

Towards Open-Vocabulary Semantic Segmentation Without Semantic Labels

Heeseong Shin¹ Chaehyun Kim¹ Sunghwan Hong² Seokju Cho¹
Anurag Arnab^{†,3} Paul Hongsuck Seo^{†,2} Seungryong Kim^{†,1}
¹KAIST ²Korea University ³Google Research
{hsshin98, kchyun, seokju.cho, seungryong.kim}@kaist.ac.kr¹
{sung_hwan, phseo}@korea.ac.kr² aarnab@google.com³

Abstract

Large-scale vision-language models like CLIP have demonstrated impressive open-vocabulary capabilities for image-level tasks, excelling in recognizing *what* objects are present. However, they struggle with pixel-level recognition tasks like semantic segmentation, which additionally require understanding *where* the objects are located. In this work, we propose a novel method, **PixelCLIP**, to adapt the CLIP image encoder for pixel-level understanding by guiding the model on *where*, which is achieved using unlabeled images and masks generated from vision foundation models such as SAM and DINO. To address the challenges of leveraging masks without semantic labels, we devise an online clustering algorithm using learnable class names to acquire general semantic concepts. PixelCLIP shows significant performance improvements over CLIP and competitive results compared to caption-supervised methods in open-vocabulary semantic segmentation. Project page is available at <https://cvlab-kaist.github.io/PixelCLIP>

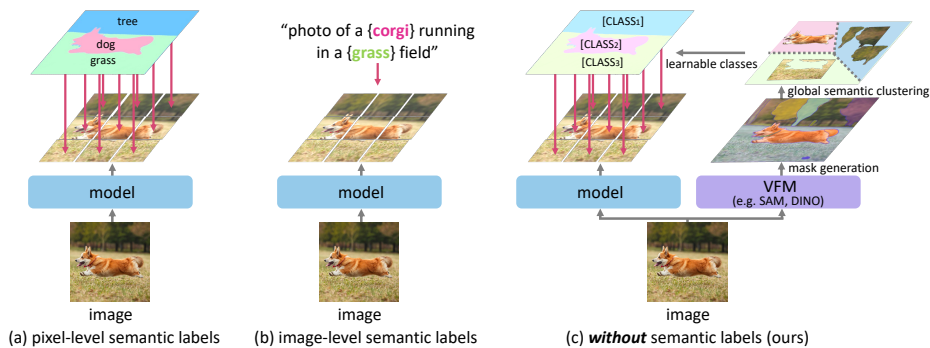


Figure 1: **Illustration of different approaches for open-vocabulary semantic segmentation.** In contrast to existing methods utilizing (a) pixel-level semantic labels [1, 2, 3, 4, 5, 6] or (b) image-level semantic labels [7, 8, 9, 10, 11, 12], we leverage unlabeled masks as supervision, which can be freely generated from vision foundation models such as SAM [13] and DINO [14].

1 Introduction

Semantic segmentation is a fundamental task in computer vision where the goal is to identify class labels for each pixel within the given image. However, segmentation datasets often require extensive human effort to obtain densely-annotated semantic labels, limiting their scalability. In this regard, recent advances in large-scale pre-trained vision-language models, *e.g.* CLIP [15] and ALIGN [16],

[†]Corresponding authors



Figure 2: **Visualization of masks from vision foundation models.** We visualize the masks generated by SAM [13] and by clustering image features from DINO [14]. Although such models can freely generate fine-grained masks, the resulting masks can be too small or incomplete to have semantic meaning. To address this over-segmentation issue, we employ online clustering [18] of the masks into semantically meaningful groups defined globally for given images.

have facilitated open-vocabulary semantic segmentation [1, 3, 2, 17, 4, 6], which aims to generalize semantic segmentation into unbounded range of classes. Despite showing remarkable generalization capabilities, they still require pixel-level semantic labels for leveraging the image-level pre-trained vision-language models for semantic segmentation.

Recently, several studies [11, 12, 7, 8] have pioneered open-vocabulary semantic segmentation without densely-annotated semantic labels. These studies often utilize image-level semantic labels, such as image captions, to enhance the pre-trained vision-language models like CLIP for semantic segmentation. However, image captions typically provide information about *what* is in the image, but without *where* it is. Since CLIP is already effective in recognizing *what* the objects are, this causes models to only implicitly learn object locations, leading to sub-optimal performance or requiring millions of image-caption pairs to compensate for this weak supervision [8, 7]. Instead, we focus on informing CLIP about *where* objects are located to address the missing information.

In this study, we propose a novel approach to achieve open-vocabulary semantic segmentation *without* leveraging semantic labels, but through guiding the pre-trained vision-language models, such as CLIP, on *where* to look. We leverage recent vision foundation models (VFMs), such as DINO [14] and SAM [13], to partition images into fine-grained regions to indicate *where* to look. Consequently, we explore methods to effectively leverage these masks for fine-tuning the image encoder of CLIP.

In contrast to existing works that leverage semantic labels [6, 19, 7], we do not have any captions or class names that can be fed to the text encoder of CLIP. To leverage its knowledge, we devise a method that employs prompt learning [20, 21] on the text encoder of CLIP to construct *learnable classes*. Setting the learnable classes as a centroid, we propose applying the online clustering algorithm [18, 22] along the given masks to gather them into semantically meaningful groups, as shown in Fig. 2. We keep these learnable classes global across the entire images, which guides the learnable classes to contain the general semantic concepts. Despite the absence of semantic labels, our method is able to jointly leverage the image encoder and text encoder of CLIP during training, successfully achieving dense open-vocabulary recognition.

Our framework, called PixelCLIP, achieves significant improvements to CLIP, on average of +16.2 mIoU in open-vocabulary semantic segmentation. Moreover, despite not using any semantic labels, PixelCLIP shows competitive performance in comparison to image-level supervised methods using captions [7, 9, 10], demonstrating the effectiveness of unlabeled masks for. We further show the effectiveness of PixelCLIP for classifying masks from various open-vocabulary segmentation models, which can be simply done by replacing the CLIP within existing methods. We also provide extensive ablation studies to validate our choices, with a detailed analysis of our method.

We summarize our contribution as follows:

- We propose a novel formulation of learning from images *without* semantic label for open-vocabulary semantic segmentation by leveraging masks generated from DINO and SAM to fine-tune vision-language models.
- We propose to globally cluster semantically similar masks by employing an online clustering algorithm, while learning class prompts for representing semantic clusters.
- We demonstrate significant gains in open-vocabulary semantic segmentation, even surpassing methods leveraging image-level semantic labels, and provide thorough ablation studies with analysis to validate our framework.

