# CAT-Seg: Cost Aggregation for Open-Vocabulary Semantic Segmentation

Seokju Cho[1,*] Heeseong Shin[1,*] Sunghwan Hong[1]
Anurag Arnab[2] Paul Hongsuck Seo[1,†] Seungryong Kim[1,†]

[1]Korea University [2]Google Research

{seokju_cho, hsshin98, sung_hwan, phseo, seungryong_kim}@korea.ac.kr
aarnab@google.com

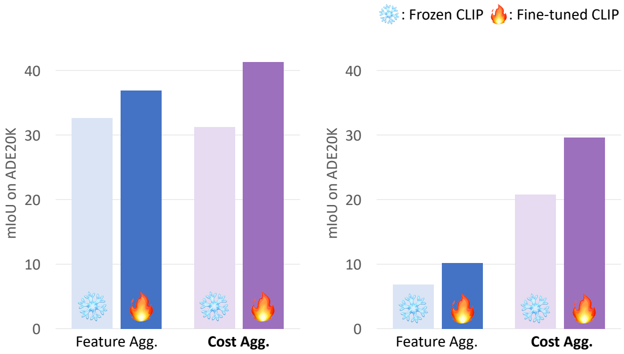
## Abstract

*Open-vocabulary semantic segmentation presents the challenge of labeling each pixel within an image based on a wide range of text descriptions. In this work, we introduce a novel cost-based approach to adapt vision-language foundation models, notably CLIP, for the intricate task of semantic segmentation. Through aggregating the cosine similarity score, i.e., the cost volume between image and text embeddings, our method potently adapts CLIP for segmenting seen and unseen classes by fine-tuning its encoders, addressing the challenges faced by existing methods in handling unseen classes. Building upon this, we explore methods to effectively aggregate the cost volume considering its multi-modal nature of being established between image and text embeddings. Furthermore, we examine various methods for efficiently fine-tuning CLIP.*

## 1. Introduction

Open-vocabulary semantic segmentation aims to assign each pixel in an image to a class label from an unbounded range, defined by text descriptions. To handle the challenge of associating an image with a wide variety of text descriptions, pre-trained vision-language foundation models, *e.g*., CLIP [43] and ALIGN [22], have drawn attention as they exerted strong open-vocabulary recognition capabilities achieved through training on extensive image-text datasets. Nonetheless, these foundation models primarily receive image-level supervision during training, which introduces a notable disparity when applying them to the pixel-level segmentation tasks [66].

To address this gap, recent works [9, 14, 30, 55–57, 60] have reformulated the task into a region-level problem by utilizing mask proposal generators. While this partially bridges the discrepancy between the pre-training and the downstream task, a discernible gap persists between the conceptualization of regions and the entire image for CLIP.



(a) mIoU of **seen** classes (b) mIoU of **unseen** classes

Figure 1. **Comparison between feature and cost aggregation for open-vocabulary semantic segmentation task.** In contrast to feature aggregation suffering severe overfitting to seen classes, cost aggregation can generalize to unseen classes and achieve significant performance improvements upon fine-tuning of CLIP.

In this work, we investigate methods to transfer the holistic understanding capability of images to the pixel-level task of segmentation. While a straightforward approach would be to fine-tune the encoders of CLIP, existing methods struggle in such attempt [57, 60, 66] as they encounter significant overfitting problems to the seen classes. This results in the misalignment of the joint embedding space for unseen classes, as the CLIP features undergo decoder modules for aggregating them into segmentation masks, hence losing their alignment. Consequently, most methods [9, 14, 30, 55–57, 60] opt for freezing the encoders of CLIP instead, remaining the challenge underexplored.

In this regard, we extend the exploration of adapting CLIP for open-vocabulary semantic segmentation and introduce a novel cost-based framework. We propose to aggregate the cosine similarity between image and text embeddings of CLIP, *i.e.*, the matching cost, drawing parallels to the visual correspondence literature [26]. Surprisingly, we find that fine-tuning CLIP upon this framework effec-

[*] Equal contribution. [†] Corresponding authors.

tively adapts CLIP to the downstream task of segmentation for both seen and unseen classes, as shown in Fig. 1. Noticing this, we delve into better aggregating the cost volume between image and text for segmentation.

Intuitively, the cost volume can be viewed as rough semantic masks grounded to their respective classes, as illustrated in Fig. 2. Subsequently, these rough masks can be further refined to obtain accurate predictions, being the cost aggregation process. In light of this, we aim to effectively aggregate the cost volume and configure the process into spatial and class aggregation, regarding its multi-modal nature from being established between image and text. Furthermore, by observing the effectiveness of fine-tuning CLIP for its adaptation to semantic segmentation, we explore various methods to facilitate this process efficiently.

We analyze our cost aggregation framework to be advantageous in two aspects for adapting CLIP to dense prediction: *i*) the robustness of cost aggregation against overfitting, and *ii*) the direct construction of the cost volume from image and text embeddings of CLIP. For cost aggregation, the aggregation layers operate upon similarity scores, preventing them from overfitting to the features [4, 32, 47]. Moreover, as opposed to existing methods where they often employ decoder layers upon the image embeddings of CLIP [60, 66], we do not introduce additional layers that can potentially project the embeddings to a different embedding space.

Our framework, dubbed CAT-Seg, combines our cost aggregation-based framework consisting of spatial and class aggregation, with our optimal approach for fine-tuning the encoders of CLIP. We achieve state-of-the-art results on every standard open-vocabulary benchmark with large margins, gaining +3.6 mIoU in A-847 and +8.1 mIoU in PC-459 compared to the recent state-of-the-art. Not only CAT-Seg it is effective, but is also efficient both for training and inference compared to region-text methods, being over ×3.7 faster for inference. Furthermore, even in the extreme scenario [1] where the domain of the image and text description differs significantly from the training dataset, our model outperforms existing state-of-the-art methods with a large margin, paving the way for various domain-specific applications.

We summarize our contribution as follows:

- We propose a cost aggregation-based framework for open-vocabulary semantic segmentation, effectively adapting CLIP to the downstream task of segmentation by fine-tuning its encoders.
- To aggregate the image-text cost volume, we consist of our framework with spatial and class aggregation to reason the multi-modal cost volume and explore various methods to enhance our cost aggregation framework.
- Our framework, named CAT-Seg, establishes state-of-the-art performance for standard open-vocabulary benchmarks, as well as for extreme case scenarios [1], demonstrating versatility and practicality.

