Open-Vocabulary Semantic Segmentation with Image Embedding Balancing

Xiangheng Shan, Dongyue Wu, Guilin Zhu, Yuanjie Shao*, Nong Sang, Changxin Gao
National Key Laboratory of Multispectral Information Intelligent Processing Technology,
School of Artificial Intelligence and Automation, Huazhong University of Science and Technology
{xianghengshan, dongyue_wu, gzhu, shaoyuanjie, nsang, cgao}@hust.edu.cn

Abstract

Open-vocabulary semantic segmentation is a challenging task, which requires the model to output semantic masks of an image beyond a close-set vocabulary. Although many efforts have been made to utilize powerful CLIP models to accomplish this task, they are still easily overfitting to training classes due to the natural gaps in semantic information between training and new classes. To overcome this challenge, we propose a novel framework for open-vocabulary semantic segmentation called EBSeg, incorporating an Adaptively Balanced Decoder (AdaB Decoder) and a Semantic Structure Consistency loss (SSC Loss). The AdaB Decoder is designed to generate different image embeddings for both training and new classes. Subsequently, these two types of embeddings are adaptively balanced to fully exploit their ability to recognize training classes and generalization ability for new classes. To learn a consistent semantic structure from CLIP, the SSC Loss aligns the inter-classes affinity in the image feature space with that in the text feature space of CLIP, thereby improving the generalization ability of our model. Furthermore, we employ a frozen SAM image encoder to complement the spatial information that CLIP features lack due to the low training image resolution and image-level supervision inherent in CLIP. Extensive experiments conducted across various benchmarks demonstrate that the proposed EBSeg outperforms the state-of-the-art methods. Our code and trained models will be here: https://github.com/slonetime/EBSeg.

1. Introduction

Semantic segmentation is an important computer vision task that requires the model to identify a class label for each pixel in an image. For traditional (or fully supervised) semantic segmentation, models are trained and evaluated on a fixed dataset with a specific set of classes, and the test set has the same classes and image distribution as the training set. For this task, many effective methods [3, 5, 6, 13, 21, 23, 26, 35, 36] have been proposed, and they have significantly improved the prediction accuracy. However, models designed for this task always fail to segment images in the real world well, because they were only trained on specific datasets with fixed sets of classes.

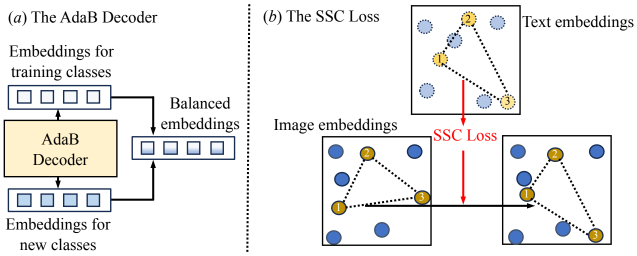

Figure 1. Illustration of our main idea. (*a*) Our AdaB Decoder(Adaptively Balanced Decoder) outputs image embeddings for both training classes(classes existing in the training set during test) and new classes(classes not existing in training set). By adaptively balancing these embeddings, our model performs better at both training and new classes. (*b*) We propose SSC Loss(Semantic Structure Consistency loss) that aligns the distribution of the image embeddings with that of the text embeddings. The SSC Loss helps our model learn the semantic structure of CLIP better and achieve better generalization capability for new classes.

To address this problem, the open-vocabulary semantic segmentation task was introduced. In this task, given an image and an arbitrary set of classes, the model is expected to classify each pixel into its most corresponding class. This task is much closer to the real-world applications. Many works utilize the CLIP [29] model for this task because CLIP was trained on a large-scale image-text pair dataset and has strong generalization capability for open-vocabulary tasks.

Some existing works like [10, 17] propose to finetune models on semantic segmentation datasets. However, models finetuned on a semantic segmentation dataset often overfit to the training classes. Some others [7, 20, 38] adopt a two-stage framework. Firstly, a category-agnostic mask generator is used to extract masks. Then, the masks are used

[*] Corresponding author.

to crop the input image to get many crops which will be fed into a frozen CLIP model for classification results. This framework incurs high computational costs as the CLIP model has to process many crops of the image. Since this framework does not leverage the discriminate features from CLIP in the mask generation process and lacks context in the classification process, it gets sub-optimal results.

Recently, some new methods adopting one-stage frameworks have emerged. ODISE [37] employs a frozen diffusion model [31] to extract image features as the input for a Mask2former [6] head. It [37] leverages the strengths of both the diffusion model and the CLIP model to accomplish the open vocabulary segmentation task. However, ODISE faces high computational costs, as the diffusion model is very large (with about 1.5 billion parameters). MaskCLIP [8] also uses a category-agnostic mask generator like [7, 20, 38] but it [8] does not crop the input image. MaskCLIP [8] proposes a Relative Mask Attention module that applies the masks in the self-attention layers of CLIP image encoder as attention masks to produce mask attention embeddings for mask classification. SAN [39] adds a lightweight image encoder to get masks and attention masks corresponding to each mask. Like MaskCLIP [8], the attention masks are fed to the last few layers of CLIP to obtain mask attention embeddings for mask classification.

Although these methods are effective, they still face a challenge. The challenge is that training on a specific semantic segmentation dataset often makes the model overfit to the training classes, impairing the generalization ability of the model, especially for large models.