2 Related Work

2.1 Open-vocabulary semantic segmentation

Open-vocabulary semantic segmentation [2, 23] aims to label each pixel within an image into an unbounded range of classes. In this regard, recent works [1, 17, 2, 6, 24] aim to generalize to classes unseen during training through leveraging pre-trained vision-language models, such as CLIP [15]. Despite their remarkable performance, they leverage per-pixel semantic labels during their training, which requires expensive cost to annotate. Instead, we focus on the weakly-supervised setup, where the goal is to zero-shot transfer to segmentation task *without* densely-annotated class labels [11, 12, 25, 5, 26, 7, 27], utilizing image-level labels as supervision or even no labels at all.

In this regard, recent studies [11, 12, 25, 5] leverage image caption as supervision. GroupViT [11] and ViL-Seg [12] are pioneering works for identifying groups or clusters emerging from captions. Along with the advance of vision-language models, SegCLIP [9] and TCL [7] leverage pre-trained CLIP and learn additional decoder modules to learn dense vision-language alignment. PACL [8] learns additional embedding layers to enhance the patch-level alignment in vision-language models and SAM-CLIP [10] attempts to merge SAM [13] and CLIP [15] into a unified model by additionally leveraging unlabeled mask data from SAM. Apart from these approaches, we avoid employing *any* semantic labels [26, 28], but leverage vision foundation models to obtain masks as a source for supervision to fine-tune the CLIP image encoder for achieving open-vocabulary semantic segmentation.

2.2 Fine-tuning vision-language models for dense prediction

Recent large-scale pre-trained vision-language models have shown its effectiveness for jointly understanding images and language [29, 15, 16]. Notably, CLIP [15], trained with web-scale image-caption pairs, has been widely popularized for transferring its open-vocabulary recognition capabilities to various downstream tasks [26, 30, 31, 32]. However, despite its success in image-level tasks like image classification, CLIP tends to struggle in dense prediction tasks [17, 26, 6], such as object detection and semantic segmentation. This originates from CLIP being trained from image-level supervision being captions, hence exhibits bias towards the global image rather than fine-grained regions within the image [17]. While non-learnable approaches, such as MaskCLIP [26] show improvements by slightly modifying the architecture, CLIP still shows limited capabilities in dense predictions in comparison to its global understanding.

To address this, OWL-ViT [33] directly fine-tunes pre-trained vision and text encoders to downstream open-vocabulary detection task, and CAT-Seg [6] introduces a cost aggregation scheme for fine-tuning the encoders of CLIP for semantic segmentation. Alternatively, ZegCLIP [19] and Xu et al. [3] implement prompt tuning [21, 20] for tuning the image and text encoders of CLIP. Instead of fine-tuning the full model, they learn prompt tokens that serve as a global prefix for the encoders of CLIP. While such methods show remarkable results from fine-tuning the encoders of CLIP for dense downstream tasks, they require densely annotated detection and segmentation data for training.

2.3 Vision foundation models

With the advent of large-scale learning enabled by scalable vision backbone architectures [34, 35] and vast amounts of data, diverse vision foundation models are emerging in the field of computer vision. In this regard, self-supervised methods [36, 37, 38, 39] have demonstrated the effectiveness of its rich visual representations for various downstream tasks. Especially, DINO [14] exerted strengths in fine-grained semantic recognition [40, 41], making it highly effective for object detection and image segmentation. Moreover, DINO features have also been demonstrated for yielding fine-grained masks within the image through applying the k -means clustering with its features [42, 43, 44].

On the other hand, the segment anything model (SAM) [13] has demonstrated its capability for generating fine-grained, high-quality segmentation masks for any object in an image. Through its self-annotation pipeline, SAM has collected an unprecedented amount of mask annotation for achieving its capabilities. While we can freely leverage SAM to obtain detailed masks in any given image, we mainly utilize the pre-computed masks within the collected dataset, SA-1B. Both DINO and SAM, however, yield unlabeled masks without semantic labels as both models are also trained without semantic labels, presenting a challenge for leveraging their masks for achieving dense vision-language recognition.

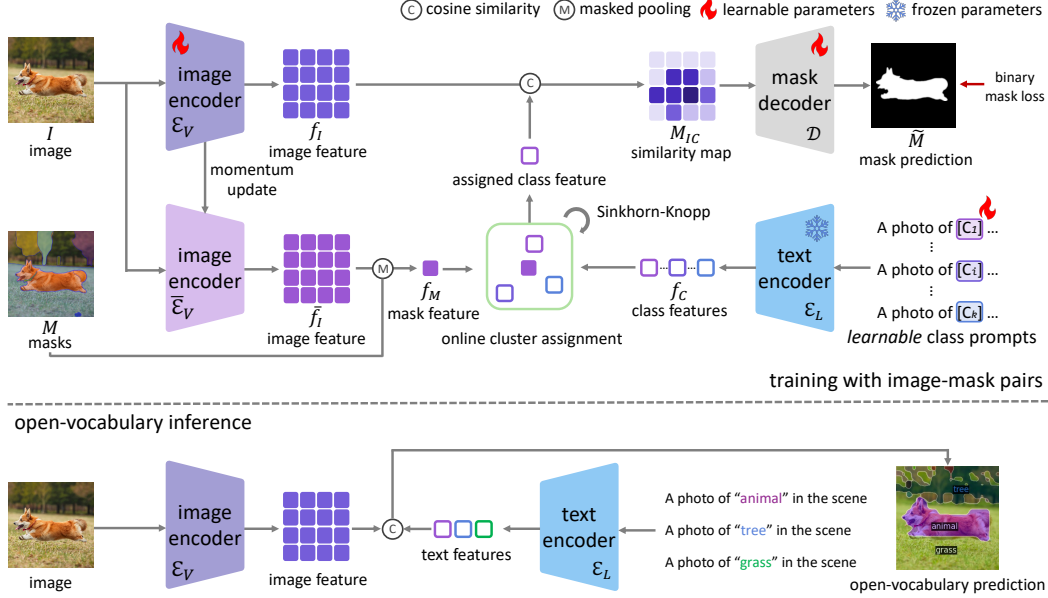


Figure 3: **Illustration of our overall framework.** We provide illustration of PixelCLIP, utilizing unlabeled images and masks for fine-tuning the image encoder of CLIP, enabling open-vocabulary semantic segmentation. We note that the momentum image encoder and the mask decoder are only leveraged during training, and inference is only done with image and text encoders of CLIP.

3 Methodology

In this section, we first establish our problem formulation of learning dense vision-language alignment from images paired with masks, generated from vision foundation models. Next, we discuss the challenges of leveraging masks as supervision for fine-tuning the image encoder of CLIP and finally, present our methodology of semantic clustering of masks to address the challenges.

