Target Refocusing via Attention Redistribution for Open-Vocabulary Semantic Segmentation: An Explainability Perspective

Jiahao Li[1], **Yang Lu**[1], **Yachao Zhang**[1], **Yong Xie**[3*], **Fangyong Wang**[4], **Yuan Xie**[2], **Yanyun Qu**[1*]

[1]Key Laboratory of Multimedia Trusted Perception and Eficient Computing.Ministry of Education of China, School of Informatics, Xiamen University
[2]School of Computer Science and Technology, East China Normal University
[3]Department of Computer, Nanjing University of Posts and Telecommunications
[4]Hanjiang National Laboratory

Abstract

Open-vocabulary semantic segmentation (OVSS) employs pixel-level vision-language alignment to associate category-related prompts with corresponding pixels. A key challenge is enhancing the multimodal dense prediction capability, specifically this pixel-level multimodal alignment. Although existing methods achieve promising results by leveraging CLIP's vision-language alignment, they rarely investigate the performance boundaries of CLIP for dense prediction from an interpretability mechanisms perspective. In this work, we systematically investigate CLIP's internal mechanisms and identify a critical phenomenon: analogous to human distraction, CLIP diverts significant attention resources from target regions to irrelevant tokens. Our analysis reveals that these tokens arise from dimension-specific over-activation; filtering them enhances CLIP's dense prediction performance. Consequently, we propose Refocusing CLIP (RF-CLIP), a training-free approach that emulates human distraction-refocusing behavior to redirect attention from distraction tokens back to target regions, thereby refining CLIP's multimodal alignment granularity. Our method achieves SOTA performance on eight benchmarks while maintaining high inference efficiency.

Introduction

Open-vocabulary semantic segmentation assigns categorical labels from an unrestricted vocabulary to each image pixel via prompt-guided multimodal semantic alignment. The core challenge lies in achieving pixel-level dense prediction, where fine-grained multimodal alignment constitutes a critical bottleneck. Existing approaches predominantly leverage CLIP's (Radford et al. 2021) vision-language alignment capability to address this challenge, falling into three paradigms: 1) **Joint Fine-tuning** (Cho et al. 2024; Jiao et al. 2023; Li et al. 2024): simultaneously fine-tuning CLIP alongside segmentation-specific components to enhance dense prediction capabilities; 2) **Pre Fine-tuning** (Wu et al. 2023, 2024; Xu et al. 2022): refining CLIP's alignment granularity through fine-grained vision-language contrastive learning; 3) **Training-Free Adaptation**(Lan et al. 2024a; Wang, Mei, and Yuille 2024; Lan et al. 2024b): modulating CLIP's final residual attention layer or integrating vision foundation models (VFMs)(Zhang et al. 2022; Oquab et al. 2023) to boost alignment granularity. Collectively, these paradigms deploy complementary strategies—augmenting CLIP with task-specific modules, re-pretraining weights, or adapting final layers while aggregating VFMs—all targeting enhanced multimodal alignment granularity. Despite promising results, existing methods rarely investigate the performance boundaries of CLIP for dense prediction from an interpretability perspective or explore the origins of its inherent inter-layer spatial misalignment, thereby limiting further improvements in OVSS performance. In this work, we focus on the training-free paradigm, exclusively modulating CLIP to enhance pixel-level dense prediction. We first systematically investigate CLIP's internal mechanisms and uncover a key phenomenon: during visual encoding, CLIP produces certain tokens irrelevant to the input query that consume substantial attention resources, causing distraction from originally focused target regions (termed the "distraction" phenomenon causing layer-wise spatial misalignment). Further analysis reveals these tokens stem from over-activation in specific dimensions; filtering them significantly improves CLIP's OVSS performance. Their presence seems to indicate that CLIP struggles to focus on the current visual targets, impairing its dense prediction ability. This insight motivates us to explore whether mimicking the human distraction-refocusing behavior — by reallocating attention resources from distraction tokens to defocused target regions — can enhance CLIP's multimodal alignment granularity.

To this end, we propose ReFocusing CLIP (RF-CLIP), a straightforward training-free modulation method that operates directly on CLIP's inter-layer attention mechanisms. The core idea lies in: 1) identifying query-irrelevant tokens consuming substantial attention resources and locating defocused target regions within CLIP's intermediate visual embeddings; 2) reallocating attention resources from distraction tokens to these defocused target regions. By exclusively modulating CLIP's attention mechanism, RF-CLIP significantly enhances OVSS performance while preserving inference efficiency (Figure 1). Our principal contributions are:

- We identify the "distraction" phenomenon and attribute it to dimension-specific over-activation inherent in CLIP, which undermines the multimodal alignment granularity.

*Corresponding author.

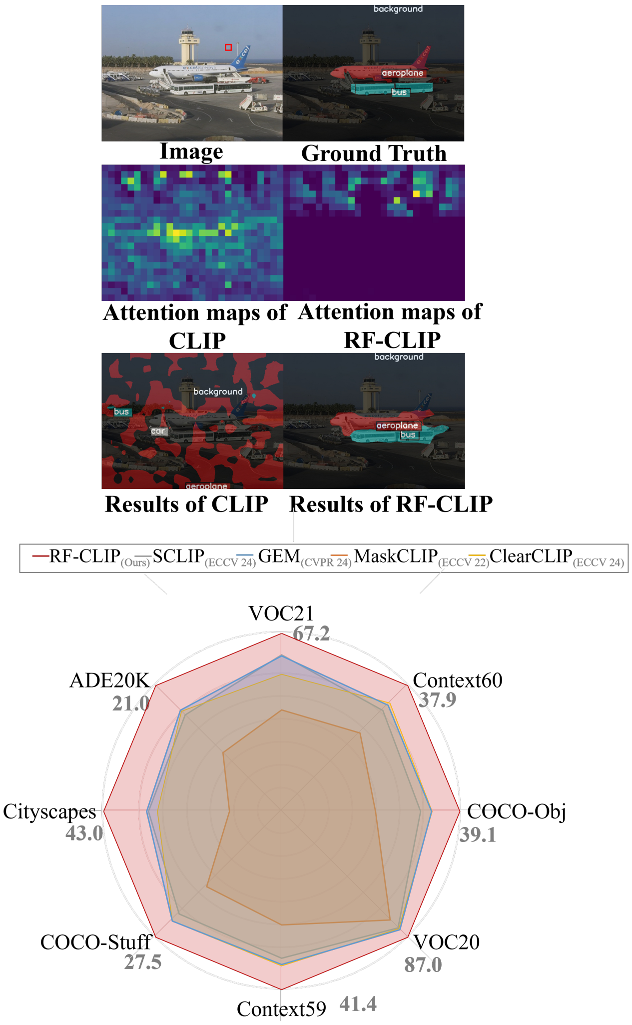


Figure 1: RF-CLIP achieves precise attention focus, which facilitates accurate segmentation of target regions. Compared to state-of-the-art methods, RF-CLIP demonstrates superior performance, achieving the highest accuracy.

