Auto-Vocabulary Semantic Segmentation

Osman Ülger[1]* Maksymilian Kulicki[2,3]* Yuki Asano[4] Martin R. Oswald[1]

[1] University of Amsterdam [2] Institute of Fundamental Technological Research, Polish Academy of Science [3] IDEAS NCBR
[4] University of Technology Nuremberg * equal contribution

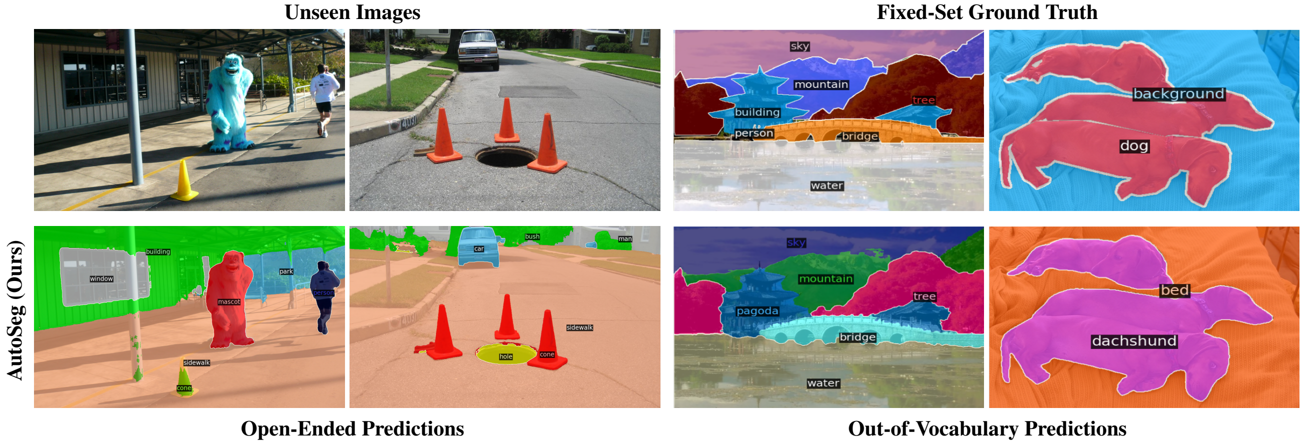

Figure 1. **AutoSeg Exemplary Results.** AutoSeg is readily applicable to unseen images for open-ended segmentation for objects such as *mascot* and *hole*, such as the two images on the left. Furthermore, where established segmentation datasets have a fixed set of annotation categories, our method is able to identify and segment with more semantically precise object categories beyond the fixed-set ground truth, such as *dachshund*, *bed* and *pagoda*. Images are from the Road Anomaly [20], PASCAL [9] and ADE20K [47] datasets.

Abstract

Open-Vocabulary Segmentation (OVS) methods are capable of performing semantic segmentation without relying on a fixed vocabulary, and in some cases, without training or fine-tuning. However, OVS methods typically require a human in the loop to specify the vocabulary based on the task or dataset at hand. In this paper, we introduce Auto-Vocabulary Semantic Segmentation (AVS), advancing open-ended image understanding by eliminating the necessity to predefine object categories for segmentation. Our approach, AutoSeg, presents a framework that autonomously identifies relevant class names using semantically enhanced BLIP embeddings and segments them afterwards. Given that open-ended object category predictions cannot be directly compared with a fixed ground truth, we develop a Large Language Model-based Auto-Vocabulary Evaluator (LAVE) to efficiently evaluate the automatically generated classes and their corresponding segments. With AVS, our method sets new benchmarks on datasets PASCAL VOC, Context, ADE20K, and Cityscapes, while showing competitive performance to OVS methods that require specified class names. All code is released here.

1. Introduction

While humans possess an open-ended understanding of scenes, recognizing thousands of distinct categories, semantic segmentation methods [25] typically rely on a fixed vocabulary of predefined semantic categories. They require large human-annotated datasets and have limited capabilities for handling a broad range of classes or unknown objects. Recent studies have focused on addressing these limitations [23, 28] with models that leverage Vision-Language Models (VLMs) as an emerging category [3, 5, 7, 10, 17, 19, 43, 45, 46]. Such VLMs learn rich multi-modal features from large numbers of image-text pairs. Due to the immense computational training costs of VLMs, it is common to build upon pre-trained VLMs such as CLIP [29] or BLIP [18]. However, applying these VLMs on per-pixel tasks to obtain precise locality information is non-trivial, as they are trained on full images and lack the ability to directly reason over local regions in the image. This makes them less useful for obtaining precise segmentation boundaries, which is crucial for downstream tasks like robot grasping. Moreover, current Open-Vocabulary segmentation (OVS) methods still require a human in the loop, inherently limiting the scal-

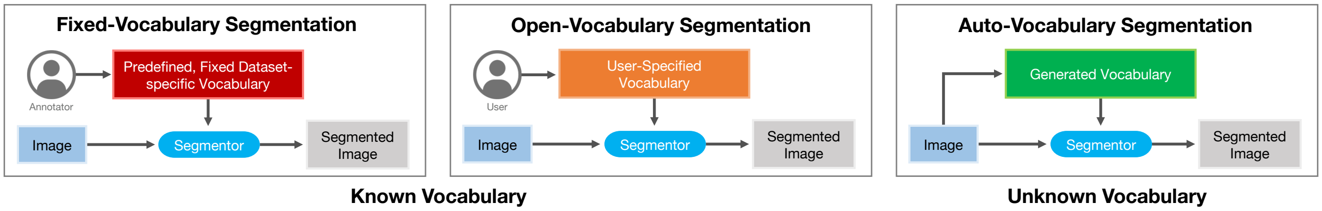
Fixed-Vocabulary Segmentation
Annotator
Predefined, Fixed Dataset-specific Vocabulary
Image
Segmentor
Segmented Image
Open-Vocabulary Segmentation
User
User-Specified Vocabulary
Image
Segmentor
Segmented Image
Known Vocabulary
Auto-Vocabulary Segmentation
Generated Vocabulary
Image
Segmentor
Segmented Image
Unknown Vocabulary