3.1 Preliminaries

Given an input image $I \in \mathbb{R}^{H \times W \times 3}$, open-vocabulary semantic segmentation [6, 7] aims to label each pixel within an image with classes given in free-form text. As a training signal, semantic labels offer a set of S textual descriptions for a semantic class $T = \{T_i\}_{i=1}^S$ related to I . This can be directly utilized with the CLIP text encoder $\mathcal{E}_L(\cdot)$ to obtain text features $f_T = \mathcal{E}_L(T) \in \mathbb{R}^{S \times d}$, where d is the hidden dimension. Dense image features $f_I = \mathcal{E}_V(I) \in \mathbb{R}^{h \times w \times d}$, where $h \times w$ is the output feature resolution, are then extracted. We finally obtain dense image-text similarity map $M_{IT} \in \mathbb{R}^{h \times w \times S}$:

$$M_{IT}(x, y, n) = \frac{f_I(x, y) \cdot f_T(n)}{\|f_I(x, y)\| \|f_T(n)\|}. \quad (1)$$

This can be interpreted as soft binary masks predicted from image and text features of CLIP, and be supervised with binary mask loss $\mathcal{L}_{\text{mask}}$ in a pixel-level manner to fine-tune CLIP [6].

3.2 Integrating masks into CLIP features

In this work, we do not have any access to T , but are only given unlabeled masks $M \in \mathbb{R}^{H \times W \times N}$, where N denotes the number of masks for the given image I . Hence, we devise methods to predict masks by incorporating M into CLIP features. We aim to fine-tune the CLIP image encoder $\mathcal{E}_V(\cdot)$ through leveraging unlabeled masks M as supervision. Since M is generated from vision foundation models, *e.g.* DINO or SAM, this presents us with the challenge of not having any semantic labels.

In order to integrate masks into CLIP, a straightforward approach would be employing the masks M with the CLIP image feature map f_I to obtain per-mask CLIP features. While there could be various methods to extract regional CLIP features [26, 45, 5], we apply mask pooling over f_I to obtain mask

pooled features $f_M = \text{MaskPool}(f_I, M) \in \mathbb{R}^{N \times d}$. Consequently, we can leverage f_M to obtain image-mask similarity map denoted $M_{IM} \in \mathbb{R}^{h \times w \times N}$:

$$M_{IM}(x, y, n) = \frac{f_I(x, y) \cdot f_M(n)}{\|f_I(x, y)\| \|f_M(n)\|}. \quad (2)$$

This allows us to supervise the model with a binary mask loss $\mathcal{L}_{\text{mask}}$ for fine-tuning CLIP with given image I and unlabeled masks M . In practice, since M_{IM} has the same resolution as the feature map from the CLIP image encoder f_I , we employ a light-weight decoder \mathcal{D} to mitigate the resolution gap between M_{IM} and M , as shown in Fig. 3. This can be written as $\mathcal{D} : \mathbb{R}^{h \times w} \rightarrow \mathbb{R}^{h' \times w'}$, where $h' \times w'$ is resolution for the upsampled mask. Therefore, the output of the model can be updated as $\tilde{M} = \mathcal{D}(M)$.

3.3 Semantic clustering of masks

Upon using mask pooled CLIP image features f_M to predict M_{IM} , however, we find the masks generated from DINO and SAM to often over-segment the image, resulting in too small or incomplete masks as seen in Fig. 2. This would require CLIP to forcefully discriminate regions that are semantically similar, impeding the training process.

In this regard, we propose to group semantically similar masks into *clusters* and predict based on the clusters rather than individual masks. Moreover, we aim to define this cluster *globally*, which is shared across the entire training process rather than for each image or iteration. This would be analogous to constructing pixel-level semantic labels, where a fixed set of classes defined over the dataset is equivalent to each cluster. However, the difference is that there is no pre-defined set of classes that we can define the clusters with. While we could heuristically pre-define such classes, we describe our learnable method for globally clustering masks into semantically meaningful groups.

Online clustering via learnable class prompts. To globally cluster masks into semantic categories, we propose representing these clusters using CLIP *text* features as centroids for clustering mask features. Given that the CLIP text encoder is trained with a broad understanding of natural language semantics, we expect these clusters to capture meaningful semantics by leveraging its comprehensive pre-trained knowledge. In this regard, we take a learnable approach, where each cluster is defined by class-specific learnable prompts fed into the CLIP text encoder. Unlike existing prompt learning methods, which typically focus on learning a task-specific prefix [20, 21, 3], we aim to learn prompt tokens that represent each class. For instance, in the sentence ‘‘A photo of an object’’, traditional prompting methods would learn the tokens for the ‘‘A photo of a’’ prefix, whereas our method focuses on learning the token for the ‘‘object.’’

Specifically, given the number of clusters k , we can define prompt tokens as $C \in \mathbb{R}^{k \times l \times d_e}$, where l is the token length of the prompt and d_e is the dimension of the token embeddings. From this, we can utilize the CLIP text encoder $\mathcal{E}_L(\cdot)$ to obtain a set of *class* features $f_C = \mathcal{E}_L(P^*, C) \in \mathbb{R}^{k \times d}$ in the form of CLIP text features, where P^* is a fixed template for the CLIP text encoder, such as ‘‘A photo of a { } in the scene.’’ While we could assign each mask f_M with f_C in a winner-takes-all manner, we desire the classes to encode general semantics across all images. Therefore, we assume that we can equally divide m masks within a minibatch [18, 14], into k clusters given a sufficient amount of masks.

Consequently, we aim to find an assignment $Q \in \mathbb{R}_+^{k \times m}$ based on the image-text similarity between the mask pooled features f_M and the class text features, which can be defined as:

$$\max_{Q \in \mathcal{Q}} \text{Tr}(Q^\top F_M^\top f_C) + \varepsilon H(Q), \quad \text{s.t.} \quad Q \in \mathbb{R}_+^{k \times m}, \quad Q^\top \mathbb{1}_k = \frac{1}{m} \mathbb{1}_m, \quad Q \mathbb{1}_m = \frac{1}{k} \mathbb{1}_k, \quad (3)$$

where F_M is the set of all m mask features f_M within the minibatch, and $\mathbb{1}_k$ denotes the k -dimensional vector of ones. H is the entropy function, $H(Q) = -\sum_{ij} Q_{ij} \log Q_{ij}$ with ε as a hyperparameter. The solution Q from Eq. 3 is an assignment matrix defining which of the k clusters each m mask should belong to, hence ε determines the smoothness of this mapping Q by scaling the entropy regularization from H . The equipartition constraint, $Q^\top \mathbb{1}_k = \frac{1}{m} \mathbb{1}_m$, $Q \mathbb{1}_m = \frac{1}{k} \mathbb{1}_k$ encourages the class features f_C to be selected at least m/k times on average, allowing to learn general concepts represented by the masks within the dataset. In practice, with the soft assignment relaxation [46], Q

can be solved as follows:

$$Q = \text{diag}(u) \exp\left(\frac{F_M^\top f_C}{\varepsilon}\right) \text{diag}(v), \quad (4)$$

where $u \in \mathbb{R}^k$, $v \in \mathbb{R}^m$ denote renormalization vectors, which can be efficiently computed by Sinkhorn-Knopp algorithm [46].