- We propose RF-CLIP, a training-free attention modulator that enhances alignment granularity by reallocating misallocated attention resources to defocused target regions.
- Our method achieves SOTA performance on eight OVSS benchmark datasets while simultaneously preserving superior computational efficiency during inference.

Preliminaries

Problem definition. Given an image $\boldsymbol{I}$ and a set of class-specific textual descriptions $\{\boldsymbol{T}_i\}_{i=1}^{N_c}$, where $\boldsymbol{T}_i$ denotes the textual description of the i-th class and N_c represents the total number of classes, CLIP aligns each pixel in $\boldsymbol{I}$ with the most semantically relevant $\boldsymbol{T}_i$, thereby assigning the corresponding class label i to the pixel. Note that the total number of classes N_c is dynamic during inference.

Dense inference for training-free OVSS. The ViT-based CLIP model primarily comprises multiple stacked residual attention layers. Each residual layer contains a self-attention module $\mathcal{S}(\cdot)$ and a feedforward module $\mathcal{F}(\cdot)$. Given an image $\boldsymbol{I}$, a visual residual layer processes it to extract visual embeddings $\boldsymbol{f} \in \mathbb{R}^{N\times d}$, where N represents the number of patches and d is the latent space dimension. (In actual, $\boldsymbol{f}$ comprises a global visual embeddings $\boldsymbol{f}^{'} \in \mathbb{R}^{1\times d}$ and a local dense embeddings $\boldsymbol{f}^{''} \in \mathbb{R}^{N\times d}$; for simplicity, we refer to $\boldsymbol{f}$ as the local dense embeddings $\boldsymbol{f}^{''}$ and ignore the global visual embeddings). The computation process within a visual residual layer is defined as (normalization operations omitted for simplicity):

$$\boldsymbol{f}^l \leftarrow \boldsymbol{f}^l + \mathcal{S}(\boldsymbol{f}^l), \tag{1}$$

$$\boldsymbol{f}^{l+1} \leftarrow \boldsymbol{f}^l + \mathcal{F}(\boldsymbol{f}^l), \tag{2}$$

where $l \in [1, L]$ denote the current layer index. The computation process of self-attention module $\mathcal{S}(\cdot)$ is defined as:

$$\mathcal{S}(\boldsymbol{f}^l) = \mathrm{Attn}_{qk}^l \cdot \boldsymbol{V}^l,\ \mathrm{Attn}_{qk}^l = \mathrm{softmax}(\frac{\boldsymbol{Q}^l {\boldsymbol{K}^l}^\top}{\sqrt{d}}), \tag{3}$$

$$\boldsymbol{Q}^l = \boldsymbol{f}^l \boldsymbol{W}_q^l, \quad \boldsymbol{K}^l = \boldsymbol{f}^l \boldsymbol{W}_k^l, \quad \boldsymbol{V}^l = \boldsymbol{f}^l \boldsymbol{W}_v^l, \tag{4}$$

where $\boldsymbol{W}_q, \boldsymbol{W}_k, \boldsymbol{W}_v$ are linear projection matrices, and Attn_{qk} are self-attention maps. Similarly, given a set of textual descriptions $\{\boldsymbol{T}_i\}_{i=1}^{N_c}$, CLIP textual encoder extracts its textual embeddings $\boldsymbol{f}_t \in \mathbb{R}^{N_c\times d}$. Thus, the final open-vocabulary semantic segmentation map $\boldsymbol{M} \in \mathbb{R}^{H\times W}$ (or $\in \mathbb{R}^{N\times 1}$ if flattened) is:

$$\boldsymbol{M} = \underset{N_c}{\arg\max} \cos(\boldsymbol{f}^L, \boldsymbol{f}_t). \tag{5}$$

In addition, due to the residual connection (including the feedforward module) in the final residual layer often producing "noisy" segmentation maps (Lan et al. 2024a), the output of its self-attention module is typically used as the final output visual embeddings $\boldsymbol{f}^L$. The computation within the last visual residual layer is: $\boldsymbol{f}^L \leftarrow \mathcal{S}(\boldsymbol{f}^{L-1})$.

Training-free adaptation for OVSS. Omission of the residual connection in the final layer and reliance solely on $\mathcal{S}(\cdot)$ markedly enhance the perceptual granularity of visual embeddings. Consequently, existing research focuses on refining $\mathcal{S}(\cdot)$ to achieve finer-grained spatial alignment. These methods primarily aim to optimize the final self-attention matrices $\mathrm{Attn}_{q,k}^L \in \mathbb{R}^{N\times N}$ for more precise modeling of local spatial relationships among image patches. They fall into two main categories:

- **VFM-proxy.** These methods leverage the dense representation capabilities of powerful visual foundation models to enhance CLIP. ProxyCLIP (Lan et al. 2024b) utilizes DINO (Zhang et al. 2022)'s visual representations to compute self-similarity, replacing CLIP's original self-attention matrices. CASS (Kim et al. 2025) employs spectral graph distillation to integrate DINO's dense visual features, thereby strengthening CLIP's contextual coherence.
- **Self-proxy.** They construct novel self-attention matrices from their own embeddings, replacing the original

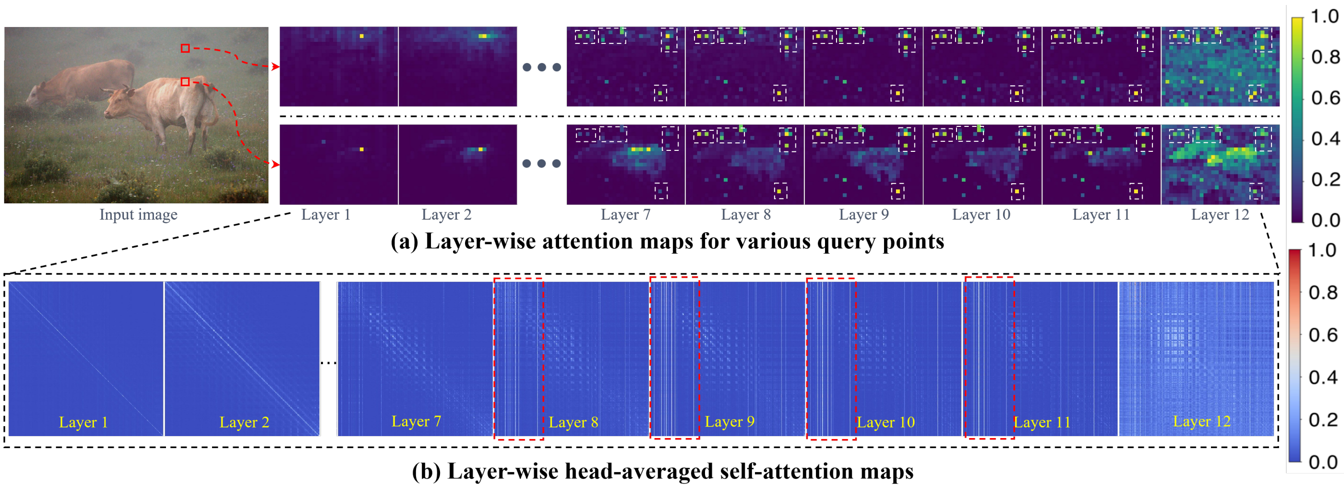


Figure 2: Illustration of "distraction" phenomenon. (a) Layer-wise attention maps for two query locations reveal query-to-tokens relevance across the entire image. (b) Layer-wise head-averaged self-attention maps characterize token-to-token relationships. Red solid boxes denote query points; white dashed boxes indicate distraction tokens; and red dashed boxes represent attention weights of distraction tokens.