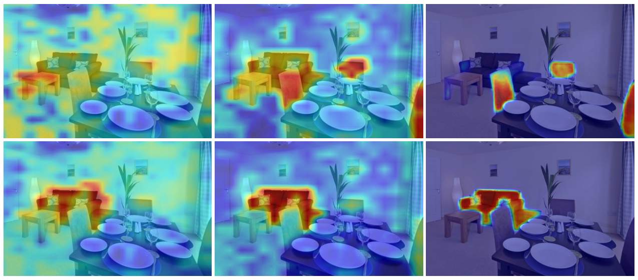

(a) CLIP (b) Fine-tuned CLIP (c) Aggregated Cost

Figure 2. **Visualization of the cost volume.** We visualize the raw cost volume obtained from frozen CLIP in (a) and fine-tuned CLIP in (b), and the aggregated cost in (c) through CAT-Seg. The top row correspond to the seen class "chair" and the bottom row correspond to the unseen class "sofa".

## 2. Related Work

**Open-vocabulary semantic segmentation.** Classical approaches to the task [2, 54, 64] attempt to learn visual embeddings that align with pre-defined text embeddings [36, 37]. However, the limited vocabulary of the words has been the major bottlenecks. To address this, LSeg [28] leveraged CLIP for learning pixel-level visual embeddings aligned with the text embeddings of CLIP. Alternatively, OpenSeg [14] proposed to identify local regions within the image and correlate with the text embeddings with class-agnostic region proposals. Similarly, ZegFormer [9] and ZSseg [56] proposed two-stage frameworks for dealing with the task. Typically, they first learn to predict class-agnostic region proposals similar to [14], and feed them to CLIP for final predictions. To better recognize these regions, OVSeg [30] collects region-text pairs to fine-tune the CLIP encoder, while MaskCLIP [10] leverages the self-attention map from CLIP to refine the region proposals. Alternatively, ODISE [55] leverages pre-trained Stable Diffusion [45] model for generating high-quality class-agnostic masks. However, these region-to-text matching methods [9, 14, 30, 55–57, 60] require a region generator, which is trained on a limited scale of annotated datasets.

More recently, ZegCLIP [70] and SAN [57] proposed one-stage frameworks, where they attempt to leverage the embeddings from CLIP to predict masks instead of having class-agnostic mask generators parallel to CLIP. Although these methods can better leverage the pre-trained knowledge from CLIP, they introduce learnable tokens or adapter layers to the CLIP image encoder, which can be only trained on the seen classes. FC-CLIP [60] implements CLIP as the visual backbone for the segmentation model but opts for a frozen image encoder as they find fine-tuning the image encoder hinders performance for unseen classes. In contrast, we refrain from adding external layers to CLIP and achieve fine-tuning of its encoders by aggregating the cost volume, which is obtained solely from the embeddings of CLIP.

**Fine-tuning vision-language models.** Along with the advance of large-scale vision-language models, *e.g.* CLIP, numerous attempts have been made to adapt CLIP to various downstream tasks [52]. CoOp [68] and CoCoOp [67] learn prompt tokens instead of optimizing the full model. Another stream of work is CLIP-Adapter [13] and TIP-Adapter [63], where they aggregate the image and text embeddings from CLIP through adapter layers instead of tuning the encoder itself. However, such methods mainly focus on few-shot settings rather than zero-shot evaluation. We explore end-to-end fine-tuning of CLIP for zero-shot pixel-level prediction, which has failed in numerous attempts [57, 60, 66].

**Cost aggregation.** Cost aggregation [5, 8, 15, 19, 21, 26, 47, 58] is a popular technique adopted for the process of establishing correspondence between visually or semantically similar images [7, 15, 18, 26, 58] by reducing the impact of errors and inconsistencies in the matching process. A matching cost, an input to cost aggregation, is typically constructed between dense features extracted from a pair of images [44], and often cosine-similarity [32, 44] is used. In this work, we view the cosine-similarity score between image and text embeddings of CLIP from the viewpoint of establishing the matching cost volume. Especially, we find the robustness of the cost aggregation layers to be favorable to open-vocabulary semantic segmentation, as these layers operate upon the similarity scores rather than the embeddings itself [32, 47]. However, our approach diverges from traditional methods as the cost volume obtained from CLIP is inherently multi-modal, originating from both image and text modalities. This contrasts with conventional cost aggregation techniques [7, 15, 18, 26, 58]. Consequently, we explore methods to effectively aggregate the multi-modal cost volume.

## 3. Methodology

Given an image $I$ and a set of candidate class categories $\mathcal{C} = \{T(n)\}$ for $n = 1, \dots, N_{\mathcal{C}}$, where $T(n)$ denotes textual description of $n$-th category and $N_{\mathcal{C}}$ is the number of classes, open-vocabulary semantic segmentation assigns a class label for each pixel in image $I$. Different from classical semantic segmentation tasks [16, 17, 24, 34, 59, 61, 69], open-vocabulary segmentation is additionally challenged by varying $\mathcal{C}$, given as free-form text description.

In this section, we describe our cost-based approach for open-vocabulary semantic segmentation. In specific, we refine the cosine-similarity scores from image and text embedding of CLIP, as illustrated in Fig. 2. The process of refining the cosine-similarity scores, or cost aggregation [26], was initially developed for the image correspondence problem and specifically designed to process an image-to-image cost volume. Consequently, traditional cost aggregation methods leverage image-specific priors, such as the assumption of local smoothness of images [27, 38, 39] for aggregating the cost volume.

On the other hand, we aim to aggregate the image-to-text cost volume, hence need to consider the multi-modality of the cost volume and the respective characteristics of each modality. In this regard, as shown in Fig. 3, we break down the aggregation stage into two separate modules, *i.e.*, spatial and class aggregation, reasonably addressing the unique challenges presented by the task of open-vocabulary semantic segmentation. This includes aspects such as handling varying numbers of classes during inference and guaranteeing the permutation invariance between classes. Specifically, we perform spatial aggregation followed by class aggregation and alternate both aggregations. In the following section, we describe the cost aggregation process in detail, as well as introduce additional techniques for enhancing the cost aggregation framework.

### 3.1. Cost Computation and Embedding

Given an image $I$ and a set of classes $\mathcal{C}$, we extract the dense image embeddings $D^V = \Phi^V(I) \in \mathbb{R}^{(H \times W) \times d}$ and the text embeddings $D^L = \Phi^L(T) \in \mathbb{R}^{N_{\mathcal{C}} \times d}$, where $\Phi^V(\cdot)$ and $\Phi^L(\cdot)$, denotes the image and text encoders of CLIP respectively. For extracting dense CLIP image embeddings, we follow the method described in [66], wherein we modify the last attention layer of the image encoder to eliminate the pooling effect. We use the image and text embeddings $D^V(i)$ and $D^L(n)$, where $i$ denotes 2D spatial positions of the image embedding and $n$ denotes an index for a class, to compute a cost volume $C \in \mathbb{R}^{(H \times W) \times N_{\mathcal{C}}}$ by cosine similarity [44]. Formally, this is defined as:

$$C(i, n) = \frac{D^V(i) \cdot D^L(n)}{\|D^V(i)\| \|D^L(n)\|}. \tag{1}$$

To enhance the processing of cost in high dimensional feature space, we feed the cost volume to a single convolution layer that processes each cost slice $C(:, n) \in \mathbb{R}^{(H \times W) \times 1}$ independently to obtain initial cost volume embedding $F \in \mathbb{R}^{(H \times W) \times N_{\mathcal{C}} \times d_F}$, where $d_F$ is the cost embedding dimension, as shown in Fig. 3.

