to refine and inject hierarchical context.

As shown in Fig. 4, we first remove the CLS token from the i-th layer output $\mathbf{F}^i$ $(i \in [K{+}1, N])$, and reshape the remaining tokens into a 2D feature map $\mathbf{F}_w^i \in \mathbb{R}^{H_w \times W_w \times C}$, where $H_w = H/S_w$ and $W_w = W/S_w$ represent the spatial dimensions determined by the patch size S_w. With this structure, a sliding-window operation is applied to $\mathbf{F}_w^i$ to capture local context while preserving spatial structure. Specifically, we apply a sliding window of size $S \times S$ with stride T to divide $\mathbf{F}_w^i$ into local feature blocks, resulting in a feature $\mathbf{F}_w^i \in \mathbb{R}^{(H_{\text{out}} \times W_{\text{out}}) \times S^2 \times C}$. Here, the total number of output blocks is $H_{\text{out}} \times W_{\text{out}}$, where $H_{\text{out}} = \frac{H_w - S}{T} + 1$ and $W_{\text{out}} = \frac{W_w - S}{T} + 1$. Each local block consists of S^2 tokens, corresponding to spatial patches of size $S \times S$ within the input feature map. To refine intra-region context, we normalize the features and compute pairwise affinities among features in each block, forming a similarity matrix $\mathbf{W}$ that encodes local contextual dependencies.

$$\mathbf{W} = \text{Softmax}\left(\frac{\mathbf{F}_w^i \cdot (\mathbf{F}_w^i)^\top}{\tau}\right), \tag{2}$$

where τ is a temperature factor controlling similarity sharpness and promoting precise feature differentiation. The similarity matrix $\mathbf{W}$ is then used to aggregate local features within each region, updating the refined feature $\tilde{\mathbf{F}}_w^i$ as: $\tilde{\mathbf{F}}_w^i = \mathbf{W} \cdot \mathbf{F}_w^i$. To capture multi-scale context from different CLIP layers, we adopt a pyramid-style sliding window strategy with progressively larger kernel sizes from layers $K{+}1$ to $N-1$. This allows the context refinement module to operate effectively across scales. Specifically, shallow layers use smaller sliding windows to preserve fine-grained local details, while deeper layers apply larger windows to incorporate broader contextual information. The context-refined features from all layers are then summed to form a unified feature map $\mathbf{F}_a$, which serves as the aggregated multi-scale contextual features. CPR

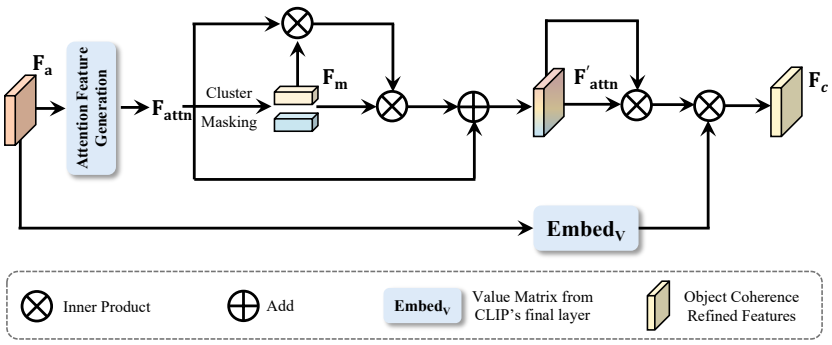

Fig. 5: The overview of Object Coherence Refinement (OCR). The context-refined features $\mathbf{F}_a$ are first used to generate $\mathbf{F}_{attn}$ for attention computation. $\mathbf{F}_{attn}$ is then refined using its clustering centroids $\mathbf{F}_m$, yielding $\mathbf{F}'_{attn}$. In parallel, $\mathbf{F}_a$ is fed into Embed$_v$ to generate value features, and the final output $\mathbf{F}_c$ is produced by re-aggregating these features based on the similarity of $\mathbf{F}'_{attn}$.

is compatible with both training-free approaches [11–13, 25] and weakly supervised training-based methods [29], demonstrating our flexibility.

3.3 Object Coherence Refinement (OCR)

Although CPR effectively leverages hierarchical context within the CLIP visual encoder, CLIP still produces fragmented segmentation results. In addition to limited contextual awareness, CLIP also lacks object-level coherence, as its attention maps often fail to aggregate features corresponding to the same object. To address this issue, we propose the Object Coherence Refinement (OCR) module.

An overview of the OCR module is presented in Fig. 5. Given the CPR-refined feature map $\mathbf{F}_a$ from the CLIP visual encoder, we first generate an attention-guided feature map $\mathbf{F}_{attn}$ using an attention feature extraction function (*e.g.*, the query matrix [13] or key matrix [11] in CLIP's final layer). We then perform clustering on $\mathbf{F}_{attn}$ to obtain a set of clustering centroids. SLIC [73] is chosen to cluster due to its ability to produce coherent segments that reflect structural similarity in $\mathbf{X}$. However, the resulting masks may include fragmented or overly small regions, resulting in inconsistencies. To mitigate this issue, we refine the masks using the mask-merging algorithm from [74], which combines adjacent regions with similar features to reduce fragmentation and improve coherence. The final refined masks are denoted as $\mathbf{M}_c \in [0,1]^{U_c \times H \times W}$, where U_c denotes the number of masks. We then compute the clustering centroids $\mathbf{F}_m$ by averaging the values of $\mathbf{F}_{attn}$ within each mask:

$$\mathbf{F}_m = \left\{ f_m \mid f_m = \frac{\sum_{(H,W)} \mathbf{F}_{attn}[\mathbb{1}(x=l)]}{\sum_{(H,W)} \mathbb{1}(x=l)},\ x \in \mathbf{M}_c \right\}, \tag{3}$$

where $\mathbb{1}(x=l)$ is an indicator function that selects the pixels assigned to the lth mask. Using $\mathbf{F}_m$, we refine the features for attention map computation $\mathbf{F}_{attn}$ to produce an

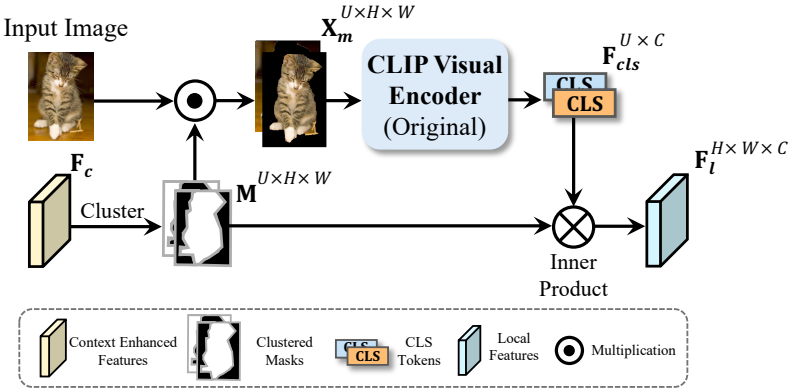

Fig. 6: The overview of Object Semantic Refinement (OSR). First, $\mathbf{F}c$ is clustered to generate the mask $\mathbf{M}$. The mask is then applied to the input image to produce sub-images $\mathbf{X}m$, which are fed into the original CLIP visual encoder to obtain object-level CLS tokens $\mathbf{F}cls$. Finally, $\mathbf{F}cls$ is mapped back to the corresponding regions via inner product, yielding the object-consistent features $\mathbf{F}_l$.

object-coherence enhanced feature $\mathbf{F}'_{attn}$, as follows:

$$\mathbf{F}'_{attn} = \text{Softmax}\left(\frac{\mathbf{F}_{attn}\cdot\mathbf{F}_m^{\top}}{\sqrt{C}}\right)\cdot\mathbf{F}_m + \mathbf{F}_{attn}. \tag{4}$$

Inspired by prior works [11, 12, 64], we construct an alternative attention map based on $\mathbf{F}'_{attn}$ to replace the original query-key attention in CLIP. Specifically, we compute pairwise feature similarities in $\mathbf{F}'_{attn}$ to replace the original Q-K attention. Meanwhile, the context-refined features $\mathbf{F}_a$ are projected through the final-layer value embedding Embed_v to obtain the value features. The final refined features $\mathbf{F}_c$ are obtained by aggregating the value features using feature similarities. By enhancing object coherence, OCR calibrates the attention maps with object-awareness, enabling more effective aggregation of features within one object and reducing fragmentation.