$\mathrm{Attn}_{q,k}^{L}$. For instance, SCLIP (Wang, Mei, and Yuille 2024) utilizes the summation of Attn_{qq}^{L} and Attn_{kk}^{L} as the final self-attention matrix. ClearCLIP (Lan et al. 2024a) and NACLIP (Hajimiri, Ayed, and Dolz 2025) respectively substitute Attn_{qk}^{L} with Attn_{qq}^{L} and Attn_{kk}^{L}. Later we refer to Attn_{kk}-proxy CLIP as CLIP with Attn_{kk}^{L} replacing Attn_{qk}^{L}.

Despite the impressive achievements of the aforementioned paradigms, they exhibit a fundamental limitation: these approaches exclusively modulate the final residual attention layer (to prevent model collapse), while ignoring patch-level spatial misalignments within intermediate residual layers. This oversight causes accumulated error propagation across the network. This raises a critical question: Could direct correction of spatial misalignments in intermediate residual layers enhance CLIP's dense prediction performance? To this end, in the next section, we systematically analyze the self-attention matrices layer-wise to probe CLIP's underlying mechanisms from an interpretability perspective.

"Distraction" Phenomenon

In this section, we first delineate the "distraction" phenomenon, which reveals pronounced spatial misalignment artifacts within CLIP. Furthermore, our empirical analysis establishes that this phenomenon originates from excessive activation in specific dimensions. Building on this analysis, we precisely identify distraction tokens. Finally, we systematically evaluate diverse suppression strategies on these distraction tokens, thereby inspiring our proposed method.

Key observations. Figure 2 presents layer-wise attention maps $\mathrm{Attn}_{qk}^{l}[i]$ capturing token relevance to diverse query points i, alongside head-averaged self-attention maps revealing token-to-token relationships. Attention maps visualization (Figure 2 (a)) demonstrates that shallow layers (1-2) predominantly attend to query-relevant tokens. In contrast, deeper layers (7-12) exhibit numerous high-attention tokens unrelated to target queries (demarcated by white dashed boxes) - a phenomenon we term "distraction," with these tokens designated as distraction tokens $\mathcal{T}_{dis}$. The emergence of distraction tokens progressively diminishes saliency around query-related regions, defocusing originally concentrated target areas. Crucially, these distraction tokens consistently occupy identical spatial positions across two varying query points, perhaps suggesting their universal correlations with all queries points. Self-attention maps visualization (Figure 2 (b)) corroborate this behavior: distraction tokens manifest distinct vertical stripes (within red dashed boxes) due to uniformly high attention values between all tokens and them, i.e., their attention weights $\sum_j \mathrm{Attn}_{qk}^{l}[j,i]_{,i\in\mathcal{T}_{dis}}$ are large, reflecting the sum of the relevance of them to all tokens. Therefore, these tokens fundamentally induce spatial misalignment, propagating errors through residual layers, and ultimately constraining CLIP's dense prediction ability.

How do distraction tokens surface? Given distraction tokens' persistent high correlation with all tokens, we hypothesize they exhibit massive embedding weights. We compute visual dense embeddings averaged across residual layers as: $\bar{\boldsymbol{f}} = \frac{1}{L}\sum_{l=1}^{L}(\frac{\boldsymbol{f}^l}{\sum_{j=1}^{d}\boldsymbol{f}^l[:,j]}) \in \mathbb{R}^{N\times d}$, where L denotes total layers. The embedding weights of i-th token in j-th dimension can be defined as $\bar{\boldsymbol{f}}[i,j]$, representing its weights distribution across diverse dimensions. Figure 3 (a) displays the mean embedding weights averaged across the entire dataset, reflecting the weights distribution of the entire dataset in CLIP's high-dimensional space. We find that three large-scale OVSS benchmark datasets exhibit consistent weights distribution, with several peaks at identical dimensions (e.g., 4, 162, 474, etc., denoted red), which we designate as distraction dimensions $\mathcal{D}_{dis}$. This indicates CLIP intrinsically

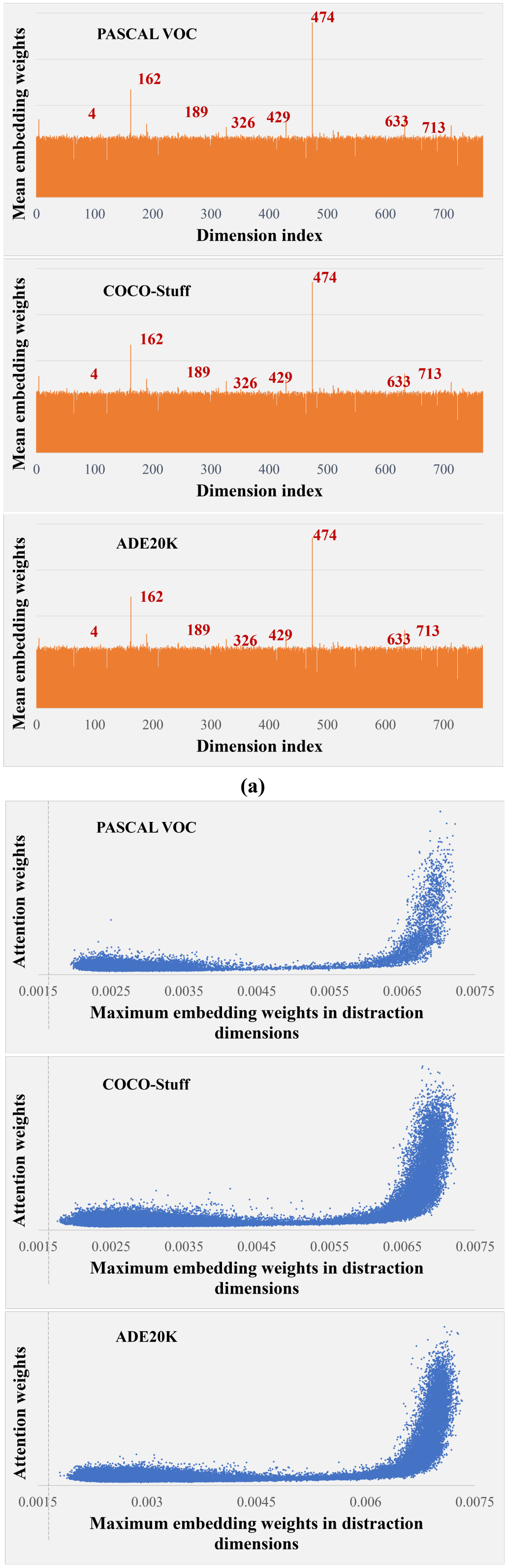


Figure 3: Weights analysis. (a) Histogram of per-dimension mean embedding weights averaged across the entire dataset. (b) Scatter plot of tokens' attention weights versus their maximum embedding weights in distraction dimensions.

