CLIP-DINOiser: Teaching CLIP a few DINO tricks for open-vocabulary semantic segmentation

Monika Wysoczańska[1], Oriane Siméoni[2], Michaël Ramamonjisoa[3]*,
Andrei Bursuc[2], Tomasz Trzciński[1,4,5], and Patrick Pérez[2]

[1]Warsaw University of Technology, [2]valeo.ai, [3]Meta AI, [4]Tooploox, [5]IDEAS NCBR

Abstract. The popular CLIP model displays impressive zero-shot capabilities thanks to its seamless interaction with arbitrary text prompts. However, its lack of spatial awareness makes it unsuitable for dense computer vision tasks, e.g., semantic segmentation, without an additional fine-tuning step that often uses annotations and can potentially suppress its original open-vocabulary properties. Meanwhile, self-supervised representation methods have demonstrated good localization properties without human-made annotations nor explicit supervision. In this work, we take the best of both worlds and propose an open-vocabulary semantic segmentation method, which *does not require any annotations*. We propose to locally improve dense MaskCLIP features, which are computed with a simple modification of CLIP's last pooling layer, by integrating localization priors extracted from self-supervised features. By doing so, we greatly improve the performance of MaskCLIP and produce smooth outputs. Moreover, we show that the used self-supervised feature properties can directly be learnt from CLIP features. Our method CLIP-DINOiser needs only a single forward pass of CLIP and two light convolutional layers at inference, *no extra supervision nor extra memory* and reaches state-of-the-art results on challenging and fine-grained benchmarks such as COCO, Pascal Context, Cityscapes and ADE20k. The code to reproduce our results is available at https://github.com/wysoczanska/clip_dinoiser.

Keywords: open-vocabulary semantic segmentation · self-supervised features · annotation-free segmentation

1 Introduction

Semantic segmentation is a key visual perception task for many real-world systems, e.g., self-driving cars, and industrial robots. Typically tackled in a dataset-oriented manner, best methods require a training dataset which is manually annotated for a *specific and finite* set of classes. The advent of powerful Vision-Language Models (VLM) [24,43,63] is stimulating a shift from a closed-vocabulary

[*] Work done outside of Meta and Meta was not involved in the research discussed here.

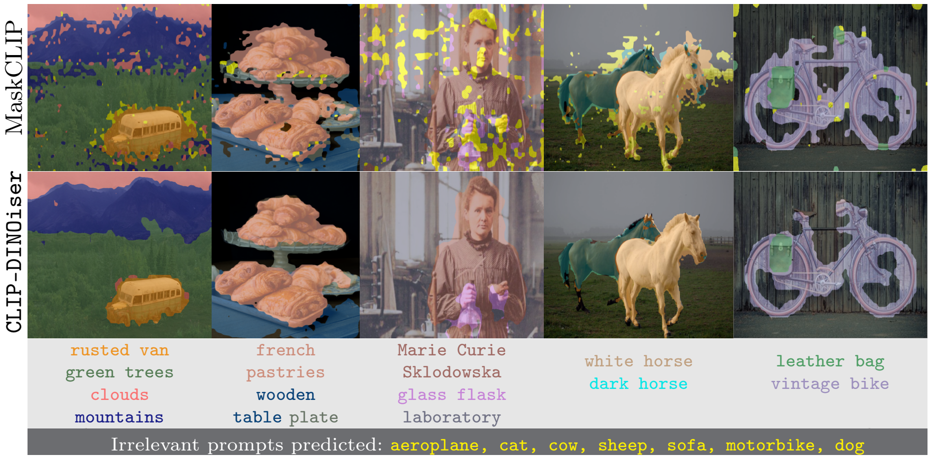


Fig. 1: Examples of open-vocabulary semantic segmentation results obtained with our method `CLIP-DINOiser` on 'in-the-wild' images vs. those of MaskCLIP [67]. Our method improves MaskCLIP features with a smart pooling strategy which does *not alter the original* open-vocabulary properties. We use self-supervised DINO [5] as a guide to *teach CLIP* [22] to produce DINO-like localization features through two light convolutional layers. Our method, which achieves state-of-the-art results, only requires a *single forward* pass through CLIP model and our two layers. In addition to the correct prompts (light grey row) we list the irrelevant prompts predicted (in yellow) that we query in all images shown here.

paradigm to an *open-world* one. Such models are trained with a simple but scalable objective: to align pairs of image and coarse text captions that can be obtained in large amounts with limited manual supervision. VLMs excel at associating *global* image content with arbitrary text inputs with remarkable generalization capabilities [17, 34], but struggle to provide dense *open-vocabulary features* [18, 67]. Obtaining such an alignment between pixels and language can lead to open-vocabulary extensions for multiple other modalities, such as point clouds [8, 23, 38, 41], 3D scenes [54], 3D shapes [1], radiance fields [27], inter-modality alignment [19, 23], with multiple potential applications for which the construction of training datasets is even more challenging and where CLIP-derived models show promising results.

Different strategies have been recently proposed towards improving CLIP's patch-level feature extraction abilities by modifying the original CLIP architecture for dense pooling and retraining [6, 37, 44, 61, 62] or finetuning on an annotated segmentation dataset with pre-defined classes [32, 67]. The former requires long training and/or large collections of annotated data, while the latter leads to an alteration of the vision-language associations of the CLIP features. An alternative line of approaches freezes the CLIP encoder and directly densifies its features with different heuristics, often with multiple forward passes [1, 23, 27, 49, 50, 59], but are less practical due to the extensive computational overhead. MaskCLIP [67] arises as a computationally efficient dense CLIP extractor. It converts CLIP's global self-attention layer into a convolutional one to produce patch features with original vision-language qualities. If such features are local, they appear to be too noisy for high-quality segmentation mask extraction (see Fig. 2b middle column).