Figure 2. **Semantic Segmentation Tasks in Comparison.** In traditional Semantic Segmentation, an image is segmented into fixed, predefined set of classes (fixed vocabulary). In Open-Vocabulary Segmentation, the user specifies which object categories (from the open vocabulary) should be segmented: 1) either via a human-provided prompt at runtime, or 2) the OV-method is trained to output the vocabulary of a human-annotated target dataset. In contrast, Auto-Vocabulary Segmentation automatically generates relevant object categories directly from the image. This enables true open-ended scene understanding without needing human input.

ability of these methods (see Fig. 2). OVS methods can be separated into two categories: **1) prompt-based methods** (*e.g.* [13, 16, 48]) require per-image user input to provide a *known* vocabulary; and **2) dataset-based methods** (*e.g.* [10, 17, 19, 41]) require human-annotated datasets in combination with training in which the *known* vocabulary is baked into the method as its defined output categorization. For both categories, a human is providing a *known* and fixed vocabulary, which drastically simplifies the segmentation problem as the ground truth vocabulary is always provided, but it also limits the application use-cases when a human needs to be in the loop. Imagine a kitchen robot that needs to distinguish a large variety of tools and ingredients when cooking a recipe and requiring precise segmentations for grasping. It is undesirable to require a human-provided prompt for every grasp or to re-label a dataset and re-train the segmentation method every time a new recipe contains a new tool or ingredient.

To eliminate the human in the loop and fully rely on pre-trained foundation models, we propose **Auto-Vocabulary Semantic Segmentation (AVS)**. In AVS, the *unknown* vocabulary is automatically generated and integrated into a semantic segmentation method thus allowing pixel-level classifications for **any** class without the need for textual input from the user, predefined class names, additional data, training or fine-tuning. To address this task, we introduce AutoSeg, a zero-shot method that first identifies image-specific target categories and then predicts masks for them. See Fig. 1 for exemplary results. We introduce BBoost for vocabulary generation, which provides a simple yet effective semantic clustering strategy to enhance locality and semantic precision when captioning with BLIP.

Furthermore, we address the performance evaluation of open-ended segmentation models, following the same goal of eliminating human input for targeting the true scalability of segmentation models. While VLMs have a *continuous* latent representation of semantics, the inherent nature of segmentation is to *discretize* semantic categories to a particular task-specific vocabulary. Comparing different discrete category sets is challenging, especially for larger vocabularies as exact matching can be difficult due to annotation ambiguities like synonyms (*e.g.* table↔desk), semantic hierarchies (*e.g.* person→man→groom), or spatial hierarchies (*e.g.* car→wheel→rim). Human-annotated datasets typically have an annotation bias towards these ambiguities and can be inconsistently annotated across multiple annotators, making it nearly impossible for a segmentation algorithm to infer the intended hierarchy level. Classical discrete semantic evaluation also neglects semantic vicinities among different labels, treating such ambiguities equal to misclassifications. Since our methods' output is not within a predefined, but an auto-generated vocabulary, we enable evaluations on labeled datasets with our proposed **L**arge Language Model-based **A**uto-**V**ocabulary **E**valuator (LAVE) which maps a generated vocabulary to a provided one.

In summary, our **contributions** are as follows: **1)** we introduce AutoSeg, a novel framework to automatically determine and segment open-ended classes in an image; **2)** we propose BBoost, a novel method for generating image-specific target vocabularies, leveraging text decoding from enhanced BLIP embeddings; and **3)** we propose LAVE, a novel evaluation approach for auto-vocabulary semantic segmentation which utilizes an LLM. Through qualitative and quantitative analyses, we demonstrate the effectiveness of our framework for AVS on multiple public datasets.

2. Related Work

Open-Vocabulary Segmentation (OVS). OVS tackles the challenging task of segmenting images based on arbitrary text queries rather than a predefined class set. Early OVS methods, like ZS3Net [1] and SPNet [38], treated it as a zero-shot task by training modules that align visual and linguistic embeddings, segmenting by comparing word vectors with local image features. Recently, Vision-Language Models (VLMs) pre-trained on large image-text dataset, such as Contrastive Language-Image Pre-training (CLIP)[29], have unified image and text features. CLIP has advanced the field of OVS significantly, with methods such as LSeg[17], which compares per-pixel embeddings with class embed-

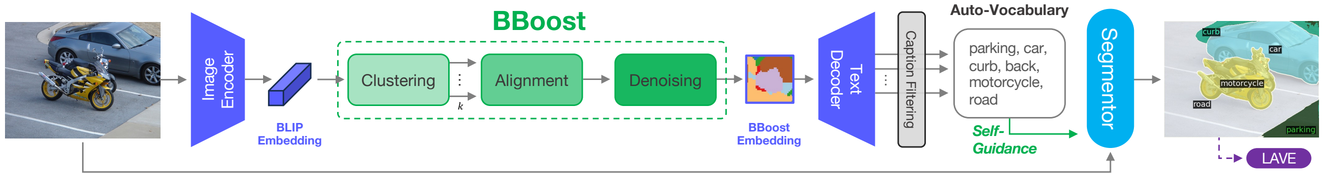

Figure 3. **Method Overview.** BLIP encodings are clustered, aligned and denoised before being decoded into nouns by BBoost. Generated nouns serve as self-guidance to a segmentor, which predicts the final mask. When evaluating (purple), our custom evaluator LAVE processes the output, mapping predicted nouns to the fixed-vocabulary annotations.

dings, and two-stage approaches like OpenSeg [10], OPSNet [3], OVSeg [19], ZSSeg [42], and POMP [31], which first obtain class-agnostic masks and then classify each mask with CLIP. Other works, such as MaskCLIP [7], FC-CLIP [46], and SAN [43], utilize intermediate CLIP representations. Other approaches leverage CLIP to relate image semantics to class labels [5], predict binary masks [45] or ensure segmentation consistency at multiple granularities [35]. Beyond VLMs, recent models use text-to-image diffusion [12, 41] or transformers [22, 39, 40, 48], with X-Decoder providing a generalized model for various vision-language tasks.One common and significant limitation of the described methods is the need for textual input as a form of guidance by the user. Our work presents a novel framework which enables self-guidance by automatically identifying relevant object categories via localized image captioning. Closest and concurrent to our work is Zero-Guidance Segmentation [32], in which clustered DINO [2] embeddings are combined with CLIP to achieve zero guidance by the user. Despite the similar setting, there are key differences: we use BLIP [18] for both image clustering and caption generation, while they use a complex pipeline involving DINO clustering, CLIP embeddings, a custom attention-masking module, and CLIP-guided GPT-2 for captioning, requiring conversions between three latent representations. In contrast, we achieve superior segmentation quality with self-guidance using any out-of-the-box OVS model.