generates massive embedding weights in $\mathcal{D}_{dis}$ — an inherent property independent of dataset specifics. Figure 3 (b) validates our hypothesis, plotting for each token i: Attention weights ($\sum_j \mathrm{Attn}[j,i]$[1]) and Maximum embedding weights ϕ_i in $\mathcal{D}_{dis}$ ($\phi_i = \max_{j \in \mathcal{D}_{dis}} \bar{\boldsymbol{f}}[i,j]$). We observe that tokens with massive ϕ_i consistently exhibit high attention weights, and that growth in ϕ_i causes exponential growth in attention weights. Therefore, we conclude that **tokens with massive maximum embedding weights in the distraction dimensions inevitably evolve into distraction tokens under self-attention calculation mechanisms**.

Where are distraction tokens? Building upon the above analysis, we define $\tau = 5/d$ as the identification threshold, where token i satisfying $\phi_i > \tau$ are identified as distraction tokens. As shown in Figure 4 (a), the scatter plot maps of two images show that above-threshold tokens exhibit high attention weights. Figure 4 (b) visually confirms these tokens' spatial distribution, revealing near-perfect alignment with observed distraction patterns. Therefore, distraction token identification reduces to a simple threshold operation on maximum embedding weights within $\mathcal{D}_{dis}$.

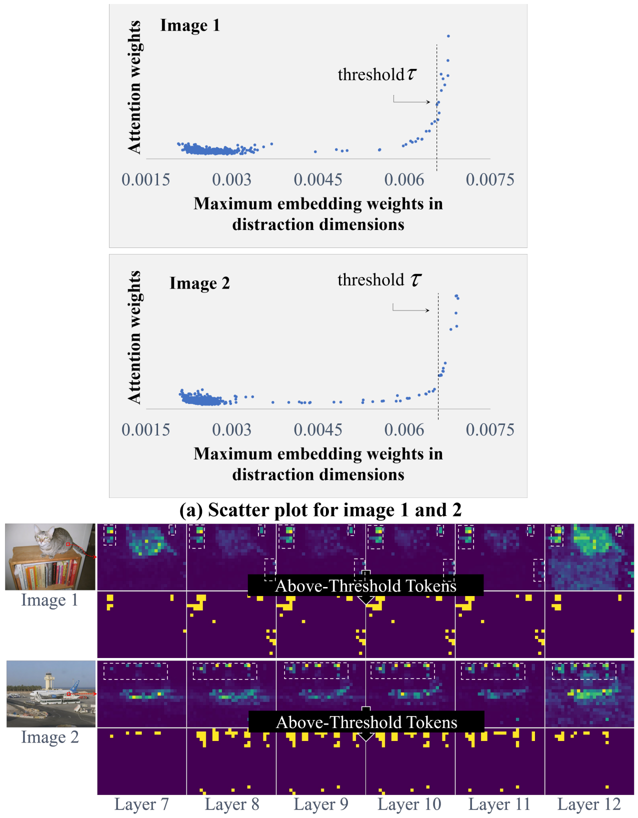
Suppression strategies on distraction tokens. To explore whether suppressing distraction tokens enhances CLIP's dense prediction capability, we establish an Attn_{kk}-proxy CLIP baseline and evaluate four suppression strategies: $-\infty$ masking[2] - nullifying layer-wise distraction tokens embeddings, low-pass filtering - clamping these embeddings to τ, mean filtering - replacing with mean of 3×3 local neighborhoods, and median filtering - substituting with median of 3×3 neighborhoods. As shown in Table 1, both $-\infty$ masking and low-passing filtering degrade performance, whereas mean and median filtering yield measurable improvements. This indicates that distraction tokens should maintain spatial consistency with adjacent regions: Direct elimination disrupts the topological structure of CLIP's high-dimensional space, inducing model collapse. Consequently, we are motivated to investigate whether reallocating attention resources from distraction tokens to defocused target regions can enhance CLIP's dense prediction capability.

Approach

Our objective is to improve CLIP's dense prediction performance by mitigating layer-wise spatial misalignment. Building on the above analysis, we introduce <u>Re</u><u>F</u>ocusing CLIP (RF-CLIP) – a training-free approach that simulates human distraction-refocusing behavior to redirect layer-wise scattered attention resources toward defocused target regions. As shown in Figure 5, RF-CLIP consists of three key components: (a) Distractor Localization, identifying attention-rich distraction tokens; (b) Defocus Localization, detecting attention-poor defocused tokens; (c) Weight Redistribution, refocusing distracted attention back to defocused target tokens. Below, we use the l-th residual layer as an illustrative example to elucidate these components.

[1]Here, $\mathrm{Attn} = \frac{1}{L \cdot H} \sum_{l,h=1}^{l,h=L,H} \mathrm{Attn}_{qk}^{l,h}$ denotes self-attention matrices averaged across all L layers and H heads.

[2]$-\infty$ forces softmax-attention value to zero



(a) Scatter plot for image 1 and 2

(b) Layer-wise attention maps and above-threshold tokens visualization

Figure 4: Distraction tokens identification. (a) Scatter plot of attention weights versus maximum embedding weights in $\mathcal{D}_{dis}$. (b) Layer-wise attention maps and above-threshold tokens visualizations.

Distractor localization. Given the input embedding $\boldsymbol{f}_i^l \in \mathbb{R}^d$ of the i-th visual token, we define its maximum embedding weight in the distraction dimensions can be defined as:

$$\phi_i^l = \max_{j \in \mathcal{D}_{dis}} \frac{\boldsymbol{f}_i^l[j]}{\sum_{k=1}^{d} \boldsymbol{f}_i^l[k]} \tag{6}$$

Using the threshold $\tau = 5/d$, we identify distraction tokens statisfying $\phi_i^l > \tau$.