To overcome the challenge, we present EBSeg (image Embedding Balancing for open-vocabulary semantic Segmentation). It consists of the Adaptively Balanced Decoder (AdaB Decoder) and the Semantic Structure Consistency loss (SSC Loss). The AdaB Decoder generates mask attention embeddings, fully supervised embeddings specialized for training classes and frozen embeddings with excellent generalization for new classes. The mask attention embeddings come from a Mask Attention module introduced in MaskCLIP [8] and SAN [39]. The fully supervised embeddings are directly supervised by a cross-entropy loss with the training classes. The frozen embeddings are extracted from a frozen CLIP image encoder. These three types of embeddings are then adaptively balanced to form a final representation of the input image for the final prediction. Thus, AdaB Decoder could take full advantage of the superior features learned on training classes and the excellent generalizing ability on new classes at the same time. On the other hand, SSC Loss aims at aligning the class-level affinity between the image features and text features. Hence, our model could encode a more consistent class-level semantic structure from the CLIP feature space to enhance the generalization on new classes. Additionally, we utilize a frozen SAM [15] image encoder to complement the spatial information of CLIP features to address the issue that image feature maps of CLIP lack important spatial details for semantic segmentation.

We conduct extensive experiments on challenging open-vocabulary segmentation datasets to prove the effectiveness of our method EBSeg. Following previous works [20, 38, 39], we train our model on COCO-Stuff [2] and evaluate the model on VOC [9], Pascal Context-59 [25], Pascal Context-459 [25], ADE20K-150 [42] and ADE20K-847 [42]. Our method achieves state-of-the-art results, with an average of 2.3% mIoU improvements on these 5 benchmarks when using the CLIP ViT-B/16 model and an average of 2.3% mIoU improvements with the CLIP ViT-L/14 model.

Our contributions are as follows:

- We propose Adaptively Balanced Decoder (AdaB Decoder). By adaptively balancing different image embeddings, AdaB Decoder can fully leverage their ability to recognize training classes and generalization capability for new classes that do not exist in the training set.
- We introduce the Semantic Structure Consistency loss (SSC Loss). The SSC Loss aligns the inter-classes affinity in the image feature space with that in the text feature space of CLIP. This loss helps our model learn a consistent semantic structure from CLIP and improves the generalization ability of our model.
- In our model, we propose to fuse the image features of SAM and CLIP to complement the spatial information of CLIP image features.
- Our method EBSeg establishes a new state-of-the-art in the open-vocabulary semantic segmentation task.

2. Related works

CLIP and its transfer learning on downstream tasks. CLIP [29] is proposed to align images and texts in a shared semantic space, enabling cross-modal understanding and transfer learning. It is trained on a large dataset of image-text pairs with contrastive loss to get a strong open-vocabulary recognition ability. CLIP can be directly used in the zero-shot image classification task.

After CLIP was released, many works have explored using it in various downstream tasks. For few-shot image classification, CoOp [44] and CoCoOp [43] use prompt tuning [16, 18, 41] to adapt the text embeddings of CLIP to task-specific classes with relatively low cost. For fully supervised dense prediction, DenseCLIP [30] directly finetunes the CLIP model and proposes a vision-to-language prompting method to leverage the prior knowledge of image contexts. CLIP Surgery [19] explores using surgery-like modifications for CLIP inference architecture and features, achieving good explainability and enhancement in multiple open-vocabulary tasks.

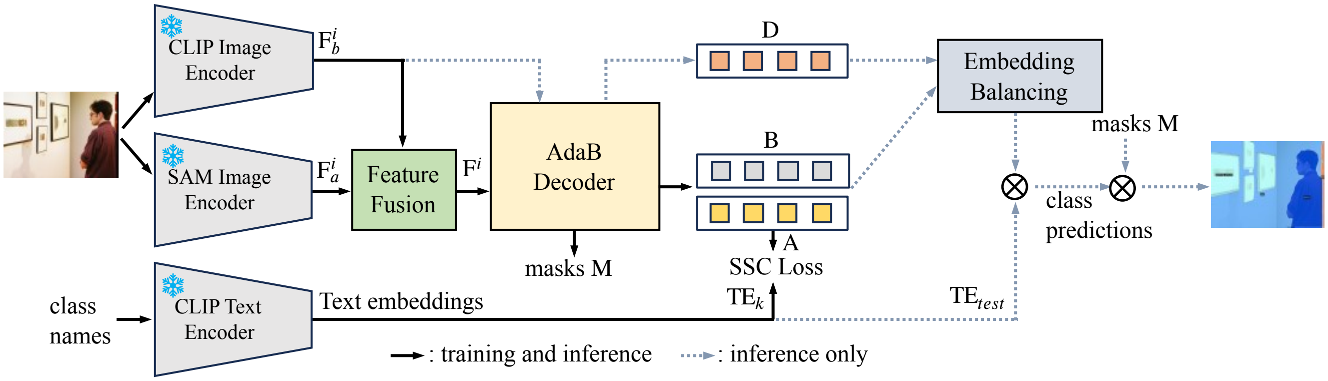

Figure 2. The architecture of our model EBSeg. We first obtain image features from two frozen image encoders and fuse them in a feature fusion module. After that, the fused features are input into our AdaB Decoder, which outputs masks $\mathbf{M}$ and image embeddings (including mask attention embeddings $\mathbf{B}$, fully supervised embeddings $\mathbf{A}$ and frozen embeddings $\mathbf{D}$). During training, we apply the SSC Loss to learn a consistent semantic structure from CLIP. During inference, we adaptively balance the three embeddings output by AdaB Decoder and obtain semantic segmentation results with the masks, balanced image embeddings, and text embeddings.

Our work explores how to fully leverage the powerful image-text alignment capability of CLIP in the challenging task of open-vocabulary semantic segmentation.