### 3.2. Spatial Cost Aggregation

For spatial aggregation, we aim to consider the characteristics of images within the image-text cost volume, such as spatial smoothness within the image. Specifically, we apply spatial aggregation for each class, respectively. Considering that we pursue the holistic understanding of images of CLIP to effectively transfer to segmentation, we adopt Transformer [33, 51] over CNNs for its global [51]

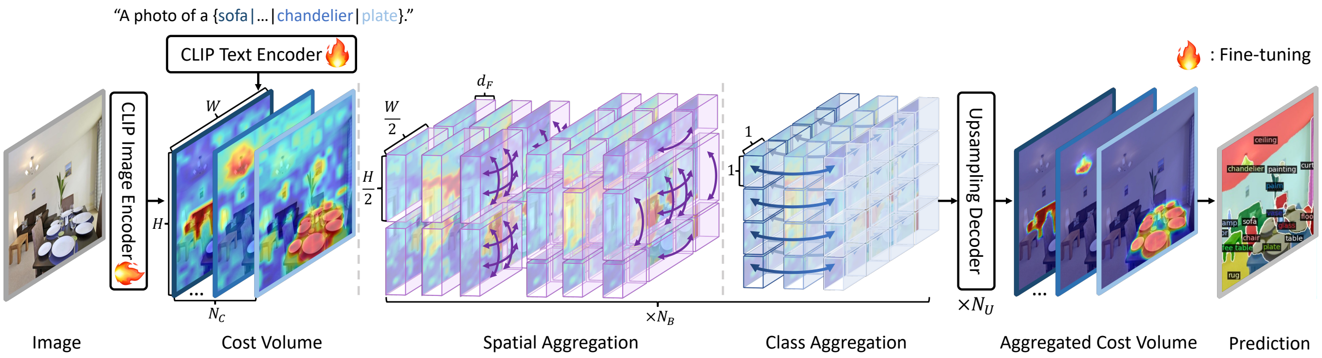


Figure 3. **Overview of CAT-Seg.** Our cost aggregation framework consists of spatial aggregation and class aggregation, followed by an upsampling decoder. Please refer to the supplementary material for a detailed illustration.

or semi-global [18, 33] receptive fields. In practice, we employ Swin Transformer [33] for computational efficiency. We define this process as follows:

$$F'(:,n) = \mathcal{T}^{\mathrm{sa}}(F(:,n)), \tag{2}$$

where $F(:,n) \in \mathbb{R}^{(H\times W)\times d_F}$, and $\mathcal{T}^{\mathrm{sa}}(\cdot)$ denotes a pair of two consecutive Swin transformer block for spatial aggregation, where the first block features self-attention within a local window, followed by the second block with self-attention within shifted window. Note that we treat $d_F$ as channel dimensions for each token, and attention is computed within individual classes separately. Intuitively, we can roughly relate the process of spatial aggregation to the bottom row of Fig. 2, where the cost volume for "sofa" is well-refined after aggregation, and the noise in the background region is suppressed.

### 3.3. Class Cost Aggregation

Subsequent to spatial aggregation, class aggregation is applied to consider the text modality, explicitly capturing relationships between different class categories. We also consider the unique challenges of open-vocabulary semantic segmentation of handling varying numbers of categories $\mathcal{C}$ while being invariant to their ordering. To address these challenges, we employ a Transformer [51] layer without position embedding for aggregation, as this can achieve both of the aforementioned criteria. This process is defined as:

$$F''(i,:) = \mathcal{T}^{\mathrm{ca}}(F'(i,:)), \tag{3}$$

where $F'(i,:) \in \mathbb{R}^{N_{\mathcal{C}}\times d_F}$, and $\mathcal{T}^{\mathrm{ca}}(\cdot)$ denotes a transformer block for class aggregation. In contrast to spatial aggregation, we instead employ a linear transformer [25] as we do not need to consider spatial structure of the input tokens in this aggregation, as well as benefitting from the linear computational complexity with respect to the number of the tokens. The class aggregation process can be related to the top row of Fig. 2, where the aggregated cost volume depicts its prediction to only chairs and excluding the sofa, as both classes are given together for reasoning.

### 3.4. CAT-Seg Framework

Upon the aggregated cost volume through spatial and class aggregation, we further enhance our methodology by incorporating an upsampling and aggregation process to derive semantic segmentation predictions. Additionally, drawing insights from state-of-the-art cost aggregation techniques [7, 8, 18], we refine our cost aggregation strategy by leveraging guidance derived from the embeddings of CLIP. Finally, we examine various methods to fine-tune the encoders of CLIP, in pursuit of effectively, yet efficiently adapting CLIP for open-vocabulary semantic segmentation. Altogether, we introduce **C**ost **A**ggrega**T**ion approach for open-vocabulary semantic **Seg**mentation (CAT-Seg). We describe the upsampling decoder, embedding guidance, and our fine-tuning approach in detail in the subsequent sections. For detailed illustrations of the architecture for each component, please refer to the supplementary materials.

**Upsampling decoder.** Similar to FPN [31], we employ bilinear upsampling on the aggregated cost volume and concatenate it with the corresponding level of feature map extracted from CLIP, followed by a convolutional layer with a 3×3 kernel of fixed size. We iterate this process $N_U$ times, generating a high-resolution output which is fed into the prediction head for final inference. To extract the high-resolution feature map, we avoid using an additional feature backbone that would introduce a heavy computational burden. Instead, similarly to [29], we extract these maps from the middle layers of the CLIP image encoder. Specifically, we extract the feature map from the output of intermediate layers of CLIP ViT [11] and then upsample them using a single learnable transposed convolution layer. This approach allows us to efficiently leverage the well-learned representations of CLIP for obtaining detailed predictions. For additional details, refer to the supplementary materials.

**Embedding guidance.** As a means to enhance the cost aggregation process, we additionally leverage the embeddings $D_L$ and $D_V$ to provide spatial structure or contextual information of the inputs. Intuitively, we aim to guide the process with embeddings, based on the assumption that visually or semantically similar input tokens, e.g., color or category, have similar matching costs, inspired by cost volume filtering [20, 48] in stereo matching literature [46]. Accordingly, we redefine Eq. 2 and Eq. 3 as:

$$\begin{aligned} F'(:,n) &= \mathcal{T}^{\mathrm{sa}}([F(:,n);\mathcal{P}^V(D^V)]), \\ F''(i,:) &= \mathcal{T}^{\mathrm{ca}}([F'(i,:);\mathcal{P}^L(D^L)]), \end{aligned} \tag{4}$$

where $[\cdot]$ denotes concatenation, $\mathcal{P}^V$ and $\mathcal{P}^L$ denote linear projection layer, $D^V \in \mathbb{R}^{(H\times W)\times d}$, and $D^L \in \mathbb{R}^{N_\mathcal{C}\times d}$, where $d$ denotes the feature dimension. Notably, we only provide the embeddings to query and key as we find this is sufficient for embedding guidance.