Image Captioning with Image-Text Embeddings. Image Captioning is the task of describing the content of an image using natural language. A number of approaches have been proposed for the task, including ones utilizing VLMs. Approaches such as CLIP-Cap [26] and CLIP-S [4] utilize CLIP embeddings to guide text generation. More recently, the Large Language and Vision Assistant (LLaVA) was introduced with similar capabilities by connecting the visual encoder of CLIP with a language decoder, as well as training on multimodal instruction-following data [21]. Other approaches rely on BLIP [18], a VLM in which the task of decoding text from image features was one of the tasks of the pretraining process. This enables BLIP to perform image captioning without additional components. Furthermore, the text decoder is guided by local patch embeddings in a way that enables local captioning based on a specific area. This unique capability, discovered during the development of our method, emerges organically in BLIP despite being trained solely with image-level captions.

Visual Phrase Grounding. Visual phrase grounding aims to connect different entities mentioned in a caption to corresponding image regions [33]. This task resembles OVS, as it aims to find correspondences between text and image regions. However, predicted areas do not have to precisely outline object boundaries and can be overlapping. Visual grounding has been approached in a self-guided manner, in which heatmaps of regions of the image are generated from image-level CLIP embeddings, which are then captioned by the BLIP encoder [34]. One disadvantage of this approach is that the region determined by CLIP can indicate multiple objects at once, hence multiple objects are captioned for that region, preventing pixel-level class-specific predictions. In our work, BLIP serves both as encoder and decoder, enabling category-specific regions and captioning.

3. Method

This work aims to perform semantic segmentation without any additional data, training, finetuning or pre-defined target categories. To this end, we propose to identify relevant object categories in an image by captioning with locality (Sect. 3.1), filter the generated captions into a meaningful vocabulary and use the vocabulary as target categories for a segmentor (Sect. 3.2). An overview is depicted in Fig. 3.

3.1. Local Region Captioning

To identify relevant object categories in the image, we employ Bootstrapped Language Image Pretraining (BLIP) [18]. BLIP is a powerful VLM capable of performing accurate and detailed image captioning. Its image encoder is designed as a Vision Transformer [8], dividing the image into patches and encoding them into embeddings in which image-text features reside. The set of embeddings is then passed through transformer layers to capture contextual information across the image. Finally, all contextualized embeddings are decoded into text descriptions. Though effective in describing salient objects in the image, captioning models like BLIP generally fail to describe all elements of a scene comprehensively. This is likely due to the captions in the training data, which mostly focus on salient foreground objects. In turn, this limits its direct application to downstream tasks such as open-ended recognition, since many non-salient objects are missed. As a solution to this limitation, we propose BBoost for exhaustive, local captioning. BBoost clusters the encoded BLIP features into semantically meaningful feature clusters, which are enhanced to capture the object-specific feature representations more effectively. Afterwards, each individual feature cluster is fed to a pre-trained BLIP text decoder. This enables generating a description of each semantically distinct and meaningful area in the image, resulting in a more comprehensive, specific and accurate description of the image overall. Its components are detailed in the next paragraphs.

Clustering. By default, BLIP expects an RGB image $\mathbf{X}_D \in \mathbb{R}^{384\times384\times3}$. Since common datasets often have images with higher resolution, we additionally process the image at a higher resolution $\mathbf{X}_H \in \mathbb{R}^{512\times512\times3}$. The multi-resolution set of images $\mathbf{X} = \{\mathbf{X}_D, \mathbf{X}_H\}$ is fed to the BLIP encoder to obtain the set of BLIP patch embeddings $\hat{\mathbf{B}} = \{\hat{\mathbf{B}}^{\mathbf{X}_D}, \hat{\mathbf{B}}^{\mathbf{X}_H}\}$ at the two resolutions:

$$\mathbf{P}^{\mathbf{X}_D} = \left\{\mathbf{X}^D_{ij} \,|\, \mathbf{X}^D_{ij} \in \mathbb{R}^{16\times16\times3}, 1 \le i,j \le 24\right\} \tag{1}$$

$$\mathbf{P}^{\mathbf{X}_H} = \left\{\mathbf{X}^H_{ij} \,|\, \mathbf{X}^H_{ij} \in \mathbb{R}^{16\times16\times3}, 1 \le i,j \le 32\right\} \tag{2}$$

$$\begin{aligned}\hat{\mathbf{B}}^{\mathbf{X}_R}_n &= T\left(f_{\text{MLP}}\left(\mathbf{P}^{\mathbf{X}_R}_n\right)\right) \parallel \mathbf{z}^{\mathbf{X}_R}_n,\\ &\forall R \in \{D,H\}, \quad n \in \{1,\dots,N_R\}\end{aligned} \tag{3}$$

where $\mathbf{P}^{\mathbf{X}_D} \in \mathbb{R}^{576\times16\times16\times3}$ and $\mathbf{P}^{\mathbf{X}_H} \in \mathbb{R}^{1024\times16\times16\times3}$ denote the sets of patches for resolutions D and H respectively, and $\mathbf{P}^{\mathbf{X}_R}_n$ is the n-th patch. $f_{\text{MLP}}(\cdot)$ represents a shared fully connected Multi-Layer Perceptron (MLP), LN is LayerNorm, and T is a Transformer encoder with L alternating layers of Multi-Head Self-Attention (MHA) and an MLP, sequentially propagated:

$$T_l(X) = X + \text{MHA}\big(\text{LN}(X), \text{LN}(X), \text{LN}(X)\big) \tag{4}$$

$$\hat{T}_l(X) = T_l(X) + f_{\text{MLP}}\big(\text{LN}(T_l(X))\big) \tag{5}$$

$$T(X) = (\hat{T}_{L-1} \circ \hat{T}_{L-2} \circ \ldots \circ \hat{T}_0)(X). \tag{6}$$

In Eq. (3), we concatenate, *i.e.* $\parallel$, a sinusoidal positional encoding [37] $\mathbf{z}^{\mathbf{X}_R}_n \in \mathbb{R}^{256}$ to each patch embedding to encode spatial information. Next, we cluster the patches in $\hat{\mathbf{B}}^{\mathbf{X}_R}$ for each resolution R using k-means clustering [24]:

$$C^R_k = \underset{C}{\text{argmin}} \sum_{i=1}^{N_R} \min_{\mu_j \in C} \|\hat{\mathbf{B}}^{\mathbf{X}_R}_i - \mu_j\|^2 \tag{7}$$

where C^R_k is the set of k clusters for resolution R and μ_j are the cluster centroids. Running the clustering procedure with $k \in \{2,\dots,8\}$ on two different image resolutions results in 14 unique cluster assignments.