Meanwhile, recent self-supervised learning (SSL) approaches [4, 5, 9, 68] produce strong visual representations displaying object localization properties, and such without requiring any manual annotation. DINO [5] stands out with its semantically meaningful features which have been exploited for unsupervised object discovery [51, 52, 56, 57]. DINO features prove useful also for zero-shot semantic segmentation [25, 27, 59], but require expensive sliding window sampling [27,59] or building concept-specific prototypes and ensemble strategies [25].

In this work, we aim for unaltered patch-level CLIP features with minimal runtime overhead. To this end, we re-examine the localization properties of MaskCLIP features and observe that it is possible to easily refine them with guidance from SSL models. In detail, we train a simple convolutional layer on unlabeled data to produce pooling weights to perform correlation-guided dense feature pooling from CLIP without distorting the vision-language alignment. This layer is optimized to mimic the patch correlations of DINO [5] that indicate likely layouts of visual concepts in the images. Furthermore, we show that the unsupervised objectness information given by FOUND [52] from DINO features can be also directly learned from CLIP features again in a fully-unsupervised fashion with a single convolutional layer and helps improve the segmentation of the ill-defined 'background' prompt. With `CLIP-DINOiser`, we obtain high-quality masks in *a single forward pass* on CLIP (see Fig. 1). `CLIP-DINOiser` is amenable to producing dense semantic maps.

To summarize, our contributions are: **(1)** We propose a light pooling mechanism to refine MaskCLIP features by leveraging guidance from SSL features without degrading its original open-vocabulary properties. `CLIP-DINOiser` does not require any annotations, nor retraining CLIP from scratch, but only a single CLIP forward pass. **(2)** We show that CLIP *already contains good localization properties* which can be exploited. We leverage simple convolutional layers to emphasize visual concept layouts from dense CLIP features. We train them without any annotation on only 1k of raw images randomly sampled in ImageNet [13]. We believe that this finding could be further exploited in different contexts. **(3)** Our method achieves state-of-the-art results on complex semantic segmentation datasets such as COCO [3], Pascal Context [16], Cityscapes [11] and ADE20K [66].

2 Related Work

Zero-shot semantic segmentation. This task has been typically approached by methods which aim at generalizing from *seen* classes to *unseen* ones [2, 20, 21, 26, 30, 40, 60, 64]. Such strategies train models with full supervision on the set of seen classes and propose different solutions to extend them to unseen ones without new images (labeled or unlabeled), e.g., by exploiting class information and relationships encapsulated in popular word embeddings [35,42]. While they produce fine segmentations without computational overhead, these methods require pixel-level annotations for the seen classes.

From CLIP to open-vocabulary segmentation. The surge of VLMs with aligned image-language representations [22, 24, 43] brought back into the spotlight the zero-shot classification task. However, the extension to zero-shot segmentation is not obvious as the CLIP architecture is not equipped to yield dense vision-language features [18, 67]. To produce dense CLIP features, several approaches fine-tune or train from scratch pixel-aligned CLIP-like models with additional modules, mechanisms or supervision objectives [6,37,44,61,62] on datasets with annotations of varying granularity and quality: dense annotations [29,31], class-agnostic object masks [14, 18, 45], coarse captions [6, 18, 31–33, 37, 44, 61, 62, 65] or pseudo-labels [67]. Recent works leverage image-level captions to align text to regions (obtained without supervision): PACL [37] trains an embedder module to learn patch-to-text affinity, TCL [6] proposes a local contrastive objective to align well-selected patches to the text and ViewCO [46] leverages multi-view consistency. On the downside, such models require long training on millions of images or specific types of very costly annotations. Also, fine-tuning CLIP with a defined vocabulary is more computationally appealing [29, 31, 67], but alters the open-vocabulary properties of the features [23].

Most related to us is a line of works that investigate how to directly densify CLIP features [1, 23, 27, 59, 67] to obtain per-patch CLIP features. Such densification can be performed by aggregating features from multiple views [1, 27] or from sliding windows [23, 59] at the extra-cost of multiple forward passes. MaskCLIP [67] drops the global pooling layer of CLIP and matches the projected features directly to text via a 1×1 convolution layer. By doing so they achieve dense predictions, however noisy.

With a concept-driven perspective, some methods [25,49,50] build codebooks of visual prototypes per concept, including negative prototypes [25], and then perform co-segmentation [49]. While such an approach yields good results, it is however at the cost of building expensive *class-specific prototypes*, therefore diverging from open-vocabulary scenarios. Instead, we aim to remain *open* to avoid retraining a model or building new expensive prototypes whenever a new concept is considered. To that end, we devise a dense CLIP-feature extraction method that preserves the open-vocabulary quality.