Defocus localization. We conceptualize defocused tokens as foreground instances and formulate defocus localization as a bipartition graph cut problem. Adopting spectral clustering on the similarity matrix Attn_{kk}^l, we represent each token i as a node n_i, partitioning the graph into foreground $\mathcal{A}$ and background $\mathcal{B}$ sets. This optimization problem aims to minimize the normalized cut energy:

$$\frac{\mathcal{E}(\mathcal{A},\mathcal{B})}{\mathcal{E}(\mathcal{A},\mathcal{V})} + \frac{\mathcal{E}(\mathcal{A},\mathcal{B})}{\mathcal{E}(\mathcal{B},\mathcal{V})}, \quad \mathcal{V} = \mathcal{A} \cup \mathcal{B}, \tag{7}$$

where $\mathcal{E}(\cdot,\cdot)$ denotes set similarity. Using *key-key* attention Attn_{kk}^l as token-wise similarity measure, we define $\mathcal{E}(\mathcal{A},\mathcal{B}) = \sum_{n_i \in \mathcal{A}, n_j \in \mathcal{B}} \mathrm{Attn}_{kk}^l[i,j]$. Following Shi and Malik (2000), Equation 7 is optimized by solving:

Method	VOC21	COCO-Stuff	Cityscapes	ADE20k	Avg.
baseline	58.1	23.0	31.1	16.3	32.1
$-\infty$ masking	3.5 (-54.6)	0.1 (-22.9)	2.0 (-29.1)	0.1 (-16.2)	1.4
low-pass filtering	7.9 (-50.2)	1.1 (-21.9)	6.2 (-24.9)	1.4 (-14.9)	4.2
mean filtering	59.3 (+1.2)	24.0 (+1.0)	35.4 (+4.3)	18.2 (+1.9)	34.2
median filtering	58.6 (+0.5)	23.7 (+0.7)	34.5 (+3.4)	17.6 (+1.3)	33.6

Table 1: Quantitative evaluation for four suppression strategies (unit: %). Red and green fonts denote descending and ascending values. **Baseline**: Attn_{kk}-proxy CLIP.

$$\boldsymbol{y}_1^l = \mathop{\arg\min}_{\boldsymbol{y}^{l\top}\boldsymbol{D}\mathbf{1}=0} \frac{\boldsymbol{y}^{l\top}(\boldsymbol{D}^l - \mathrm{Attn}_{kk}^l)\boldsymbol{y}^l}{\boldsymbol{y}^{l\top}\boldsymbol{D}^l\boldsymbol{y}^l}, \tag{8}$$

where $\boldsymbol{y}_1^l \in \mathbb{R}^N$ denotes the Fiedler vector — the eigenvector corresponding to the second smallest eigenvalue of the generalized eigensystem $(\boldsymbol{D}^l - \mathrm{Attn}_{kk}^l)\boldsymbol{y}^l = \lambda\boldsymbol{D}^l\boldsymbol{y}^l$. Here, $\boldsymbol{D}$ is a diagonal matrix with $\sum_j \mathrm{Attn}_{kk}^l[:,j]$ on its diagonal. Thus, defocused tokens are identified as those satisfying $\boldsymbol{y}_1^l[i] > \frac{1}{N}\sum_{j=1}^{N}\boldsymbol{y}_1^l[j]$.

Weight redistribution. We formalize weight redistribution through two complementary mechanisms: (1) attention weight redistribution across spatial locations, and (2) embedding weight redistribution across embedding dimensions. Let $\mathcal{T}_{dis}$ and $\mathcal{T}_{def}$ denote the sets of distraction tokens and defocused tokens, respectively. The attention redistribution process transfers attention weights from $\mathcal{T}_{dis}$ to $\mathcal{T}_{def}$ while preserving original contribution distribution, thereby maintaining the topological structure of layer-wise self-attention matrices and preventing model collapse. This is achieved through two sequential operations. First. attention weights for distraction tokens are scaled down, with the decrement preserved as a redistribution budget Ω:

$$\mathrm{Attn}_{qk}^{l,h}[i,j] \leftarrow (1-\beta)\cdot\mathrm{Attn}_{qk}^{l,h}[i,j], {\scriptstyle \forall j \in \mathcal{T}_{dis}}, \tag{9}$$

$$\Omega[i] = \beta \cdot \sum_{j \in \mathcal{T}_{dis}} \mathrm{Attn}_{qk}^{l,h}[i,j], \tag{10}$$

where $\beta \in (0,1)$ is the attenuation factor. Second, the budget is distributed to defocused tokens proportional to their original attention weights:

$$\mathrm{Attn}_{qk}^{l,h}[i,j] \leftarrow \mathrm{Attn}_{qk}^{l,h}[i,j] + \Omega[i]\cdot\rho[i,j], {\scriptstyle \forall j \in \mathcal{T}_{def}}, \tag{11}$$

$$\rho[i,j] = \frac{\mathrm{Attn}_{qk}^{l,h}[i,j]}{\sum_{j \in \mathcal{T}_{def}} \mathrm{Attn}_{qk}^{l,h}[i,j]}, \tag{12}$$

where ρ represents the original contribution distribution determining each defocused token's allocation share. This process maintains column-normalization ($\sum_j \mathrm{Attn}_{qk}^{l,h}[i,j] = 1$) and preserves the original attention distribution, thus effectively mitigating model collapse while enhancing focus on defocused tokens. The embedding redistribution process uses a local 3×3 neighborhood centered on each distraction

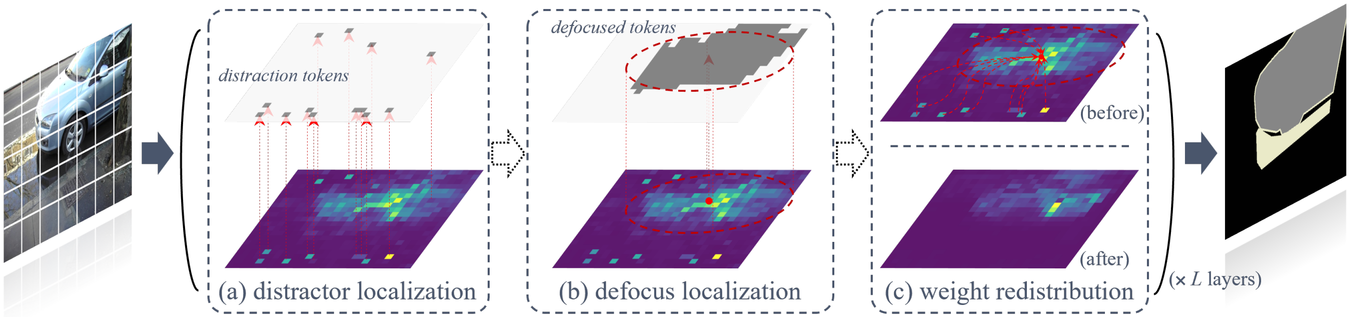


Figure 5: Overview of RF-CLIP. Our RF-CLIP directly correct layer-wise spatial misalignment in CLIP to enhance dense prediction capabilities. Each layer's correction mechanism comprises three key components: (a) distractor localization, identifying attention-rich distraction tokens irrelevant to target objects; (b) defocus localization, detecting attention-poor defocused tokens relevant to targets; (c) weight redistribution, reallocating attention resources from distraction tokens to defocused tokens.