Cross-clustering Consistency. Each run of k-means clustering labels its clusters independently from others, yielding a correspondence mismatch between clusters across runs. To resolve this, we relabel the cluster indices to a common reference frame with the following steps:

1. Select C with the most clusters after k-means as a reference set S. As some clusters end up empty during the k-means iterations, this is not always the set with highest initial k. The reference set determines the indices used for all other sets of clusters $\mathcal{C}$, each with its number of clusters denoted by $|C_i|$:

$$S = \underset{C_i \in \mathcal{C}}{\text{argmax}} \, |C_i| \tag{8}$$

2. Sets of clusters are aligned to the reference set using Hungarian matching [15]. We calculate pairwise Intersection over Union (IoU) between the clusters from S and C. Then, each cluster from C is assigned a new index, matching the cluster with the highest IoU from S:

$$\begin{aligned}&\text{For each cluster } c_j \in C, \quad j \in \{1,\ldots,|C|\} :\\ &\text{Assign index } i \text{ to } c_j \text{ where } i = \underset{i \in \{1,\ldots,|S|\}}{\text{argmax}} \ \text{IoU}(c_j, s_i)\end{aligned} \tag{9}$$

3. With the labeled sets of clusters, a probability distribution over the clusters is assigned to each image patch. For a given patch p, let $\mathbf{L}(p) = \{\mathbf{L}_1(p), \mathbf{L}_2(p), \ldots,$ $\mathbf{L}_m(p)\}$ be the set of labels assigned to p by the m different sets of clusters. The probability $P(n|p)$ of p being assigned to a particular cluster n is defined as the relative frequency of n among labels $\mathbf{L}(p)$:

$$P(n|p) = \frac{\sum_{i=1}^{m} \mathbb{1}}{m}, \text{with } \mathbb{1} = \begin{cases} 1, & \text{if } \mathbf{L}_i(p) = n \\ 0, & \text{else} \end{cases}. \tag{10}$$

The predictions of a single k-means predictor tend to contain high levels of noise, likely due to the high dimensionality of the embeddings. Our method can be seen as an ensemble, reducing the variance present in each individual predictor. The areas that are consistently clustered together by various predictors are likely to be semantically connected. In addition, our method enables for a flexible number of output clusters. Some of the initial clusters can disappear if they are not well-supported by multiple predictors. This property is highly desired due to the variety of input images and the number of objects in them.

Cluster Denoising. To further improve the locality and semantic meaningfulness of clustered feature representations, we apply a Conditional Random Field (CRF) [14] and majority filter. CRF is a discriminative statistical method that is used to denoise predictions based on local interactions between them. In our case, the predictions are a 2D grid of cluster assignment probabilities of the image patches. Our implementation is specifically tailored for refining 2D segmentation map, using a mean field approximation with a convolutional approach to iteratively adjust the probability distributions of each image patch's cluster indices. In the pairwise potentials, we use a Gaussian filter to ensure spatial smoothness and consistency in the segmentation. The application of the CRF yields embeddings which are less noisy and more cohesive than the original aligned k-means result. To address remaining noise in the embeddings, a neighborhood majority filter is applied as a final step. For each image patch, we consider the set of patches $\mathcal{N}(i,j)$ in its square neighborhood: $\mathcal{N}(i,j) = \{(i+\delta_i, j+\delta_j) \,|\, \delta_i, \delta_j \in \{-1,0,1\}\}$. The mode value from the cluster indices in that neighborhood is calculated and assigned as the new index of the central patch:

$$\text{mode}\big(\mathcal{N}(i,j)\big) = \underset{k \in K}{\text{argmax}} \sum_{m \in \mathcal{N}(i,j)} \mathbb{1}_{\text{index}(m)=k} \tag{11}$$

This step is applied recursively until convergence or 8 times at most. In the supplementary material, we visualize the effect of each step on the embeddings.

Captioning. The next step involves turning clustered, denoised and enhanced embeddings into text. The BLIP text decoder is a transformer architecture capable of processing unordered sets of embeddings of arbitrary size. We leverage this feature and feed flattened subsets of patch embeddings, each corresponding to a cluster, to the text decoder. Spatial information is preserved due to the presence of positional embeddings added in the clustering step. With this technique, our method essentially infers semantic categories captured by clusters and represented by BLIP embeddings. To the best of our knowledge, we are the first to use the text decoder in this manner, enabling local captioning without specifically training for it. The caption generation is stochastic, with different object namings appearing in the captions depending on initialization. To obtain a rich, unbiased and diverse set of object names, we regenerate captions with each embedding with multiple inference *cycles*.

Caption Filtering. The captions generated in the previous step are sentences in natural language. For our task, we are only interested in the class names present in each sentence. To obtain these, we filter the sentence down to relevant class names by extracting all nouns using part-of-speech labels for each word in the caption. Nouns are kept and converted into their singular form through lemmatization. We collect all nouns generated by different clusters and cycles into one target vocabulary and remove any duplicates, as well as nouns which do not appear in the WordNet dictionary.

3.2. Segmentation through Self-Guidance

The output of the clustered BLIP embeddings is a 32x32 grid with cluster index assignments (see Sect. 3.1), with each element corresponding to an image patch from the original image. This enables the extraction of segmentation masks - defined as a union of areas covered by image patches with the same index - essentially for free. For instance, Fig.3 and 4 show the clustered output as a 2D mask that partially captures relevant objects. However, the low resolution of the segmentations are unsatisfactory, and upsampling leads to oversegmented objects or unsharp boundaries. To perform effective and accurate auto-vocabulary semantic segmentation, we leverage BBoost's strength in generating an elaborate set of relevant class names from clusters instead - and use this as textual guidance for a pre-trained OVS model capable of producing high-resolution outputs. BBoost is model-agnostic, allowing our approach to integrate with any OVS model that accepts an image and a set of class labels. In this work, we focus on X-Decoder [48], a popular and well-performing OVS model.