**Efficient fine-tuning of CLIP** While we aim to fully adapt CLIP to the downstream task through fine-tuning its image and text encoders, fine-tuning such foundation models can scale up to hundreds of millions of parameters, being computationally expensive and memory-intensive. On the other hand, freezing some of its layers, not only would be more efficient but also can help CLIP preserve its original embedding space, allowing it to be more robust to overfitting. To this end, we extensively investigate which layers should be frozen within CLIP [11], among examining various approaches for fine-tuning pre-trained models. We provide a detailed analysis of our exploration in Sec. 4.4.

# 4. Experiments

## 4.1. Datasets and Evaluation

We train our model on the COCO-Stuff [3], which has 118k densely annotated training images with 171 categories, following [30]. We employ the mean Intersection-over-Union (mIoU) as the evaluation metric for all experiments. For the evaluation, we conducted experiments on two different sets of datasets [12, 40, 65]: a commonly used in-domain datasets [14], and a multi-domain evaluation set [1] containing domain-specific images and class labels.

**Datasets for standard benchmarks.** For in-domain evaluation, we evaluate our model on ADE20K [65], PASCAL VOC [12], and PASCAL-Context [40] datasets. ADE20K has 20k training and 2k validation images, with two sets of categories: A-150 with 150 frequent classes and A-847 with 847 classes [9]. PASCAL-Context contains 5k training and validation images, with 459 classes in the full version (PC-459) and the most frequent 59 classes in the PC-59 version. PASCAL VOC has 20 object classes and a background class, with 1.5k training and validation images. We report PAS-20 using 20 object classes. We also report the score for PAS-20$^b$, which defines the “background” as classes present in PC-59 but not in PAS-20, as in Ghiasi et al. [14].

**Datasets for multi-domain evaluation.** We conducted a multi-domain evaluation on the MESS benchmark [1], specifically designed to stress-test the real-world applicability of open-vocabulary models with 22 datasets. The benchmark includes a wide range of domain-specific datasets from fields such as earth monitoring, medical sciences, engineering, agriculture, and biology. Additionally, the benchmark contains a diverse set of general domains, encompassing driving scenes, maritime scenes, paintings, and body parts. We report the average scores for each domain in the main text for brevity. For the complete results and details of the 22 datasets, please refer to the supplementary material.

## 4.2. Implementation Details

We train the CLIP image encoder and the cost aggregation module with per-pixel binary cross-entropy loss. We set $d_F = 128$, $N_B = 2$, $N_U = 2$ for all of our models. We implement our work using PyTorch [41] and Detectron2 [53]. AdamW [35] optimizer is used with a learning rate of $2\cdot10^{-4}$ for our model and $2\cdot10^{-6}$ for the CLIP, with weight decay set to $10^{-4}$. The batch size is set to 4. We use 4 NVIDIA RTX 3090 GPUs for training. All of the models are trained for 80k iterations.

## 4.3. Main Results

**Results of standard benchmarks.** The evaluation of standard open-vocabulary semantic segmentation benchmarks is shown in Table 1. Overall, our method significantly outperforms all competing methods, including those [14, 30] that leverage additional datasets [6, 42] for further performance improvements. To ensure a fair comparison, we categorize the models based on the scale of the vision-language models (VLMs) they employ. First, we present results for models that use VLMs of comparable scale to ViT-B/16 [11], and our model surpasses all previous methods, even achieving performance that matches or surpasses those using the ViT-L/14 model as their VLM [57]. For models employing the ViT-L/14 model as their VLM, our model demonstrates remarkable results, achieving a 16.0 mIoU in the challenging A-847 dataset and a 23.8 mIoU in PC-459. These results represent a 29% and 52% increase, respectively, compared to the previous state-of-the-art. We also present qualitative results of PASCAL-Context with 459 categories in Fig. 4, demonstrating the efficacy of our proposed approach in comparison to the current state-of-the-art methods [9, 30, 56].

**Results of multi-domain evaluation.** In Table 2, we present the qualitative results obtained from the MESS benchmark [1]. This benchmark assesses the real-world

| Model | VLM | Additional Backbone | Training Dataset | Additional Dataset | A-847 | PC-459 | A-150 | PC-59 | PAS-20 | PAS-20$^b$ |
|---|---|---|---|---|---|---|---|---|---|---|
| SPNet [54] | - | ResNet-101 | PASCAL VOC | ✗ | - | - | - | 24.3 | 18.3 | - |
| ZS3Net [2] | - | ResNet-101 | PASCAL VOC | ✗ | - | - | - | 19.4 | 38.3 | - |
| LSeg [28] | CLIP ViT-B/32 | ResNet-101 | PASCAL VOC-15 | ✗ | - | - | - | - | 47.4 | - |
| LSeg+ [14] | ALIGN | ResNet-101 | COCO-Stuff | ✗ | 2.5 | 5.2 | 13.0 | 36.0 | - | 59.0 |
| ZegFormer [9] | CLIP ViT-B/16 | ResNet-101 | COCO-Stuff-156 | ✗ | 4.9 | 9.1 | 16.9 | 42.8 | 86.2 | 62.7 |
| ZegFormer† [9] | CLIP ViT-B/16 | ResNet-101 | COCO-Stuff | ✗ | 5.6 | 10.4 | 18.0 | 45.5 | 89.5 | 65.5 |
| ZSseg [56] | CLIP ViT-B/16 | ResNet-101 | COCO-Stuff | ✗ | 7.0 | - | 20.5 | 47.7 | 88.4 | - |
| OpenSeg [14] | ALIGN | ResNet-101 | COCO Panoptic | ✓ | 4.4 | 7.9 | 17.5 | 40.1 | - | 63.8 |
| OVSeg [30] | CLIP ViT-B/16 | ResNet-101c | COCO-Stuff | ✓ | 7.1 | 11.0 | 24.8 | 53.3 | 92.6 | - |
| ZegCLIP [70] | CLIP ViT-B/16 | - | COCO-Stuff-156 | ✗ | - | - | - | 41.2 | 93.6 | - |
| SAN [57] | CLIP ViT-B/16 | - | COCO-Stuff | ✗ | 10.1 | 12.6 | 27.5 | 53.8 | 94.0 | - |
| CAT-Seg (ours) | CLIP ViT-B/16 | - | COCO-Stuff | ✗ | **12.0**<br>(+1.9) | **19.0**<br>(+6.4) | **31.8**<br>(+4.3) | **57.5**<br>(+3.7) | **94.6**<br>(+0.6) | **77.3**<br>(+11.8) |
| LSeg [28] | CLIP ViT-B/32 | ViT-L/16 | PASCAL VOC-15 | ✗ | - | - | - | - | 52.3 | - |
| OpenSeg [14] | ALIGN | Eff-B7 | COCO Panoptic | ✓ | 8.1 | 11.5 | 26.4 | 44.8 | - | 70.2 |
| OVSeg [30] | CLIP ViT-L/14 | Swin-B | COCO-Stuff | ✓ | 9.0 | 12.4 | 29.6 | 55.7 | 94.5 | - |
| SAN [57] | CLIP ViT-L/14 | - | COCO-Stuff | ✗ | 12.4 | 15.7 | 32.1 | 57.7 | 94.6 | - |
| ODISE [55] | CLIP ViT-L/14 | Stable Diffusion | COCO-Stuff | ✗ | 11.1 | 14.5 | 29.9 | 57.3 | - | - |
| CAT-Seg (ours) | CLIP ViT-L/14 | - | COCO-Stuff | ✗ | **16.0**<br>(+3.6) | **23.8**<br>(+8.1) | **37.9**<br>(+5.8) | **63.3**<br>(+5.6) | **97.0**<br>(+2.4) | **82.5**<br>(+12.3) |