3.4 Object Semantic Refinement (OSR)

Beyond attention map limitations, the lack of object coherence also arises from insufficient constraints at the feature level, which hinders the enforcement of semantic consistency among features belonging to the same object. To mitigate this issue, we propose the Object Semantic Refinement (OSR) module.

The overview of OSR is illustrated in Fig. 6. Given an input image $\mathbf{X}$ and CPR-, OCR-refined features $\mathbf{F}_c$, OSR enhances object-level semantics by injecting region-aware CLS tokens into $\mathbf{F}_c$, yielding more explicit and consistent region features. Specifically, we apply the same clustering and region-merging algorithm from OCR to $\mathbf{F}_c$ to obtain object-awareness masks $\mathbf{M} \in [0,1]^{U\times H\times W}$, where U is the number of clustered regions. Each mask corresponds to a potential semantic object and is applied to the original image $\mathbf{X}$ to retain only the pixels of that region, setting all others to

zero. The resulting masked images $\mathbf{X}_m \in \mathbb{R}^{U\times 3\times H\times W}$ are obtained via element-wise multiplication, $\mathbf{X}_m = \mathbf{M}\odot\mathbf{X}$, where $\odot$ denotes pixel-wise multiplication. Each masked image is then independently processed by the frozen CLIP visual encoder (excluding CPR and OCR) to extract a region-level CLS token $\mathbf{F}_{cls} \in \mathbb{R}^{U\times C}$. These CLS tokens act as semantic prototypes, summarizing each segmented region into a consistent feature. To project these region-level embeddings back into the spatial domain, we compute the dot product between the CLS tokens and their corresponding soft masks: $\mathbf{F}_l = \mathbf{F}_{cls}^{\top}\cdot\mathbf{M}$. This yields $\mathbf{F}_l \in \mathbb{R}^{C\times H\times W}$, injecting region-level semantics into the spatial domain. Finally, the region-consistent feature $\mathbf{F}_l$ is fused with the context- and object-coherence refined feature $\mathbf{F}_c$ using a weighted sum:

$$\mathbf{F}_f = \alpha\cdot\mathbf{F}_c + (1-\alpha)\cdot\mathbf{F}_l, \tag{5}$$

where α is a hyperparameter that balances object-aware contextual features and region-level semantic consistency. OSR explicitly integrates region semantics into dense features, promoting consistent predictions and reducing fragmentation within each object, thereby improving open-vocabulary segmentation.

While both our method and [74] utilize the same clustering and mask fusion to localize potential object regions, their downstream usage diverges significantly. In [74], a Transformer decoder is trained to combine trainable features with the local CLS token, producing pseudo text embeddings that mimic textual semantics. These are used to supervise a new zero-shot segmentation model. In contrast, our approach is fully training-free and does not construct any pseudo embeddings. We directly extract local CLS tokens within each region as semantic abstractions, leveraging CLIP's inherent vision–language alignment to enhance dense features with stronger object-level coherence, yielding improved segmentation performance without any training.

4 Experiment Results

4.1 Experiment Setups

Datasets. To evaluate the generalization ability of our method, we follow the standard protocols established in prior works [11, 12, 29], and conduct extensive experiments across eight datasets. The primary benchmarks include PASCAL VOC [75] (VOC20), PASCAL Context [76] (Context59), COCO-Stuff [77] (COCO), ADE20K [78] (ADE), and Cityscapes [79] (City). To further expand the evaluation scope, we additionally include VOC21, Context60, and COCO-Object (Object) [77]. VOC21 and Context60 are extended versions of VOC20 and Context59, respectively, with an added "background" category. COCO-Object is constructed by selecting the first 80 object-related classes from the COCO-Stuff subset, focusing solely on object-level categories. Together, these eight datasets provide a comprehensive and diverse testbed for evaluating the generalization performance of our approach.

Implementation Details. We adopt ProxyCLIP [12] as our baseline and replace the attention feature $\mathbf{F}_{attn}$ in Eq. 4 with features from DINO [72]. We follow prior works [11–13] and adopt a sliding inference strategy, where each image is divided into sub-images for independent processing and the results are subsequently merged. This

enables arbitrary input resolutions and avoids long-context limitations, generalizing well to high-resolution datasets such as Cityscapes (2048×1024). The window size for sliding inference is set to 336×336 pixels, with a stride of 112×112 pixels. Context Pyramid Refinement (CPR) aggregates features from the last three layers of the CLIP visual encoder using sliding window sizes of 2, 4, and 8, respectively. CPR leverages PyTorch's built-in unfold/fold operations to enable fully parallel window processing, with self-attention applied in batch after unfold, ensuring high parallel efficiency without sequential bottlenecks. We set the fusion weight α to 0.8, and fix the sharpness parameter τ in Eq. 2 at 0.05. For class names, we adopt the same template prompts as in the original CLIP [7], generating semantic embeddings by feeding them into the ImageNet-style templates. The SLIC algorithm in OCR and OSR is applied in two stages: seed initialization and iterative clustering. We initialize a uniform 7×7 grid of seeds across all datasets to control the number of superpixels. Following [74], the mask merging threshold is set to 0.8, enabling the fusion of masks with similar semantics. Unless otherwise specified, all experiments are conducted with the ViT-B/16 [44] backbone. We report results on the validation sets without any additional training or post-processing, using mIoU as the evaluation metric.

To accommodate different application requirements for efficiency and accuracy, our framework supports both full and lightweight configurations. The full configuration incorporates all three modules, CPR, OCR, and OSR, to maximize segmentation performance through joint modeling of multi-scale context, object-level coherence, and semantic consistency. In contrast, the lightweight configuration includes only CPR and OCR, substantially reducing computational overhead while preserving contextual reasoning and object-aware attention. This configuration is especially suitable for resource-constrained or real-time scenarios.

4.2 Comparison with SOTA Methods

We conduct comprehensive evaluations on eight open-vocabulary benchmarks, as shown in Table 1. Among existing methods, ProxyCLIP [12] achieves the highest average baseline score of 42.3%, followed by SCLIP [11] and ClearCLIP [13], which generate attention features using the query matrix from CLIP's final layer, achieving 38.2% and 38.1%, respectively. In contrast, methods such as ReCo [80] and vanilla CLIP show inferior performance, highlighting the importance of advanced adaptation strategies for open-vocabulary segmentation. Nevertheless, these approaches still face generalization challenges across diverse segmentation tasks. Integrating our approach into ClearCLIP [13], SCLIP [11], and ProxyCLIP [12] leads to consistent performance improvements across all datasets, demonstrating its effectiveness in enhancing segmentation quality. Our method achieves the highest overall average score of 45.1%, outperforming ProxyCLIP [12] by 2.8%, with improvements on Context59 and ADE.