Leveraging self-supervised models & CLIP. Recent self-supervised ViTs [4,5,9,12, 68] have demonstrated features with good localization properties [51, 52, 56, 57]. Such features have also been exploited in the context of open-vocabulary segmentation methods, e.g. for pre-training for the visual backbone [7, 44, 62], co-segmentation [49], clustering patches into masks [47], representing object prototypes [25]. Related to us is the recent CLIP-DIY [59] which computes patch-level representations from CLIP features from different image crops with guidance from an unsupervised saliency segmenter [52] FOUND. While we also leverage the latter, in contrast with CLIP-DIY which runs multiple forward passes to build their dense CLIP features, our method requires only a *single forward pass* of CLIP. Furthermore, our method mitigates the limits of FOUND in cluttered scenarios by integrating an uncertainty constraint. Finally, we leverage the informative patch correlation properties of DINO [5] and show that it is possible to *teach CLIP* to produce DINO-like features through light convolutional layers.

3 Method

We present in this section `CLIP-DINOiser`, a simple and efficient strategy to improve MaskCLIP using localization information extracted from CLIP—with a lightweight model trained to mimic some of DINO's properties. We first set the goal in Sec. 3.1 and present MaskCLIP [67] in Sec. 3.2. We then introduce our strategy which leverages self-supervised features localization information to consolidate MaskCLIP features in Sec. 3.3 and discuss how such localization information can directly be learnt from CLIP in Sec. 3.4 (we visualize both steps in Fig. 3). We also propose a way to improve the 'background' filtering in Sec. 3.5.

3.1 Problem statement

In this work, we aim to produce open-vocabulary[1] semantic segmentation of an image. We consider an image $X \in \mathbb{R}^{H \times W \times 3}$ which we split into a sequence of N patches of dimensions $P \times P \times 3$ with $P \times P$ the patch size and $N = \lceil \frac{H}{P} \rceil \cdot \lceil \frac{W}{P} \rceil$. A class token, noted `CLS`, is added to the input sequence and we feed the $N+1$ patches to a ViT [15] model. We aim at producing dense visual features $F \in \mathbb{R}^{N \times d}$, with d the feature dimension, that can later be matched to *any* set of text inputs embedded in the same space. In particular, the goal is to produce a segmentation map per textual query.

3.2 Preliminaries on MaskCLIP

Extracting dense open-vocabulary features. The popular CLIP [22] model pre-trained on image/caption pairs produces good *global* image features, but was not trained to generate high-quality 2D feature maps. In order to extract such dense feature maps relevant to semantic segmentation, Zhou et al. [67] revisit the global attention pooling layer of the last attention layer of the model. The authors discard the *query* and *key* embeddings of the layer and transform both the *value* projection and the last linear layer into a conv 1×1 layer. With this new model, named MaskCLIP and denoted $\phi(\cdot)$, we extract d-dimensional features $\phi^L(X) \in \mathbb{R}^{N \times d}$ from the last layer L which retains most of the open-vocabulary properties of CLIP [67].

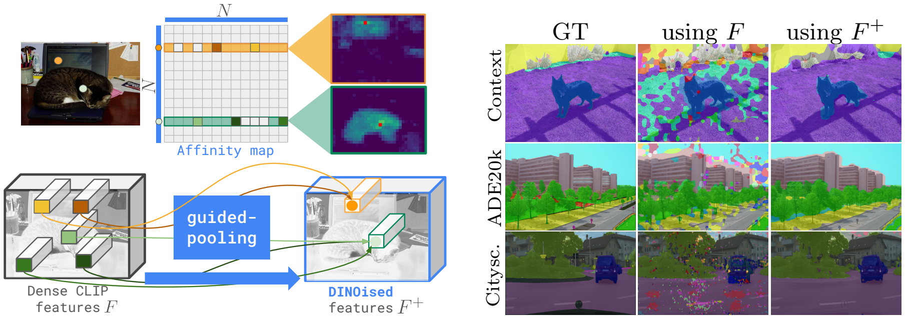
[1] We adopt the taxonomy defined in the recent survey [58] and define our method as 'open-vocabulary', with capabilities to generalize to unseen datasets.

Semantic segmentation given textual queries. We also extract CLIP textual features $\phi_T(t_j)$ for each text query $t_j \in \mathcal{T}$ with $j \in \{1, \ldots, |\mathcal{T}|\}$. Segmentation maps are then generated by computing the cosine similarity between each of the visual patch features and of the textual prompts, after L2-normalization. The most similar prompt is assigned to each patch. Note that a query 'background' can be added in order to obtain *negative* patches. Using MaskCLIP allows us to produce dense segmentation maps with a single forward pass of the classic CLIP model, but its outputs are noisy, as visible in Fig. 2b (middle column).

3.3 DINOising open-vocabulary features

In this work, we aim to improve MaskCLIP's open-vocabulary features described above. To do so, we propose to leverage the known good localization properties of self-supervised features [5, 39, 51–53, 57] .

Extracting self-supervised correlation information. Recent works [51, 57] have shown that the patch correlation information of the embeddings from the last attention layer of the self-supervised model, DINO [5] can help highlight objects in images. We use here the *value* embeddings which we observe have finer correlation than those of key and query (more discussion in supplementary material). We extract such self-supervised features $\xi(X) \in \mathbb{R}^{N \times d_\xi}$ and discard the `CLS` token. We then compute the per-patch cosine-similarity and produce the affinity map $A^\xi \in [-1, 1]^{N \times N}$. We compare in Fig. 4 the patch-similarities obtained for a patch *seed* with MaskCLIP and DINO features and observe that the self-supervised features are more densely and accurately correlated than those of CLIP.



(a) Our guided pooling **(b)** Impact of the pooling

Fig. 2: We present in (a) is our *guided pooling* strategy defined in Eq. (1). The $N \times N$ affinity matrix is computed from patch features and is used to refine MaskCLIP features (bottom left). In (b) we compare our results with F^+ (right) versus those obtained with MaskCLIP features (middle).