tokens in $\mathcal{T}_{dis}$ to compute averaged embedding representations. For dimension $j \in \mathcal{D}_{dis}$, the embedding weights are updated through spatial averaging:

$$\boldsymbol{f}_i^l[j] = \frac{1}{8} \cdot \sum_{\hat{i} \in \mathcal{O}_i} \boldsymbol{f}_{\hat{i}}^l[j]_{,\forall j \in \mathcal{D}_{dis} \text{ and } i \in \mathcal{T}_{dis}} , \tag{13}$$

where $\mathcal{O}_i$ denote 3×3 neighboring regions centered token i. This process only adjusts the embedding in $\mathcal{D}_{dis}$ without destroying the distribution in normal dimensions, thus effectively ensuring the integrity of the original embeddings.

Dense prediction. Given the layer-wise spatial misalignment correction, we replace Attn_{qk}^L with the layer-averaged attention $\overline{\text{Attn}}_{kk} = \frac{1}{L} \sum_{l=1}^{L} \text{Attn}_{kk}^l$ to produce the final visual embeddings. The segmentation map is computed as:

$$\boldsymbol{M} = \arg\max_{N_c} \cos(\boldsymbol{f}^L, \boldsymbol{f}_t), \quad \boldsymbol{f}^L = \overline{\text{Attn}}_{kk} \cdot \boldsymbol{V}^L. \tag{14}$$

Please refer to the Appendix for the pseudocode of RF-CLIP.

Experiments

Experiments settings. Following the existing training-free works, we conduct evaluations on eight standard benchmark datasets: VOC21 (Everingham et al. 2012), VOC20 (Everingham et al. 2012), Context60 (Mottaghi et al. 2014), Context59 (Mottaghi et al. 2014), COCO-Suff (Caesar, Uijlings, and Ferrari 2018), COCO-Obj (Caesar, Uijlings, and Ferrari 2018), ADE20K (Zhou et al. 2019) and Cityscapes (Cordts et al. 2016). Performance is measured using mean *Intersection-over-Union* (mIoU). All experiments are implemented in PyTorch (Paszke 2019) with MMSegmentation (Contributors 2020). We employ CLIP models with ViT-B/16 and ViT-L/14 architectures, and set β to 0.7. Evaluation configurations include: sliding window inference with a 224×224 window and a 112×112 stride, input resolution resizing with a short side of 336 (560 for Cityscapes), and text descriptions $\{\boldsymbol{T}_i\}_{i=1}^{N_c}$ constructed by combining standard ImageNet prompts (Radford et al. 2021) with category names. See Appendix for more details.

System level comparison. To evaluate the performance of our proposed method, we compare it with unsupervised approaches, including both training-free and pre-training modes. As shown in Table 2, quantitative evaluation results demonstrate that: 1) Our method achieves the SOTA performance, outperforming same-baseline approaches by 1.6% mIoU and even surpassing methods incorporating additional visual foundational models (VFMs) by $+1.1\%$ mIoU; 2) While methods like FreeDa (Barsellotti et al. 2024) and CASS (Kim et al. 2025) achieve SOTA results on specific benchmarks, they exhibit some performance drops on others. In contrast, our approach maintains top-2 performance across all benchmarks, highlighting superior generalization and robustness; 3) Without utilizing the post-processing module, i.e., PAMR (Araslanov and Roth 2020), our method remains highly competitive and still achieves SOTA performance against same-baseline approaches.

Efficiency analysis. To verify the efficiency of our method, we conduct efficiency analysis on an NVIDIA RTX 3090 GPU. As shown in the Table 3, our approach achieves 46.2% higher mIoU while maintaining inference speed comparable to CLIP. Compared with ProxyCLIP (Lan et al. 2024b) which introduces an additional VFM, our method doubles the inference speed while improving mIoU by 5.7%. In summary, these results demonstrate that our method achieves the optimal performance-efficiency trade-off.

Components analysis. To validate the effectiveness of the proposed components, we adopt layer-averaged *key-key* attention $\overline{\text{Attn}}_{kk}$-proxy CLIP as the baseline (I), and then incrementally integrate our components to measure performance gains. As shown in Table 4, we first perform 3×3 mean filtering on 10 randomly-selected tokens (II), and then the performance drops by 1.3% mIoU compared to the baseline. In contrast, performing same operation on distraction tokens improves performance by 1.1% mIoU (III). These two controlled experiments establish the importance of distraction-aware processing. Next, we reallocate the abundant attention resources from distraction tokens to all non-distraction tokens (IV, V, VI), yielding progressive mIoU gains of $+2.1\%$, $+2.8\%$, and $+3.7\%$. This demon-