In terms of performance on background-free datasets, our method achieves the highest accuracy across all benchmarks, particularly excelling on those with complex semantics and multiple categories, such as Context59, Stuff, Cityscapes, and ADE20K. Although the gain on VOC is relatively small, we still obtain the best result. We attribute this to VOC's limited scale, where many methods already saturate the performance. For datasets with background, our method does not outperform all prior

Table 1: Comparison with state-of-the-art training-free methods. **Bold** and underlined numbers indicate the best and second-best performances.

Methods	Backbone	w. background			w/o. background					Avg.
		VOC	Context	Object	VOC20	Context59	Stuff	City	ADE	
CLIP [7]	ViT-B [44]	16.4	8.4	5.6	41.9	9.2	4.4	5.0	2.9	11.7
ReCo [80]		25.1	19.9	15.7	57.7	22.3	14.8	21.1	11.2	23.5
MaskCLIP [25]		38.8	23.6	20.6	74.9	26.4	16.4	12.6	9.8	27.9
CLIPSurgery [30]		-	29.3	-	-	-	21.9	31.4	-	-
GEM [26]		46.2	32.6	-	-	-	-	-	15.7	-
CLIPtrase [70]		53.0	30.8	**44.8**	81.0	-	24.0	-	17.0	-
CLIPSeg [81]		-	35.3	37.9	83.5	39.7	27.1	36.4	19.7	-
NACLIP [21]		64.1	35.0	36.2	83.0	38.4	25.7	38.3	19.1	42.5
ResCLIP [28]		61.1	33.5	35.0	86.0	36.8	24.7	35.9	18.0	41.4
ClearCLIP [13]		51.8	32.6	33.0	80.9	35.9	23.9	30.0	16.7	38.1
ClearCLIP [13] + Ours(light)		51.5	34.4	33.0	77.1	37.9	24.5	34.9	18.1	38.9 (↑ 0.8)
ClearCLIP [13] + Ours(full)		52.3	33.2	34.1	79.9	39.2	24.6	35.0	18.5	39.6 (↑ 1.5)
SCLIP [11]		59.1	30.4	30.5	80.4	34.2	22.4	32.2	16.1	38.2
SCLIP [11] + Ours (light)		61.6	33.7	33.6	80.2	37.7	24.6	37.6	18.7	41.0 (↑ 1.8)
SCLIP [11] + Ours (Full)		61.9	35.6	35.9	82.1	39.2	25.3	38.2	19.3	42.2 (↑ 4.0)
ProxyCLIP [12]		61.3	35.3	37.5	80.3	39.1	26.5	38.1	20.2	42.3
ProxyCLIP [12] + Ours (light)		65.3	37.3	38.7	82.0	41.4	27.7	40.3	21.2	44.2 (↑ 1.9)
ProxyCLIP [12] + Ours (full)		**66.5**	**37.7**	39.8	**84.3**	**42.2**	**28.0**	**40.7**	**21.8**	**45.1** (↑ 2.8)
CLIP [7]	ViT-L [44]	8.2	4.1	2.7	15.6	4.4	2.4	2.5	1.7	5.2
MaskCLIP [25]		23.3	11.7	7.2	29.4	12.4	8.8	11.5	7.2	13.9
SCLIP [11]		43.5	22.3	25.0	69.1	25.2	17.6	18.6	10.9	29.0
ClearCLIP [13]		48.7	28.3	29.7	80.0	29.6	19.9	27.9	15.0	34.9
ResCLIP [28]		54.1	30.9	32.5	85.5	34.5	23.4	33.7	18.2	39.1
ProxyCLIP [12]		60.6	34.5	39.2	83.2	37.7	25.6	40.1	22.6	43.0
ProxyCLIP [12] + Ours (light)		62.0	36.4	38.5	84.1	39.8	27.1	**41.0**	**24.0**	44.1 (↑ 1.1)
ProxyCLIP [12] + Ours (full)		**63.0**	**36.5**	**40.5**	**84.8**	**40.2**	**27.3**	40.2	23.8	**44.5** (↑ 1.5)
OpenCLIP [9]	ViT-H [44]	8.8	5.0	5.3	21.7	5.5	3.2	4.3	2.8	7.1
MaskCLIP [25]		31.4	13.3	16.2	41.7	15.8	8.4	17.7	10.4	19.3
SCLIP [11]		43.8	23.5	24.6	67.5	25.6	16.8	19.5	11.3	29.1
ProxyCLIP [12]		65.0	35.4	38.6	83.3	39.6	26.8	**42.0**	24.2	44.4
ProxyCLIP [12] + Ours (light)		66.3	36.4	39.6	85.1	40.9	27.7	41.5	25.0	45.3 (↑ 0.9)
ProxyCLIP [12] + Ours (full)		**67.4**	**36.5**	**40.7**	**86.7**	**41.0**	**28.0**	41.8	**25.3**	**45.9** (↑ 1.5)

work. Specifically, CLIPtrase [70] achieves the best performance on COCO-Object. This can be explained by the fact that COCO-Object only retains 80 foreground (thing) classes while removing most background (stuff) categories, such as sky and grass. As a result, it forms relatively simple scenes that align well with CLIPtrase [70] patch-reconstruction-based strategy. In contrast, our method demonstrates consistent advantages across both background-free and background-included datasets, reflecting stronger generalization and robustness in diverse open-vocabulary segmentation.

We further apply our method to different backbones, *i.e.*, ViT-L, and ViT-H. As shown in Table 1, our method consistently improves with stronger backbones and achieves the best performance with ViT-H across all datasets, confirming its robustness to backbone scale. Notably, the performance of ViT-L is inferior to ViT-B, which is observed in other baselines such as ProxyCLIP [12], SCLIP [11] and ClearCLIP [13]. For example, when using ViT-B, ProxyCLIP [12] obtains mIoU of 61.3 on VOC, 35.3 on Context59, and 37.5 on Stuff, all surpassing the corresponding results obtained with ViT-L. Similar or worse degradation appears in other baselines, suggesting that the drop originates from the backbone itself rather than our method. We hypothesize this reflects an intrinsic limitation of ViT-L in CLIP: as the model scales from

Table 2: Comparison with state-of-the-art training-free methods with different image resolutions. **Bold** and underline indicate the best and second-best performances.

Dataset	Resolution	SCLIP [11]	SCLIP [11] + Ours	ProxyCLIP [12]	ProxyCLIP [12] + Ours
VOC	468 × 386	59.1	61.9	61.3	**66.5**
Context	468 × 386	30.4	35.6	35.3	**37.7**
Object	577 × 485	30.5	35.9	37.5	**39.8**
VOC20	468 × 386	80.4	82.1	80.3	**84.3**
Context59	468 × 386	34.2	39.2	39.1	**42.2**
Stuff	573 × 483	22.4	25.3	26.5	**28.0**
City	2048 × 1024	32.2	38.2	38.1	**40.7**
ADE	616 × 621	16.1	19.3	20.2	**21.8**
Avg.	-	38.2	42.2	42.3	**45.1**

Table 3: Comparison with training-based methods. **Bold** and underlined numbers indicate the best and second-best performances.

