CP-CLIP: Customized Parameter Generation for Open-vocabulary Semantic Segmentation

Zelin Peng[1], **Zhengqin Xu**[1], **Feilong Tang**[2], **Wei Shen**[1*]

[1]MoE Key Lab of Artifcial Intelligence, AI Institute, Shanghai Jiao Tong University
[2]Mohamed bin Zayed University of Artificial Intelligence

Abstract

Open-vocabulary semantic segmentation aims to assign pixel-level labels to images based on textual descriptions, even for categories beyond predefined closed sets. While vision-language foundation models like CLIP are widely used for this task, fine-tuning them for pixel-level predictions often compromises their generalization capabilities. To address this, we propose a novel fine-tuning strategy, CP-CLIP, which generates customized parameters for CLIP without sacrificing its generalization. Our method employs a customized parameter generator that produces newly added parameters based on random noise, using local visual features from CLIP's image encoder as conditions, enabling generalization to new images from unseen scenarios. Additionally, we introduce an orthogonal adaptation technique to ensure the update direction is orthogonal to the pre-trained weights, largely preserving the initial generalization ability. Extensive experiments demonstrate that CP-CLIP achieves state-of-the-art performance across multiple benchmarks in open-vocabulary semantic segmentation.

Introduction

The objective of open-vocabulary semantic segmentation is to create a segmentation model that can assign pixel-level labels to images based on textual descriptions, even for categories beyond a predefined closed set. Vision-language foundation models (Tan and Bansal 2019; Lu et al. 2019; Zhao et al. 2024; Tang et al. 2025; Tang et al.), especially CLIP (Radford et al. 2021), are commonly utilized to enable open-vocabulary recognition. Thus, open-vocabulary semantic segmentation essentially involves adapting these vision-language foundation models, which are initially trained with image-level supervision, to perform pixel-level predictions.

Existing studies (Yu et al. 2023; Xie et al. 2023; Cho et al. 2024; Xu et al. 2023b; Shao et al. 2024; Zhang et al. 2023; Xu et al. 2022a; Wu et al. 2024; Yang et al. 2025a,b,c; Peng et al. 2025b,a) often fine-tune CLIP on closed-set datasets with segmentation annotations, e.g., COCO-stuff (Caesar, Uijlings, and Ferrari 2018), to enhance its segmentation capabilities. A common belief is that freezing CLIP's text encoder helps preserve its robust text embeddings for a wide range of classes, which is thought to improve generalization on open sets (Li et al. 2022; Liang et al. 2023; Xu et al. 2023b). However, since the pre-trained text encoder is optimized for image-level classification, it is not aware of the dense knowledge for pixel-level recognition, leading to degraded segmentation performance. Recent works (Cho et al. 2024; Xie et al. 2023) attempt to fine-tune the text encoder with a small learning rate, balancing the preservation of CLIP's generalization capabilities with improved segmentation performance compared to former approaches. However, fine-tuning CLIP's text encoder remains a double-edged sword: while it significantly enhances segmentation performance, it risks undermining the model's pre-trained generalization capabilities. This occurs because CLIP's text encoder is fine-tuned on specific classes, thus its parameters after fine-tuning, cannot adapt to generalizable features for recognizing unseen classes. Faced with this dilemma, we pose a question by the light of nature: Is there a way to fine-tune CLIP without sacrificing its generalization capabilities?

In this paper, to offer a solution to this question, we propose two principles for fine-tuning CLIP: (1) the pre-trained weights should remain frozen, and the newly added parameters for CLIP's text encoder should be customized for each image, enabling generalization to new images from unseen classes; and (2) the update direction of the newly added parameters should be orthogonal to the pre-trained parameters, thereby largely preserving the initial generalization capabilities. To achieve this, we draw inspiration from recent methods for neural network parameter generation (Ha, Dai, and Le 2017; Wang et al. 2024; Jin et al. 2024), which generate parameters using diffusion models (Rombach et al. 2022). As a result, the model's parameters are not fixed after training on specific datasets. By conditioning on different scenarios or tasks, the model can generate distinct parameters from random noise, enabling generalization to new scenarios.

Based on this motivation, we propose a novel fine-tuning strategy, named CP-CLIP, for generating customized parameters for CLIP. CP-CLIP achieve the newly added parameters $\Delta\boldsymbol{\omega}$ by adopting a low-rank adaptation (LoRA)-like (Hu et al. 2022) fine-tuning framework to reduce computation cost, i.e., $\Delta\boldsymbol{\omega} = \boldsymbol{\alpha} \cdot \boldsymbol{\beta}$, where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are low-rank weights. Then, we introduce a customized parameter generator that takes random noise as input to generate the value of $\boldsymbol{\alpha}$. Specifically, the generator consists of multiple layers,

*Corresponding Author: Wei Shen

each containing a cross-attention block and a feed-forward network. The random noise and local visual features, extracted from the final layer of CLIP's vision encoder, interact within the cross-attention block as conditions. Subsequently, a linear projection is applied to derive the value of $\boldsymbol{\alpha}$. Finally, the value of $\boldsymbol{\beta}$ is orthogonally parameterized and fixed to ensure that $\Delta\boldsymbol{\omega}$ maintains an orthogonal direction relative to the pre-trained weights of CLIP's text encoder. Interestingly, the behavior of our method aligns with existing multi-modal large language models (Alayrac et al. 2022; Chen et al. 2022; Dai et al. 2023; Liu et al. 2023; Ma et al. 2024) designed for visual question-answering tasks (Lu et al. 2022; Goyal et al. 2017). Instead of focusing on specific classes, these models dynamically inject visual features into large language models, enhancing their generalization ability to address zero-shot vision-related problems. Additionally, our method can be viewed as a form of test-time adaptation (Wang et al. 2020, 2022), as it dynamically adjusts parameters to adapt to new images during inference. Extensive experiments demonstrate that CP-CLIP achieves state-of-the-art performance in open-vocabulary semantic segmentation across multiple benchmarks.