Open-vocabulary semantic segmentation. Some early works [1, 34, 40] on this task focused on how to project image features and text features into a shared feature space, which is hard because images and texts are in two different modalities. Recently, benefiting from the powerful open-vocabulary recognition ability of CLIP, many works have attempted to apply the CLIP model on this task. [7, 20, 38] adopt a two-stage framework, where a mask generator is leveraged to extract category-agnostic masks. Then the masks are used to get many crops of the input image and the crops will be fed into CLIP for mask classification results. MaskCLIP [8] also uses a category-agnostic mask generator, but it does not use the masks to crop the input image. MaskCLIP [8] proposes a Relative Mask Attention module where it uses the masks as attention masks in the self-attention layers of CLIP image encoder to get mask attention embeddings. ODISE [37] explores using a frozen diffusion model [31] to extract image features as the input for a Mask2former [6] head. SAN [39] adds a lightweight image encoder to get masks and attention masks corresponding to each mask. Like MaskCLIP [8], the attention masks are fed to the last few layers of CLIP to obtain mask attention embeddings for mask classification. Different from MaskCLIP [8], the attention masks in SAN [39] are per-head, which means SAN [39] produces different attention masks for each attention head in CLIP image encoder.

Our work explores how to overcome the overfitting challenge faced by CLIP-based methods. To achieve this goal, we design the AdaB Decoder and SSC Loss for better generalization on new classes.

3. Method

3.1. Method Overview

In Fig. 2, we present the architecture of our open-vocabulary semantic segmentation model EBSeg. In this framework, we first input the image into both frozen CLIP and SAM image encoders. The obtained image features from the two image encoders are then fused in a fusion module (Sec. 3.2). After that, the fused features are fed into our AdaB Decoder which outputs masks and image embeddings (including mask attention embeddings $\mathbf{B}$, fully supervised embeddings $\mathbf{A}$ and frozen embeddings $\mathbf{D}$) (Sec. 3.3). In our model, we apply the Semantic Structure Consistency loss (SSC Loss) to learn a consistent semantic structure from CLIP (Sec. 3.4). During inference, we propose to adaptively balance the three different image embeddings to fully exploit their ability to recognize training classes and the excellent generalization ability for new classes (Sec. 3.5). Finally, we obtain semantic segmentation results with the masks, balanced image embeddings, and text embeddings.

3.2. Image Feature Extraction and Fusion

As mentioned before, the feature maps of CLIP lack important spatial information for semantic segmentation. Therefore, we propose to utilize a frozen SAM image encoder to complement the spatial information.

Given an image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, we input it into the SAM image encoder and get image features $\mathbf{F}_a^i \in \mathbb{R}^{h_a\times w_a\times C_a}(i = 1, 2, 3)$ from the last three global attention blocks. Meanwhile, we downsample $\mathbf{I}$ to $\mathbf{I}^{'} \in \mathbb{R}^{\frac{H}{p}\times\frac{W}{p}\times 3}$ (where p is the downsample ratio). We then feed $\mathbf{I}^{'}$ into CLIP image encoder to get image features $\mathbf{F}_b^i \in \mathbb{R}^{h_b\times w_b\times C_b}(i = 1, 2, 3)$ from the number $L/4, L/2$, and $3L/4$ blocks of CLIP (L is the total number of blocks in

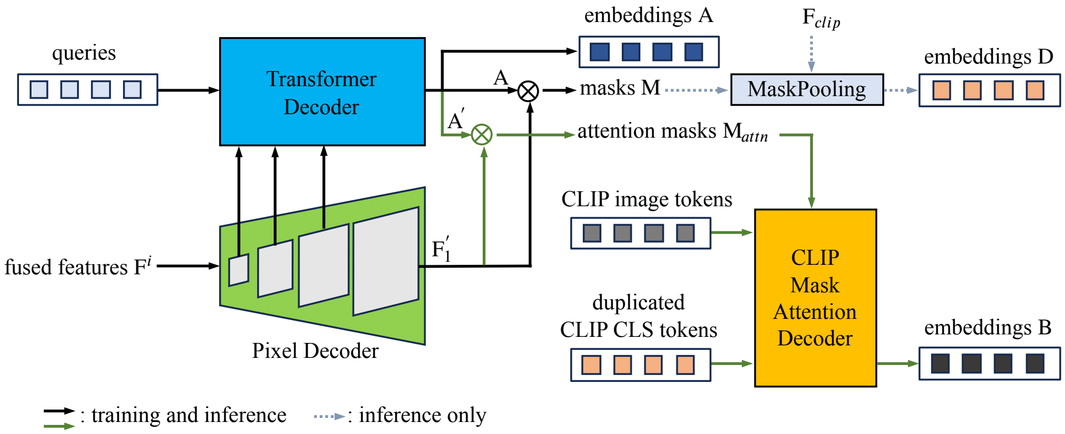

Figure 3. Detailed structure of AdaB Decoder. We first input fused image features into the Pixel Decoder. The outputs of the first three stages are then fed to the Transformer Decoder which outputs image embeddings $\mathbf{A}$ and $\mathbf{A}'$. Then we obtain masks $\mathbf{M}$ with $\mathbf{A}$ and the largest feature map $\mathbf{F}'_1$ from the Pixel Decoder. We obtain per-head attention masks $\mathbf{M}_{attn}$ with per-head embeddings $\mathbf{A}'$ and $\mathbf{F}'_1$. Finally, we perform masked self-attention in the last few blocks of CLIP image encoder with $\mathbf{M}_{attn}$ to get mask attention embeddings $\mathbf{B}$.

CLIP image encoder; $L = 12$ for CLIP ViT-B and $L = 24$ for CLIP ViT-L).