4. Evaluation of Auto-Classes

As discussed in Sect. 2, previous works in OVS have mainly focused on a setting where the target class names are provided by the user. Hence, evaluation is possible by having access to the ground truth of those target class names during evaluation. However, in scenarios where the target class names are discovered, as in our framework, rather than prespecified there may be a lack of direct alignment between the semantics of these categories and the classes used in the annotations. In zero-guidance segmentation [32], the authors have proposed to align class names based on the cosine similarity in the latent space of Sentence-BERT [30] or CLIP [29]. In our initial tests, however, this approach misaligned obvious class pairs (e.g., *taxi* mapped to *road* instead of *car*), thereby reducing segmentation accuracy despite promising qualitative results. Relations between two class names can be complex and ambigious [44], such as due to synonymity, hyponymy, or hypernymy, which exceed the capabilities of a cosine-similarity criterium in the latent space. To address this, we propose an **L**LM-based **A**uto-**V**ocabulary **E**valuator (LAVE), leveraging the Llama-2-7B language model [36], to map predicted auto-vocabulary categories to target dataset classes. LAVE gathers all predicted auto-vocabulary classes and maps each category to the most relevant or similar class in the known vocabulary. It then updates all pixel values in the predicted segmentation masks according to the mapping, after which the mean Intersection over Union (mIoU) is calculated using the updated mask. Though not an integral part of our method, LAVE greatly reduces the mapping effort which is often infeasible to do manually, requiring $C \times A$ comparisons between C known classes and A auto-classes. Pseudocode and prompts for LAVE are included in the supplementary material.

5. Experiments

5.1. Experimental Setup

Datasets. AutoSeg is evaluated on four popular semantic segmentation validation datasets: PASCAL VOC [9] and

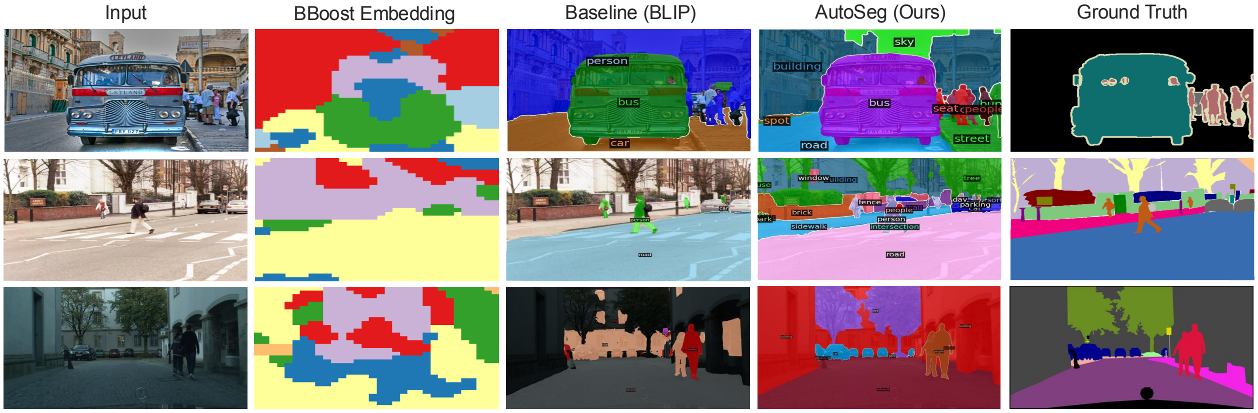

Figure 4. **Segmentation with VLMs.** Example outputs on PASCAL VOC/Context (top), ADE (middle) and Cityscapes (bottom) by (left to right) directly using BBoost embeddings as masks, feeding plain BLIP embeddings to X-Decoder or AutoSeg. Notably, our method segments images in the most comprehensive and semantically accurate manner.

Table 1. **Ablations on Segmentation with VLMs (cmIoU).** Leveraging BBoost semantic embeddings for segmentation with AutoSeg outperforms baselines on all four datasets.

Method	**VOC** [9]	**PC** [27]	**ADE** [47]	**CS** [6]
BBoost Embeddings	16.3	16.3	11.3	0.85
X-Decoder [48] + BLIP [18]	38.0	35.3	26.7	29.2
AutoSeg (Ours)	**71.8**	**47.7**	**29.2**	**35.8**

Context (PC) [27], ADE20K (ADE) [47] and Cityscapes (CS) [6] with 20, 459, 847 and 20 classes respectively, covering a wide range of difficulty and class diversity. For instance, ADE is challenging with its many and infrequently appearing classes, while CS contains many instances per image (see Tab. 2 for detailed statistics).

Evaluation. For quantitative comparison with previous works, we use mIoU as the main metric. As mentioned in Sec. 4, auto-class predictions in the segmentation mask are mapped using LAVE before the mIoU is computed. To evaluate class-agnostic segmentation performance of different VLMs, we report the class-agnostic mean Intersection over Union (cmIoU). For each ground-truth segment g_i, we compute its IoU with all predicted segments p_j and select the best match $p_{\max_i} = \arg\max_j \text{IoU}(g_i, p_j)$. The cmIoU is then given by cmIoU $= \frac{1}{N}\sum_{i=1}^{N} \text{IoU}(g_i, p_{\max_i})$, where N is the total number of ground-truth segments.

Implementation details. For our experiments, we use the BLIP model built with ViT-Large backbone and finetuned for image captioning on the COCO dataset in combination with the Focal-L variant of X-Decoder. For both models, we use publicly available pretrained weights and do not perform any additional finetuning. Parameter tuning is performed for the clustering and captioning modules. Parameters of the clustering include image scales encoded by BLIP (384 × 384 and 512 × 512), the k values of k-means clustering (2 to 8), the parameters of Gaussian smoothing in the CRF (smoothness weight of 6 and smoothness θ of 0.8), the number of iterations of majority filtering (8) and the feature dimension size of the positional embeddings (256). For text generation, we use nucleus sampling with a minimum length of 4 tokens and maximum of 25, top P value of 1 and repetition penalty of 100 to ensure as many unique nouns as possible. These parameters were determined using Bayesian optimization on VOC with the goal of maximizing the mIoU. For CRF denoising and part-of-speech tagging, we use the crfseg [14] and spaCy [11] libraries respectively.