Related Work

Open-vocabulary Semantic Segmentation

Previous research on open-vocabulary semantic segmentation often relies on CLIP (Radford et al. 2021) as a foundational model. Initial approaches (Yu et al. 2023; Xie et al. 2023; Cho et al. 2024; Xu et al. 2023b; Shao et al. 2024; Zhang et al. 2023), such as (Zhou, Loy, and Dai 2022), directly fine-tune CLIP on standard segmentation datasets like COCO (Caesar, Uijlings, and Ferrari 2018). However, these methods face criticism because fully fine-tuning CLIP's encoder can significantly reduce its ability to generalize to unseen classes. To address this, alternative methods (Ghiasi et al. 2022; Ding et al. 2022; Xu et al. 2022b, 2023a) adopt a different strategy: they freeze CLIP to preserve its generalization capabilities and instead fine-tune an additional mask generator (Cheng et al. 2022) for segmentation. Despite this, the frozen parameter space in such approaches often results in a lack of segmentation awareness, causing misalignment between image regions and textual descriptions (Liang et al. 2023). More advanced techniques (Yu et al. 2023; Xu et al. 2023b; Cho et al. 2024) propose selectively fine-tuning specific parameters, such as certain layers of CLIP, to enable pixel-level predictions while keeping most of CLIP's parameters fixed. Although these methods show promise, their reliance on very small learning rates to minimize deviations from the pre-trained CLIP can constrain segmentation performance. In summary, while fine-tuning typically improves segmentation performance, it often compromises CLIP's generalization capabilities. To address this, we propose a novel fine-tuning pipeline that generates customized parameters for CLIP's text encoder, dynamically controlled by individual images, thereby preserving its generalization ability when encountering new scenarios.

Neural Network Parameter Generation

Neural network parameter generation (Ha, Dai, and Le 2017; Wang et al. 2024) represents a significant advancement in artificial intelligence-generated content. As an early pioneer in the deep learning era, HyperNetworks (Ha, Dai, and Le 2017) is proposed to generate parameters for various architectures of larger networks, often applied to fine-tune models for personalized downstream tasks, such as subject-driven generation (Ruiz et al. 2023). With the emergence of diffusion models, e.g., latent diffusion model (Rombach et al. 2022) (LDM), recent works (Erkoç et al. 2023; Peebles et al. 2022) have leveraged diffusion models for neural network parameter generation. For example, Tina (Peebles et al. 2022) introduces text-controlled parameter generation methods, broadening the scope of personalized model adaptation. However, the diffusion process typically requires manually pre-collecting a large set of parameters optimized through traditional back-propagation as targets, which is time-consuming and limits its broader applicability. In this paper, we simplify the diffusion process of LDMs and propose a customized parameter generator to equip CLIP with segmentation ability, largely preserving CLIP's open-set recognition ability for open-vocabulary semantic segmentation.

Preliminary Background

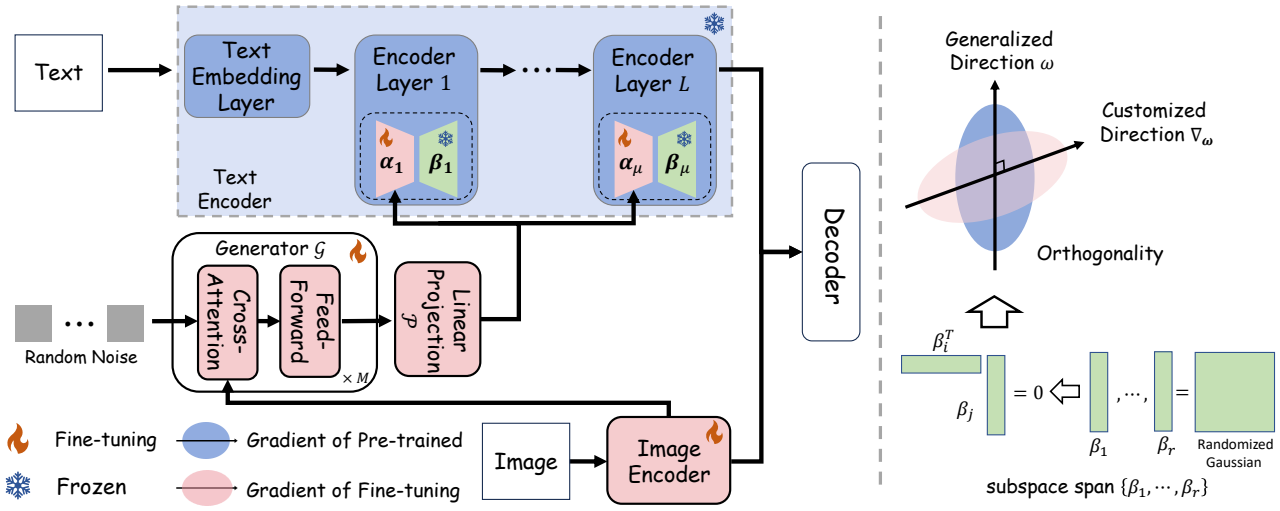
In this section, we first describe the learning paradigm for fine-tuning CLIP (Radford et al. 2021), followed by an overview of one of the most widely used fine-tuning paradigms: parameter-efficient fine-tuning, upon which our framework is built.

Fine-tuning CLIP. Given a visual-language model $\mathcal{F}$, e.g., CLIP (Radford et al. 2021), the goal of fine-tuning CLIP $\mathcal{F}(\mathbf{I};\boldsymbol{\omega})$ is to adapt it to a new downstream task, e.g., open-vocabulary semantic segmentation, where $\mathbf{I}$ is an input image from a dataset of the new task and $\boldsymbol{\omega} \in \boldsymbol{\Omega}_l$ denotes the trainable parameters, where $\boldsymbol{\Omega}_l$ refers to the parameter set of $\mathcal{F}$. Accordingly, the objective function of fine-tuning is formulated as:

$$\boldsymbol{\omega}^* = \arg\min_{\boldsymbol{\omega}} \mathcal{L}(\mathbf{I}, \mathbf{Y}), \tag{1}$$

where $\mathbf{Y}$ is a full dense label map. For segmentation tasks, the loss function $\mathcal{L}$ is commonly selected as a cross-entropy loss. To preserve CLIP's pre-trained generalization capabilities, we draw inspiration from parameter-efficient fine-tuning (PEFT) paradigms, which typically introduce trainable parameters $\Delta\boldsymbol{\omega}$ as the trainable parameters and keep the $\boldsymbol{\omega}$ frozen. The updated parameters are then computed as $\boldsymbol{\omega}^* = \boldsymbol{\omega} + \Delta\boldsymbol{\omega}$ through optimization. Here, we briefly review LoRA (Hu et al. 2022), a representative PEFT method for CLIP adaptation, as our approach builds upon it.

LoRA. LoRA (Hu et al. 2022) introduced a learnable low-rank matrix $\Delta\boldsymbol{\omega}$ that works in parallel to the original weight matrix $\boldsymbol{\omega} \in \mathbb{R}^{D\times U}$, which is frequently associated with the parameter matrix in the self-attention block of each transformer layer. $\Delta\boldsymbol{\omega}$ is derived by a matrix factorization decomposition, denoted as $\Delta\boldsymbol{\omega} = \boldsymbol{\alpha} \cdot \boldsymbol{\beta}$, where $\boldsymbol{\alpha} \in \mathbb{R}^{D\times r}$, $\boldsymbol{\beta} \in \mathbb{R}^{r\times U}$. $r \ll \min\{D, U\}$.