Model	Baseline	Additional VFM	VOC21	Context60	COCO-Obj	VOC20	Context59	COCO-Stuff	Cityscapes	ADE20K	Avg.
ReCo (Shin, Xie, and Albanie 2022)	CLIP ViT-B/16	MaskCLIP	25.1	19.9	15.7	57.7	22.3	14.8	21.6	11.2	23.5
LaVG (Kang and Cho 2024)	CLIP ViT-B/16	DINO	62.1	31.6	34.2	82.5	34.7	23.2	26.2	15.8	38.8
CLIP-DINOiser† (Wysoczańska et al. 2024)	CLIP ViT-B/16	DINO	62.1	32.4	34.8	80.9	35.9	24.6	31.7	20.0	40.3
FreeDa (Barsellotti et al. 2024)	CLIP ViT-B/16	DINOv2	-	-	-	85.6	**43.1**	**27.8**	36.7	**22.4**	-
ProxyCLIP‡ (Lan et al. 2024b)	CLIP ViT-B/16	DINO	59.1	35.2	36.2	78.2	38.8	26.2	38.1	19.6	41.4
CASS (Kim et al. 2025)	CLIP ViT-B/16	DINO	<u>65.8</u>	36.7	37.8	**87.8**	40.2	26.7	39.4	20.4	<u>44.4</u>
CLIP (Radford et al. 2021)	CLIP ViT-B/16	✗	18.6	7.8	6.5	49.1	11.2	7.2	6.7	3.2	13.8
MaskCLIP (Zhou, Loy, and Dai 2022)	CLIP ViT-B/16	✗	38.3	23.6	20.6	74.9	26.4	16.4	12.6	9.8	27.9
GroupViT (Xu et al. 2022)	CLIP ViT-B/16	✗	50.4	18.7	27.5	79.7	23.4	15.3	11.1	9.2	29.4
CLIPtrase (Shao et al. 2024)	CLIP ViT-B/16	✗	50.9	29.9	**43.6**	81.0	33.8	22.8	21.3	16.4	32.7
TCL (Cha, Mun, and Roh 2023)	CLIP ViT-B/16	✗	55.0	30.4	31.6	83.2	33.9	22.4	24.0	17.1	37.2
CLIPSurgery (Li et al. 2023)	CLIP ViT-B/16	✗	55.2	30.3	29.7	77.5	33.4	22.2	33.1	16.1	37.2
GEM (Bousselham et al. 2024)	CLIP ViT-B/16	✗	58.7	32.0	32.9	81.7	35.6	23.9	32.6	16.9	39.3
CaR (Sun et al. 2024)	CLIP ViT-B/16	✗	48.6	13.6	15.4	73.7	18.4	-	-	5.4	-
ClearCLIP (Lan et al. 2024a)	CLIP ViT-B/16	✗	51.8	32.6	33.0	80.9	35.9	23.9	30.0	16.7	38.1
SCLIP (Wang, Mei, and Yuille 2024)	CLIP ViT-B/16	✗	59.1	30.4	30.5	80.4	34.1	22.4	32.2	16.1	38.2
NACLIP (Hajimiri, Ayed, and Dolz 2025)	CLIP ViT-B/16	✗	58.9	32.2	33.2	79.7	35.2	23.3	35.5	17.4	39.4
SC-CLIP (Bai et al. 2024)	CLIP ViT-B/16	✗	64.6	<u>36.8</u>	37.7	84.3	40.1	26.6	<u>41.0</u>	20.1	43.9
RF-CLIP (Ours)	CLIP ViT-B/16	✗	64.8	36.4	37.9	87.0	39.8	26.3	41.3	20.4	44.2
with PAMR			**67.2**	**37.9**	<u>39.1</u>	<u>87.0</u>	<u>41.4</u>	<u>27.5</u>	**43.0**	<u>21.0</u>	**45.5**
ProxyCLIP‡ (Lan et al. 2024b)	CLIP ViT-L/14	DINO	58.1	34.1	37.4	82.0	37.3	25.5	38.1	21.2	41.7
CLIP (Radford et al. 2021)	CLIP ViT-L/14	✗	10.3	4.5	4.4	19.9	5.7	3.2	3.2	1.9	6.6
MaskCLIP (Zhou, Loy, and Dai 2022)	CLIP ViT-L/14	✗	24.8	9.7	10.2	30.1	13.0	9.0	12.1	7.1	14.5
SCLIP (Wang, Mei, and Yuille 2024)	CLIP ViT-L/14	✗	44.4	22.3	24.9	70.6	25.2	16.5	21.3	10.9	29.5
GEM (Bousselham et al. 2024)	CLIP ViT-L/14	✗	45.2	25.5	28.3	83.7	28.1	19.2	27.1	13.2	33.8
CLIPSurgery (Li et al. 2023)	CLIP ViT-L/14	✗	47.9	27.3	28.1	84.3	31.0	21.4	29.7	17.3	35.9
PnP-OVSS (Luo et al. 2024)	CLIP ViT-L/14	✗	-	-	36.2	51.3	28.0	17.9	-	14.2	-
NACLIP (Hajimiri, Ayed, and Dolz 2025)	CLIP ViT-L/14	✗	52.1	28.7	29.9	78.6	32.1	21.4	31.4	17.3	36.4
ClearCLIP (Lan et al. 2024a)	CLIP ViT-L/14	✗	48.6	28.0	28.6	84.8	31.5	21.2	32.1	16.9	36.5
SC-CLIP (Bai et al. 2024)	CLIP ViT-L/14	✗	<u>65.0</u>	<u>36.9</u>	<u>40.5</u>	<u>88.3</u>	<u>40.6</u>	<u>26.9</u>	<u>41.3</u>	<u>21.7</u>	<u>45.2</u>
RF-CLIP (Ours)	CLIP ViT-L/14	✗	65.8	36.7	41.8	89.1	40.2	26.7	41.4	22.4	45.4
with PAMR			**68.1**	**38.1**	**42.0**	**89.1**	**40.8**	**27.8**	**43.0**	**23.4**	**46.5**

Table 2: Quantitative evaluation on standard benchmarks (unit: %). †: Re-implementation with OpenAI's weights. ‡: Re-implementation with its DINO-B/16 variant. ***PAMR***: Pixel-adaptive mask refinement (Araslanov and Roth 2020). Here, the best results are shown in bold and the second-best results are underlined.

Model	FLOPs(G)↓	Params(M)↓	Speed(FPS)↑	mIoU(%)↑
baseline	**16.7**	149.6	**12.7**	58.1
CLIP	17.4	149.6	12.0	18.6
ProxyCLIP	34.1	235.4	6.1	<u>59.1</u>
RF-CLIP (Ours)	<u>17.1</u>	149.6	<u>12.0</u>	**64.8**

Table 3: Efficiency analysis on VOC21 benchmark. **Baseline**: Attn_{kk}-proxy CLIP as the same in Table 1.

	Components	VOC21	COCO-Stuff	Cityscapes	ADE20K	Avg.
(I)	baseline	59.1	23.6	32.1	16.9	32.9
(II)	mean filtering	58.8	21.4	31.6	14.7	31.6
(III)	**(II)** + Dis Loc.	60.3	24.4	33.6	17.5	34.0
(IV)	**(III)** + Attn Red.	61.5	24.8	35.3	18.3	35.0
(V)	**(III)** + Embed Red.	62.1	25.2	36.7	18.9	35.7
(VI)	**(IV)** + **(V)**	63.2	25.4	38.5	19.3	36.6
(VII)	**(VI)** + Def Loc.	**64.8**	**26.3**	**41.3**	**20.4**	**38.2**

Table 4: Effect of different components (unit: %). **Baseline**: layer-averaged *key-key* attention $\overline{\mathrm{Attn}}_{kk}$-proxy CLIP. **Dis Loc.**: Distractor localization. **Attn Red.**: Attention weight redistribution. **Embed Red.**: Embedding weight redistribution. **Def Loc.**: Defocus localization.

strates the importance of reallocating the abundant resources occupied by distraction tokens. Finally, defocus localization (VII) directs these liberated resources to target regions, providing an additional $+1.6\%$ mIoU improvement. Cumulatively, our components deliver $+1.1\%$, $+2.6\%$, and $+1.6\%$ mIoU gains per stage (total $+5.3\%$ mIoU), confirming their effectiveness.