We employ a simple addition approach to fuse the image features from two frozen models. Firstly, we use a linear layer to match the channel number of $\mathbf{F}_b$ to that of $\mathbf{F}_a$. Then, we upsample or downsample $\mathbf{F}_a$ and $\mathbf{F}_b$ to $\mathbf{F}_v^i \in \mathbb{R}^{\frac{H}{s}\times\frac{W}{s}\times C_a}(v = a, b; i = 1, 2, 3; s = 2^{i+2})$. Finally we perform element-wise addition with $\mathbf{F}_a$ and $\mathbf{F}_b$, obtaining fused features $\mathbf{F}^i \in \mathbb{R}^{\frac{H}{s}\times\frac{W}{s}\times C_a}(i = 1, 2, 3; s = 2^{i+2})$:

$$\mathbf{F}^i = \mathbf{F}_a^i + \mathbf{F}_b^i. \tag{1}$$

3.3. AdaB Decoder

In this section, we show the detailed architecture of our AdaB Decoder (Fig. 3). Note that we will present how to adaptively balance the image embeddings later in Sec. 3.5.

The AdaB Decoder consists of three components: Pixel Decoder, Transformer Decoder, and CLIP Mask Attention Decoder. The Pixel Decoder and Transformer Decoder follow Mask2former [6], and the CLIP Mask Attention Decoder follows the mask attention module in SAN [39].

Similar to Mask2former [6], we input the fused multi-scale image features $\mathbf{F}^i$ into the Pixel Decoder. The outputs of the first three stages in Pixel Decoder are then fed into the Transformer Decoder, from which we get the fully supervised mask image embeddings $\mathbf{A} \in \mathbb{R}^{N\times C}$ (N is the number of queries of the Transformer Decoder). We then perform matrix multiplication between $\mathbf{A}$ and the largest feature map $\mathbf{F}'_1 \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times C}$ from Pixel Decoder to obtain masks $\mathbf{M} \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times N}$:

$$\mathbf{M} = \mathbf{F}'_1 \times \mathbf{A}^T. \tag{2}$$

Furthermore, we increase the channel numbers of $\mathbf{A}$ to $C \times n$ using a linear layer (n is the number of heads in the multi-head self-attention layers of CLIP image encoder), getting per-head image embeddings $\mathbf{A}' \in \mathbb{R}^{N\times(C\times n)}$. We then perform matrix multiplication between $\mathbf{A}'$ and $\mathbf{F}'_1$, obtaining per-head attention masks $\mathbf{M}_{attn} \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times n\times N}$ which will be used in the CLIP Mask Attention Decoder.

In the CLIP Mask Attention Decoder, we use the last l ($l = \frac{L}{4}$) transformer blocks from CLIP image encoder. Firstly, we get the output tokens $\mathbf{U}$ from the last $(l + 1)$th block of CLIP image encoder. Then we duplicate the CLS token of $\mathbf{U}$ N times as queries for the CLIP Mask Attention Decoder. Then, we concatenate the queries and $\mathbf{U}$ together and input them into the CLIP Mask Attention Decoder, with $\mathbf{M}_{attn}$ as the per-head attention masks for the multi-head self-attention layers of those l CLIP transformer blocks. The output of CLIP Mask Attention Decoder is mask attention embeddings $\mathbf{B} \in \mathbb{R}^{N\times C}$.

During inference, to improve the recognition ability for new classes that do not exist in the training set, we perform mask pooling with $\mathbf{M}$ and the final output $\mathbf{F}_{clip} \in \mathbb{R}^{\frac{H}{16}\times\frac{W}{16}\times C}$ of CLIP image encoder to get frozen image embeddings $\mathbf{D} \in \mathbb{R}^{N\times C}$:

$$\mathbf{D} = MaskPooling(\mathbf{M}, \mathbf{F}_{clip}). \tag{3}$$

3.4. SSC Loss

In this section, we will introduce the Semantic Structure Consistency Loss (SSC Loss). Our SSC Loss draws inspiration from unsupervised segmentation methods like STEGO [11]. Works like STEGO [11] mainly focus on the unsupervised segmentation task and usually aim to distill image feature semantic similarity from a pretrained image encoder. Designed for OVSS, our SSC loss is proposed to enhance generalization ability for new classes by distilling semantic similarity from text features output by frozen CLIP text encoder to image features. By introducing the SSC Loss, our model learns more about the latent knowledge from the CLIP feature space. Thus our model can learn a consistent semantic structure from CLIP and gain a stronger generalization ability for new classes that do not exist in the training set.

During training, assuming that an image has k ground truth masks and classes, after Hungarian matching, we match these k ground truth masks with k prediction masks (from $\mathbf{M}$) and k image embeddings $\mathbf{IE}_k \in \mathbb{R}^{k\times C}$ (from $\mathbf{A}$).

After that, we calculate the cosine similarities $\mathbf{CS}^{i,j}_{text} \in \mathbb{R}^1$ between the text embeddings $\mathbf{TE}_k \in \mathbb{R}^{k\times C}$ (produced by the CLIP text encoder) of the k ground truth classes:

$$\mathbf{CS}^{i,j}_{text} = \frac{\mathbf{TE}^i_k \cdot \mathbf{TE}^j_k}{|\mathbf{TE}^i_k| * |\mathbf{TE}^j_k|}, \tag{4}$$

where $\mathbf{TE}^i_k$, $\mathbf{TE}^j_k \in \mathbb{R}^{1\times C}$ from $\mathbf{TE}_k$, i, $j \le k$, and $\cdot$ denotes dot product. Then, we calculate the cosine similarities $\mathbf{CS}^{i,j}_{image} \in \mathbb{R}^1$ between the matched k image embeddings $\mathbf{IE}_k \in \mathbb{R}^{k\times C}$ (from $\mathbf{A} \in \mathbb{R}^{N\times C}$):

$$\mathbf{CS}^{i,j}_{image} = \frac{\mathbf{IE}^i_k \cdot \mathbf{IE}^j_k}{|\mathbf{IE}^i_k| * |\mathbf{IE}^j_k|}, \tag{5}$$

where $\mathbf{IE}^i_k$, $\mathbf{IE}^j_k \in \mathbb{R}^{1\times C}$ from $\mathbf{IE}_k$ and $i, j \le k$.