(a) Customized Parameter Generator (b) Orthogonal Adaptation

Figure 1: Overview of the proposed CP-CLIP framework. (a) Illustration of the pipeline of our customized parameter generator. We use μ random noises with local visual features obtained from CLIP's image encoder as conditions to generate $\{\boldsymbol{\alpha_i}\}_{i=1}^{\mu}$ values via a generator $\mathcal{G}$ for updating the entire parameter space of CLIP's text encoder. Then, M orthogonal subspace spans $\{\beta_1, \cdots, \beta_r\}$ are employed to transform these $\boldsymbol{\alpha}$ values into customized parameters $\Delta\boldsymbol{\omega}$. (b) Orthogonal adaptation introduces a simple method to achieve approximate orthogonality. It ensures that the orthogonal subspace spans $\{\beta_1, \cdots, \beta_r\}$ prevent interference between the update direction of customized parameters $\Delta\boldsymbol{\omega}$ and the update direction of pre-trained parameters $\boldsymbol{\omega}$.

Methodology

Fig. 1 illustrates the proposed CP-CLIP framework, which consists of two core components: (1) a customized parameter generator and (2) orthogonal adaptation. To obtain the $\Delta\boldsymbol{\omega} = \boldsymbol{\alpha} \cdot \boldsymbol{\beta}$ in CP-CLIP, the customized parameter generator leverages local visual features from CLIP's image encoder as conditions to generate $\boldsymbol{\alpha}$. Orthogonal adaptation initializes $\boldsymbol{\beta}$ using orthogonal parameterization and keeps it fixed, ensuring that the update direction of $\Delta\boldsymbol{\omega}$ is orthogonal to the original $\boldsymbol{\omega}$, preserving semantic information in CLIP's text encoder and maintaining its generalization ability.

Customized Parameter Generator

Inspired by the denoising U-Net architecture in latent diffusion model (LDM) (Rombach et al. 2022), which generates images from random noise by incorporating external knowledge (e.g., images) as conditions, we adapt this concept to generate customized parameters for CLIP's text encoder tailored to each image. Specifically, this generator $\mathcal{G}$ comprises M layers, each featuring a cross-attention block (`Attn`) and a feed-forward network (`FFN`), to generate $\boldsymbol{\alpha} \in \mathbb{R}^{D \times r}$. Given an input random noise $\mathbf{x} \in \mathbb{R}^{D}$ and local visual features—specifically, N patch tokens with D feature dimensions, $\mathbf{C} \in \mathbb{R}^{N \times D}$—from the image encoder $\mathcal{V}$, the random noise and local visual features interact within the cross-attention block. Similar to LDM, we designate the random noise as queries and the local visual features as keys and values. Formally, this process is formulated as:

$$\texttt{Attn}(\mathbf{x}^q, \mathbf{C}^k, \mathbf{C}^v) = \texttt{softmax}\left(\frac{\mathbf{x}^q \mathbf{C}^{k^T}}{\sqrt{D}}\right)\mathbf{C}^v, \quad (2)$$

where $\mathbf{x}^q = \mathbf{x}\boldsymbol{\omega}_Q$, $\mathbf{C}^k = \mathbf{C}\boldsymbol{\omega}_K$, and $\mathbf{C}^v = \mathbf{C}\boldsymbol{\omega}_V$. Here, $\boldsymbol{\omega}_Q \in \mathbb{R}^{D \times U}$, $\boldsymbol{\omega}_K \in \mathbb{R}^{D \times U}$ and $\boldsymbol{\omega}_V \in \mathbb{R}^{D \times U}$ are learnable weights. U represents the dimension of each attention head. Then, the output features from h attention heads are concatenated and linearly transformed using a learnable weight $\boldsymbol{\omega}_O \in \mathbb{R}^{D \times D}$ to generate the input of `FFN`. The entire forward process of $\mathcal{G}$ is summarized as:

$$\mathbf{x}_m = \mathcal{G}_m(\mathbf{x}_{m-1}, \mathbf{C}), m = 1, 2, ..., M, \quad (3)$$

where $\mathbf{C} = \mathcal{V}(\mathbf{I})$, with $\mathbf{I}$ being an input image fed into $\mathcal{V}$, and $\mathcal{G}_m = \texttt{Attn}_m \circ \texttt{FFN}_m$. After that, $\mathbf{x}_m$ is transformed through a linear projection $\mathcal{P}$ to obtain $\boldsymbol{\alpha}$ in our proposed CP-CLIP for fine-tuning.

Notably, updating the entire parameter space of CLIP's text encoder requires different variations of newly added weights for distinct locations, e.g., shallow and deep layers. However, a single input random noise can only produce one $\boldsymbol{\alpha}$ for updating $\Delta\boldsymbol{\omega}$, resulting in identical weights across all locations. To address this, we initialize a set of random noise $\mathbf{X}_0 \in \mathbb{R}^{\mu M \times D}$, where μ represents the number of locations requiring newly added weights in each layer. Then, we can rewrite Eq. (3) as:

$$\mathbf{X}_m = \mathcal{G}_m(\mathbf{X}_{m-1}, \mathbf{C}), m = 1, 2, ..., M. \quad (4)$$

Finally, we utilize the $\mathcal{P}$ to generate distinct $\boldsymbol{\alpha}$ values for updating the entire parameter space of CLIP's text encoder.

Orthogonal Adaptation

In this section, we begin by analyzing the update mechanism of CLIP's text encoder. Building on this analysis, we introduce a simple method to ensure that the update direction of $\Delta\boldsymbol{\omega}$ remains orthogonal to the original parameter space defined by $\boldsymbol{\omega}$.