Methods	VOC	Context	Object	VOC20	Context59	Stuff	City	ADE	Avg.
OVSegmentor [82]	53.8	20.4	25.1	-	-	-	-	5.6	-
CoCu [83]	51.4	23.6	22.7	-	-	22.1	15.2	12.3	-
SAM-CLIP [84]	60.6	29.2	-	-	-	-	-	17.1	-
GroupViT [63]	50.4	18.7	27.5	79.7	23.4	15.3	11.1	9.2	29.4
TCL [85]	51.2	24.3	30.4	77.5	30.3	19.6	23.1	14.9	33.9
CLIP-DINOiser [29]	62.2	32.4	35.0	80.2	35.9	24.6	31.7	20.0	40.3
CLIP-DINOiser [29] + Ours (light)	62.5	**33.6**	34.0	80.5	37.9	25.5	34.0	21.1	40.9 (↑ 0.6)
CLIP-DINOiser [29] + Ours (full)	**62.9**	33.4	**35.7**	**80.9**	**38.0**	**25.6**	**34.6**	**21.2**	**41.6** (↑ 1.3)

ViT-B to ViT-L, global alignment improves but local perception weakens due to limited capacity. This harms downstream dense prediction tasks like segmentation, which require both. In contrast, ViT-H restores the balance between global and local representations, enabling our method to fully leverage its rich multi-scale features.

To evaluate whether our method can adapt to different image resolutions, we compute the average image resolution of each dataset and evaluate the corresponding performance, as shown in Table 2. The results show that our method consistently improves upon baseline training-free approaches across datasets with varying image resolutions. On low-resolution datasets (*i.e.*, VOC and Context), our method consistently improves upon representative training-free baselines, including SCLIP [11] and ProxyCLIP [12]. On medium-resolution datasets (*i.e.*, VOC20, Context59, Stuff, and ADE), we observe similar gains over both baselines. More importantly, these improvements also hold on high-resolution datasets such as Cityscapes (2048×1024), indicating that the performance gains are consistent across low-, medium-, and high-resolution inputs rather than being tied to a particular resolution or baseline method. This robustness can be attributed to the sliding-window inference strategy, which enables resolution-agnostic inference without resizing or additional fine-tuning.

Table 3 compares our method with representative training-based open-vocabulary segmentation approaches. Without any parameter updates, directly integrating our modules into the frozen CLIP-DINOiser [29] yields consistent gains, improving the average mIoU from 40.3% to 41.6% (+1.3%). Our full variant achieves the best results on 7 datasets, while the light variant also delivers solid improvements compared with baseline. Compared with prior methods that require retraining, our approach demonstrates strong cross-dataset generalization on Context59, City, and ADE.

Table 4: Comparison of computational cost (GFLOPs) between baseline methods and their counterparts with the proposed method.

Method	SCLIP	ClearCLIP	ProxyCLIP
Original	39.7	39.6	253.3
+ Ours(light)	44.0	43.9	263.7
+ Ours(full)	128.0	127.8	348.3

Table 5: Ablation studies of the proposed method, where "B" denotes the baseline (ProxyCLIP [12]), "P" denotes CPR, "C" denotes OCR, and "R" denotes OSR.

Methods	GFLOPs	FPS	VOC	Context	Object	VOC20	Context59	Stuff	City	ADE	Avg.
B	**253.3**	**26.9**	61.3	35.3	37.5	80.3	39.1	26.5	38.1	20.2	42.3
B+P	254.5	22.4	65.2	37.0	38.7	81.8	41.1	27.5	39.9	21.1	44.0
B+C	262.4	17.2	64.3	35.9	38.5	81.1	40.4	27.3	38.5	20.8	43.4
B+P+C (light)	263.7	15.9	65.3	37.3	38.7	82.0	41.4	27.7	40.3	21.2	44.2
B+P+C+R (full)	348.3	7.8	**66.5**	**37.7**	**39.8**	**84.3**	**42.2**	**28.0**	**40.7**	**21.8**	**45.1**

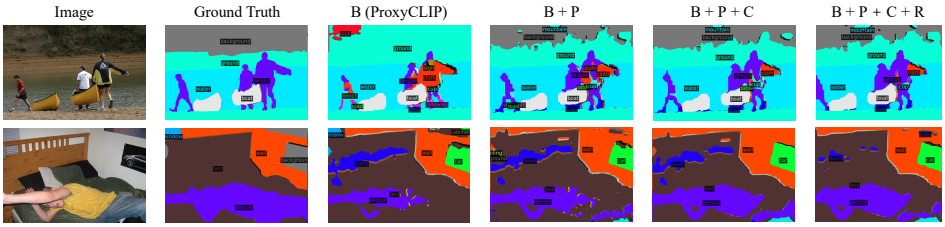

Fig. 7: Qualitative ablations of our method which improves contextual understanding and object coherence, *e.g.*, clothes worn by a person are correctly labeled as person.

We further compare the computational cost (GFLOPs) between baseline methods and our approach. For all three baselines, SCLIP [11], ClearCLIP [13], and ProxyCLIP [12], integrating our approach introduces additional computation, as reflected by the increased GFLOPs, as shown in Table 4. However, the increase remains moderate under the lightweight configuration. For instance, SCLIP's GFLOPs increase from 39.7 to 44.0 with the lightweight version and to 128.0 with the full version. ClearCLIP shows a similar trend, increasing from 39.6 to 43.9 (light) and to 127.8 (full). ProxyCLIP, a heavier model, increases from 253.3 to 263.7 (light) and to 348.3 (full), demonstrating our flexibility with minimal computational overhead.

4.3 Analysis on Proposed Method

To further evaluate the effectiveness of each proposed component, we use ProxyCLIP with a ViT-B backbone [44] as the baseline. In the following experiments, CPR is applied to the last three layers of CLIP using sliding window sizes of 2, 4, and 8.

Ablations studies on the proposed method. We present the quantitative analysis in Table 5. Starting from the baseline ProxyCLIP ("B"), adding CPR ("P") improves the mIoU from 42.3 to 44.0, while increasing the computational cost only marginally, from 253.3 to 254.5 GFLOPs, and reducing the inference speed from 26.9 to 22.4 FPS. This demonstrates a favorable trade-off between performance and efficiency. Incorporating the OCR module ("C") further raises the accuracy to 44.2, with the computation increasing to 263.7 GFLOPs and the inference speed remaining at a

Table 6: Comparison of our method with different CLIP in ViT-B backbone.

CLIP Type	VOC	Context	Object	VOC20	Context59	Stuff	City	ADE	Avg.
OpenCLIP [9]	61.9	35.6	34.3	84.4	39.9	28.0	**42.3**	**23.8**	43.8
MetaCLIP [8]	65.2	**38.7**	**42.5**	82.5	**42.5**	**28.7**	39.9	23.5	**45.4**
EVA02-CLIP [10]	61.1	35.6	38.9	**86.3**	38.2	25.9	36.6	23.6	43.2
CLIP [7]	**66.5**	37.7	39.8	84.3	42.2	28.0	40.7	21.8	45.1

Table 7: Ablation study on window sizes (x, y, z) applied to the last three layers (3rd, 2nd, and last) of the CLIP visual encoder.

Window size	VOC	Context	Object	VOC20	Context59	Stuff	City	ADE	Avg.
None	66.4	36.7	39.9	82.6	42.0	28.2	38.5	21.7	44.5
2,2,2	**66.8**	37.5	**39.9**	82.9	42.5	**28.3**	40.3	**21.8**	45.0
4,4,4	**66.8**	**37.7**	**39.9**	83.2	42.5	28.1	39.9	21.6	45.0
8,8,8	64.5	37.3	38.6	83.1	**42.6**	28.0	40.0	21.4	44.4
8,4,2	64.6	34.9	35.4	79.1	41.2	26.8	38.0	20.3	42.5
2,4,8	66.5	**37.7**	39.8	**84.3**	42.2	28.0	**40.7**	**21.8**	**45.1**

reasonable 15.9 FPS, highlighting its effectiveness in balancing segmentation quality and efficiency. Notably, OCR alone already achieves a strong performance–efficiency balance. Finally, integrating the OSR module ("R") leads to the full configuration ("B+P+C+R"), which attains the highest accuracy of 45.1, at the cost of 348.3 GFLOPs and 7.8 FPS. This progressive gain–cost trend confirms that each module contributes complementary improvements.