Table 1. **Quantitative evaluation on standard benchmarks.** The best-performing results are presented in bold, while the second-best results are underlined. Improvements over the second-best are highlighted in green. †: Re-implementation trained on full COCO-Stuff.

| Model | VLM | Additional Backbone | General | Earth Monit. | Medical Sciences | Engineering | Agri. and Biology | Mean |
|---|---|---|---|---|---|---|---|---|
| *Random (LB)* | - | - | *1.17* | *7.11* | *29.51* | *11.71* | *6.14* | *10.27* |
| *Best supervised (UB)* | - | - | *48.62* | *79.12* | *89.49* | *67.66* | *81.94* | *70.99* |
| ZSSeg [56] | CLIP ViT-B/16 | ResNet-101 | 19.98 | 17.98 | 41.82 | 14.0 | 22.32 | 22.73 |
| ZegFormer [9] | CLIP ViT-B/16 | ResNet-101 | 13.57 | 17.25 | 17.47 | 17.92 | 25.78 | 17.57 |
| X-Decoder [71] | UniCL-T | Focal-T | 22.01 | 18.92 | 23.28 | 15.31 | 18.17 | 19.8 |
| OpenSeeD [62] | UniCL-B | Swin-T | 22.49 | 25.11 | **44.44** | 16.5 | 10.35 | 24.33 |
| SAN [57] | CLIP ViT-B/16 | - | 29.35 | 30.64 | 29.85 | **23.58** | 15.07 | 26.74 |
| CAT-Seg (ours) | CLIP ViT-B/16 | - | **38.69**<br>(+9.34) | **35.91**<br>(+5.27) | 28.09<br>(-16.35) | 20.34<br>(-3.24) | **32.57**<br>(+6.79) | **31.96**<br>(+5.22) |
| OVSeg [30] | CLIP ViT-L/14 | Swin-B | 29.54 | 29.04 | **31.9** | 14.16 | 28.64 | 26.94 |
| SAN [57] | CLIP ViT-L/14 | - | 36.18 | 38.83 | 30.27 | 16.95 | 20.41 | 30.06 |
| CAT-Seg (ours) | CLIP ViT-L/14 | - | **44.69**<br>(+8.51) | **39.99**<br>(+1.16) | 24.70<br>(-7.2) | **20.20**<br>(+3.25) | **38.61**<br>(+9.97) | **34.70**<br>(+4.64) |

Table 2. **Quantitative evaluation on MESS [1].** MESS includes a wide range of domain-specific datasets, which pose significant challenges due to their substantial domain differences from the training dataset. We report the average score for each domain. Please refer to the supplementary material for the results of all 22 datasets. *Random* is the result of uniform distributed prediction which represents the lower-bound, while *Best supervised* represents the upper-bound performance for the datasets.

| | Methods | A-847 | PC-459 | A-150 | PC-59 | PAS-20 | PAS-20$^b$ |
|---|---|---|---|---|---|---|---|
| **(I)** | Feature agg. + Freeze | 3.1 | 8.7 | 16.6 | 46.8 | 92.3 | 69.7 |
| **(II)** | Feature agg. + F.T. | 5.6 | 12.8 | 23.6 | 58.1 | 96.3 | 77.7 |
| **(III)** | Cost agg. + Freeze | 10.0 | 14.5 | 26.0 | 46.9 | 94.2 | 65.1 |
| **(IV)** | Cost agg. + F.T. | **14.7** | **23.2** | **35.3** | **60.3** | **96.7** | **78.9** |

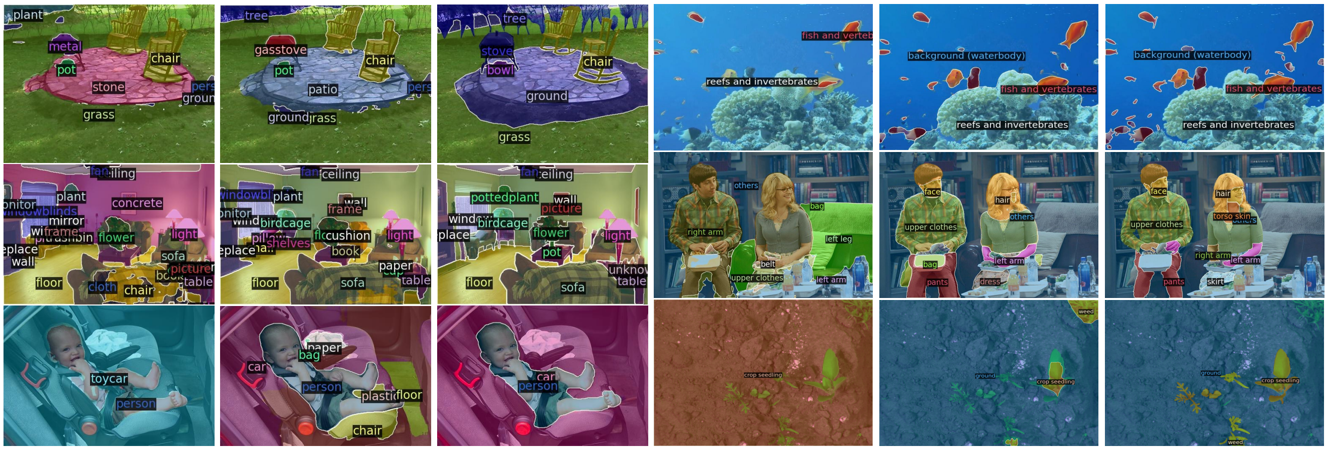
Table 3. **Quantitative comparison between feature and cost aggregation.** Cost aggregation acts as an effective alternative to direct fine-tuning of CLIP image encoder. *F.T.: Fine-Tuning.*

performance of a model across a wide range of domains. Notably, our model demonstrates a significant performance boost over other models, achieving the highest mean score. It particularly excels in the general domain as well as in agriculture and biology, showing its strong generalization ability. However, in the domains of medical sciences and engineering, the results exhibit inconsistencies with respect to the size of the VLM. Additionally, the scores for medical sciences are comparable to random predictions. We speculate that CLIP may have limited knowledge in these particular domains [43].