Analysis. Without loss of generality. We first detail the updating process of CLIP's text encoder as follows:

$$\mathbf{h} = \boldsymbol{\omega}^*\mathbf{t} = (\boldsymbol{\omega} + \Delta\boldsymbol{\omega})\mathbf{t}, \tag{5}$$

where $\mathbf{t}$ is a text feature embedding derived from the input text description, and $\mathbf{h}$ is the output text embedding. Notably, the introduced learnable parameters $\Delta\boldsymbol{\omega}$ often influence the pre-trained parameters $\boldsymbol{\omega}$, and thus lead to the drop of generalization capabilities. In CP-CLIP, although the customized parameters $\boldsymbol{\alpha}$, generated via Eq. (4) and integrated into CLIP's text encoder, enable adaptation to unseen scenarios, there is no explicit constraint on how $\boldsymbol{\alpha}$ adapts. Similarly, this lack of constraint may lead to conflicts between the newly added customized parameters and the pre-trained parameters, potentially compromising the generalization capabilities of $\boldsymbol{\omega}^*$. To address this issue, we introduce an orthogonal adaptation strategy to minimize interference between the customized parameter $\boldsymbol{\alpha}$ and the pre-trained parameter $\boldsymbol{\omega}$. Specifically, we begin by analyzing the gradient of $\boldsymbol{\omega}$ to illustrate the derivation process, which is expressed as follows:

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\omega}^*} = \frac{\partial\mathcal{L}}{\partial\mathbf{h}}\frac{\partial\mathbf{h}}{\partial\boldsymbol{\omega}^*} = \frac{\partial\mathcal{L}}{\partial\mathbf{h}}\mathbf{t}^T, \tag{6}$$

where $\mathcal{L}$ is a loss function. The change of $\boldsymbol{\omega}^*$ can then be obtained as:

$$\bigtriangledown_{\boldsymbol{\omega}}\boldsymbol{\omega}^* = -\eta\frac{\partial\mathcal{L}}{\partial\mathbf{h}}\mathbf{t}^T, \tag{7}$$

where η is the learning rate. For our proposed CP-CLIP, when updating the customized parameter $\boldsymbol{\alpha}$ in Eq. (5), the gradient of $\boldsymbol{\alpha}$ is computed as:

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\alpha}} = \frac{\partial\mathcal{L}}{\partial\mathbf{h}}\frac{\partial\mathbf{h}}{\partial\boldsymbol{\alpha}} = \frac{\partial\mathcal{L}}{\partial\mathbf{h}}\mathbf{t}^T\boldsymbol{\beta}^T. \tag{8}$$

In analogy, the change of $\boldsymbol{\omega}^*$ can be rewritten as follows:

$$\begin{aligned}\bigtriangledown_{\boldsymbol{\alpha}}\boldsymbol{\omega}^* &= -\eta\frac{\partial\mathcal{L}}{\partial\boldsymbol{\alpha}}\boldsymbol{\beta} = -\eta\frac{\partial\mathcal{L}}{\partial\mathbf{h}}\mathbf{t}^T\boldsymbol{\beta}^T\boldsymbol{\beta} \\ &= \bigtriangledown_{\boldsymbol{\omega}}\boldsymbol{\omega}^*\boldsymbol{\beta}^T\boldsymbol{\beta}.\end{aligned} \tag{9}$$

According to Eq. (9), the each row vector of $\bigtriangledown_{\boldsymbol{\omega}}\boldsymbol{\omega}^*$ can be projected into a subspace span $\{\beta_1, \cdots, \beta_r\}$ by the projection matrix $\boldsymbol{\beta}^T\boldsymbol{\beta}$, where β_i is the i-th row vector of $\boldsymbol{\beta}$.

Our analysis shows that updating $\boldsymbol{\alpha}$ is equivalent to modifying the pre-trained parameters $\boldsymbol{\omega}$ within the subspace span$\{\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_r\}$. Since we expect $\boldsymbol{\alpha}$ to be customized, we would like to ensure $\boldsymbol{\alpha}$ and $\boldsymbol{\omega}$ to be non-interfering of each other. Then, prior to generating the customized parameter $\boldsymbol{\alpha}$, we can strategically design an orthogonal subspace span $\{\beta_1, \cdots, \beta_r\}$ to avoid the conflicts between the newly added customized parameters and the pre-trained parameters, while achieving a good trade-off between stability and plasticity. The core challenge of this strategy is to design a set of basis vectors $\{\beta_i\}_{i=1}^r$ that are orthogonal to each other, i.e., $\beta_i^T\beta_j = 0$. This is particularly difficult because the pre-trained dataset is unknown, making it infeasible to strictly enforce orthogonality among the designed basis vectors β_i. To address this, we propose a simple method to achieve approximate orthogonality by leveraging a zero-mean Gaussian distribution.

Proposition 1 (Approximate Orthogonality). *Random Gaussian sampling can yield approximately orthogonal matrices.*

Proof. Let $\mathbf{a}_i \sim \mathcal{N}(0, \sigma^2 I)$ and $\mathbf{b}_i \sim \mathcal{N}(0, \sigma^2 I)$ for all $i \in [1, d]$ independently, then $\mathbb{E}[\mathbf{a}^T\mathbf{b}] = \mathbb{E}[\sum_{i=1}^d \mathbf{a}_i\mathbf{b}_i] = \sum_{i=1}^d \mathbb{E}[\mathbf{a}_i\mathbf{b}_i] = \sum_{i=1}^d \mathbb{E}[\mathbf{a}_i]\mathbb{E}[\mathbf{b}_i] = \sum_{i=1}^d 0 \cdot 0 = 0$. Similarly, let $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{B} \in \mathbb{R}^{m\times n}$, where all entries of $\mathbf{A}$ and $\mathbf{B}$ are independently sampled from $\mathcal{N}(0, \sigma^2 I)$, then $\mathbb{E}[\mathbf{A}^T\mathbf{B}] = \mathbf{0} \in \mathbb{R}^{m\times m}$. ■

Building upon Proposition 1, we develop a simple strategy to achieve approximate orthogonality. Specifically, we sample the basis vectors β_i from a zero-mean Gaussian distribution with standard deviation σ: $\beta_i \sim \mathcal{N}(0, \sigma^2 I)$. Naturally, the resulting projection matrix $\boldsymbol{\beta}^T\boldsymbol{\beta}$ is approximately orthogonal, as it satisfies the expectation condition $\mathbb{E}[\beta_i^T\beta_j] = 0$ for $i \neq j$. Notably, this design method does not require any prior knowledge of the pre-training data.

Experiments

Experimental Setup

Datasets. Following previous studies (Cho et al. 2024; Xie et al. 2023), we utilize the COCO-Stuff dataset (Caesar, Uijlings, and Ferrari 2018) for training. For evaluation, we benchmark our method ADE20K (Zhou et al. 2019), PASCAL VOC (Everingham et al. 2010), and PASCAL-Context (Mottaghi et al. 2014).

Implementation Details. Our method is implemented using the Transformer-based CLIP model. Following the protocol established in (Cho et al. 2024), we evaluate our results on two versions of the CLIP model: ViT-B/16 and ViT-L/14. For training, we follow the same experimental setup as (Cho et al. 2024). Besides, we use the Adam optimizer (Kingma and Ba 2014) with an initial learning rate of 1×10^{-5} for CLIP, and a weight decay of 10^{-4}. Training is performed with one image per mini-batch. We inject the newly generated weights into the $\boldsymbol{\omega}_Q$ and $\boldsymbol{\omega}_V$ linear projections within the self-attention block of each CLIP layer. The number of layers M is set to 3. The rank r is set to 32 for CLIP's text encoder. σ is set to 0.01. We also employ the LoRA-like framework to update CLIP's image encoder, with a rank of 128. The decoder follows the cost-based approach from (Cho et al. 2024).