Strengthening features with guided pooling. In order to locally consolidate MaskCLIP features $\phi^L(X)$, now noted F, we propose to perform a *concept-aware* linear combination of the features per patch with guidance from the patch affinity A^ξ. The feature combination strategy can be seen as a form of voting mechanism that enforces similar patches to have similar CLIP features (and prediction) while attenuating noisy features. Specifically, we compute the new features $F^{+} \in \mathbb{R}^{N \times d}$ as an average of MaskCLIP features F weighted by A^{ξ}, presented in Fig. 2a. We zero-out A^{ξ} correlations below a threshold γ, following [51,57], and compute the new features for patch $p \in \{1, \ldots, N\}$:

$$F_p^{+} = \frac{1}{\sum_{q=1}^{N} A_{p,q}^{\xi}} \sum_{q=1}^{N} A_{p,q}^{\xi} \cdot F_q. \tag{1}$$

We then produce the segmentation maps $S \in [-1, 1]^{N \times |\mathcal{T}|}$, by comparing the new features F^{+} to each textual queries in $\mathcal{T}$. As shown in Fig. 2b, when using such consolidated features, we obtain more accurate outputs and the high-frequency predictions observed in MaskCLIP are smoothed out, showing the benefit of the pooling.

3.4 Teaching CLIP a first DINO trick: object correlations

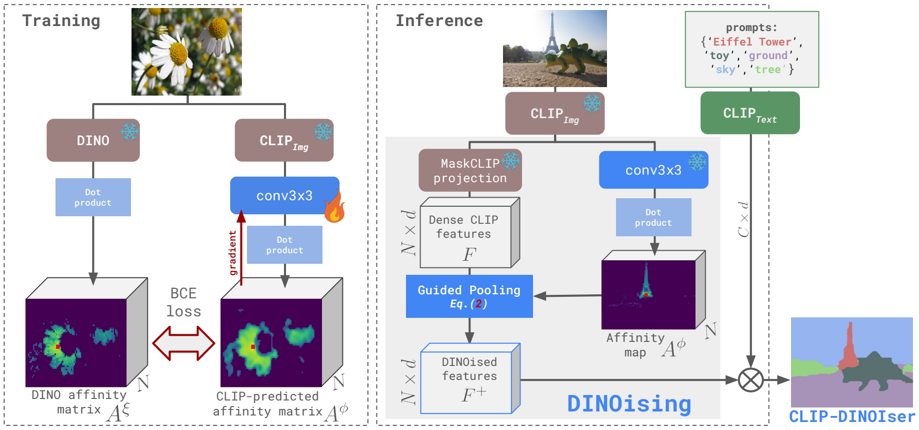


Fig. 3: Overview of CLIP-DINOiser which leverages the quality of self-supervised features to improve the notoriously noisy MaskCLIP feature maps. We use DINO as a teacher which 'teaches' CLIP how to extract localization information. We train (left) a conv3 × 3 layer to reproduce the patch correlations obtained with DINO. At inference (right), an input image is forwarded through the frozen CLIP image backbone and MaskCLIP projection. The produced features are then improved with our *pooling* strategy which is guided by correlations predicted with the trained convolutional layer applied on CLIP. With this light 'DINOising' process, we obtain 'DINOised' features which are matched against the prompts features to produce CLIP-DINOiser outputs.

We have shown in the previous section that self-supervised correlation information can successfully be used to improve the dense quality of open-vocabulary features. If the difficulty of densifying CLIP is well-known, we show here that CLIP features already contain *good localization information* which can be extracted with a light model. We indeed predict DINO correlations A^{ξ} from CLIP with a single convolutional layer.

In order to predict the DINO affinity map A^{ξ} from CLIP features, we train a *single* 3×3 *convolutional layer* $g(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^{d_g}$ which projects intermediate features $\phi^l(X)$–extracted from layer l–into a smaller space of dimension $d_g < d$. We enforce the patch correlations of the generated features $A^{\phi} \in [-1, 1]^{N \times N}$:

$$A^{\phi} = \frac{g(\phi^l(X))}{\|g(\phi^l(X))\|} \otimes \left(\frac{g(\phi^l(X))}{\|g(\phi^l(X))\|} \right)^{\top}, \tag{2}$$

with $\otimes$ denoting the outer product, to be close to the binarized correlations $D = A^{\xi} > \gamma$ (we use here the same γ as defined above), using the binary cross-entropy loss $\mathcal{L}^c$:

$$\mathcal{L}^c = \sum_{p=1}^{N} \left[D_p \log A_p^{\phi} + (1 - D_p) \log(1 - A_p^{\phi}) \right]. \tag{3}$$

We present our layer training in Fig. 3 (left part) and observe the quality of CLIP-predicted affinity matrix A^{ϕ}. We also show in Fig. 4 another example of obtained A^{ϕ} and observe their similarity to DINO-based correlations. We use the CLIP-produced correlations A^{ϕ} to replace A^{ξ} in Eq. (1) to weight the pooling and observe a similar boost over MaskCLIP, thus showing that good patch correlations can indeed be extracted directly from CLIP. We can now discard DINO and we name `CLIP-DINOiser` the guided-pooling strategy which uses CLIP-based correlation. As shown in Fig. 3 (*inference* step), our method runs with a single forward pass of CLIP model and a small extra layer.