Finally, we can re-write the prediction of our model to be a cosine-similarity map between f_I and f_C :

$$M_{IC}(x, y, i) = \frac{f_I(x, y) \cdot f_C(i)}{\|f_I(x, y)\| \|f_C(i)\|}, \quad (5)$$

thereby predicting masks for $f_C(i)$ being the i -th class feature from f_C , which we have obtained from clustering mask pooled features f_M . Accordingly, ground truth masks M are also clustered according to Q by converting it into hard assignment with the argmax operator [47, 22]. This can be written as $\bar{M} \in \mathbb{R}^{k \times H \times W}$ where \bar{M}_i is the union of masks assigned into the cluster represented by i -th learned class $f_C(i)$.

Momentum encoder for integrating mask features. Since we jointly optimize the CLIP image encoder $\mathcal{E}_V(\cdot)$ as well as the learnable class feature f_C , we may experience instability during our training process, or forgetting of the pre-trained knowledge [48]. To stabilize the training, we keep a momentum encoder [39, 38] for obtaining f_M as seen in Fig. 3. Therefore, we update f_M as $f_M = \text{MaskPool}(\bar{\mathcal{E}}_V(I), M)$, where $\bar{\mathcal{E}}_V$ is the momentum encoder of the CLIP image encoder, updated with momentum γ . This can be denoted as $\theta_{\bar{V}}^l \leftarrow \gamma \theta_{\bar{V}}^l + (1 - \gamma) \theta_V^l$, where $\theta_{\bar{V}}^l, \theta_V^l$ are model parameters of $\bar{\mathcal{E}}_V$ and \mathcal{E}_V , respectively.

4 Experiments

4.1 Implementation details

For training, we employ per-pixel binary cross-entropy loss as $\mathcal{L}_{\text{mask}}$ to jointly train all of the components [6]. For all our experiments, we use a single text prompt ‘‘A photo of {} in the scene’’ for P^* , including for our learnable class prompts while training and for inference, we apply prompt ensemble strategy [30] with 7 additional prompts originally curated from CLIP [15]. We train our model on SA-1B [13] dataset, where we randomly sample 5% of the images. We train for 10000 iterations with a batch size of 48 for all experiments. For experiments using masks from DINO, we obtain masks with k -means clustering where we set $k = 16$. For experiments using masks from SAM, we use the unlabeled mask annotation in the SA-1B dataset. Without specification, we report results on ConvNeXt-B [49] backbone with mask annotation from SAM, which takes approximately 6 hours to train with 4 NVIDIA A6000 GPUs. We provide more details in the supplementary materials.

4.2 Experimental setting

Following Cha et al. [7], we evaluate our model on zero-shot transfer to semantic segmentation on the validation sets of COCO-Stuff [50], ADE-20K [51], PASCAL-Context [52], PASCAL VOC [53], and CityScapes [54]. For CLIP [15], we apply MaskCLIP [26] for ViT backbone for extracting image features, and remove the global pooling layer for OpenCLIP [55] with ConvNeXt [49] backbone. We note that we do not apply any post-processing to the predictions and for the compared methods. For the evaluation metric, we employ the mean Intersection over Union (mIoU).

4.3 Results

Open-vocabulary semantic segmentation. We provide results for quantitative comparisons in Tab. 1. We first compare with CLIP, and demonstrate remarkable gains in all benchmarks, bringing in an average of +16.2 mIoU improvement. Since we do not have comparable baselines without leveraging semantic labels, we further provide a comparison with image-level supervised methods [7, 9, 10]. Surprisingly, PixelCLIP surpasses TCL [7] and SegCLIP [9] in all benchmarks while using only a fraction of the images *without* semantic labels. Furthermore, we show competitive performance compared to SAM-CLIP, which uses not only 40 million image-level semantic labels, but also leverages the SA-1B dataset on a similar scale to our framework.

Table 1: **Quantitative comparison on open-vocabulary semantic segmentation.** We compare in open-vocabulary semantic segmentation with vision-language models, as well as image-level supervised methods. *: Images were seen during training. †: Masks from SA-1B [13] were used.

Method	Training Dataset	Backbone	Additional Labels	VFM	Evaluation Dataset				
					COCO-St.	ADE-150	Context	CityScapes	VOC
GroupViT [11]	CC12M [56], YFCC15M [57]	ViT-S/16	-	-	15.3	9.2	23.4	11.1	79.7
CLIPpy [25]	HQITP-134M [25]	ViT-B/16	-	-	-	13.5	-	-	52.2
OVSegmentor [58]	CC4M [58]	ViT-B/16	-	DINO	-	-	20.4	-	53.8
CLIP [15]	WIT-400M [15]	ViT-B/16	-	-	16.5	13.2	25.6	14.9	73.9
OpenCLIP [55]	LAION-2B [59]	ConvNeXt-B	-	-	12.8	13.1	16.5	16.2	34.8
<i>Training with additional image-level semantic labels</i>									
SegCLIP [9]	COCO [60], CC12M [56]	ViT-B/16	Captions	CLIP	26.5*	-	24.7	-	52.6
TCL [7]	CC3M, CC12M [56]	ViT-B/16	Captions	CLIP	19.6	14.9	30.3	23.1	77.5
SAM-CLIP [10]	Merged-41M [10]	ViT-B/16	Captions	CLIP, SAM	-	17.1	29.2	-	60.6
<i>Training without additional semantic labels</i>									
ZeroSeg [28]	ImageNet-1K [61]	ViT-B/16	-	CLIP	20.2	-	20.4	-	40.8
		ViT-B/16	-	CLIP, DINO	22.2	17.4	34.3	22.9	83.8
		ViT-B/16	-	CLIP, SAM†	23.6	18.7	37.9	27.2	85.9
PixelCLIP (Ours)	5% SA-1B [13] (0.5M)	ConvNeXt-B	-	CLIP, DINO	20.2	19.4	32.7	30.0	62.9
		ConvNeXt-B	-	CLIP, SAM†	21.4	20.3	35.4	34.8	67.2

Table 2: **Quantitative comparison on zero-shot mask classification.** We compare the results for mask classification using ground truth masks and generated masks from ZegFormer [1] and FC-CLIP [24]. To evaluate zero-shot mask classification from CLIP, we report the results from the zero-shot branch for both methods.