Comparisons with State-of-the-arts

We compare CP-CLIP with several state-of-the-art approaches, as shown in Table 1. Overall, our method achieves the best performance. Following the classification of fine-tuning methods outlined in (Han et al. 2024), most existing approaches fall into two categories: (1) selective fine-tuning and (2) additive fine-tuning. The former typically fine-tunes a subset of CLIP's parameters, e.g., specific layers with a small learning rate, to enhance its segmentation ability. However, directly modifying CLIP's pre-trained parameter space can compromise its generalization capabilities. The latter often augments CLIP with segmentation-oriented architectures, aiming to preserve its generalization ability. However, this approach retains CLIP's pre-trained

Model	VLM	Additional Backbone	A-847	PC-459	A-150	PC-59	PAS-20	PAS-20^b
Selective Fine-Tuning								
ZS3Net (Bucher et al. 2019)	-	ResNet-101	-	-	-	19.4	38.3	-
OpenSeg (Ghiasi et al. 2022)	ALIGN	ResNet-101	4.4	7.9	17.5	40.1	-	63.8
ZegCLIP (Zhou et al. 2023)	CLIP ViT-B/16	-	-	-	-	41.2	93.6	-
SED (Xie et al. 2023)	CLIP ConvNeXt-B	-	11.4	18.6	31.6	57.3	94.4	-
CAT-Seg (Cho et al. 2024)	CLIP ViT-B/16	-	<u>12.0</u>	<u>19.0</u>	<u>31.8</u>	<u>57.5</u>	<u>94.6</u>	**77.3**
Additive Fine-Tuning								
ZegFormer (Ding et al. 2022)	CLIP ViT-B/16	ResNet-101	4.9	9.1	16.9	42.8	86.2	62.7
ZSseg (Xu et al. 2022b)	CLIP ViT-B/16	ResNet-101	7.0	-	20.5	47.7	88.4	-
SAN (Xu et al. 2023b)	CLIP ViT-B/16	Side Adapter	10.1	12.6	27.5	53.8	94.0	-
OVSeg (Liang et al. 2023)	CLIP ViT-B/16	ResNet-101c	7.1	11.0	24.8	53.3	92.6	-
Reparameterized Fine-Tuning								
CP-CLIP	CLIP ViT-B/16	-	**12.8**	**19.3**	**32.5**	**57.9**	**95.1**	<u>77.1</u>
Selective Fine-Tuning								
OpenSeg (Ghiasi et al. 2022)	ALIGN	Eff-B7	8.1	11.5	26.4	44.8	-	70.2
SED (Xie et al. 2023)	CLIP ConvNeXt-L	-	13.9	22.6	35.2	60.6	96.1	-
CAT-Seg (Cho et al. 2024)	CLIP ViT-L/14	-	<u>16.0</u>	<u>23.8</u>	<u>37.9</u>	<u>63.3</u>	<u>97.0</u>	<u>82.5</u>
Additive Fine-Tuning								
SAN (Xu et al. 2023b)	CLIP ViT-L/14	Side Adapter	12.4	15.7	32.1	57.7	94.6	-
OVSeg (Liang et al. 2023)	CLIP ViT-L/14	Swin-B	9.0	12.4	29.6	55.7	94.5	-
ODISE (Xu et al. 2023a)	CLIP ViT-L/14	Stable Diffusion	11.1	14.5	29.9	57.3	-	-
FC-CLIP (Yu et al. 2023)	CLIP ConvNeXt-L	-	14.8	18.2	34.1	58.4	95.4	-
Reparameterized Fine-Tuning								
CP-CLIP	CLIP ViT-L/14	-	**16.9**	**24.0**	**38.8**	**63.7**	**97.3**	**83.1**

Table 1: Comparison with SOTA methods on several benchmarks. The best results are presented in bold, while the second-best results are underlined. We categorize these fine-tuning methods based on the classification outlined in (Han et al. 2024).

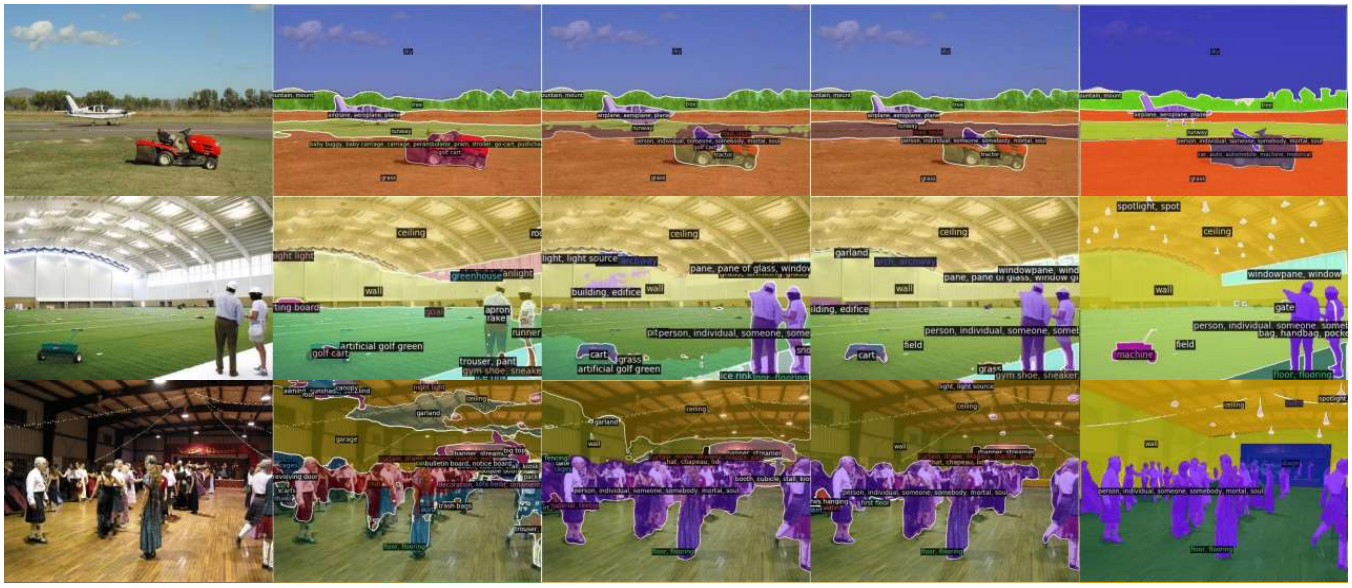
parameter space, which is optimized for image-level classification, potentially limiting its performance in pixel-level recognition tasks. Unlike most previous methods, our proposed CP-CLIP introduces a reparameterized fine-tuning approach for CLIP, preserving its initial generalization capabilities while enhancing segmentation ability. CP-CLIP significantly outperforms the latest SOTA method, CAT-Seg (Cho et al. 2024), achieving a 0.9% improvement on the A-847 dataset, which includes 847 unseen classes, using ViT-B/16 as the base model. These results underscore the effectiveness of our method in maintaining generalization capabilities.