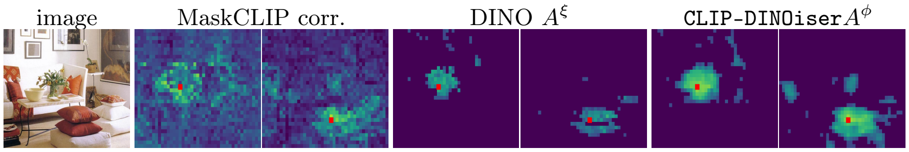
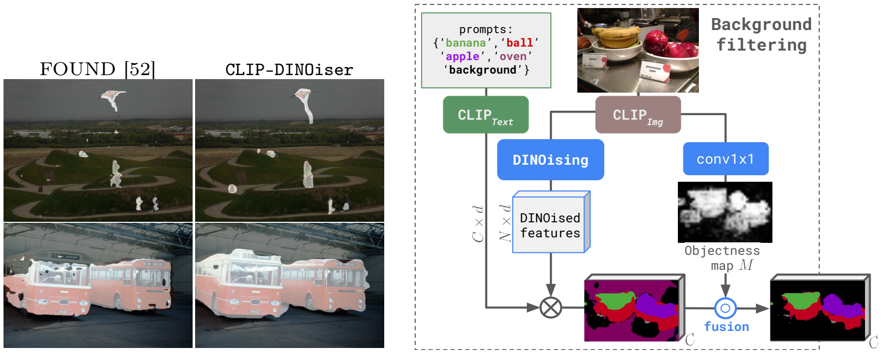


Fig. 4: Comparison of the affinity maps between a *seed* (one on the 'plant' and the other on a 'pillow') and the other patch features when using features of MaskCLIP, DINO and ours after training.

3.5 Teaching CLIP a second DINO trick: background filtering

Moreover, as discussed earlier, a 'background' query may be added to the set of textual queries $\mathcal{T}$ in order to help filter out patches falling in the *background* and not corresponding to any objects. We do not assume here any prior knowledge about classes of interest and focus rather on the foreground/background paradigm [52]. We argue that relying solely on the textual prompt 'background'



(a) Comparison of background filtering **(b)** Background filtering

Fig. 5: We present in (a) a comparison of *objectness* mask generated by FOUND [52] and with our layer using CLIP features. We carefully define the fusion operation and the simple training strategy of the conv1 × 1 again using DINO as a teacher in Sec. 3.5. In (b) is an overview of our *background filtering* which is applied when a 'background' prompt is provided and helps reduce hallucinations.

to catch all non-salient patches is underperforming and, similarly to [59], we propose to use a very light-weight *unsupervised* foreground/background segmentation method, namely FOUND [52] which also relies on DINO self-supervised features. We run FOUND on the entire image and extract a prediction mask $M \in \{0,1\}^N$ in which a patch is assigned the value 1 if falling into the foreground and 0 otherwise. We also observe that saliencies produced by FOUND can be too restrictive and discard objects which are partially visible or in a clutter. To mitigate this behaviour, we propose to relax the background selection by integrating an additional uncertainty constraint. To this end, we fuse the background information from both modalities by assigning the 'background' prompt to patches p which are both *uncertain*, e.g. have low confidence score $\sigma(S)_p < \delta$, with $\sigma(\cdot)$ the softmax operation, *and* which fall in the background in M.

Learning FOUND objectness. Moreover, we are also able to learn the predictions of FOUND [52] directly from CLIP features. To do so, we train a *single* 1 × 1 convolutional layer $h(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}$ which predicts from the features $\phi^l(X)$ an objectness map $M^\phi = h(\phi^l(X)) \in \mathbb{R}^N$. We train the model to predict the FOUND binary mask M with the binary cross-entropy loss $\mathcal{L}^m$:

$$\mathcal{L}^m = \sum_{p=1}^{N} \left[M_p \log(M_p^\phi) + (1 - M_p) \log(1 - M_p^\phi) \right].$$

We show examples of predicted CLIP-based objectness in Fig. 5a and observe their very high similarity to those produced with DINO. Moreover, we can now replace M defined above with the binarized CLIP-based scores $\zeta(M^\phi) > 0.5$, with $\zeta(\cdot)$ the sigmoid operation, and observe a minimal drop in performances. We show an example of the background filtering with trained objectness in Fig. 5b.

4 Experiments

We detail in Sec. 4.1 the experimental setup used in our evaluation. We produce state-of-the-art results on the task of open-vocabulary semantic segmentation in Sec. 4.2 and ablation studies in Sec. 4.3.

4.1 Experimental setup

Technical details. We use in all experiments a *frozen* CLIP ViT-B/16 pre-trained following OpenCLIP [22]. Our method `CLIP-DINOiser` uses two convolutional layers to extract DINO-like information from CLIP layer $l = 10$ (the 3rd before the last which was shown to provide the best results [55]). The first layer $g(\cdot)$ has a kernel 3×3 and output dimension $d_g = 256$ and $h(\cdot)$ a kernel 1×1 with $d_h = 1$. The first is trained to match the correlation information extracted from the *value* embeddings of the last layer of a ViT-B/16 model trained following DINO [5]. The second layer is trained to replicate the unsupervised object localization predictions of FOUND [52]–which also uses DINO model. We train both layers with a binary cross-entropy loss on *only 1k raw images* randomly sampled from ImageNet [13] dataset *without any annotation.* We report average scores over 3 runs with different sampling seeds and provide standard deviations in the supplementary material. We follow [57] and binarize the correlations with $\gamma = 0.2$. In the background filtering step, we use a high confidence score, i.e., $\delta = 0.99$. We train our model for 6k iterations with a batch size of 16 images using Adam optimizer [28], which takes approximately 3 hours on a single NVIDIA RTX A5000 GPU. We decrease the learning rate for both heads by a factor of 0.1 after 5k iterations. We apply data augmentations during training (random scale and cropping, flipping and photometric distortions).