To further assess the effectiveness of our method, we visualize the predictions at each ablation stage in Fig. 7. The segmentation quality progressively improves as each component is added. Starting from the baseline (B), the model struggles to maintain object coherence (*e.g.*, the person in the canoe or on the bed). With CPR (B+P), region-level semantics are enhanced, leading to better segmentation on the cloth worn by people and reduced background noise. Incorporating coherence refinement (B+P+C) further enhances spatial consistency and improves separation of adjacent objects. Finally, integrating semantic refinement (B+P+C+R) results in more complete segmentation, further demonstrating our effectiveness.

Impact on different CLIP variants. Table 6 compares three CLIP variants used in our framework: OpenCLIP [9], MetaCLIP [8], and the original CLIP [7]. The original CLIP achieves competitive performance across most datasets, with an overall average of 45.1%. MetaCLIP [8] slightly underperforms on certain datasets but achieves the highest overall average of 45.4%, suggesting a more accurate feature of localized regions. OpenCLIP [9] consistently underperforms compared to the other variants, achieving the lowest average score and indicating weaker generalization. In addition to these methods, we further evaluate our approach with different CLIP variants by replacing the CLIP visual encoder with EVA-CLIP [10]. As shown in the table, our method remains fully compatible and achieves strong performance, demonstrating its ability to adapt to diverse CLIP variants and benefit from stronger visual encoders.

Table 8: Comparison between intra-layer and inter-layer CPR.

Methods	VOC	Context	Object	VOC20	Context59	Stuff	City	ADE	Avg.
CPR (only in final layer)	65.8	37.0	38.9	84.2	41.5	27.4	39.5	21.4	44.5
CPR (across layers)	**66.5**	**37.7**	**39.8**	**84.3**	**42.2**	**28.0**	**40.7**	**21.8**	**45.1**

Table 9: Comparison on different applications of Context Pyramid Refinement, where x indicates that the last x layers are used.

Stage	VOC	Context	Object	VOC20	Context59	Stuff	City	ADE	Avg.
1	65.9	37.0	39.1	84.2	41.5	27.6	39.8	21.5	44.6
2	66.0	37.1	39.1	84.2	41.7	27.8	40.0	21.6	44.7
3	**66.5**	**37.7**	**39.8**	**84.3**	**42.2**	**28.0**	**40.7**	**21.8**	**45.1**
4	65.4	37.5	39.7	84.0	42.0	27.9	40.5	21.7	44.8

4.3.1 Analysis on Context Pyramid Refinement

To evaluate the impact of CPR on different types of context modeling, we conduct experiments to analyze: **(1)** its effect on inter-layer integration and **(2)** its effect on intra-layer context modeling.

4.3.1.1 Analysis on Inter-layer Context Modeling

Impact of sliding window design sizes different layers. We first conduct a quantitative analysis of sliding window sizes, as shown in Table 7. Without sliding windows, simply averaging the features from the last three stages results in a significant performance drop, from 45.1% to 44.5% in average mIoU. We then apply a uniform window size across all three stages. With a window size of 2, performance improves on Context59 and Stuff compared to the default setting. However, performance on VOC20 and City remains lower. Increasing the window size further leads to continued performance degradation. Finally, reversing the window size order (from 2, 4, 8 to 8, 4, 2) also results in degraded performance.

To further analyze the behavior of different layers, we perform qualitative analysis by visualizing the predictions at each stage, as shown in Fig. 8. Without CPR, the model misclassifies key regions, for example, confusing mountains with background or missing some people. Applying CPR to deeper layers progressively improves recognition of object boundaries and regional semantics. Notably, applying CPR to all stages results in more accurate segmentation by providing richer contextual information.

Impact on multiple-layer feature utilization. Table 8 presents a quantitative comparison between two strategies for applying our CPR: using multi-scale windows only at the final layer versus applying them across all layers. We observe that applying CPR across multiple layers consistently outperforms the final-layer-only variant on all datasets. For instance, the average mIoU increases from 44.5% to 45.1%. These results suggest that multi-scale context aggregation is more effective when integrated throughout the network hierarchy, rather than limited to the final layer.

To better understand these findings, we perform qualitative analysis. When CPR is applied only at the final layer, misclassifications still occur due to limited context. For example, motorcycles may be mistaken for bicycles, and humans for sofas, owing to local visual similarities. In contrast, inter-layer CPR aggregates multi-scale context across the entire network hierarchy, as shown in Fig. 9.

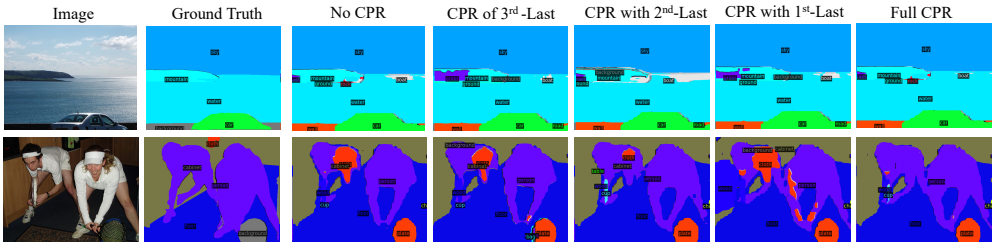

Fig. 8: Qualitative analysis of CPR results across different layers. Compared with no CPR or single-layer CPR, combining multi-layer CPR features yields more complete object-coherent predictions and richer contextual understanding.

Table 10: Comparison on different strides of the sliding windows.

stride	VOC	Context	Object	VOC20	Context59	Stuff	City	ADE	Avg.
1	66.5	**37.7**	**39.8**	**84.3**	**42.2**	**28.0**	**40.7**	**21.8**	**45.1**
2	66.2	36.5	39.7	83.8	41.6	27.6	39.9	21.7	44.5
S / 2	**66.8**	36.5	39.2	82.1	42.0	27.9	37.3	21.4	44.2
S	65.7	34.2	37.4	81.4	40.8	26.8	35.6	21.0	42.9

Table 11: Comparison on different ways to aggregate the intra-layer context.

Fusion	VOC	Context	Object	VOC20	Context59	Stuff	City	ADE	Avg.
Avg	44.6	28.4	28.8	82.8	31.5	17.5	27.6	21.7	35.4
Max	37.8	11.2	14.3	56.8	18.4	11.7	9.1	9.1	21.1
Self-attn (ours)	**66.5**	**37.7**	**39.8**	**84.3**	**42.2**	**28.0**	**40.7**	**21.8**	**45.1**

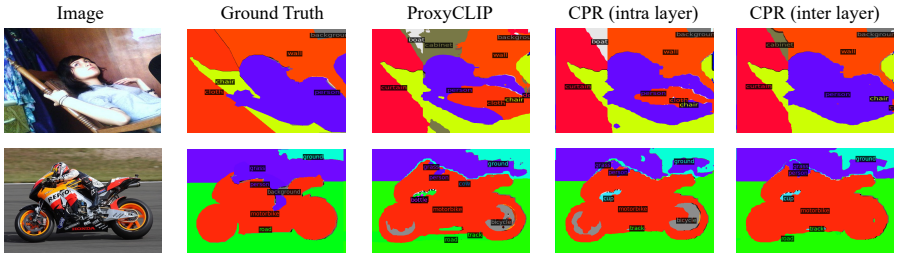

Fig. 9: Qualitative comparison of intra-layer vs. inter-layer CPR, where integrating features from multiple layers produces more coherent object predictions and enhanced contextual awareness, *e.g.*, wheels of motorbike is correctly segmented.