VLM	Method	Backbone	Evaluation Dataset				
			COCO-St.	ADE-150	Context	CityScapes	VOC
OpenCLIP [55]	Zegformer [1]	ConvNeXt-B	15.3	19.1	24.7	26.5	51.8
PixelCLIP (Ours)	Zegformer [1]	ConvNeXt-B	23.9 (+8.6)	21.5 (+2.4)	38.5 (+13.8)	34.2 (+7.7)	71.5 (+19.7)
OpenCLIP [55]	FC-CLIP [24]	ConvNeXt-L	37.3	27.4	42.8	35.8	91.4
PixelCLIP (Ours)	FC-CLIP [24]	ConvNeXt-L	46.8 (+9.5)	30.1 (+2.7)	52.2 (+9.4)	48.1 (+12.3)	90.7 (-0.7)
OpenCLIP [55]	Ground Truth	ConvNeXt-B	23.8	30.2	31.4	32.8	68.3
PixelCLIP (Ours)	Ground Truth	ConvNeXt-B	34.2 (+10.4)	34.6 (+4.4)	51.2 (+18.4)	41.4 (+8.6)	85.4 (+17.1)

Zero-shot mask classification. We provide results for evaluating mask classification in Tab. 2. We consider ZegFormer [1] and FC-CLIP [24] as baselines since they first predict masks, then employ CLIP as a zero-shot mask classifier within their framework, and also provide results with ground-truth masks to simulate having oracle mask predictions. For all methods, we apply masked pooling to CLIP image feature map to classify masks. For ZegFormer [1] and FC-CLIP [24], reported results are only from the zero-shot prediction branch to solely ablate our gains. We highlight that PixelCLIP can be readily applied to existing frameworks that leverage CLIP as a zero-shot mask classifier, and bring instantaneous improvements by simply replacing the model and weights of CLIP.

Qualitative results. We provide qualitative results for open-vocabulary semantic segmentation in Fig. 4 compared with results from CLIP, highlighting the dense open-vocabulary recognition capabilities of our framework. We further provide qualitative results in the supplementary materials.

4.4 Ablation studies

In Tab. 3, we show ablation studies on open-vocabulary semantic segmentation to validate our design choices. We report results without prompt ensembling for ablations, and also report results from OpenCLIP [55] as a baseline.

Component analysis. In Tab. 3 (a), we provide results for ablating our key components. Notably, we observe that without global semantic clustering of masks, the framework collapses and loses the pre-trained knowledge of CLIP. This validates the challenge presented by leveraging unlabeled masks and demonstrates the crucial role of our proposed clustering approach. Moreover, we observe constant improvements over all datasets with our learnable class prompt, proving our approach of leveraging the text encoder of CLIP to define the clusters in the form of prompt learning. We also observe constant gains with the momentum encoder for extracting mask pooled features f_M .

Number of clusters. In Tab. 3 (b), we compare the results of the variants of the proposed method by varying the number of clusters k . We find that scaling k does not necessarily guarantee performance boosting, but it generally improves until k is set to 64 and tends to degrade as k grows. Considering that with an extremely large number for k , we can assign each of the masks to individual clusters (e.g. 1 billion for SA-1B.) This scenario would virtually be identical to not having semantic clustering as

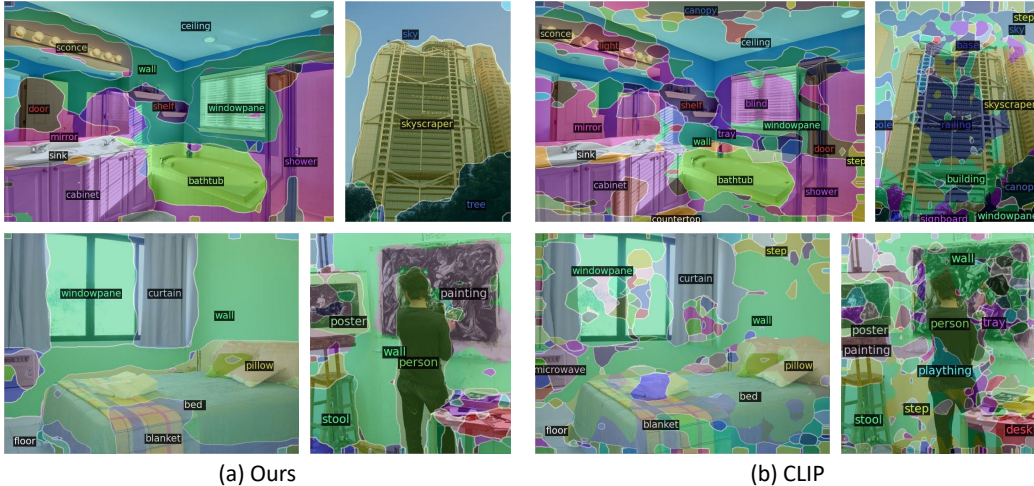


Figure 4: **Comparison between PixelCLIP and CLIP.** We provide qualitative comparison on ADE-20K [51] dataset with PixelCLIP and CLIP. We demonstrate the dense visual recognition capabilities achieved from fine-tuning CLIP, whereas CLIP shows results with significant noise.

Table 3: **Ablation studies.** We show results on open-vocabulary semantic segmentation for validating our design choices. We also report results from OpenCLIP [55] as baseline in the results.

Component	Evaluation Dataset				
	COCO	ADE-150	Context	CityScapes	VOC
Baseline	12.8	13.1	16.5	16.2	34.8
Ours	21.1	20.2	34.2	33.2	66.0
w/o Semantic Clustering	0.8	2.1	4.2	4.4	6.0
w/o CLIP Text Encoder	17.9	18.5	29.9	28.9	53.5
w/o Class Prompt	18.2	18.8	30.1	28.1	54.4
w/o Momentum	19.4	18.5	28.8	27.2	58.2

(a) **Component analysis.** We validate the core components of our framework by ablating each components. Notably, global clustering of masks shows its importance for facilitating the framework.

l	Evaluation Dataset				
	COCO	ADE-150	Context	CityScapes	VOC
Baseline	12.8	13.1	16.5	16.2	34.8
1	20.2	19.7	32.7	30.8	64.5
4	21.1	20.2	34.2	33.2	66.0
10	20.4	19.6	33.2	30.2	63.3
20	19.9	19.7	32.6	33.8	62.8

(c) **Length of learnable prompt token.** For varying l , we find that $l = 4$ shows best overall performance.

seen in Table. 3 (a), and progressively growing k would slowly converge to this scenario. We further provide analysis in 4.5, studying the different aspects from varying k .

Length to represent learnable class prompts. Tab. 3 (c) compares the effects of varying the length of the learnable class prompt, l . We find that $l = 1$ shows lower scores in comparison to other lengths. We can interpret this as only describing a class with a single word, whereas having multiple words would better describe the depicted class. However, for $l = 4$ and larger, we find that increasing l does not result in a gain of performance, hence, we adopt $l = 4$ as default.

Effects of learnable prompt token. Finally, we compare PixelCLIP to having a pre-defined set of classes instead of using learnable prompt tokens. Specifically, we use 171 classes from COCO-Stuff [50], and do not apply online clustering for assignment when utilizing classes from COCO-Stuff, as it already yields text features with semantic meanings. We find apparent improvements over all the datasets as shown in Tab. 3 (d). We speculate that since the classes defined in COCO-Stuff are heuristically chosen, it is hard to ideally encompass various semantics and concepts that may appear in images, hence restricting the perception of the model to the finite set of classes.