Conclusion

Our findings reveal that shallow-layer features exhibit strong spatial coherence and high target saliency; however, in deep layers, distraction tokens emerge that consume significant attention resources, thereby disrupting spatial coherence and diverting attention from highly salient target regions. We designate this as the "distraction" phenomenon. Further experiments demonstrate that these distraction tokens originate from excessive activation in specific dimensions; based on this insight, we effectively localize distraction tokens and achieve performance improvements by filtering them out. Inspired by this, we propose RF-CLIP, a training-free approach that redistributes attention resources from distraction tokens back to target regions. Comprehensive experimental evaluations demonstrate significant improvements in dense prediction accuracy and inference speed.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grant 62176224, 62176092, 62222602, 62306165, 62172234, 62306165, and 62376233, in part by Science and Technology on Sonar Laboratory under grant 2024-JCJQ-LB-32/07, in part by Xiaomi Young Talents Program award, in part by Natural Science Foundation of Shanghai under Grant 23ZR1420400; and in part by Natural Science Foundation of Chongqing under Grant CSTB2023NSCQ-JQX0007.

References

Araslanov, N.; and Roth, S. 2020. Single-stage semantic segmentation from image labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4253–4262.

Bai, S.; Liu, Y.; Han, Y.; Zhang, H.; and Tang, Y. 2024. Self-calibrated clip for training-free open-vocabulary segmentation. *arXiv preprint arXiv:2411.15869*.

Barsellotti, L.; Amoroso, R.; Cornia, M.; Baraldi, L.; and Cucchiara, R. 2024. Training-free open-vocabulary segmentation with offline diffusion-augmented prototype generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3689–3698.

Bousselham, W.; Petersen, F.; Ferrari, V.; and Kuehne, H. 2024. Grounding everything: Emerging localization properties in vision-language transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3828–3837.

Caesar, H.; Uijlings, J.; and Ferrari, V. 2018. COCO-Stuff: Thing and stuff classes in context. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE conference on*. IEEE.

Cha, J.; Mun, J.; and Roh, B. 2023. Learning to generate text-grounded mask for open-world semantic segmentation from only image-text pairs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11165–11174.

Cho, S.; Shin, H.; Hong, S.; Arnab, A.; Seo, P. H.; and Kim, S. 2024. Cat-seg: Cost aggregation for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4113–4123.

Contributors, M. 2020. MMSegmentation: OpenMMLab Semantic Segmentation Toolbox and Benchmark. https://github.com/open-mmlab/mmsegmentation.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223.

Everingham, M.; Van Gool, L.; Williams, C. K. I.; Winn, J.; and Zisserman, A. 2012. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results.

Hajimiri, S.; Ayed, I. B.; and Dolz, J. 2025. Pay attention to your neighbours: Training-free open-vocabulary semantic segmentation. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 5061–5071. IEEE.

Jiao, S.; Wei, Y.; Wang, Y.; Zhao, Y.; and Shi, H. 2023. Learning mask-aware clip representations for zero-shot segmentation. *Advances in Neural Information Processing Systems*, 36: 35631–35653.

Kang, D.; and Cho, M. 2024. In defense of lazy visual grounding for open-vocabulary semantic segmentation. In *European Conference on Computer Vision*, 143–164. Springer.

Kim, C.; Ju, D.; Han, W.; Yang, M.-H.; and Hwang, S. J. 2025. Distilling spectral graph for object-context aware open-vocabulary semantic segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 15033–15042.

Lan, M.; Chen, C.; Ke, Y.; Wang, X.; Feng, L.; and Zhang, W. 2024a. Clearclip: Decomposing clip representations for dense vision-language inference. In *European Conference on Computer Vision*, 143–160. Springer.

Lan, M.; Chen, C.; Ke, Y.; Wang, X.; Feng, L.; and Zhang, W. 2024b. Proxyclip: Proxy attention improves clip for open-vocabulary segmentation. In *European Conference on Computer Vision*, 70–88. Springer.

Li, J.; Lu, Y.; Xie, Y.; and Qu, Y. 2024. Relationship prompt learning is enough for open-vocabulary semantic segmentation. *Advances in Neural Information Processing Systems*, 37: 74298–74324.

Li, Y.; Wang, H.; Duan, Y.; and Li, X. 2023. Clip surgery for better explainability with enhancement in open-vocabulary tasks. *arXiv e-prints*, arXiv–2304.

Luo, J.; Khandelwal, S.; Sigal, L.; and Li, B. 2024. Emergent open-vocabulary semantic segmentation from off-the-shelf vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4029–4040.

Mottaghi, R.; Chen, X.; Liu, X.; Cho, N.-G.; Lee, S.-W.; Fidler, S.; Urtasun, R.; and Yuille, A. 2014. The Role of Context for Object Detection and Semantic Segmentation in the Wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Paszke, A. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PmLR.

Shao, T.; Tian, Z.; Zhao, H.; and Su, J. 2024. Explore the potential of clip for training-free open vocabulary semantic segmentation. In *European Conference on Computer Vision*, 139–156. Springer.

Shi, J.; and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8): 888–905.

Shin, G.; Xie, W.; and Albanie, S. 2022. Reco: Retrieve and co-segment for zero-shot transfer. *Advances in Neural Information Processing Systems*, 35: 33754–33767.

Sun, S.; Li, R.; Torr, P.; Gu, X.; and Li, S. 2024. Clip as rnn: Segment countless visual concepts without training endeavor. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13171–13182.

Wang, F.; Mei, J.; and Yuille, A. 2024. Sclip: Rethinking self-attention for dense vision-language inference. In *European Conference on Computer Vision*, 315–332. Springer.

Wu, S.; Zhang, W.; Xu, L.; Jin, S.; Li, X.; Liu, W.; and Loy, C. C. 2023. Clipself: Vision transformer distills itself for open-vocabulary dense prediction. *arXiv preprint arXiv:2310.01403*.

Wu, S.; Zhang, W.; Xu, L.; Jin, S.; Liu, W.; and Loy, C. C. 2024. Clim: Contrastive language-image mosaic for region representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 6117–6125.

Wysoczańska, M.; Siméoni, O.; Ramamonjisoa, M.; Bursuc, A.; Trzciński, T.; and Pérez, P. 2024. CLIP-DINOiser: Teaching CLIP a few DINO tricks for open-vocabulary semantic segmentation. In *European Conference on Computer Vision*, 320–337. Springer.

Xu, J.; De Mello, S.; Liu, S.; Byeon, W.; Breuel, T.; Kautz, J.; and Wang, X. 2022. Groupvit: Semantic segmentation emerges from text supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 18134–18144.

Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L. M.; and Shum, H.-Y. 2022. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*.

Zhou, B.; Zhao, H.; Puig, X.; Xiao, T.; Fidler, S.; Barriuso, A.; and Torralba, A. 2019. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127: 302–321.

Zhou, C.; Loy, C. C.; and Dai, B. 2022. Extract free dense labels from clip. In *European Conference on Computer Vision*, 696–712. Springer.