5.2. Ablations

To assess our framework we investigate various segmentation methods with captions, the impact of captioning cycles, generated/fixed vocabulary similarity, alternative mappers and the effects of cluster denoising. The latter three ablation studies are detailed in the supplementary material.

Segmentation with Vision Language Models. We compare a baseline that uses upsampled BBoost embeddings directly as segmentation masks, leveraging their semantic clustering (see Sec. 3.2), to AutoSeg, which refines low-resolution semantics into high-resolution outputs. Segmentation quality is evaluated using class-agnostic cmIoU, with results in Tab. 1 and Fig. 4. Across all datasets, AutoSeg performs best, with the baseline X-Decoder + BLIP showing issues of false positives and negatives that explain the performance gap. For example, it incorrectly segments areas around objects in VOC/PC (*e.g. person* for *building*) and ignores significant regions in ADE and CS. In contrast, AutoSeg effectively captures smaller objects and details. As expected, low-resolution BBoost masks are inaccurate but can still segment general areas in VOC/PC and ADE, though they struggle in CS due to higher object density.

Captioning Cycles. A key feature of our model is its ability to repeatedly enrich the vocabulary with BBoost, enabling it to cover more objects classes in the scene. This

Table 2. **Ablations on Captioning Cycles (mIoU) and Generated Classes.** The optimal number of captioning cycles depends on the dataset, influenced by the image content and annotation method. Our method identifies a substantially larger variety of unique classes (**Gen.**) than those annotated by humans (**Ann.**), highlighting its capacity to capture finer context. $\overline{I}$ and $\overline{C}$ denote the average number of instances and classes per image.

	Number of Captioning Cycles						**Data Properties**			
Dataset	**1**	**5**	**10**	**15**	**20**	**25**	**Gen.**	**Ann.**	$\overline{I}$	$\overline{C}$
PASCAL VOC [9]	85.3	84.4	**87.1**	85.7	83.3	83.3	938	20	3.5	3.5
PASCAL Context [27]	10.1	9.7	11.2	**11.7**	9.6	10.8	721	459	18.9	6.2
ADE20K [47]	**6.0**	5.8	5.1	5.2	5.1	5.6	1578	847	19.5	10.5
Cityscapes [6]	28.2	28.1	28.4	28.3	27.9	**30.0**	395	20	34.3	17

Table 3. **Ablation on denoising components (mIoU).** The combination of our denoising components outperforms individual parts.

Denoising Component	**VOC**
CRF only	77.8
Majority filter only	74.5
CRF & Majority filter	**87.1**

Table 4. **Quantitative Auto-Vocabulary Results (mIoU).** The wide availability of captioning and OVS methods allows various combinations to design an auto-vocabulary method, but they are not necessarily performing well. AutoSeg performs superior over the only Auto-Vocabulary method ZeroSeg, as well as over AVS-adapted methods with alternative captioners or segmentors.

Auto-Vocabulary Segmentation Method	**VOC** [9]	**PC** [27]	**CS** [6]
LLaVA + LISA [16, 21]	7.7	0.2	1.5
ZeroSeg [32]	20.1	11.4	-
SAM + BLIP + X-Decoder [13, 18, 48]	41.1	11.3	27.4
LLaVA + X-Decoder [21, 48]	56.7	11.4	23.4
AutoSeg (Ours)	**87.1**	**11.7**	**30.0**

Table 5. **Open/Auto-Vocabulary State of the Art Comparison (mIoU).** AutoSeg surpasses ZeroSeg [32] in human-free segmentation and remains competitive with some well-known OVS methods that rely on human input, while outperforming others (3 out of 8 methods on VOC and 3 out of 7 methods on PC). Results with dashes indicate unpublished or unavailable data.

Method	**Unknown Vocabulary**	**VOC** [9] (20)	**PC** [27] (459)	**ADE** [47] (847)	**CS** [6] (20)
OVS Segmentation methods with prompted (*known* = ground truth) vocabulary					
LSeg [17]	✗	47.4	-	-	-
OpenSeg [10]	✗	72.2	9.0	8.8	-
OVSeg [19]	✗	94.5	11.0	9.0	-
ODISE [41]	✗	82.7	13.8	11.0	-
SAN [43]	✗	94.6	12.6	12.4	-
CAT-Seg [5]	✗	**97.2**	**19.0**	13.3	-
X-Decoder [48]	✗	96.2	16.1	6.4	50.8
FC-CLIP [46]	✗	95.4	12.8	**14.8**	**56.2**
AVS Segmentation methods with auto-generated (*unknown*) vocabulary					
ZeroSeg [32]	✓	20.1	11.4	-	-
AutoSeg + LAVE mapper (Ours)	✓	87.1	11.7	6.0	30.0
AutoSeg + Manual mapper (Ours)	✓	**88.2**	**12.8**	**6.2**	**31.1**

approach proves especially beneficial for instances where objects are initially overlooked or described with less precise semantic terms. We assess the effects of various captioning iterations using the mIoU metric. Tab. 2 reveals that the optimal number of captioning cycles varies with the dataset in question. We observe that for CS, which features a high average number of instances and unique classes per image ($\overline{I}$ and $\overline{C}$ in Tab. 2), increasing the number of captioning cycles improves performance the most. In contrast, datasets with fewer instances and classes per image such as VOC and ADE require fewer cycles for BBoost to identify relevant categories. As expected, additional cycles are also beneficial for PC given its additional 439 object categories. Remarkably, AutoSeg accurately handles ADE, the dataset with the highest number of unique classess, using only a single captioning cycle, demonstrating its efficiency in open-ended settings. Furthermore, through its captioning framework, AutoSeg identifies significantly more distinct classes than hand-crafted fixed vocabularies.

Denoising Components. To investigate the effectiveness of the denoising components, we measure their individual performance, as well as the combination as in AutoSeg. Tab. 3 shows that including both components in the pipeline significantly improves individual performance. Additional analyses are provided in the supplementary material.