(b) **Number of clusters.** For varying k , we find that scaling k larger than 64 does not show much improvements, while $k = 32$ also show competitive results.

k	Evaluation Dataset				
	COCO	ADE-150	Context	CityScapes	VOC
Baseline	12.8	13.1	16.5	16.2	34.8
32	19.8	19.4	33.0	31.3	60.5
64	21.1	20.2	34.2	33.2	66.0
128	21.0	20.3	33.5	30.1	64.1
256	21.3	20.4	33.6	30.0	64.1
512	21.2	20.2	32.7	29.8	62.7

(d) **Effects of utilizing learnable classes.** We compare our method of learnable class prompts to having fixed set of classes from COCO-Stuff [50].

Text.	Evaluation Dataset				
	COCO	ADE-150	Context	CityScapes	VOC
Baseline	12.8	13.1	16.5	16.2	34.8
Ours	21.1	20.2	34.2	33.2	66.0
COCO	19.5	17.7	30.0	24.8	63.3

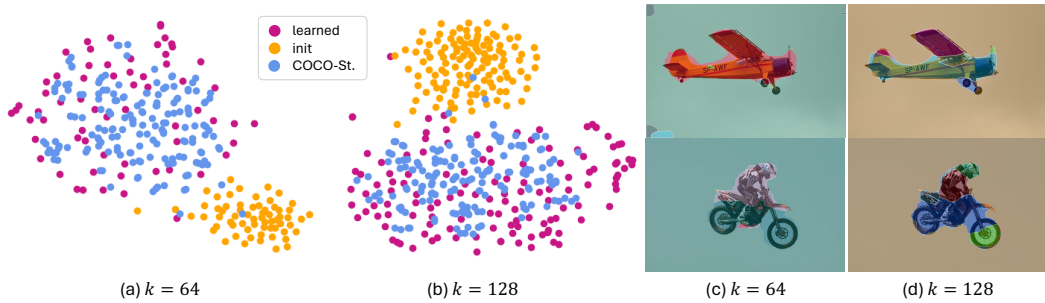


Figure 5: **Visualization of learned class prompts.** We visualize the text features from our learned class prompts, as well as text features from classnames of COCO-Stuff with t -SNE visualization in (a-b). We also visualize images inferred with the learned class prompts in (c-d).

4.5 Analysis

Learnable class prompt. We further analyze the learned class prompt in Fig. 5 (a-b) with t -SNE visualization on the text features encoded from the learned class prompts, as well as text features obtained from class names of COCO-Stuff. Since we initialize the class prompt tokens as random tokens, we observe that they are in a skewed distribution in the initial state. However, the learned prompts show that they are well-dispersed among the text features from COCO-Stuff, indicating that the class prompts have well-learned diverse semantic concepts within the text features. We observe well-distributed features both for $k = 64$ and $k = 128$.

Since the learned prompts should act as implicit class names, we visualize the results from inference with learned class prompts in Fig. 5 (c-d). Although both $k = 64$ and $k = 128$ show similar performance when evaluated, we observe that the prompts have learned more fine-grained semantics for $k = 128$. We generally observe human parts to be well distinguished; this could come from the SA-1B dataset, as there are numerous images with fine-grained masks representing human parts as annotations.

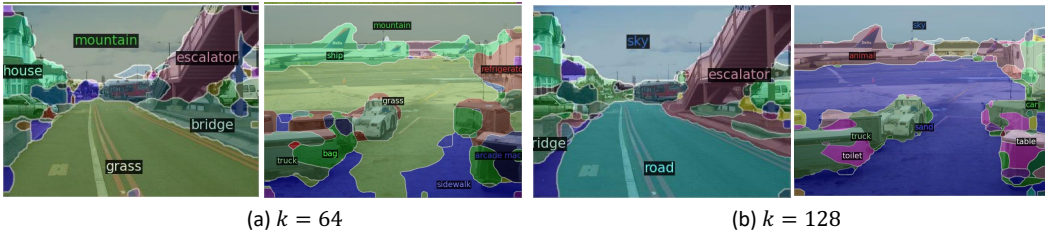


Figure 6: **Visualization of interpreting learned text prompt.** We provide visualization on results for predicting with learned class prompts, then mapping the results to classes in the dataset with the highest similarity to the prompt.

Interpreting learned classes. Considering the learned class prompts represent semantic concepts, we further study the learned embeddings by mapping each class embeddings to class names in COCO-Stuff with the highest cosine-similarity score. Fig. 6 shows results when we first inference the image features with learned class prompts, then map the results with the closest COCO classes. We can observe that with $k = 128$, as the prompt learns more diverse semantics, we observe more accurately mapped classes. However, we still see predictions with large disparity to the actual ground truth. We leave a more in-depth analysis of the learned classes for future investigation.

5 Conclusion

In this paper, we introduced PixelCLIP, a framework for leveraging unlabeled images and masks for fine-tuning the pre-trained vision-language models for open-vocabulary semantic segmentation. To address the unique challenges posed by incorporating unlabeled masks generated by vision foundation models into our framework, we propose global semantic clustering of the masks, with learnable class prompts to represent each cluster. We demonstrated PixelCLIP to show remarkable improvements to CLIP and its applicability to existing methods, providing instantaneous improvements, as well as surpassing methods that leverage image-level semantic labels such as image captions.

Acknowledgement. This research was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2019-II190075, RS-2024-00509279, RS-2020-II201819, RS-2024-00398115, Research on the reliability and coherence of outcomes produced by Generative AI) and the Culture, Sports, and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism (RS-2024-00348469, RS-2023-00266509), and National Research Foundation of Korea (RS-2024-00346597).

References

- [1] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11583–11592, 2022.
- [2] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVI*, pages 540–557. Springer, 2022.
- [3] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX*, pages 736–753. Springer, 2022.
- [4] 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.
- [5] 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.
- [6] Seokju Cho, Heeseong Shin, Sunghwan Hong, Anurag Arnab, Paul Hongsuck Seo, and Seungryong Kim. Cat-seg: Cost aggregation for open-vocabulary semantic segmentation, 2024.
- [7] Junbum Cha, Jonghwan Mun, and Byungseok Roh. Learning to generate text-grounded mask for open-world semantic segmentation from only image-text pairs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11165–11174, 2023.
- [8] Jishnu Mukhoti, Tsung-Yu Lin, Omid Poursaeed, Rui Wang, Ashish Shah, Philip HS Torr, and Ser-Nam Lim. Open vocabulary semantic segmentation with patch aligned contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19413–19423, 2023.
- [9] Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li. Segclip: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. In *International Conference on Machine Learning*, pages 23033–23044. PMLR, 2023.
- [10] Haoxiang Wang, Pavan Kumar Anasosalu Vasu, Fartash Faghri, Raviteja Vemulapalli, Mehrdad Farajtabar, Sachin Mehta, Mohammad Rastegari, Oncel Tuzel, and Hadi Pouransari. Sam-clip: Merging vision foundation models towards semantic and spatial understanding. *arXiv preprint arXiv:2310.15308*, 2023.
- [11] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18134–18144, 2022.
- [12] 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 *European Conference on Computer Vision*, pages 275–292. Springer, 2022.