Impact of number of stages in CPR. Table 9 summarizes the effect of incorporating different CLIP stages in our hierarchical refinement framework. We evaluate the impact of incorporating the last 1 to 4 layers (denoted as Stages 1 to 4) for multi-layer feature enhancement. Among all configurations, incorporating the last three layers (Stage 3) achieves the best overall performance, with an average mIoU of 45.1%. In contrast, using fewer stages (*e.g.*, Stage 1 or 2) results in insufficient feature abstraction. Incorporating four layers (Stage 4) slightly degrades performance.

Table 12: Comparison on different clustering strategies.

Clustering	VOC	Context	Object	VOC20	Context59	Stuff	City	ADE	Avg.
KMeans	65.8	37.1	39.4	84.8	41.6	27.8	39.5	21.5	44.7
HDBscan	65.4	37.1	39.3	83.5	42.1	28.1	39.9	21.5	44.6
Felzenszwalb-Huttenlocher	66.8	36.8	39.3	83.0	42.0	27.9	40.1	21.4	44.7
SLIC (ours)	**66.5**	**37.7**	**39.8**	**84.3**	**42.2**	**28.0**	**40.7**	**21.8**	**45.1**

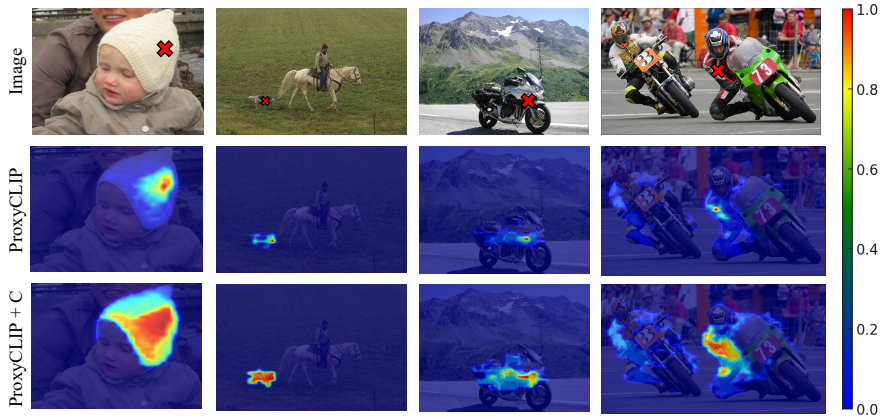

Fig. 10: Qualitative analysis of the OCR attention map, showing that features can be aggregated from all parts of each individual object.

4.3.1.2 Analysis on Intra-layer Context Modeling

Impact on the stride of windows. Table 10 presents the performance of our method under various stride settings applied in the pyramid sliding window. We evaluate four stride variants: fixed strides of 1 and 2, and feature-resolution-dependent strides $S/2$ and S, where S denotes the spatial resolution of the feature map. Results show that a stride of 1 yields the best overall performance, achieving the highest average mIoU of 45.1%, and outperforming all other settings across most datasets. Specifically, it achieves the highest scores on Context, Object, and Stuff. Although $S/2$ achieves a slightly higher score on VOC, its performance declines on other datasets.

Impact on the how to aggregate context. Table 11 compares different fusion strategies for aggregating intra-layer contextual information: average pooling (Avg), max pooling (Max), and self-attention (Ours). Our method consistently outperforms both baselines across all benchmarks, achieving the highest average mIoU of 45.1. In contrast, average pooling yields a moderate mIoU of 35.4, while max pooling performs poorly with only 21.1, demonstrating the effectiveness of our approach.

4.3.2 Analysis on OCR and OSR

Impact on clustering strategies. Table 12 compares different clustering algorithms used in the OCR and OSR modules, including KMeans, HDBSCAN, Felzenszwalb-Huttenlocher, and our applied method (SLIC). While all methods deliver comparable overall performance, our method consistently achieves the best results on most datasets. In particular, it obtains the highest average mIoU of 45.1, outperforming

Table 13: Ablation studies on the mask merging.

Methods	VOC	Context	Object	VOC20	Context59	Stuff	City	ADE	Avg.
w/o. merging	65.4	35.8	39.1	82.2	41.6	28.0	39.4	21.3	44.1
w. merging	**66.5**	**37.7**	**39.8**	**84.3**	**42.2**	**28.0**	**40.7**	**21.8**	**45.1**

Table 14: Comparison on different α in OSR.

α	VOC	Context	Object	VOC20	Context59	Stuff	City	ADE	Avg.
0.6	66.5	36.8	39.2	**85.6**	41.8	27.7	40.1	**22.0**	45.0
0.7	**66.7**	37.4	**39.8**	85.0	**42.2**	**28.0**	40.5	**22.0**	**45.2**
0.8	66.5	**37.7**	**39.8**	84.3	**42.2**	**28.0**	**40.7**	21.8	45.1
0.9	66.1	37.6	39.4	83.2	41.9	27.9	40.6	21.5	44.8

the baselines on Context, Context59, and ADE. Although Felzenszwalb-Huttenlocher performs well on VOC, it falls short on Object and ADE.

Qualitative analysis on OCR. Fig. 10 compares the attention maps of ProxyCLIP [12] and ProxyCLIP + Ours, illustrating the impact of our proposed method on context-aware attention. For each input image (top row), the red cross marks the location where self-attention is centered. The second row presents attention maps from ProxyCLIP, showing limited and fragmented object coherence, often confined to a small neighborhood around the marked point. In contrast, the third row illustrates results from ProxyCLIP + Ours, demonstrating more coherent and spatially expansive attention that captures relevant object parts and surrounding context. This broader and more consistent attention distribution confirms the effectiveness of our method in promoting richer feature aggregation and improving semantic understanding.

Impact on the mask merging. Table 13 presents an ablation study evaluating the effectiveness of the mask merging step. We compare model performance with and without this component to isolate its contribution. The results show that integrating mask merging consistently improves segmentation quality across all benchmarks. Specifically, the average mIoU increases from 44.1% to 45.1%, with particularly notable gains observed on VOC, Context, and VOC20 datasets. These improvements can be attributed to mask merging to consolidate fragmented predictions into more coherent object regions, thereby reducing over-segmentation.

Impact on different α in OSR. Table 14 summarizes model performance across various datasets, VOC, Context, Object, VOC20, Context59, Stuff, City, and ADE, under different values of α. The table also includes the average performance (Avg.). The results show that the choice of α significantly impacts segmentation quality across datasets. Specifically, $\alpha = 0.7$ yields the highest average score, with consistent improvements on VOC, Context, and Object compared to other values. For ADE, the best result (22.0%) is observed at both $\alpha = 0.6$ and $\alpha = 0.7$. In contrast, increasing α to 0.9 leads to a performance drop, resulting in the lowest average score and noticeable degradation on datasets such as VOC20 and ADE. These findings highlight the importance of proper hyperparameter tuning, with $\alpha = 0.7$ offering effective setting.