Qualitative results. We present visual comparisons of our method's segmentation results against leading approaches, e.g., CAT-Seg (Cho et al. 2024) and SAN (Xu et al. 2023b), on the A-847 dataset. As shown in Fig. 2, our method achieves superior accuracy in predicting the semantics of unseen objects and maintaining internal consistency within objects. This highlights the preservation of CLIP's pre-trained generalization capabilities.

Ablation Study

Ablation of fine-tuning CLIP. We assess the effectiveness of different fine-tuning design variants for CP-CLIP, as shown in Table 2. We also compare the performance with several commonly used fine-tuning methods, such as LoRA (Hu et al. 2022) and HyperNetwork (Ha, Dai, and Le 2017), since our proposed method is inspired by them. Compared to the baseline, i.e., fully freezing CLIP's encoders (see row 1), fine-tuning with each existing method (see rows 2 to 5) leads to significant improvements, demonstrating that fine-tuning effectively equips CLIP with segmentation capabilities for pixel-level recognition. When compared to existing fine-tuning strategies, our method also achieves optimal performance (see row 8). Moreover, if we use random initialization for β and keep it fixed, i.e., "w/o using OA" (see row 6), the performance drops significantly on the A-847 dataset. This is because random adaptation could interfere with CLIP's pre-trained weights, resulting in a loss of generalization ability. Additionally, if we do not use local visual features as conditions for the proposed customized parameter generator, i.e., "w/o using CPG", there will be no interaction between the input random noise and local visual features in the generator. As a result, the parameters remain fitted to specific classes, leading to a lack of generalization capabilities for adapting to new unseen classes after fine-tuning. Finally, by integrating these two core designs in our CP-CLIP, we significantly preserve CLIP's generalization and achieve the best performance, as shown in Table 2.

Why also fine-tune CLIP's image encoder? We evaluate the effectiveness of updating the parameters of CLIP's image encoder, as shown in Table 3. As detailed in the implementation details paragraph, we also adopt a LoRA-like parameter-efficient fine-tuning strategy for the image encoder. The reason for fine-tuning CLIP's image encoder is that freezing it can only preserve the original image-to-text alignment, while the alignment of text encoder is already shifting from image-to-text to pixel-to-text. This alignment discrepancy may impede optimization, and visual features



Input Image SAN (Xu et al. 2023b) CAT-Seg (Cho et al. 2024) Ours Ground Truth

Figure 2: Comparison of qualitative reults on A-847 (Zhou et al. 2019). We compare our method with other two state-of-the-art methods, i.e., SAN (Xu et al. 2023b) and CAT-Seg (Cho et al. 2024). The results demonstrate that our method achieves more accurate segmentation, particularly in terms of predicting the semantics of unseen objects and region consistency.

Method	CPG	OA	A-847	PC-459	A-150	PC-59	PAS-20	PAS-20^b
Baseline (Freeze)	-	-	4.4	6.6	24.8	49.4	92.5	71.9
Reparameterized Fine-Tuning								
LoRA (Hu et al. 2022)	-	-	11.4	17.6	28.6	55.1	94.2	76.7
Additive Fine-Tuning								
VPT (Jia et al. 2022)	-	-	5.7	10.2	23.7	54.3	93.8	75.1
Adapter (Houlsby et al. 2019)	-	-	10.4	16.5	28.8	54.9	94.2	75.2
HyperNetwork (Ha, Dai, and Le 2017)	-	-	10.9	16.7	29.1	56.5	94.2	76.2
CP-CLIP (Ours)	✓	w/o using	11.5	18.0	30.9	57.1	94.2	76.5
	w/o using	✓	11.9	18.5	31.9	56.5	94.1	76.2
	✓	✓	**12.8**	**19.3**	**32.5**	**57.9**	**95.1**	**77.1**

Table 2: Ablation on the fine-tuning encoder in CP-CLIP and comparison of universal fine-tuning methods. The base model is ViT-B/16. "CPG": Customized parameter generator. "OA": Orthogonal adaptation. "w/o using CPG": Without using local visual features as conditions. "w/o using OA": Without orthogonal adaptation.

Fine-tuned Encoder	A-847	PC-459	A-150	PC-59	PAS-20^b
Text only	<u>11.2</u>	<u>17.2</u>	<u>30.3</u>	55.9	75.6
Image only	10.4	16.5	28.9	<u>56.2</u>	<u>76.1</u>
Both (CP-CLIP)	**12.8**	**19.3**	**32.5**	**57.9**	**77.1**

Table 3: Ablation on the different fine-tuning strategies.

M	A-847	PC-459	A-150	PC-59	PAS-20^b
1	11.2	17.2	30.8	56.5	75.9
2	12.4	**19.4**	31.7	57.2	76.6
3	**12.8**	<u>19.3</u>	**32.5**	<u>57.9</u>	<u>77.1</u>
4	12.7	19.2	32.3	**58.5**	**77.9**

Table 4: Ablation on the number of layers M in $\mathcal{G}$.

at image-level granularity are also sub-optimal for segmentation tasks. (2) Fine-tuning CLIP's text encoder is essential, as it enables the task adaptation of CLIP from text-to-image to text-to-pixel recognition. Based on this, fine-tuning CLIP's image encoder can further enhances performance by providing visual features better suited for pixel-level understanding. Importantly, using these refined visual features as conditions facilitates improved parameter generation for the text encoder, culminating in our model, CP-CLIP.

Ablation of the number of layers M in $\mathcal{G}$. The number of layers M represents the interaction times between the in-

r	A-847	PC-459	A-150	PC-59	PAS-20^b
8	11.0	17.7	31.0	55.9	75.6
16	12.1	18.5	31.8	56.9	76.4
32	**12.8**	19.3	**32.5**	**57.9**	77.1
64	12.6	**19.5**	32.3	57.8	**78.0**

Table 5: Ablation on the value of r in $\boldsymbol{\alpha}$.

Layer	A-847	PC-459	A-150	PC-59	PAS-20^b
1	11.6	18.2	31.2	57.3	76.8
6	12.4	19.2	32.3	57.8	76.7
12	**12.8**	**19.3**	**32.5**	**57.9**	**77.1**

Table 6: Discussion on selecting visual features from which layer of CLIP's image encoder as conditions.