- [13] 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.
- [14] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.
- [15] 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.
- [16] 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.
- [17] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yanan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. *arXiv preprint arXiv:2210.04150*, 2022.
- [18] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.
- [19] Ziqin Zhou, Bowen Zhang, Yinjie Lei, Lingqiao Liu, and Yifan Liu. Zegclip: Towards adapting clip for zero-shot semantic segmentation. *arXiv preprint arXiv:2212.03588*, 2022.
- [20] 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.
- [21] 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.
- [22] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019.
- [23] Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. *Advances in Neural Information Processing Systems*, 32, 2019.
- [24] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. *Advances in Neural Information Processing Systems*, 36, 2024.
- [25] Kanchana Ranasinghe, Brandon McKinzie, Sachin Ravi, Yinfei Yang, Alexander Toshev, and Jonathon Shlens. Perceptual grouping in contrastive vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5571–5584, 2023.
- [26] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *ECCV*, pages 696–712. Springer, 2022.
- [27] Nir Zabari and Yedid Hoshen. Semantic segmentation in-the-wild without seeing any segmentation examples. *arXiv preprint arXiv:2112.03185*, 2021.
- [28] Jun Chen, Deyao Zhu, Guocheng Qian, Bernard Ghanem, Zhicheng Yan, Chenchen Zhu, Fanyi Xiao, Mohamed Elhoseiny, and Sean Chang Culatana. Exploring open-vocabulary semantic segmentation without human labels. *arXiv preprint arXiv:2306.00450*, 2023.
- [29] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

- [30] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021.
- [31] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094, 2021.
- [32] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.
- [33] Matthias Minderer, Alexey Gritsenko, Maxim Neumann Austin Stone, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. Simple open-vocabulary object detection with vision transformers. *ECCV*, 2022.
- [34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [35] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.
- [36] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021.
- [37] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.
- [38] Jean-Bastien Grill, Florian Strub, Florent Alché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent—a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.
- [39] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.
- [40] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2(3):4, 2021.
- [41] Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence. *Advances in Neural Information Processing Systems*, 36, 2024.
- [42] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *CVPR*, 2022.
- [43] Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L. Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14543–14553, June 2022.
- [44] Yin Zhaoyun, Wang Pichao, Wang Fan, Xu Xianzhe, Zhang Hanling, Li Hao, and Jin Rong. Transfgu: A top-down approach to fine-grained unsupervised semantic segmentation. In *European Conference on Computer Vision*, pages 73–89. Springer, 2022.
- [45] Zheng Ding, Jieke Wang, and Zhuowen Tu. Open-vocabulary panoptic segmentation with maskclip. *arXiv preprint arXiv:2208.08984*, 2022.

- [46] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.
- [47] Tianfei Zhou, Wenguan Wang, Ender Konukoglu, and Luc Van Gool. Rethinking semantic segmentation: A prototype view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2582–2593, 2022.
- [48] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7959–7971, 2022.
- [49] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.
- [50] 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.
- [51] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019.
- [52] 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.
- [53] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–308, 2009.
- [54] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.
- [55] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023.
- [56] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021.
- [57] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.
- [58] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2935–2944, 2023.
- [59] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.
- [60] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014.

- [61] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [62] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [63] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
- [64] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019.
- [65] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [66] Sehban Omer. fast-pytorch-kmeans, September 2020.
- [67] Terrance DeVries. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.
- [68] Lihe Yang, Wei Zhuo, Lei Qi, Yinghuan Shi, and Yang Gao. St++: Make self-training work better for semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4268–4277, 2022.
- [69] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. *arXiv preprint arXiv:2303.15343*, 2023.

Appendix

A Further Implementation Details

We set $\gamma = 0.999$, input resolution as $H = W = 640$, which results in $h = w = 20$, and set $h' = w' = 80$ for ConvNeXt [49] backbones. For ViT [62] backbones, we set $H = W = 320$, which also results in $h = w = 20$. For global clustering, we set $\varepsilon = 1$ for ConvNeXt backbones and $\varepsilon = 0.01$ for ViT backbones. We implement our work using PyTorch [63] and Detectron2 [64]. AdamW [65] optimizer is used with a learning rate of $2 \cdot 10^{-4}$ for the decoder, $2 \cdot 10^{-5}$ for the prompt tokens and $2 \cdot 10^{-6}$ for CLIP, with weight decay set to 10^{-4} . Prompt tokens are initialized as random word tokens with $l = 4$, and $k = 64$ as default. We use GPU implementation [66] of k -means clustering for our experiments with DINO masks. For \mathcal{E}_V , we apply CutOut [67] and color augmentations [68] during training. For the prompt ensemble strategy during inference, we use the prompts curated originally from CLIP [15] in their repository, which results in total of 8 text prompt as follows:

“itap of a {}.”,
“a bad photo of the {}.”,
“a origami {}.”,
“a photo of the large {}.”,
“a {} in a video game.”,
“art of the {}.”,
“a photo of the small {}.”,
“a photo of a {} in the scene”.

B Additional Experiments

Table 4: **Quantitative results with various backbones.** We show results on open-vocabulary semantic segmentation of various CLIP backbones with the addition of ViT-B from SigLIP [69] and ConvNeXt-L.

Method	Backbone	Evaluation Dataset				
		COCO-St.	ADE-150	Context	CityScapes	VOC
SigLIP [69]	ViT-B/16 [62]	12.4	11.8	18.3	19.2	46.8
PixelCLIP (Ours)	ViT-B/16 [62]	20.0 (+7.6)	19.2 (+7.4)	33.1 (+14.8)	31.6 (+12.4)	72.3 (+25.5)
CLIP [26]	ViT-B/16 [62]	16.5	13.2	25.6	14.9	73.9
PixelCLIP (Ours)	ViT-B/16 [62]	21.4 (+4.9)	16.7 (+3.5)	34.9 (+9.3)	23.8 (+8.9)	83.1 (+9.2)
OpenCLIP [55]	ConvNeXt-B [49]	12.8	13.1	16.5	16.2	34.8
PixelCLIP (Ours)	ConvNeXt-B [49]	21.1 (+8.3)	20.2 (+7.1)	34.2 (+17.7)	33.2 (+17.0)	66.0 (+31.2)
OpenCLIP [55]	ConvNeXt-L [49]	16.9	15.2	22.9	17.1	57.2
PixelCLIP (Ours)	ConvNeXt-L [49]	24.8 (+7.9)	22.6 (+7.4)	39.4 (+16.5)	34.3 (+17.2)	78.9 (+21.7)

B.1 Results on Different Backbones

In Tab. 4, we show results for PixelCLIP when applied to different backbones. We note that since the ViT backbone has a larger output feature resolution scale compared to ConvNeXt models, we set the input image resolution to match the output feature resolution, and report results without prompt ensembling. In general, we observe noticeable gains across all backbones, with CLIP ViT-B/16 outperforming ConvNeXt-B on several datasets. Through testing with various pre-trained CLIP models, we demonstrate that our method can effectively fine-tune CLIP for dense prediction regardless of the backbone architecture.