## 4.4. Analysis and Ablation Study

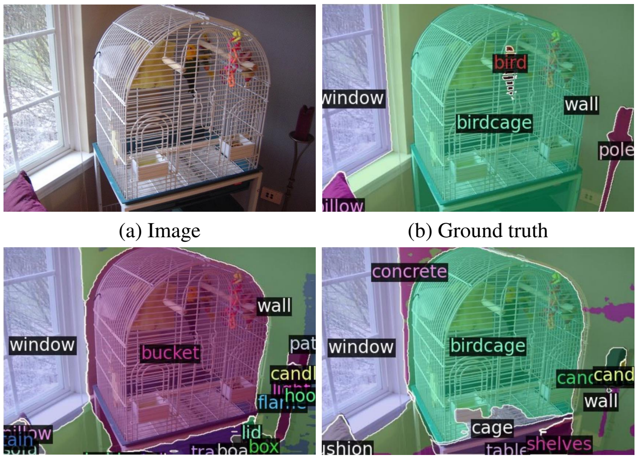
**Comparison between feature and cost aggregation.** We provide quantitative and qualitative comparison of two aggregation baselines, feature aggregation, and cost aggregation, in Table 3. For both of baseline architectures, we simply apply the upsampling decoder and note that both methods share most of the architecture, but differ in whether they aggregate the concatenated features or aggregate the cosine similarity between image and text embeddings of CLIP.

For **(I)** and **(III)**, we freeze the encoders of CLIP and only optimize the upsampling decoder. Subsequently, in **(II)** and **(IV)**, we fine-tune the encoders of CLIP on top of



(a) SAN (b) **Ours** (c) GT (d) SAN (e) **Ours** (f) GT

Figure 4. **Qualitative comparison to SAN [57].** We visualize the results of PC-459 dataset in (a-c). For (d-f), we visualize the results from the MESS benchmark [1] across three domains: underwater (top), human parts (middle), and agriculture (bottom).



(a) Image (b) Ground truth

(c) Feature aggregation (d) Cost aggregation

Figure 5. **Qualitative comparison between feature and cost aggregation.** Our approach (d) successfully segments the previously unseen class, such as "birdcage," whereas approach (c) fails.

**(I)** and **(III)**. Our results show that feature aggregation can benefit from fine-tuning, but the gain is only marginal. On the other hand, cost aggregation benefits significantly from fine-tuning, highlighting the effectiveness of cost aggregation for adapting CLIP to the task of segmentation.

For the qualitative results in Fig. 5, we show the prediction results from **(II)** and **(IV)**. As seen in Fig. 5(c-d), we observe that feature aggregation shows overfitting to the seen class of "bucket," while cost aggregation successfully identifies the unseen class "birdcage."

**Component analysis.** Table 4 shows the effectiveness of the main components within our architecture through quantitative results. First, we introduce the baseline models in **(I)** and **(II)**, identical to the fine-tuned baseline models from Table 3. We first add the proposed spatial and class aggregations to the cost aggregation baseline in **(III)** and **(IV)**, respectively. In **(V)**, we interleave the spatial and class aggregations. Lastly, we add the proposed embedding guidance to **(V)**, which becomes our final model.

| | Components | A-847 | PC-459 | A-150 | PC-59 | PAS-20 | PAS-$20^b$ |
|---|---|---|---|---|---|---|---|
| **(I)** | Feature Agg. | 5.6 | 12.8 | 23.6 | 58.1 | 96.3 | 77.7 |
| **(II)** | Cost Agg. | 14.7 | 23.2 | 35.3 | 60.3 | 96.7 | 78.9 |
| **(III)** | **(II)** + Spatial agg. | 14.9 | 23.1 | 35.9 | 60.3 | 96.7 | 79.5 |
| **(IV)** | **(II)** + Class agg. | 14.7 | 21.5 | 36.6 | 60.6 | 95.5 | 80.5 |
| **(V)** | **(II)** + Spatial and Class agg. | 15.5 | 23.2 | 37.0 | 62.3 | 96.7 | 81.3 |
| **(VI)** | **(V)** + Embedding guidance | **16.0** | **23.8** | **37.9** | **63.3** | **97.0** | **82.5** |

Table 4. **Ablation study for CAT-Seg.** We conduct ablation study by gradually adding components to the cost aggregation baseline.

| Methods | A-847 | PC-459 | A-150 | PC-59 | PAS-20 | PAS-$20^b$ |
|---|---|---|---|---|---|---|
| CAT-Seg w/o upsampling decoder | 9.9 | 16.1 | 28.4 | 52.9 | 93.2 | 73.3 |
| CAT-Seg (ours) | **12.0** | **19.0** | **31.8** | **57.5** | **94.6** | **77.3** |

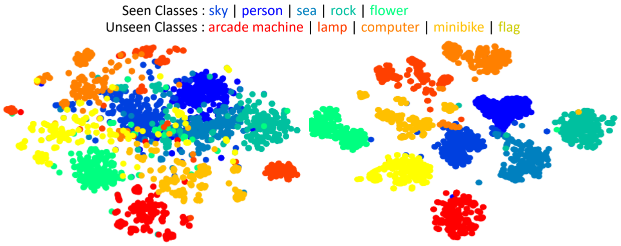
Table 5. **Ablation study of upsampling decoder.** CLIP with ViT-B is used for ablation.

As shown, we stress the gap between **(I)** and **(II)**, which supports the findings presented in Fig. 5. Given that PAS-20 shares most of its classes with the training datasets[56], the performance gap between **(I)** and **(II)** is minor. However, for challenging datasets such as A-847 or PC-459, the difference is notably significant, validating our cost aggregation framework for its generalizability. We also highlight that as we incorporate the proposed spatial and class aggregation techniques, our approach **(V)** outperforms **(II)**, demonstrating the effectiveness of our design. Finally, **(VI)** shows that our embedding guidance further improves performance across all the benchmarks. Furthermore, we provide quantitative results of adopting the upsampling decoder in Table 5. The results show consistent improvements across all the benchmarks.