Freeze	A-847	PC-459	A-150	PC-59	PAS-20^b
✗	11.1	18.2	31.0	57.6	77.0
✓	**12.8**	**19.3**	**32.5**	**57.9**	**77.1**

Table 7: Discussion on whether $\boldsymbol{\beta}$ should be frozen or not. ✗: fine-tuning. ✓: keep it fixed.

put random noise and local visual features. To investigate its impact, we conduct an ablation study on the M by systematically adding the number of layers in $\mathcal{G}$. As shown in Table 4, accuracy improves incrementally on all datasets. However, on datasets with more classes (e.g., A-847), the accuracy performance plateaus around 3 layers. This indicates that while increasing the number of layers allows for a more comprehensive utilization of local visual features, the denoising process may become over-parameterized for analyzing new images from unseen scenarios.

Ablation of value of r in $\boldsymbol{\alpha}$. The value of r in $\boldsymbol{\alpha}$ for CLIP's text encoder represents the demand of CLIP's text encoder for segmentation ability. The smaller the value, the closer CLIP's text encoder operates to a zero-shot setting, i.e., when $r = 0$. To explore its impact, we conduct an ablation study on r by systematically varying its value from 8 to 64. As shown in Table 5, at first, accuracy improves globally on all datasets, which indicates that CLIP's text encoder is equipped with the pixel-level recognition ability. Then, on datasets with more classes (e.g., A-847), the accuracy performance reaches a saturation point around a value of 32. This demonstrates that even for the largest open-vocabulary semantic segmentation dataset, with a value of only 32, our model can effectively equip CLIP's text encoder with segmentation capabilities without compromising the generalization abilities of the pre-trained parameters.

Which layer's visual features should be selected as conditions? We are also interested in determining which layer's visual features should be selected as conditions. As shown in Table 6, we choose layers 1, 6, and 12 to represent low-, middle-, and high-level local visual features, respectively. Consistent with prior studies (Zhang et al. 2021), high-level features often contain more semantic information, which is particularly beneficial for CLIP's text encoder when encountering new images in unseen scenarios. In contrast, low-level and middle-level features contain more detailed information, offering less benefit to CLIP's text encoder.

Parameter	Memory (GB)	FLOPs (G)	Latency (s)
Frozen	15.3	1815.9	0.165
Customized (Ours)	15.8 (+0.5)	1993.7 (+9.8%)	0.184 (+0.019)

Table 8: Efficiency comparison during inference in terms of (Memory, FLOPs, and Latency) compared to baseline (Freeze), i.e., fixed parameters.

Whether $\boldsymbol{\beta}$ should be frozen or not? Lastly, we investigate whether $\boldsymbol{\beta}$ should be frozen, providing deeper insights into the design of orthogonal adaptation. As shown in Table 7, making $\boldsymbol{\beta}$ trainable leads to a significant performance drop across all datasets. This occurs because if $\boldsymbol{\beta}$ is learnable, the update of $\boldsymbol{\Delta\omega} = \alpha \cdot \boldsymbol{\beta}$ becomes unconstrained, potentially interfering with the generalization capabilities embedded in CLIP's pre-trained weights $\boldsymbol{\omega}$. Consequently, the generalization ability on unseen classes is compromised, particularly for datasets with a larger number of unseen classes, such as A-847. More importantly, We quantify the update directions by computing the cosine similarity between $\Delta\boldsymbol{\omega}$ and $\boldsymbol{\omega}$ across layers, all are smaller than 10^{-3} ($\approx 89.94°$), confirming that the update directions are **approximately orthogonal** to the original weights.

Efficiency of CP-CLIP. Although we introduce an additional side network to generate visual-feature-conditional parameters, this network comprises only three cross-attention layers and FFNs, making it highly lightweight ($\sim$30MB) compared to CLIP's backbone. Furthermore, we compare the efficiency of CP-CLIP with its baseline, i.e., freezing CLIP's pre-trained weights. The comparison, summarized in Table 8, shows that our method incurs only a small inference overhead in terms of memory, FLOPs, and Latency, while achieving significant improvement.

Conclusion

We propose CP-CLIP, a novel fine-tuning strategy that generates customized parameters for open-vocabulary semantic segmentation by conditioning on local visual features from CLIP's image encoder. This approach ensures adaptability to new images from unseen scenarios while preserving the model's initial generalization ability. Additionally, we introduce an orthogonal adaptation technique to ensure that the update direction remains orthogonal to the pre-trained weights, further enhancing generalization. Extensive experiments demonstrate that CP-CLIP achieves superior performance across multiple dense prediction benchmarks.

Acknowledgments

This work was supported by the NSFC under Grant 62322604, 62576207 and 62401361.

References

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35: 23716–23736.

Bucher, M.; Vu, T.-H.; Cord, M.; and Pérez, P. 2019. Zero-shot semantic segmentation. *Advances in Neural Information Processing Systems*, 32.

Caesar, H.; Uijlings, J.; and Ferrari, V. 2018. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1209–1218.

Chen, X.; Wang, X.; Changpinyo, S.; Piergiovanni, A.; Padlewski, P.; Salz, D.; Goodman, S.; Grycner, A.; Mustafa, B.; Beyer, L.; et al. 2022. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*.

Cheng, B.; Misra, I.; Schwing, A. G.; Kirillov, A.; and Girdhar, R. 2022. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1290–1299.

Cho, S.; Shin, H.; Hong, S.; Arnab, A.; Seo, P. H.; and Kim, S. 2024. CAT-Seg: Cost Aggregation for Open-Vocabulary Semantic Segmentation. arXiv:2303.11797.

Dai, W.; Li, J.; Li, D.; Tiong, A. M. H.; Zhao, J.; Wang, W.; Li, B.; Fung, P.; and Hoi, S. 2023. InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. arXiv:2305.06500.

Ding, J.; Xue, N.; Xia, G.-S.; and Dai, D. 2022. Decoupling zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11583–11592.

Erkoç, Z.; Ma, F.; Shan, Q.; Nießner, M.; and Dai, A. 2023. Hyperdiffusion: Generating implicit neural fields with weight-space diffusion. In *ICCV*.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88: 303–338.

Ghiasi, G.; Gu, X.; Cui, Y.; and Lin, T.-Y. 2022. Scaling open-vocabulary image segmentation with image-level labels. In *European Conference on Computer Vision*, 540–557. Springer.

Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; and Parikh, D. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6904–6913.

Ha, D.; Dai, A. M.; and Le, Q. V. 2017. HyperNetworks. In *International Conference on Learning Representations*.