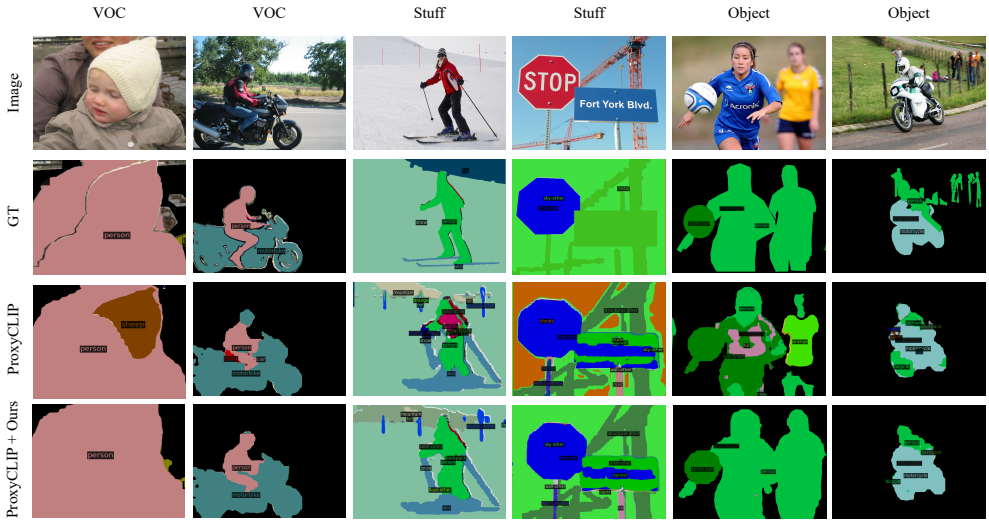

Fig. 11: Visualization comparison. The four rows show the input images, ground truth, ProxyCLIP [12], and our predicted masks. Our method achieves superior performance by enhancing contextual awareness and improving object coherence.

4.3.3 Prediction Visualization

We present qualitative results comparing segmentation performance across various datasets, including VOC, Stuff, and Objectc. The results are shown in Fig. 11. Each column corresponds to a distinct dataset and scenario, with selected examples chosen to emphasize challenging cases. Compared to the baseline method, *i.e.*, ProxyCLIP [12], our approach consistently produces more accurate segmentation boundaries and improved label consistency, particularly in complex scenes with overlapping objects, as exemplified by images from the Stuff and Object datasets.

5 Conclusion

In this paper, we introduce a training-free framework for open-vocabulary semantic segmentation that refines CLIP's visual features from both contextual and object-coherence perspectives. To enhance contextual awareness, we proposed Context Pyramid Refinement (CPR), which captures intra- and inter-layer contexts across multiple CLIP layers. To strengthen object-level coherence, we developed Object Coherence Refinement (OCR) and Object Semantic Refinement (OSR), jointly promoting consistent feature grouping and semantic consistency within individual objects. Our plug-and-play design requires no training and can be seamlessly integrated into existing segmentation frameworks including training-based and training-free methods. Extensive experiments on eight benchmarks demonstrate consistent mIoU improvements and a substantial reduction in fragmented predictions, demonstrating the potential of leveraging CLIP's features for dense prediction in training-free open-vocabulary segmentation.

Future Works: In future work, we will explore the underlying causes of computational overhead and seek more efficient designs. Our method is compatible with CLIP variants that include a global CLS token, such as the original CLIP [7] and EVA-CLIP [10], enabling effective alignment for open-vocabulary segmentation. However, CLS-token-free models like SigLIP [86], which rely on attention pooling, show a significant drop in performance. This issue also affects other training-free approaches, suggesting a strong reliance on global representations. In future work, we aim to extend our framework to CLS-token-free vision-language models for training-free open-vocabulary segmentation.

Acknowledgment

Support for this work was given by the Toyota Motor Corporation (TMC) and JSPS KAKENHI Grant Number 23K28164 and JST CREST Grant Number JPMJCR22D1. However, note that this article solely reflects the opinions and conclusions of its authors and not TMC or any other Toyota entity.

6 Data availability statements

The datasets used in this study are all publicly available. Specifically: The COCO-Stuff dataset can be accessed at https://github.com/nightrome/cocostuff. The ADE20K dataset is available at http://groups.csail.mit.edu/vision/datasets/ADE20K. The PASCAL VOC 2012 dataset is available at http://host.robots.ox.ac.uk/pascal/VOC/voc2012. The PASCAL Context dataset can be obtained from https://cs.stanford.edu/ roozbeh/pascal-context/.

References

[1] Chen, J., Deguchi, D., Zhang, C., Zheng, X., Murase, H.: Frozen is better than learning: A new design of prototype-based classifier for semantic segmentation. PR (2024)

[2] Zheng, X., Luo, Y., Zhou, P., Wang, L.: Distilling efficient vision transformers from cnns for semantic segmentation. PR (2025)

[3] Bi, H., Feng, Y., Mao, Y., Pei, J., Diao, W., Wang, H., Sun, X.: Agmtr: Agent mining transformer for few-shot segmentation in remote sensing. IJCV (2024)

[4] Xie, H., Wang, C., Zhao, J., Liu, Y., Dan, J., Fu, C., Sun, B.: Prcl: Probabilistic representation contrastive learning for semi-supervised semantic segmentation. IJCV (2024)

[5] Siam, M.: Temporal transductive inference for few-shot video object segmentation. IJCV (2025)

[6] Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR (2017)

[7] 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: ICML (2021)

[8] Hu, X., Saining, X., Xiaoqing, E.T., Po-Yao, H., Russell, H., Vasu, S., Shang-Wen, L., Gargi, G., Luke, Z., Christoph, F.: Demystifying clip data. arXiv preprint arXiv:2309.16671 (2023)

[9] Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., Jitsev, J.: Reproducible scaling laws for contrastive language-image learning. In: CVPR (2023)

[10] Sun, Q., Fang, Y., Wu, L., Wang, X., Cao, Y.: Eva-clip: Improved training techniques for clip at scale. arXiv (2023)

[11] Wang, F., Mei, J., Yuille, A.: Sclip: Rethinking self-attention for dense vision-language inference. In: ECCV (2024)

[12] Lan, M., Chen, C., Ke, Y., Wang, X., Feng, L., Zhang, W.: Proxyclip: Proxy attention improves clip for open-vocabulary segmentation. In: ECCV (2024)

[13] Lan, M., Chen, C., Ke, Y., Wang, X., Feng, L., Zhang, W.: Clearclip: Decomposing clip representations for dense vision-language inference. In: ECCV (2024)

[14] 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)

[15] Ye, C., Zhuge, Y., Zhang, P.: Towards open-vocabulary remote sensing image semantic segmentation. In: AAAI (2025)

[16] 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: CVPR (2023)

[17] Ghiasi, G., Gu, X., Cui, Y., Lin, T.-Y.: Scaling open-vocabulary image segmentation with image-level labels. In: ECCV (2022)

[18] Shi, H., Dao, S.D., Cai, J.: Llmformer: Large language model for open-vocabulary semantic segmentation. IJCV (2025)

[19] Wang, Z., Feng, T., Lyu, F., Shang, F., Feng, W., Wan, L.: Dual semantic guidance for open vocabulary semantic segmentation. In: CVPR (2025)

[20] Kim, C., Ju, D., Han, W., Yang, M.-H., Hwang, S.J.: Distilling spectral graph for

object-context aware open-vocabulary semantic segmentation. In: ICCV (2025)

[21] Hajimiri, S., Ben Ayed, I., Dolz, J.: Pay attention to your neighbours: Training-free open-vocabulary semantic segmentation. In: WACV (2025)

[22] Xu, M., Zhang, Z., Wei, F., Hu, H., Bai, X.: Side adapter network for open-vocabulary semantic segmentation. In: CVPR (2023)

[23] Han, K., Liu, Y., Liew, J.H., Ding, H., Liu, J., Wang, Y., Tang, Y., Yang, Y., Feng, J., Zhao, Y., *et al.*: Global knowledge calibration for fast open-vocabulary segmentation. In: ICCV (2023)