Finally, we compute the $L2$ distance between $\mathbf{CS}^{i,j}_{image}$ and $\mathbf{CS}^{i,j}_{text}$ as the SSC Loss:

$$L_{SSC} = \frac{1}{k^2}\sum_{i=1}^{k}\sum_{j=1}^{k} \|\mathbf{CS}^{i,j}_{text} - \mathbf{CS}^{i,j}_{image}\|_2. \tag{6}$$

During training, we apply the semantic segmentation loss in Mask2former [6]. Our total loss during training is:

$$L_{total} = L_{sem\_seg} + \lambda L_{SSC}. \tag{7}$$

3.5. Adaptively Balancing and Inference

In this section, we show how to adaptively balance the image embeddings (mask attention embeddings $\mathbf{B}$, fully supervised embeddings $\mathbf{A}$ and frozen embeddings $\mathbf{D}$) for better recognition ability on both training and new classes.

During inference, we assume that there are K classes C_{test} in the test set, where f ($f < K$) classes exist in the training classes. Firstly, we use the CLIP text encoder to extract text embeddings $\mathbf{TE}_{test} \in \mathbb{R}^{K\times C}$ for C_{test}:

$$\mathbf{TE}_{test} = \Theta(C_{test}), \tag{8}$$

where Θ denotes the CLIP Text Encoder.

Then, we adaptively balance image embeddings ($\mathbf{A}$, $\mathbf{B}$ and $\mathbf{D} \in \mathbb{R}^{N\times C}$) with weights ($\alpha$, β and γ) for both training and new classes:

$$\mathbf{E}_{train} = \alpha * \mathbf{A} + \beta * \mathbf{B} + (1 - \alpha - \beta) * \mathbf{D}, \tag{9}$$

$$\mathbf{E}_{new} = \gamma * \mathbf{A} + \beta * \mathbf{B} + (1 - \gamma - \beta) * \mathbf{D}, \tag{10}$$

where $\mathbf{E}_{train}$, $\mathbf{E}_{new} \in \mathbb{R}^{N\times C}$ are balanced embeddings for training and new classes respectively; $\odot$ denotes element-wise multiplication. In Eq. (9) and Eq. (10), we use arithmetic mean to obtain balanced embeddings. However, we find that using geometric mean achieves slightly higher accuracy. Please refer to the supplementary materials for more details and experiments about this.

Next, we get mask classification predictions for both training and new classes:

$$\mathbf{P}_{train} = \mathbf{E}_{train} \times \mathbf{TE}^T_{train}, \tag{11}$$

$$\mathbf{P}_{new} = \mathbf{E}_{new} \times \mathbf{TE}^T_{new}, \tag{12}$$

where $\mathbf{TE}_{train} \in \mathbb{R}^{f\times C}$ and $\mathbf{TE}_{new} \in \mathbb{R}^{(K-f)\times C}$ from $\mathbf{TE}_{test}$, $\mathbf{P}_{train} \in \mathbb{R}^{N\times f}$ and $\mathbf{P}_{new} \in \mathbb{R}^{N\times (K-f)}$. The mask classification predictions $\mathbf{P} \in \mathbb{R}^{N\times K}$ for all test classes are:

$$\mathbf{P} = Concatenate(\mathbf{P}_{train}, \mathbf{P}_{new}). \tag{13}$$

Please note that we rearrange the order of $\mathbf{P}$ along the second dimension to match the order of the ground truth labels.

Finally, we obtain the semantic segmentation results $\mathbf{S} \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times K}$:

$$\mathbf{S} = \mathbf{M} \times \mathbf{P}. \tag{14}$$

Note that ”adaptively” means using different preset balancing weights specific for training and test classes to achieve better open vocabulary recognition ability, rather than automatically setting balancing weights. We will try using learnable balancing weights in our future work.

4. Experiments

4.1. Experiment Setup

Datasets and evaluation protocol. We train our model on the COCO-Stuff [2] dataset, and evaluate the model on five other benchmarks, including Pascal VOC [9], Pascal Context-59 (PC-59) [25], Pascal Context-459 (PC-459) [25], ADE20K-150 (A-150) [42], ADE20K-847 (A-847)[42]. There are 171 densely annotated classes in the COCO-stuff dataset, which contains 118K training images, 5K validation images, and 41K test images. We train our model on the training set of COCO-Stuff. Pascal VOC has 20 classes, with 1464 training and 1449 validation images. Pascal Context has 5K training images and 5K validation images, with two types of annotations: 59 classes annotated in Pascal Context-59 and 459 classes annotated in Pascal Context-459. ADE20K contains 20K training images and 2K validation images, and it has two sets of annotated classes: ADE20K-150 with 150 classes and ADE20K-847