Han, Z.; Gao, C.; Liu, J.; Zhang, J.; and Zhang, S. Q. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-efficient transfer learning for NLP. In *International conference on machine learning*, 2790–2799. PMLR.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Jia, M.; Tang, L.; Chen, B.-C.; Cardie, C.; Belongie, S.; Hariharan, B.; and Lim, S.-N. 2022. Visual prompt tuning. In *European Conference on Computer Vision*, 709–727. Springer.

Jin, X.; Wang, K.; Tang, D.; Zhao, W.; Zhou, Y.; Tang, J.; and You, Y. 2024. Conditional LoRA Parameter Generation. *arXiv preprint arXiv:2408.01415*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, B.; Weinberger, K. Q.; Belongie, S.; Koltun, V.; and Ranftl, R. 2022. Language-driven Semantic Segmentation. In *International Conference on Learning Representations*.

Liang, F.; Wu, B.; Dai, X.; Li, K.; Zhao, Y.; Zhang, H.; Zhang, P.; Vajda, P.; and Marculescu, D. 2023. Open-vocabulary semantic segmentation with mask-adapted clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7061–7070.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Lu, J.; Batra, D.; Parikh, D.; and Lee, S. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32.

Lu, P.; Mishra, S.; Xia, T.; Qiu, L.; Chang, K.-W.; Zhu, S.-C.; Tafjord, O.; Clark, P.; and Kalyan, A. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35: 2507–2521.

Ma, F.; Xue, H.; Zhou, Y.; Wang, G.; Rao, F.; Yan, S.; Zhang, Y.; Wu, S.; Shou, M. Z.; and Sun, X. 2024. Visual perception by large language model's weights. *Advances in Neural Information Processing Systems*, 37: 28615–28635.

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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 891–898.

Peebles, W.; Radosavovic, I.; Brooks, T.; Efros, A. A.; and Malik, J. 2022. Learning to learn with generative models of neural network checkpoints. *arXiv preprint arXiv:2209.12892*.

Peng, Z.; Xu, Z.; Zeng, Z.; Huang, Y.; Wang, Y.; and Shen, W. 2025a. Parameter-efficient Fine-tuning in Hyperspherical Space for Open-vocabulary Semantic Segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 15009–15020.

Peng, Z.; Xu, Z.; Zeng, Z.; Wen, C.; Huang, Y.; Yang, M.; Tang, F.; and Shen, W. 2025b. Understanding Fine-tuning CLIP for Open-vocabulary Semantic Segmentation in Hyperbolic Space. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 4562–4572.

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.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Ruiz, N.; Li, Y.; Jampani, V.; Pritch, Y.; Rubinstein, M.; and Aberman, K. 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 22500–22510.

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.

Tan, H.; and Bansal, M. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*.

Tang, F.; An, X.; Yang, H.; Xie, Y.; Yang, K.; Hu, M.; Cheng, Z.; Zhou, X.; Ran, Z.; Razzak, I.; et al. ???? UniViT: Unifying Image and Video Understanding in One Vision Encoder. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Tang, F.; Liu, C.; Xu, Z.; Hu, M.; Huang, Z.; Xue, H.; Chen, Z.; Peng, Z.; Yang, Z.; Zhou, S.; et al. 2025. Seeing Far and Clearly: Mitigating Hallucinations in MLLMs with Attention Causal Decoding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 26147–26159.

Wang, D.; Shelhamer, E.; Liu, S.; Olshausen, B.; and Darrell, T. 2020. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*.

Wang, K.; Tang, D.; Zeng, B.; Yin, Y.; Xu, Z.; Zhou, Y.; Zang, Z.; Darrell, T.; Liu, Z.; and You, Y. 2024. Neural network diffusion. *arXiv preprint arXiv:2402.13144*.

Wang, Q.; Fink, O.; Van Gool, L.; and Dai, D. 2022. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7201–7211.

Wu, J.; Li, X.; Xu, S.; Yuan, H.; Ding, H.; Yang, Y.; Li, X.; Zhang, J.; Tong, Y.; Jiang, X.; et al. 2024. Towards open vocabulary learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(7): 5092–5113.

Xie, B.; Cao, J.; Xie, J.; Khan, F. S.; and Pang, Y. 2023. SED: A Simple Encoder-Decoder for Open-Vocabulary Semantic Segmentation. arXiv:2311.15537.

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

Xu, J.; Liu, S.; Vahdat, A.; Byeon, W.; Wang, X.; and De Mello, S. 2023a. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2955–2966.

Xu, M.; Zhang, Z.; Wei, F.; Hu, H.; and Bai, X. 2023b. Side adapter network for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2945–2954.

Xu, M.; Zhang, Z.; Wei, F.; Lin, Y.; Cao, Y.; Hu, H.; and Bai, X. 2022b. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In *European Conference on Computer Vision*, 736–753. Springer.

Yang, Z.; Meng, Y.; Fu, K.; Tang, F.; Wang, S.; and Song, Z. 2025a. Exploring CLIP's Dense Knowledge for Weakly Supervised Semantic Segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 20223–20232.

Yang, Z.; Zhao, X.; Wang, X.; Zhang, Q.; and Xiao, J. 2025b. FFR: Frequency Feature Rectification for Weakly Supervised Semantic Segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 30261–30270.

Yang, Z.; Zhao, X.; Yao, C.; Zhang, Q.; and Xiao, J. 2025c. M-SEE: A multi-scale encoder enhancement framework for end-to-end Weakly Supervised Semantic Segmentation. *Pattern Recognition*, 162: 111348.

Yu, Q.; He, J.; Deng, X.; Shen, X.; and Chen, L.-C. 2023. Convolutions Die Hard: Open-Vocabulary Segmentation with Single Frozen Convolutional CLIP. In *NeurIPS*.

Zhang, D.; Zhang, H.; Tang, J.; Hua, X.-S.; and Sun, Q. 2021. Self-regulation for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6953–6963.

Zhang, F.; Zhou, T.; Li, B.; He, H.; Ma, C.; Zhang, T.; Yao, J.; Zhang, Y.; and Wang, Y. 2023. Uncovering prototypical knowledge for weakly open-vocabulary semantic segmentation. *Advances in Neural Information Processing Systems*, 36: 73652–73665.

Zhao, C.; Wang, Y.; Jiang, X.; Shen, Y.; Song, K.; Li, D.; and Miao, D. 2024. Learning Domain Invariant Prompt for Vision-Language Models. *IEEE Transactions on Image Processing*, 33: 1348–1360.

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.

Zhou, Z.; Lei, Y.; Zhang, B.; Liu, L.; and Liu, Y. 2023. ZegCLIP: Towards adapting CLIP for zero-shot semantic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.