Datasets and metric. We evaluate our method on eight benchmarks typically used for zero-shot semantic segmentation [6]. Following [6], we split them into two groups. The first consists in datasets with a 'background' query: PASCAL VOC [16] (noted 'VOC'), PASCAL Context [36] (noted 'Context'), and COCO Object [3] (noted 'Object') and the second without: PASCAL VOC20 [16] (noted 'VOC20'), PASCAL Context59 [36] (noted 'C59'), COCO-Stuff [3] (noted 'Stuff'), Cityscapes [11] (noted 'City'), and ADE20K [66] (noted 'ADE'). We evaluate results with the standard mIoU metric. We also follow the evaluation protocol of [6], use the implementations provided by MMSegmentation [10], employ a sliding window strategy, resize the input image to have a shorter side of 448. We also do not perform text expansions of the class names and use only the standard ImageNet prompts following [22, 61, 67].

Baselines. We compare our method against state-of-the-art methods on open-vocabulary zero-shot semantic segmentation. For a fair comparison between methods, we report results without any post-processing step. In our evaluations, we follow the taxonomy presented in [58] and compare our model with the methods relying on language-image pretraining, also called open-vocabulary.

We split the compared baselines into four categories: (1) *dataset specific* which employ pseudo-labeling and supervised training of a segmentation model on target dataset: NamedMask [50], MaskCLIP+ [67]); (2) *construct prototypes*: ReCO [49], OVDiff [25]; (3) *train with text supervision* including GroupViT [61], ZeroSeg [47], SegCLIP [33], TCL [6], CLIPpy [44], OVSegmentor [62], which all require access to additional datasets of millions of image/caption pairs (we note in the table the exact datasets used for the training); and finally *use frozen CLIP* i.e. CLIP-DIY [59] and MaskCLIP [67], which use pre-trained CLIP. Our method falls into the last category as we do not modify CLIP, and do not need access to additional caption annotations as we use only 1k unannotated images.

	Concept		**Extra**	**Inference**	**No background prompt**					**W/ bkg prompt**		
Methods	**spec.**	❄	**data**	**backbone**	VOC20	C59	Stuff	City	ADE	Cont.	Object	VOC
Dataset specific												
MaskCLIP+ [67]	✓	✗	I	DLv2	-	31.1	18.0	-	-	-	-	-
NamedMask [50]	✓	✗	I	DLv3+	-	-	-	-	-	-	27.7	59.2
Build prototypes per visual concept												
ReCo [49]	✓	✓	I	CLIP	57.8	22.3	14.8	21.1	11.2	19.9	15.7	25.1
OVDiff [25]	✓	✓	✗	CLIP+ DINO+SD	**81.7**	33.7	-	-	14.9	30.1	**34.8**	**67.1**
Text/image alignment training with captions												
GroupViT [61] [6]	✗	✗	IT	CLIP	79.7	23.4	15.3	11.1	9.2	18.7	27.5	50.4
ZeroSeg [7]	✗	✗	IT	CLIP	-	-	-	-	-	21.8	22.1	42.9
SegCLIP [33]	✗	✗	IT	CLIP	-	-	-	11.0	8.7	24.7	26.5	52.6
TCL [6]	✗	✗	IT	CLIP	77.5	30.3	19.6	23.1	14.9	24.3	30.4	51.2
CLIPpy [44]	✗	✗	IT	CLIP	-	-	-	-	13.5	-	32.0	52.2
OVSegmentor [62]	✗	✗	IT	CLIP	-	-	-	-	5.6	20.4	25.1	53.8
Frozen CLIP												
CLIP-DIY [59]*	✗	✓	✗	CLIP+ DINO	79.7	19.8	13.3	11.6	9.9	19.7	31.0	59.9
MaskCLIP [67] [6]	✗	✓	✗	CLIP	53.7	23.3	14.7	21.6	10.8	21.1	15.5	29.3
MaskCLIP*	✗	✓	✗	CLIP	61.8	25.6	17.6	25.0	14.3	22.9	16.4	32.9
MaskCLIP* †	✗	✓	✗	CLIP	71.9	27.4	18.6	23.0	14.9	24.0	21.6	41.3
CLIP-DINOiser	✗	✓	I(1k)	CLIP	80.9	**35.9**	**24.6**	**31.7**	**20.0**	**32.4**	**34.8**	62.1

Table 1: Open-vocabulary semantic segmentation quantitative comparison using the mIoU metric. We separate in two columns the evaluation datasets: those without a 'background' prompt and those with (noted 'W/ bkg prompt'), as discussed in Sec. 4.1. We report all methods without post-processing. We note with * methods for which we computed scores; we obtained MaskCLIP* scores with OpenCLIP [22] and mark with † the use of MaskCLIP refinement. The first and second best methods are respectively **bold** and underlined. We specify if a method assumes prior access to names of concepts ('Concept spec.') and if it employs a frozen backbone (❄). We specify what additional data is used at training ('Extra data') ('I' stands for images and 'IT' for image/text aligned data). Our CLIP-DINOiser only needs 1k images from ImageNet to be trained. 'SD' stands for Stable Diffusion [48]. We refer to Sec. 4.1 for more details on baselines and we detail the datasets used for training by each method in the supplementary material.