**Analysis on fine-tuning of CLIP.** In this section, we analyze the effects and methods of fine-tuning of the encoders of CLIP. In Table 6, we report the results of different approaches, which include the variant **(I)**: without fine-tuning,

| | Methods | A-847 | PC-459 | A-150 | PC-59 | PAS-20 | PAS-20$^b$ | #param. (M) | Memory (GiB) |
|---|---|---|---|---|---|---|---|---|---|
| **(I)** | Freeze | 10.4 | 15.0 | 31.8 | 52.5 | 92.2 | 71.3 | 5.8 | 20.0 |
| **(II)** | Prompt | 8.8 | 14.3 | 30.5 | 55.8 | 93.2 | 74.7 | 7.0 | 20.9 |
| **(III)** | Full F.T. | 13.6 | 22.2 | 34.0 | 61.1 | **97.3** | 79.7 | 393.2 | 26.8 |
| **(IV)** | Attn. F.T. | 15.7 | 23.7 | 37.1 | 63.1 | 97.1 | 81.5 | 134.9 | 20.9 |
| **(V)** | QK F.T. | 15.3 | 23.0 | 36.3 | 62.0 | 95.9 | 81.9 | 70.3 | 20.9 |
| **(VI)** | KV F.T. | **16.1** | **23.8** | 37.6 | 62.4 | 96.7 | 82.0 | 70.3 | 20.9 |
| **(VII)** | QV F.T. (Img.) | 13.9 | 22.8 | 35.1 | 62.0 | 96.3 | 82.0 | 56.7 | 20.9 |
| **(VIII)** | QV F.T. (Txt.) | 14.7 | 22.2 | 35.1 | 60.0 | 95.8 | 80.3 | 19.9 | 20.0 |
| **(IX)** | QV F.T. (Both) | 16.0 | **23.8** | **37.9** | **63.3** | 97.0 | **82.5** | 70.3 | 20.9 |

Table 6. **Analysis of fine-tuning methods for CLIP.** We additionally note the number of learnable parameters of CLIP and memory consumption during training. Our method not only outperforms full fine-tuning, but also requires smaller computation.



(a) CLIP

(b) Fine-tuned CLIP

Figure 6. **Effects of fine-tuning CLIP.** We show the t-SNE [50] visualization of CLIP image embeddings based on its predictions. In contrast to (a), we observe well-grouped clusters in (b), showing the adaptation of CLIP to segmentation for both seen and unseen classes.

**(II)**: adopting Prompt Tuning [23, 68], **(III)**: fine-tuning the entire CLIP, **(IV)**: fine-tuning the attention layer only [49], **(V)**: fine-tuning query and key projections only, **(VI)**: fine-tuning key and value projections only, **(VII)**: our approach for CLIP image encoder only, **(VIII)**: our approach for text encoder only, and **(IX)**: our approach for both encoders. Note that both image and text encoders are fine-tuned in **(I-VI)**. Overall, we observed that fine-tuning enhances the performance of our framework. Among the various fine-tuning methods, fine-tuning only the query and value projection yields the best performance improvement while also demonstrating high efficiency. Additionally, as can be seen in **(VII-IX)**, fine-tuning both encoders leads to better performance compared to fine-tuning only one of them in our framework.

In Fig. 6, we show the t-SNE [50] visualization of the dense image embeddings of CLIP within the A-150 [65] dataset. We color the embeddings based on the prediction with text classes. From (a), we can observe that the clusters are not well-formed for each classes, due to the image-level training of CLIP. In contrast, we observe well-formed clusters in (b) for both seen and unseen classes, showing the adaptation of CLIP for the downstream task.

**Training with various datasets.** In this experiment, we further examine the generalization power of our method in comparison to other methods [9, 56] by training our model on smaller-scale datasets, which include A-150 and PC-59, that poses additional challenges to achieve good performance. The results are shown in Table 7. As shown, we find that although we observe some performance drops, which seem quite natural when a smaller dataset is used, our work significantly outperforms other competitors. These results highlight the strong generalization power of our framework, a favorable characteristic that suggests the practicality of our approach.

| Methods | Training dataset | A-847 | PC-459 | A-150 | PC-59 | PAS-20 | PAS-20$^b$ |
|---|---|---|---|---|---|---|---|
| ZegFormer | COCO-Stuff | 5.6 | 10.4 | 18.0 | 45.5 | 89.5 | 65.5 |
| ZSseg | COCO-Stuff | 7.0 | 9.0 | 20.5 | 47.7 | 88.4 | 67.9 |
| CAT-Seg (ours) | COCO-Stuff | **12.0** | **19.0** | **31.8** | **57.5** | **94.6** | **77.3** |
| ZegFormer | A-150 | 6.8 | 7.1 | 33.1 | 34.7 | 77.2 | 53.6 |
| ZSseg | A-150 | 7.6 | 7.1 | 40.3 | 39.7 | 80.9 | 61.1 |
| CAT-Seg (ours) | A-150 | **14.4** | **16.2** | 47.7 | **49.9** | **91.1** | **73.4** |
| ZegFormer | PC-59 | 3.8 | 8.2 | 13.1 | 48.7 | 86.5 | 66.8 |
| ZSseg | PC-59 | 3.0 | 7.6 | 11.9 | 54.7 | 87.7 | 71.7 |
| CAT-Seg (ours) | PC-59 | **9.6** | **16.7** | **27.4** | 63.7 | **93.5** | **79.9** |

Table 7. **Training on various datasets.** CLIP with ViT-B is used for all methods. Our model demonstrates remarkable generalization capabilities even on relatively smaller datasets. The scores evaluated on the same dataset used for training are colored in gray.

| Methods | ZegFormer | ZSSeg | OVSeg | CAT-Seg (Ours) |
|---|---|---|---|---|
| # of learnable params. (M) | 103.3 | 102.8 | 408.9 | **70.3** |
| # of total params. (M) | 531.2 | 530.8 | 532.6 | **433.7** |
| Training time (min) | 1,148.3 | 958.5 | - | **875.5** |
| Inference time (s) | 2.70 | 2.73 | 2.00 | **0.54** |
| Inference GFLOPs | 19,425.6 | 22,302.1 | 19,345.6 | **2,121.1** |

Table 8. **Efficiency comparison.** All results are measured with a single RTX 3090 GPU.

**Efficiency comparison.** In Table 8, we thoroughly compare the efficiency of our method to recent methods [9, 30, 56]. We measure the number of learnable parameters, the total number of parameters, training time, inference time, and inference GFLOPs. Our model demonstrates strong efficiency in terms of both training and inference. This efficiency is achieved because our framework does not require an additional mask generator [9].

## 5. Conclusion

In conclusion, we introduce a cost aggregation framework for open-vocabulary semantic segmentation, aggregating the cosine-similarity scores between image and text embeddings of CLIP. Through our CAT-Seg framework, we fine-tune the encoders of CLIP for its adaptation for the downstream task of segmentation. Our method surpasses the previous state-of-the-art in standard benchmarks and also in scenarios with a vast domain difference. The success in diverse domains underscores the promise and potential of our cost aggregation framework in advancing the field of open-vocabulary semantic segmentation.