Table 5: **Additional experiments on prompt ensembling.** We show results on open-vocabulary semantic segmentation with prompt ensembling being used during only training, only inference, or both. The default setting of prompt ensembling only being used during inference is highlighted in gray.

Prompt Ensembling		Evaluation Dataset				
Training	Inference	COCO-St.	ADE-150	Context	CityScapes	VOC
		21.4	16.7	34.9	23.8	83.1
	✓	<u>23.6 (+2.2)</u>	<u>18.7 (+2.0)</u>	37.9 (+3.0)	<u>27.2 (+3.4)</u>	85.9 (+2.8)
✓		21.6 (+0.2)	17.1 (+0.4)	35.1 (+0.2)	24.9 (+1.1)	82.9 (-0.2)
✓	✓	23.7 (+2.3)	19.2 (+2.5)	37.9 (+3.0)	28.1 (+4.3)	<u>85.5 (+2.4)</u>

B.2 Analysis on Prompt Ensembling

In Tab. 5, we show results with prompt ensembling being applied during only training, only inference, and both. We report results with ViT-B/16 using SA-1B masks as supervision. Although prompt ensembling does bring slight gains when enabled during training, the computation for optimizing learnable class prompts scales along with the number of prompts used, increasing the training time and the memory consumption. On the other hand, applying prompt ensembling during inference only adds negligible cost as they can be computed once and be cached, but shows much significant gains compared to when applied training. Therefore, we adopt prompt ensembling only during inference, but noticing that the performance can be maximized with better prompts during training. In this regard we can better results with better prompt design or a learnable prefix to accompany the learnable class prompts, which we leave for future investigation.

B.3 Additional Ablation Studies

Table 6: **Additional ablation studies.** We show results on open-vocabulary semantic segmentation with a larger number of clusters and different training datasets.

γ	Evaluation Dataset				
	COCO	ADE-150	Context	CityScapes	VOC
Baseline [55]	12.8	13.1	16.5	16.2	34.8
0.99	19.9	19.5	<u>32.5</u>	29.5	62.6
0.999	21.1	20.2	34.2	33.2	66.0
0.9999	<u>20.4</u>	<u>19.7</u>	32.0	<u>29.9</u>	<u>63.0</u>

Dataset	Evaluation Dataset				
	COCO	ADE-150	Context	CityScapes	VOC
Baseline [55]	12.8	13.1	16.5	16.2	34.8
COCO-St. [50]	24.1	21.9	36.8	30.2	71.0
SA-1B [13]	21.1	20.2	34.2	33.2	66.0

(a) **Varying momentum γ .** We show additional results for varying γ for the momentum update.

(b) **Different training dataset.** We show results for leveraging ground-truth masks from COCO-Stuff while removing its class labels.

B.3.1 Ablation on the momentum update rate γ

In Tab. 6 (a), we show results for varying γ for the momentum update. While having the momentum encoder generally shows improvements, we find $\gamma = 0.999$ to show the best results for updating the momentum encoder.

B.3.2 Ablation on training dataset

In Tab. 6 (b), we show results for training with mask annotation from COCO-Stuff [50]. For COCO-Stuff, we remove the ground truth class labels and utilize them as unlabeled masks, and other hyperparameters are set identically with $k = 64$. Although the masks from COCO-Stuff show better results across all datasets, we highlight that the SA-1B [13] dataset mostly consists of automatically generated masks from SAM, whereas COCO-Stuff has human annotated masks from expert annotators.



Figure 7: **Visualization on COCO-Stuff with learned class prompts.** We provide results with learned classes with different k up to 256.



Figure 8: **Visualization on ADE-20k** We provide qualitative comparison on ADE-20K [51].

C Additional Qualitative Results

We provide qualitative results of visualization on ADE-20K [51], PASCAL-Context [52] in Fig. 8 and Fig. 9.

D Additional Visualization

In Fig. 7, we show visualization on COCO-Stuff by classifying the image features with our learned class prompts for varying k . From the first and the second row, we can observe that with larger numbers of k , different parts of human are segmented into fine-grained regions whereas $k = 64$ has



Figure 9: **Visualization on PASCAL-Context.** We provide qualitative comparison on PASCAL-Context [52].

more coarse regions. Especially for $k = 256$, in the second row, we observe the glasses, hair, and hands all classified into different classes with our learned prompt.

On the other hand, we also observe cases where a small number of k struggles to differentiate visual concepts in the last row, where the animals are partially grouped with $k = 64$ and show better groups for $k = 128$ and $k = 256$. This could indicate that with only a small number of clusters, several fine-grained visual concepts that may not be seen often in the dataset to be grouped as a whole, whereas independent clusters could be assigned with a larger number of k , allowing fine-grained recognition of semantics.

E Limitations

Although we aim to fine-tune the image encoder of CLIP for adapting to dense predictions, we initialize the mask encoder within our framework with pre-trained weights of CLIP, which yields poor results for classifying masks when applying mask pooling to its features. Consequently, the noisy mask features in the earlier stage of training may result in sub-optimal performance. While there could be alternative methods to extract per-mask CLIP image features, we consider mask pooling to be sufficient to show meaningful improvements to CLIP and consider such exploration for future directions.

F Broader Impact

Our framework facilitates open-vocabulary semantic segmentation through leveraging vision-language models, hence the recognition capabilities of our method rely on the pre-trained knowledge of the vision-language models. Considering that large-scale pre-trained vision-language models [15, 16, 55] leverage web-crawled data within its training, the models may exhibit wrongful behaviors from bias or corrupted data from the internet which calls for future research to address.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [NA].
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist",**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: We verify our claims in L54-L59 experimentally in Section 4.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We discuss limitations in Section E.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: We do not involve theory assumption and proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide exhaustive details in Section 4.1 in the main paper, as well as further details in Section A in the supplementary materials.

Guidelines:

- The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We provide exhaustive details in Section 4.1 in the main paper, as well as further details in Section A in the supplementary materials for reproducing the experimental results. We are committed to releasing our code upon acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide exhaustive details in Section 4.1 in the main paper, as well as further details in Section A in the supplementary materials.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We do not report error bars, but we fix the random seed to minimize the stochasticity for all of our experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We report the resources and training time of our method in Section 4.1.

Guidelines:

- The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines?>

Answer: [Yes]

Justification: Anonymity is kept as shown in the first page.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss broader impacts in Section F in the supplementary material.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have used only publicly available datasets. We have cited the original authors and respected the respective licenses.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New Assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We do not introduce new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and Research with Human Subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: We do not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: We do not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.