4.2 Open-vocabulary semantic segmentation

We discuss in this section state-of-the-art results on the task of open-vocabulary semantic segmentation.

Evaluation with no 'background' class. We first compare in Tab. 1 ('No background prompt' column) the results on datasets which aim at the segmentation of most of the pixels in an image and do not consider a 'background' class. We observe that our method `CLIP-DINOiser` achieves the best results on four datasets yielding +2.2, +5.0, +6.7 and +5.1 mIoU over the second best performing method. Interestingly, we outperform methods which build expensive prototypes per visual concept on fine-grained datasets, showing the benefit of our lightweight and generalizable method. The only drop (-0.8 mIoU) is seen on VOC20 with respect to OVDiff; we believe it is due to the benefit of generating per-concept negative prototypes which likely benefits this object-centric dataset. An adaptive granularity of feature correlation could help mitigate this drop, which we leave for future work.

Evaluation with 'background' class. We now compare our method on datasets which include a 'background' query in Tab. 1 ('W/ bkg prompt' column). In this setup, we also apply our background detection mechanism (detailed in Sec. 3.5) on VOC and Object in order to improve the stuff-like background detection. We observe that `CLIP-DINOiser` significantly outperforms all methods which do not construct prototypes. Moreover, we surpass OVDiff (which uses an ensemble of three models) on Context dataset by +2.3 mIoU and are on par on Object. It is to be noted that with a single feature extractor, the performance of OVDiff drops by -10 mIoU and the method requires the construction of a 'background' prototype *per concept*, otherwise losing another -10 mIoU on VOC. On the other hand, `CLIP-DINOiser` produces segmentation masks in a *single* pass of CLIP with the light addition of two convolutional layers while remaining fully open-vocabulary as it does not require *any* concept-specific constructs.

Qualitative results. We qualitatively compare in Fig. 6 `CLIP-DINOiser` with high-performing TCL [6], CLIP-DIY [59] (two recent methods which provide code) and our baseline method MaskCLIP [67] on images taken from the datasets considered in the evaluation. We observe that our method generates predictions accurate both in terms of localization and assignment. Indeed we obtain fined-grained results on the challenging datasets, e.g. in the Cityscapes example the text query 'car' and in the ADE20k example 'fountain' are accurately located when CLIP-DIY and TCL produce coarser results. Versus MaskCLIP, we can see the denoising capabilities of `CLIP-DINOiser` as MaskCLIP hallucinations grow with the number of text queries prompted at evaluation. Finally, in Fig. 1 we present 'in the wild' examples, beyond the evaluation benchmarks, and show that `CLIP-DINOiser` produces accurate segmentation masks for arbitrary and very specific prompts, such as 'wooden table' or 'leather bag'.

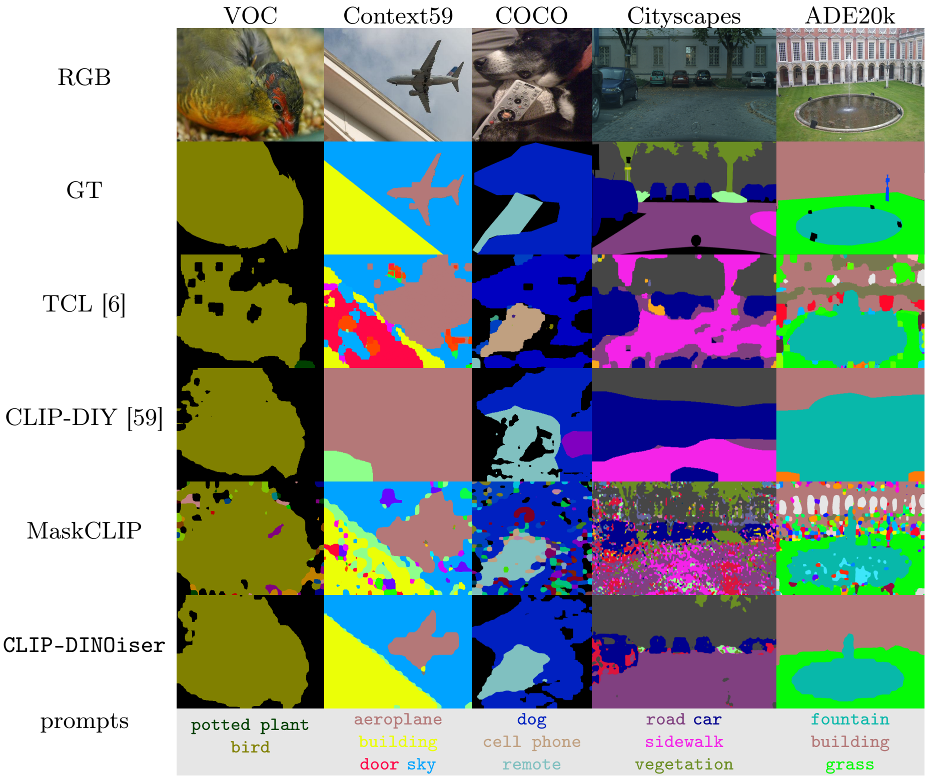


Fig. 6: Qualitative open-vocabulary segmentation results. We compare ours against CLIP-DIY [59], TCL [6] and MaskCLIP [67]. For a fair comparison, we do not apply post-processing. All pixels annotated in black are from the background class. We observe that our method achieves more accurate results both in terms of localization and class assignment.