5.3. Quantitative Analysis

Auto-Vocabulary Segmentation Setting. Our quantitative analysis begins by comparing AutoSeg with other auto-vocabulary methods, such as Zero-Guidance Segmentation (ZeroSeg [32]), as well as LLM-aided segmentation methods like LISA [16] with alternative captioning methods like LLaVA [21]. We also evaluate a configuration combining BLIP and X-Decoder, where instance crops predicted by SAM [13] are used for captioning to simulate locality. We use the same caption filtering as in our method. The results, shown in Tab. 4, indicate that our method outperforms ZeroSeg, the only other true auto-vocabulary approach. In contrast, LISA requires multiple inferences with individually prompted categories to achieve scene segmentation, a notably more complex and less efficient approach that also yields poor performance. Captioning based on smaller object crops is similarly ineffective compared to using enhanced vision-language features that maintain spatial locality. Moreover, while LISA provides LLM-guided captions, these lack the contextual specificity needed to establish an effective vocabulary for X-Decoder. Although AutoSeg also utilizes this segmentation backbone, it achieves a substantially higher performance. This comparison not only highlights the unique strengths of AutoSeg but also reveals the practical limitations of current LLM-aided approaches in segmenting scenes with multiple object categories.

Open-Vocabulary Segmentation Setting. We compare our method against existing OVS methods that require class

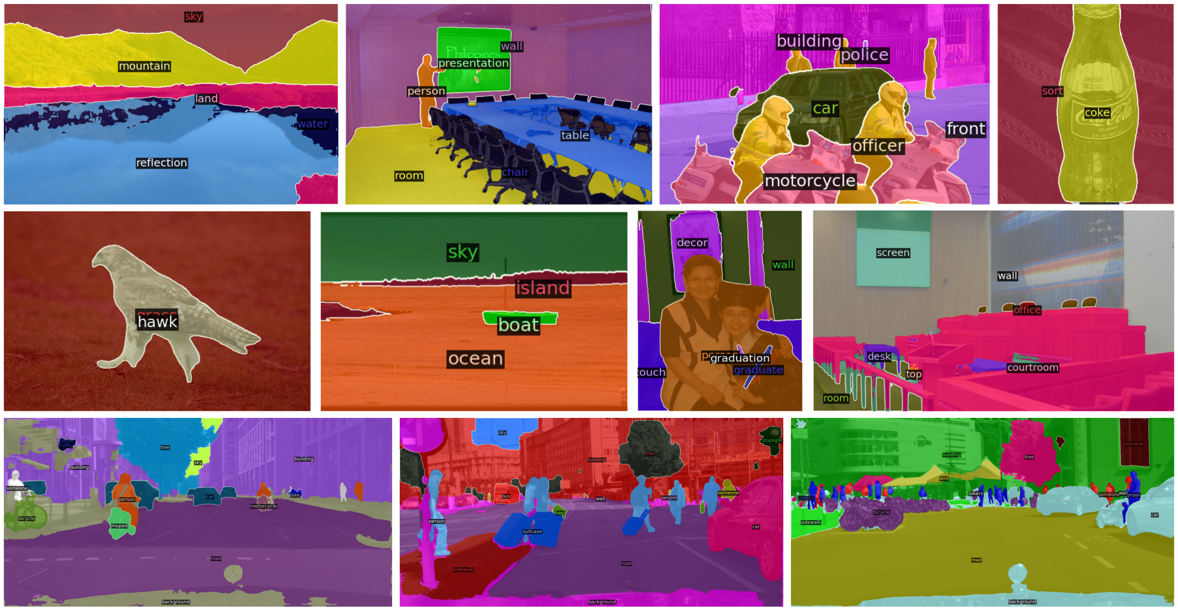

Figure 5. **Qualitative Results.** AutoSeg shows remarkable capability to identify out-of-vocabulary categories, such as *hawk* or *coke*, and segment them accurately across different datasets. Images are from the VOC/PC, ADE and CS datasets.

names to be given. Tab. 5 shows the results. In addition to the results mapped with LAVE, we provide results with one manual mapper per dataset (note that we provide a manual mapper for this table only, given the significant manual effort to construct it). Without any specification of class names through user input, AutoSeg matches 91%, 67%, 42% and 55% of the best OVS method performance on VOC, PC, ADE and CS respectively. It should be noted that this metric reflects the performance on the *known*, annotated classes, while additional open-ended classes are potentially mapped. Despite not being explicitly instructed with the known vocabulary contrary to OVS methods, AutoSeg is remarkably able to surpass six of them on VOC and PC, such as OpenSeg [10], ODISE [41] or OVSeg [19]. While still leaving room for improvement, our model compares competitively with OVS methods on ADE and CS, two very challenging datasets either high in number of unique classes or instances. Compared to ZeroSeg, AutoSeg achieves superior performance on VOC (88.2 over 20.1 mIoU) and PC (12.8 over 11.4 mIoU). Furthermore, we set the first benchmark on ADE and CS under the unknown vocabulary task setting. This outcome underscores the efficacy of our method in dealing with complex scenes that are open-ended in nature, such as ADE with its large number of rare classes. Finally, results obtained with LAVE mappings show little difference with manually constructed ones. This indicates that at marginal cost, LAVE can act as a feasible bridge between a known and unknown vocabulary. Our method deals well with high numbers of instances and can successfully identify various non-salient objects. While BBoost struggles with producing high-quality masks by itself (as seen in Tab. 2), it is highly effective in embedding target classes accurately which can be segmented with precision afterwards. Overall, our results demonstrate that the integration of BBoost with OVS harnesses the strengths of both, leading to enhanced open-ended recognition capabilities.

5.4. Qualitative Analysis

Fig. 5 displays qualitative results on the four datasets, where AutoSeg demonstrates its capability to accurately detect and segment relevant object categories present in the images but missing from the ground truth vocabulary. Remarkably, it predicts segmentation masks for classes such as *moped*, *presentation*, *coke*, *courtroom* or *hawk*. Moreover, in certain instances, AutoSeg successfully captures additional contextual details, such as *graduation* or *reflection*, illustrating the model's capabilities at semantic segmentation in a genuinely open-ended manner. Additional results, including failure cases, are included in the supplementary material.