**Acknowledgement.** This research was supported by the MSIT, Korea (IITP-2023-2020-0-01819, RS-2023-00266509).

## References

[1] Benedikt Blumenstiel, Johannes Jakubik, Hilde Kühne, and Michael Vössing. What a mess: Multi-domain evaluation of zero-shot semantic segmentation. *arXiv preprint arXiv:2306.15521*, 2023. 2, 5, 6, 7

[2] Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. *Advances in Neural Information Processing Systems*, 32, 2019. 2, 6

[3] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018. 5

[4] Changjiang Cai, Matteo Poggi, Stefano Mattoccia, and Philippos Mordohai. Matching-space stereo networks for cross-domain generalization. In *2020 International Conference on 3D Vision (3DV)*, pages 364–373. IEEE, 2020. 2

[5] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5418, 2018. 3

[6] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 5

[7] Seokju Cho, Sunghwan Hong, Sangryul Jeon, Yunsung Lee, Kwanghoon Sohn, and Seungryong Kim. Cats: Cost aggregation transformers for visual correspondence. *Advances in Neural Information Processing Systems*, 34:9011–9023, 2021. 3, 4

[8] Seokju Cho, Sunghwan Hong, and Seungryong Kim. Cats++: Boosting cost aggregation with convolutions and transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 3, 4

[9] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11583–11592, 2022. 1, 2, 5, 6, 8

[10] Zheng Ding, Jieke Wang, and Zhuowen Tu. Open-vocabulary panoptic segmentation with maskclip. *arXiv preprint arXiv:2208.08984*, 2022. 2

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4, 5

[12] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–308, 2009. 5

[13] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, pages 1–15, 2023. 3

[14] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVI*, pages 540–557. Springer, 2022. 1, 2, 5, 6

[15] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3273–3282, 2019. 3

[16] Junjun He, Zhongying Deng, Lei Zhou, Yali Wang, and Yu Qiao. Adaptive pyramid context network for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7519–7528, 2019. 3

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3

[18] Sunghwan Hong, Seokju Cho, Jisu Nam, Stephen Lin, and Seungryong Kim. Cost aggregation with 4d convolutional swin transformer for few-shot segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX*, pages 108–126. Springer, 2022. 3, 4

[19] Sunghwan Hong, Jisu Nam, Seokju Cho, Susung Hong, Sangryul Jeon, Dongbo Min, and Seungryong Kim. Neural matching fields: Implicit representation of matching fields for visual correspondence. *arXiv preprint arXiv:2210.02689*, 2022. 3

[20] Asmaa Hosni, Christoph Rhemann, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *PAMI*, 2012. 5

[21] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer: A transformer architecture for optical flow. *arXiv preprint arXiv:2203.16194*, 2022. 3

[22] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 1

[23] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pages 709–727. Springer, 2022. 8

[24] Zhenchao Jin, Tao Gong, Dongdong Yu, Qi Chu, Jian Wang, Changhu Wang, and Jie Shao. Mining contextual information beyond image for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7231–7241, 2021. 3

[25] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020. 4

[26] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry.

End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE international conference on computer vision*, pages 66–75, 2017. 1, 3

[27] Junha Lee, Seungwook Kim, Minsu Cho, and Jaesik Park. Deep hough voting for robust global registration. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15994–16003, 2021. 3

[28] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022. 2, 6

[29] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, pages 280–296. Springer, 2022. 4

[30] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. *arXiv preprint arXiv:2210.04150*, 2022. 1, 2, 5, 6, 8

[31] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 4

[32] Biyang Liu, Huimin Yu, and Guodong Qi. Graftnet: Towards domain generalized stereo matching with a broad-spectrum and task-oriented feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13012–13021, 2022. 2, 3

[33] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 3, 4

[34] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 3

[35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[36] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 2

[37] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998. 2

[38] Juhong Min and Minsu Cho. Convolutional hough matching networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2940–2950, 2021. 3

[39] Juhong Min, Dahyun Kang, and Minsu Cho. Hypercorrelation squeeze for few-shot segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6941–6952, 2021. 3

[40] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 891–898, 2014. 5

[41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5

[42] Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with localized narratives. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 647–664. Springer, 2020. 5

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 6

[44] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6148–6157, 2017. 3

[45] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 2

[46] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47:7–42, 2002. 5

[47] Xiao Song, Guorun Yang, Xinge Zhu, Hui Zhou, Zhe Wang, and Jianping Shi. Adastereo: A simple and efficient approach for adaptive stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10328–10337, 2021. 2, 3

[48] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 5

[49] Hugo Touvron, Matthieu Cord, Alaaeldin El-Nouby, Jakob Verbeek, and Hervé Jégou. Three things everyone should know about vision transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pages 497–515. Springer, 2022. 8

[50] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 8

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3, 4

[52] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok

Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7959–7971, 2022. 3

[53] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 5

[54] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero-and few-label semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8256–8265, 2019. 2, 6

[55] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2955–2966, 2023. 1, 2, 6

[56] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX*, pages 736–753. Springer, 2022. 2, 5, 6, 7, 8

[57] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2945–2954, 2023. 1, 2, 3, 5, 6, 7

[58] Gengshan Yang, Joshua Manela, Michael Happold, and Deva Ramanan. Hierarchical deep stereo matching on high-resolution images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5515–5524, 2019. 3

[59] Changqian Yu, Jingbo Wang, Changxin Gao, Gang Yu, Chunhua Shen, and Nong Sang. Context prior for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12416–12425, 2020. 3

[60] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. *arXiv preprint arXiv:2308.02487*, 2023. 1, 2, 3

[61] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 173–190. Springer, 2020. 3

[62] Hao Zhang, Feng Li, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianfeng Gao, Jianwei Yang, and Lei Zhang. A simple framework for open-vocabulary segmentation and detection. *arXiv preprint arXiv:2303.08131*, 2023. 6

[63] Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 3

[64] Hang Zhao, Xavier Puig, Bolei Zhou, Sanja Fidler, and Antonio Torralba. Open vocabulary scene parsing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2002–2010, 2017. 2

[65] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019. 5, 8

[66] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *ECCV*, pages 696–712. Springer, 2022. 1, 2, 3

[67] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. 3

[68] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 3, 8

[69] Tianfei Zhou, Wenguan Wang, Ender Konukoglu, and Luc Van Gool. Rethinking semantic segmentation: A prototype view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2582–2593, 2022. 3

[70] Ziqin Zhou, Bowen Zhang, Yinjie Lei, Lingqiao Liu, and Yifan Liu. Zegclip: Towards adapting clip for zero-shot semantic segmentation. *arXiv preprint arXiv:2212.03588*, 2022. 2, 6

[71] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15116–15127, 2023. 6