Method	VLM	Training Dataset	A-847	PC-459	A-150	PC-59	VOC
LSeg+ [17]	ALIGN RN-101	COCO-Stuff	2.5	5.2	13.0	36.0	59.0
OpenSeg [10]	ALIGN RN-101	COCO Panoptic[14]	4.0	6.5	15.3	36.9	60.0
LSeg+ [17]	ALIGN EN-B7	COCO-Stuff	3.8	7.8	18.0	46.5	-
OpenSeg [10]	ALIGN EN-B7	COCO Panoptic[14]+Loc.Narr.	8.8	12.2	28.6	48.2	72.2
ZegFormer [7]	CLIP ViT-B/16	COCO-Stuff	5.6	10.4	18.0	45.5	89.5
SimSeg [38]	CLIP ViT-B/16	COCO-Stuff	6.9	9.7	21.1	51.9	91.8
OVSeg [20]	CLIP ViT-B/16	COCO-Stuff+COCO Caption[4]	7.1	11.0	24.8	53.3	92.6
SAN [39]	CLIP ViT-B/16	COCO-Stuff	10.1	12.6	27.5	53.8	94.0
EBSeg(ours)	CLIP ViT-B/16	COCO-Stuff	**11.1**	**17.3**	**30.0**	**56.7**	**94.6**
SimSeg [38]	CLIP ViT-L/14	COCO-Stuff	7.1	10.2	21.7	52.2	92.3
MaskCLIP [8]	CLIP ViT-L/14	COCO Panoptic	8.2	10.0	23.7	45.9	-
OVSeg [20]	CLIP ViT-L/14	COCO-Stuff	9.0	12.4	29.6	55.7	94.5
ODISE [37]	SD+CLIP ViT-L/14	COCO Panoptic	11.1	14.5	29.9	57.3	-
SAN [39]	CLIP ViT-L/14	COCO-Stuff	12.4	15.7	32.1	57.7	94.6
EBSeg(ours)	CLIP ViT-L/14	COCO-Stuff	**13.7**	**21.0**	**32.8**	**60.2**	**96.4**

Table 1. Comparison with state-of-the-art methods. We use mIoU as the evaluation metric. VLM denotes vision-language model. ALIGN [12] is a vision-language model. EN-B7 [32] is the image backbone used by ALIGN [12]. Loc.Narr. stands for Localized Narrative [28], which contains detailed natural language descriptions for multiple datasets. SD denotes the stable diffusion model [31].

with 847 classes. Same as early works, we adopt mean Intersection over Union (mIoU) as the evaluation metric for all our experiments.

Implement details. We use OpenAI pretrained CLIP model [29] in our experiments, including a ViT-B/16 model and a ViT-L/14 model. The ViT-B model of SAM [15] is used for all our experiments. The input image resolution is 640^2, and the downsample ratio p for the CLIP image encoder is set to 0.5 and 0.7 for our ViT-B and ViT-L models respectively. For the CLIP and SAM image encoders, we freeze all their parameters except for their positional embeddings. For the AdaB Decoder, the Transformer Decoder in it has 9 layers and 100 queries. The hidden dimension of the Transformer Decoder is 256, and its output dimension C is set to the same as the dimension of CLIP features (512 for CLIP ViT-B and 640 for CLIP ViT-L). For adaptively image embedding balancing, we set $\alpha = 0.2$, $\beta = 0.7$ and $\gamma = 0$ by default. We use CLIP Surgery [19] to get better CLIP final output $\mathbf{F}_{clip}$ for frozen embeddings $\mathbf{D}$. For SSC loss, the loss weight λ is set to 10. An auxiliary loss (the loss L_{sem_seg}) is added to every intermediate Transformer Decoder layer. Note that we only apply our SSC Loss to fully supervised embeddings A from the last layer of the Transformer Decoder. Our models are implemented with Pytorch [27] and Detectron2 [33]. AdamW [24] optimizer is used with the initial learning rate of $1 \cdot 10^{-4}$, weight decay of $5 \cdot 10^{-2}$. We set the batch size to 16, and train models for $120k$ iterations.

AIB	AdaB Decoder	SSC Loss	mIoU
Swin-B [22]			27.9
SAM-B [15]			28.5
SAM-B [15]	✓		29.2
SAM-B [15]		✓	29.0
SAM-B [15]	✓	✓	**30.0**

Table 2. The ablation study on the proposed components. SAM-B denotes the ViT-B model of SAM [15]

4.2. Comparison with State-of-the-Art methods

In Tab. 1 we compared our method EBSeg with other methods on several datasets. We list the datasets and vision-language models (VLM) used in various methods in the table. To ensure a fair comparison, we group the results using the same vision-language model together.

Overall, compared with other methods using CLIP ViT models, our method outperforms the best of them across all test datasets. With CLIP ViT-B/16 as the vision-language model, our method gains +1.0% mIoU, + 4.7% mIoU, +2.5% mIoU, +2.9% mIoU, +0.6% mIoU improvements on A-847, PC-459, A-150, PC-59, VOC, respectively. When using CLIP ViT-L, the improvements are +1.3% mIoU, + 5.3% mIoU, +0.7% mIoU, +2.5% mIoU, +1.8% mIoU on the five datasets respectively. These results demonstrate the effectiveness of our proposed method EBSeg.

We also show some visualization results of our method on the ADE20K-150 validation set in Fig. 4. More visual-

type(s) of embeddings	embeddings	mIoU
one	**A**	17.7
	B	29.0
	D	19.7
two	**A** and **B**	29.4
	A and **D**	26.3
	B and **D**	29.7
three	**A**, **B** and **D**	**30.0**

Table 3. The ablation study on the Adaptively Balanced Decoder (AdaB Decoder).