6. Conclusion

This paper introduced AutoSeg, a novel method which leverages a vision-language model to automatically generate relevant target classes and segment them. Additionally, we proposed LAVE, a new evaluation framework which maps open-ended class names to ground-truth labels. AutoSeg shows open-ended recognition capabilities, achieving state-of-the-art performance in the zero label setting, while being competitive with open-vocabulary segmentation models which require provided ground-truth labels.

Acknowledgements. This work was supported by TomTom, the University of Amsterdam and the allowance of Top consortia for Knowledge and Innovation (TKIs) from the Netherlands Ministry of Economic Affairs and Climate Policy.

References

[1] Maxime Bucher, Tuan-Hung VU, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. In *NeurIPS*, 2019. 2

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 3

[3] Xi Chen, Shuang Li, Ser-Nam Lim, Antonio Torralba, and Hengshuang Zhao. Open-vocabulary panoptic segmentation with embedding modulation. In *ICCV*, 2023. 1, 3

[4] Jaemin Cho, Seunghyun Yoon, Ajinkya Kale, Franck Dernoncourt, Trung Bui, and Mohit Bansal. Fine-grained image captioning with clip reward. *NAACL*, 2022. 3

[5] Seokju Cho, Heeseong Shin, Sunghwan Hong, Seungjun An, Seungjun Lee, Anurag Arnab, Paul Hongsuck Seo, and Seungryong Kim. Cat-seg: Cost aggregation for open-vocabulary semantic segmentation. *CVPR*, 2024. 1, 3, 7

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 6, 7

[7] Xiaoyi Dong, Jianmin Bao, Yinglin Zheng, Ting Zhang, Dongdong Chen, Hao Yang, Ming Zeng, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Maskclip: Masked self-distillation advances contrastive language-image pretraining. *CVPR*, 2023. 1, 3

[8] 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. *ICLR*, 2021. 3

[9] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 1, 5, 6, 7

[10] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. *ECCV*, 2022. 1, 2, 3, 7, 8

[11] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020. `https://github.com/explosion/spaCy`. 6

[12] Laurynas Karazija, Iro Laina, Andrea Vedaldi, and Christian Rupprecht. Diffusion Models for Zero-Shot Open-Vocabulary Segmentation. *ECCV*, 2024. 3

[13] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *ICCV*, 2023. 2, 7

[14] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in neural information processing systems*, 24, 2011. 4, 6

[15] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. 4

[16] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. *CVPR*, 2024. 2, 7

[17] Boyi Li, Kilian Q. Weinberger, Serge J. Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *ICLR*, 2022. 1, 2, 7

[18] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 1, 3, 6, 7

[19] 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. *CVPR*, 2023. 1, 2, 3, 7, 8

[20] Krzysztof Lis, Krishna K. Nakka, Pascal Fua, and Mathieu Salzmann. Detecting the unexpected via image resynthesis. *ICCV*, 2019. 1

[21] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 3, 7

[22] Quande Liu, Youpeng Wen, Jianhua Han, Chunjing Xu, Hang Xu, and Xiaodan Liang. Open-world semantic segmentation via contrasting and clustering vision-language embedding. In *ECCV*, 2022. 3

[23] Chaofan Ma, Yuhuan Yang, Yanfeng Wang, Ya Zhang, and Weidi Xie. Open-vocabulary semantic segmentation with frozen vision-language models. In *BMVC*, 2022. 1

[24] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, pages 281–297, 1967. 4

[25] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3523–3542, 2021. 1

[26] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. 3

[27] 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 *CVPR*, 2014. 6, 7

[28] Prashant Pandey, Mustafa Chasmai, Monish Natarajan, and Brejesh Lall. A language-guided benchmark for weakly supervised open vocabulary semantic segmentation. *arXiv preprint arXiv:2302.14163*, 2023. 1

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual

models from natural language supervision. In *ICML*, 2021. 1, 2, 5

[30] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *EMNLP*, 2019. 5

[31] Shuhuai Ren, Aston Zhang, Yi Zhu, Shuai Zhang, Shuai Zheng, Mu Li, Alex Smola, and Xu Sun. Prompt pre-training with twenty-thousand classes for open-vocabulary visual recognition. In *NeurIPS*, 2023. 3

[32] Pitchaporn Rewatbowornwong, Nattanat Chatthee, Ekapol Chuangsuwanich, and Supasorn Suwajanakorn. Zero-guidance segmentation using zero segment labels. In *ICCV*, 2023. 3, 5, 7

[33] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *ECCV*, 2016. 3

[34] Tal Shaharabany, Yoad Tewel, and Lior Wolf. What is where by looking: Weakly-supervised open-world phrase-grounding without text inputs. *NeurIPS*, 2022. 3

[35] Peize Sun, Shoufa Chen, Chenchen Zhu, Fanyi Xiao, Ping Luo, Saining Xie, and Zhicheng Yan. Going denser with open-vocabulary part segmentation. In *ICCV*, 2023. 3

[36] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. 5

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 4

[38] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero- and few-label semantic segmentation. In *CVPR*, 2019. 2

[39] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. *CVPR*, 2022. 3

[40] Jilan Xu, Junlin Hou, Yuejie Zhang, Rui Feng, Yi Wang, Yu Qiao, and Weidi Xie. Learning open-vocabulary semantic segmentation models from natural language supervision. In *CVPR*, 2023. 3

[41] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-Vocabulary Panoptic Segmentation with Text-to-Image Diffusion Models. *ICCV*, 2023. 2, 3, 7, 8

[42] 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. *ECCV*, 2022. 3

[43] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *CVPR*, 2023. 1, 3, 7

[44] Apurwa Yadav, Aarshil Patel, and Manan Shah. A comprehensive review on resolving ambiguities in natural language processing. *AI Open*, 2021. 5

[45] Muyang Yi, Quan Cui, Hao Wu, Cheng Yang, Osamu Yoshie, and Hongtao Lu. A simple framework for text-supervised semantic segmentation. In *CVPR*, 2023. 1, 3

[46] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. In *NeurIPS*, 2023. 1, 3, 7

[47] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 1, 6, 7

[48] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, Nanyun Peng, Lijuan Wang, Yong Jae Lee, and Jianfeng Gao. Generalized decoding for pixel, image, and language. *CVPR*, 2023. 2, 3, 5, 6, 7