[24] Wu, S., Zhang, W., Xu, L., Jin, S., Li, X., Liu, W., Loy, C.C.: CLIPSelf: Vision transformer distills itself for open-vocabulary dense prediction. In: ICLR (2024)

[25] Zhou, C., Loy, C.C., Dai, B.: Extract free dense labels from clip. In: ECCV (2022)

[26] Bousselham, W., Petersen, F., Ferrari, V., Kuehne, H.: Grounding everything: Emerging localization properties in vision-language transformers. In: CVPR (2024)

[27] Lin, Y., Chen, M., Zhang, K., Li, H., Li, M., Yang, Z., Lv, D., Lin, B., Liu, H., Cai, D.: Tagclip: A local-to-global framework to enhance open-vocabulary multi-label classification of clip without training. In: AAAI (2024)

[28] Yang, Y., Deng, J., Li, W., Duan, L.: Resclip: Residual attention for training-free dense vision-language inference. In: CVPR (2025)

[29] Wysoczańska, M., Siméoni, O., Ramamonjisoa, M., Bursuc, A., Trzciński, T., Pérez, P.: Clip-dinoiser: Teaching clip a few dino tricks for open-vocabulary semantic segmentation. ECCV (2024)

[30] 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)

[31] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)

[32] Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018)

[33] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV (2021)

[34] Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: CVPR (2022)

[35] Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017)

[36] Xia, Z., Pan, X., Song, S., Li, L.E., Huang, G.: Vision transformer with deformable attention. In: CVPR (2022)

[37] Li, J., Chen, P., Qian, S., Jia, J.: Tagclip: Improving discrimination ability of open-vocabulary semantic segmentation. arXiv preprint arXiv:2304.07547 (2023)

[38] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)

[39] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR (2015)

[40] Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: CVPR (2022)

[41] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV (2015)

[42] Cheng, B., Schwing, A., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. In: NeurIPS (2021)

[43] Zeng, Q., Lu, Z., Xie, Y., Xia, Y.: Pick: Predict and mask for semi-supervised medical image segmentation. IJCV (2025)

[44] 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)

[45] He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020)

[46] Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML (2020)

[47] Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: ICML (2021)

[48] Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: ICML (2022)

[49] Zhang, Y., Huang, X., Ma, J., Li, Z., Luo, Z., Xie, Y., Qin, Y., Luo, T., Li, Y.,

Liu, S., et al.: Recognize anything: A strong image tagging model. arXiv preprint arXiv:2306.03514 (2023)

[50] Sain, A., Bhunia, A.K., Chowdhury, P.N., Koley, S., Xiang, T., Song, Y.-Z.: Clip for all things zero-shot sketch-based image retrieval, fine-grained or not. In: CVPR (2023)

[51] Shen, C., Liu, Y., Zhai, W.: ColCLIP: Enhancing fine-grained image retrieval with pre-trained embeddings. In: ICLR (2024)

[52] Nara, R., Lin, Y.-C., Nozawa, Y., Ng, Y., Itoh, G., Torii, O., Matsui, Y.: Revisiting relevance feedback for clip-based interactive image retrieval. In: ECCV (2024)

[53] Zhou, Z., Lei, Y., Zhang, B., Liu, L., Liu, Y.: Zegclip: Towards adapting clip for zero-shot semantic segmentation. In: CVPR (2023)

[54] Ding, J., Xue, N., Xia, G.-S., Dai, D.: Decoupling zero-shot semantic segmentation. In: CVPR (2022)

[55] Zhang, Y., Guo, M.-H., Wang, M., Hu, S.-M.: Exploring regional clues in clip for zero-shot semantic segmentation. In: CVPR (2024)

[56] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. TPAMI (2017)

[57] Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. NeurIPS (2021)

[58] Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., *et al.*: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: CVPR (2021)

[59] Guo, M.-H., Lu, C.-Z., Hou, Q., Liu, Z., Cheng, M.-M., Hu, S.-M.: Segnext: Rethinking convolutional attention design for semantic segmentation. In: NeurIPS (2022)

[60] 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: ECCV (2022)

[61] Han, C., Zhong, Y., Li, D., Han, K., Ma, L.: Zero-shot semantic segmentation with decoupled one-pass network. arXiv preprint arXiv:2304.01198 (2023)

[62] 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: CVPR (2023)

[63] Xu, J., De Mello, S., Liu, S., Byeon, W., Breuel, T., Kautz, J., Wang, X.: Groupvit: Semantic segmentation emerges from text supervision. In: CVPR (2022)

[64] Luo, H., Bao, J., Wu, Y., He, X., Li, T.: Segclip: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. In: ICML (2023)

[65] Jatavallabhula, K.M., Kuwajerwala, A., Gu, Q., Omama, M., Chen, T., Maalouf, A., Li, S., Iyer, G., Saryazdi, S., Keetha, N., et al.: Conceptfusion: Open-set multimodal 3d mapping. arXiv preprint arXiv:2302.07241 (2023)

[66] Wysoczańska, M., Ramamonjisoa, M., Trzciński, T., Siméoni, O.: Clip-diy: Clip dense inference yields open-vocabulary semantic segmentation for-free. In: WACV (2024)

[67] 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. In: ICCV (2023)

[68] Shin, H., Kim, C., Hong, S., Cho, S., Arnab, A., Seo, P.H., Kim, S.: Towards open-vocabulary semantic segmentation without semantic labels. NeurIPS (2024)

[69] Dao, S.D., Shi, H., Phung, D.Q., Cai, J.: Ca-ovs: Cluster and adapt mask proposals for open-vocabulary semantic segmentation. In: MmAsia (2024)

[70] Shao, T., Tian, Z., Zhao, H., Su, J.: Explore the potential of clip for training-free open vocabulary semantic segmentation. In: ECCV (2025)

[71] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS (2017)

[72] Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: ICCV (2021)

[73] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. TPAMI (2012)

[74] Chen, J., Deguchi, D., Zhang, C., Zheng, X., Murase, H.: Clip is also a good teacher: A new learning framework for inductive zero-shot semantic segmentation. arXiv preprint arXiv:2310.02296 (2023)

[75] Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. IJCV (2015)

[76] 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: CVPR (2014)

[77] Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: CVPR (2018)

[78] Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: CVPR (2017)

[79] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)

[80] Shin, G., Xie, W., Albanie, S.: Reco: Retrieve and co-segment for zero-shot transfer. In: NeurIPS (2022)

[81] Liu, Y., Wang, G., Zhang, J., Liu, Q., Huang, D.: Unveiling the knowledge of clip for training-free open-vocabulary semantic segmentation. In: AAAI (2025)

[82] Xu, J., Hou, J., Zhang, Y., Feng, R., Wang, Y., Qiao, Y., Xie, W.: Learning open-vocabulary semantic segmentation models from natural language supervision. In: CVPR (2023)

[83] Xing, Y., Kang, J., Xiao, A., Nie, J., Shao, L., Lu, S.: Rewrite caption semantics: Bridging semantic gaps for language-supervised semantic segmentation. NeurIPS (2024)

[84] Wang, H., Vasu, P.K.A., Faghri, F., Vemulapalli, R., Farajtabar, M., Mehta, S., Rastegari, M., Tuzel, O., Pouransari, H.: Sam-clip: Merging vision foundation models towards semantic and spatial understanding. In: CVPR (2024)

[85] Cha, J., Mun, J., Roh, B.: Learning to generate text-grounded mask for open-world semantic segmentation from only image-text pairs. In: CVPR (2023)

[86] Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: ICCV (2023)