AdaB Decoder	mIoU train	mIoU new	mIoU
×	39.5	20.5	29.0
✓	40.3(+0.8)	21.6(+1.1)	30.0(+1.0)

Table 4. The ablation study on the influence of the proposed AdaB Decoder on the mIoU of the training (mIoU training) and new (mIoU new) classes in the ADE20K-150 validation set. In the second row, we only use the mask attention embeddings **B** and in the third row we adaptively balance the image embeddings **A**, **B**, and **D**. Note that ADE20K-150 has 67 classes that exist in the COCO-Stuff dataset and 83 classes that do not exist in COCO-Stuff.

ization results can be found in the supplementary materials.

4.3. Ablation Studies

We conduct ablation studies on the ADE20K-150 dataset. In this section, all models use CLIP ViT-B/16 as the vision-language model by default.

Component Analysis. We conduct ablation studies in Tab. 2 to analyze the effect of the essential components of our method. The first row in Tab. 2 represents a model that uses a CLIP image encoder and a frozen Swin-B [22] as backbones, and leverages mask attention embeddings **B** to compute semantic segmentation results for all classes. If we replace the frozen Swin-B with a frozen SAM-B [15] image encoder, the performance is further improved by 0.6% mIoU. The AdaB Decoder improves the mIoU further by 0.7% mIoU. With the SSC Loss, the mIoU improves further by 0.8%. The experiments show that the three methods we propose are all effective for the open-vocabulary semantic segmentation task.

AdaB Decoder. In Tab. 3, we present the performances when we balance different image embeddings during inference. When using a single type of embeddings, the mIoU for **A**, **B** and **D** embeddings are 17.7%, 29.0% and 19.7% respectively. When balancing two types of embeddings, mIoU improves. The highest mIoU is achieved when we adaptively balance all image embeddings, which is 30.0%.

embeddings using SSC Loss	layers using SSC Loss	mIoU
None	None	29.2
A	all layers	29.0
	last 3 layers	29.6
	last 1 layer	**30.0**
B	all layers	28.6
	last 3 layers	28.8
	last 1 layer	29.1
both **A** and **B**	all layers	28.8
	last 3 layers	29.3
	last 1 layer	29.7

Table 5. The ablation study on the Semantic Structure Consistency loss (SSC Loss). We apply the SSC Loss to different embeddings and different layers in our Adab Decoder during training.

SSC Loss	mIoU train	mIoU new	mIoU
×	40.6	19.9	29.2
✓	40.3(-0.3)	21.6(+1.7)	30.0(+0.8)

Table 6. Ablation study on the influence of the SSC Loss to the mIoU of the training and new classes in the ADE20K-150 validation set.

In Tab. 4, we show how AdaB Decoder influences mIoU of the training and new classes. After applying the AdaB Decoder, the mIoU of both training and new classes improves by a large margin. The improvement shows that with our AdaB Decoder, the model performs better at both training and new classes.

SSC Loss. In Tab. 5, we show the importance of the SSC Loss (Semantic Structure Consistency loss). Without the SSC Loss, the performance will drop from 30.0% mIoU to 29.2% mIoU. When we apply the loss to image embeddings **A** from all layers of the Transformer Decoder in our AdaB Decoder, the performance drops. Since the image embeddings from the first few layers lack rich semantic information, we do not apply the SSC Loss to the first few layers. When we apply the SSC Loss to **A** from the last 3 layers, the mIoU is improved from 29.2% to 29.6%. After we only apply the SSC Loss to embeddings **A** generated by the last layer, the mIoU is further improved to 30.0%. We also find that when applying this loss only to embeddings **B** or both embeddings **A** and **B**, the model's performance is unsatisfactory. We believe this is because the process of obtaining mask attention embeddings **B** involves many frozen parameters, which makes it difficult to optimize the SSC Loss.

In Tab. 6, we present how SSC Loss influences the mIoU of training and new classes in the ADE20K-150 validation

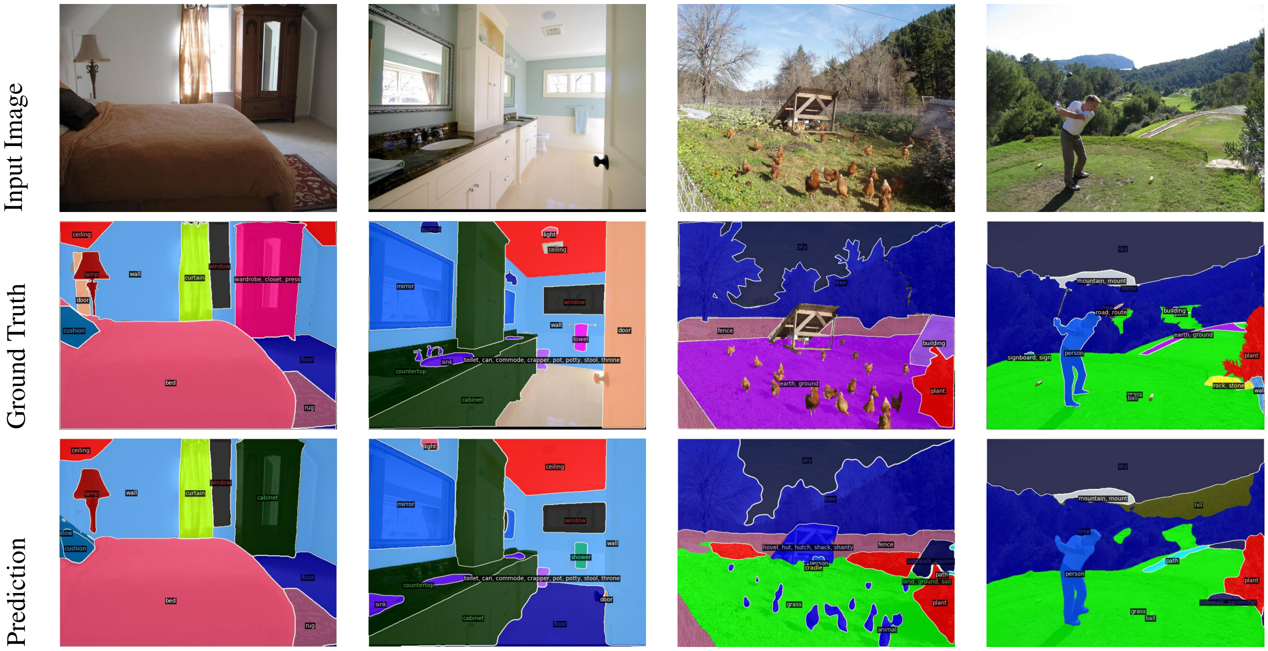

Figure 4. Visualization examples of our model on ADE20K-150 validation set.

set. The results show that SSC Loss significantly improves the mIoU of new classes. This indicates that this loss could help our model learn a more consistent semantic structure of CLIP and gain stronger generalization ability for new classes.

Additional Image Backbone. We show the advantage of using SAM as the additional image backbone in Tab. 7. We only change the additional image backbone in our model, other settings are the same as our default settings. The results show that with a trainable Swin-T backbone, the model gets 28.5% mIoU on ADE20K-150. If the Swin-T backbone is frozen, the performance will be further improved by 0.5% mIoU. When we replace the Swin-T with a Swin-B model, the mIoU is improved from 29.0% to 29.3%. Finally, a frozen SAM-B backbone brings a further improvement of 0.7% mIoU compared to a frozen Swin-B. The experiment results in Tab. 7 demonstrate that the SAM-B model can help our model perform better by complementing the spacial information. The results also denote that SAM-B is better than Swin-B as an additional image backbone in the open-vocabulary semantic segmentation task.

AIB	freeze	parameters (M)	mIoU
Swin-T [22]	×	50.2	28.5
Swin-T [22]	✓	24.5	29.0
Swin-B [22]	✓	24.9	29.3
SAM-B [15]	✓	26.6	**30.0**

Table 7. The ablation study on the additional image backbone (AIB). We only change the additional image backbone, the other settings are the same as our default settings.

5. Conclusion

In this paper, to overcome the challenge that training on semantic segmentation datasets often makes the model overfit to the classes in those datasets, we propose a novel framework for open-vocabulary semantic segmentation called EBSeg, which incorporates an Adaptively Balanced Decoder (AdaB Decoder) and a Semantic Structure Consistency loss (SSC Loss). By adaptively balancing different image embeddings, AdaB Decoder can fully leverage their ability to recognize training classes and the generalization capability for new classes. The SSC Loss aligns the inter-classes affinity in the image feature space with that in the text feature space of CLIP. This loss helps our model learn a consistent semantic structure from CLIP and improves the generalization ability of our model. We also propose to fuse the image features of SAM and CLIP to complement the spatial information of CLIP image features. Our method EBSeg establishes a new state-of-the-art in the open-vocabulary semantic segmentation task.

Acknowledgement. This work was supported by Hubei Provincial Natural Science Foundation of China No.2022CFA055.

References

[1] Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[2] 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. 2, 5

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 1

[4] 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. 6

[5] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021. 1

[6] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 1, 2, 3, 4, 5

[7] Jian Ding, Nan Xue, Guisong Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. 2022 ieee. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11573–11582, 2021. 1, 2, 3, 6

[8] Zheng Ding, Jieke Wang, and Zhuowen Tu. Open-vocabulary universal image segmentation with maskclip, 2023. 2, 3, 6

[9] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. *Pattern Anal. Stat. Model. Comput. Learn., Tech. Rep*, 2007(1-45):5, 2012. 2, 5

[10] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *European Conference on Computer Vision*, pages 540–557. Springer, 2022. 1, 6

[11] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *International Conference on Learning Representations*, 2021. 4

[12] 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. 6

[13] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6399–6408, 2019. 1

[14] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9404–9413, 2019. 6

[15] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 2, 6, 7, 8

[16] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, 2021. 2

[17] Boyi Li, Kilian Q. Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation, 2022. 1, 6

[18] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 2

[19] Yi Li, Hualiang Wang, Yiqun Duan, and Xiaomeng Li. Clip surgery for better explainability with enhancement in open-vocabulary tasks, 2023. 2, 6

[20] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7061–7070, 2023. 1, 2, 3, 6

[21] Chen Liang-Chieh, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *International Conference on Learning Representations*, 2015. 1

[22] 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. 6, 7, 8

[23] 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. 1

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[25] 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. 2, 5

[26] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015. 1

[27] 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. *Ad-*

vances in neural information processing systems, 32, 2019. 6

[28] 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. 6

[29] 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, 2, 6

[30] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18082–18091, 2022. 2

[31] 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*, pages 10684–10695, 2022. 2, 3, 6

[32] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 6

[33] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 6

[34] 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. 3

[35] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018. 1

[36] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 1

[37] 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. 2, 3, 6

[38] 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 *European Conference on Computer Vision*, pages 736–753. Springer, 2022. 1, 2, 3, 6

[39] 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. 2, 3, 4, 6

[40] 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. 3

[41] Zexuan Zhong, Dan Friedman, and Danqi Chen. Factual probing is [mask]: Learning vs. learning to recall. *arXiv preprint arXiv:2104.05240*, 2021. 2

[42] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 2, 5

[43] 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. 2

[44] 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. 2