4.3 Ablation study

We now conduct an ablation study of the different components of `CLIP-DINOiser` and investigate the impact of both our feature pooling strategy and background detection.

The impact of the pooling mechanism. We propose with `CLIP-DINOiser` to combine MaskCLIP *features* with a well-defined linear combination and compare different solutions in Tab. 2a. In [67], the authors proposed to refine the *predictions* with a combination weighted by CLIP *key* embeddings (noted 'CLIP keys (preds.)' in the table) and boost MaskCLIP results by more than +8 mIoU on VOC and VOC20, +1.8 and +1.0 and +0.6 mIoU on the other datasets. However, we show that working directly at the feature level allows us to achieve better results; we obtain consistent improvements ranging from +6 to +19 mIoU on all datasets when using DINO-based weight A^{ξ} and further improve when using trained CLIP-based weights A^{ϕ}.

Pooling strategy	VOC	VOC20	C59	Stuff	ADE
MaskCLIP [67] - *baseline*	32.9	61.8	25.6	17.6	14.3
CLIP keys (preds.) [67]	41.3	71.9	27.4	18.6	14.9
ours w. CLIP keys	39.2	73.2	23.0	12.6	7.7
ours w. DINO A^{ξ}	53.7	79.1	35.5	24.7	20.4
ours w. trained A^{ϕ}	54.0	80.9	35.9	24.6	20.0

(a) Pooling strategy

Pooling	Bkg det.	Object	VOC
MaskCLIP [67] - *baseline*		16.4	32.9
ours w. DINO A^{ξ}		29.9	53.7
ours w. DINO A^{ξ}	FOUND	32.1	60.1
ours w. DINO A^{ξ}	ours w. M	34.1	62.1
ours w. DINO A^{ξ}	ours w. M^{ϕ}	34.2	61.9
ours w. trained A^{ϕ}	ours w. M^{ϕ}	34.8	62.1

(b) Background detection

Table 2: Impact of the pooling strategy (a) and background detection (b) on diverse datasets reported with the mIoU metric.

The impact of the background detection. We now discuss the improvement provided by our background refinement strategy, which is applied when *stuff*-like background patches need to be detected. We report such results in Tab. 2b when employing our pooling strategy (either using DINO features, noted 'w. DINO A^{ξ}' or those extracted from CLIP, noted 'w. trained A^{ϕ}'). When using solely 'FOUND' for background detection, as in [59], we improve by +6.4 mIoU on VOC (achieving 60.1 mIoU), but when relaxing FOUND (see Sec. 3.5) with an uncertainty condition, we boost scores up to 62.1 on VOC, showing the limitation of using FOUND alone. We also achieve similar results when using CLIP-based predictions M^{ϕ} both with DINO-based A^{ξ} and trained CLIP-based A^{ϕ} correlations, although we observe that best results are overall obtained with trained A^{ϕ}. We visualize CLIP-based mask M^{ϕ} in Fig. 5a and see high similarity to DINO-based predictions, therefore showing the localization quality of CLIP.

5 Conclusions

In this work, we propose to make the most out of CLIP features and show that the features already contain useful *localization information.* Indeed with light convolutional layers, we are able to learn both good patch-correlation and objectness information by using DINO self-supervised model as a guide. With such information, our method `CLIP-DINOiser` performs zero-shot open-vocabulary semantic segmentation in a single pass of CLIP model and with two light extra convolutional layers. `CLIP-DINOiser` reaches state-of-the-art results on complex semantic segmentation datasets.

Limitations. Despite yielding strong results on open-vocabulary semantic segmentation, `CLIP-DINOiser` is still bounded by the capability of the CLIP model to separate classes, as it inherits its granularity. We believe that better prompt engineering paired with better image-text models could further boost the performance of `CLIP-DINOiser`.

Acknowledgments

This work was supported by the National Centre of Science (Poland) Grant No. 2022/45/B/ST6/02817 and by the grant from NVIDIA providing one RTX A5000 24GB used for this project.

References

1. Abdelreheem, A., Skorokhodov, I., Ovsjanikov, M., Wonka, P.: Satr: Zero-shot semantic segmentation of 3d shapes. In: ICCV (2023)
2. Bucher, M., Vu, T.H., Cord, M., Pérez, P.: Zero-shot semantic segmentation. In: NeurIPS (2019)
3. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: CVPR (2018)
4. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. In: NeurIPS (2020)
5. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: ICCV (2021)
6. Cha, J., Mun, J., Roh, B.: Learning to generate text-grounded mask for open-world semantic segmentation from only image-text pairs. In: CVPR (2023)
7. Chen, J., Zhu, D., Qian, G., Ghanem, B., Yan, Z., Zhu, C., Xiao, F., Culatana, S.C., Elhoseiny, M.: Exploring open-vocabulary semantic segmentation from clip vision encoder distillation only. In: ICCV (2023)
8. Chen, R., Liu, Y., Kong, L., Zhu, X., Ma, Y., Li, Y., Hou, Y., Qiao, Y., Wang, W.: Clip2scene: Towards label-efficient 3d scene understanding by clip. In: CVPR (2023)
9. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)
10. Contributors, M.: MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark (2020)
11. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
12. Darcet, T., Oquab, M., Mairal, J., Bojanowski, P.: Vision transformers need registers. arXiv preprint arXiv:2309.16588 (2023)
13. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
14. Ding, J., Xue, N., Xia, G.S., Dai, D.: Decoupling zero-shot semantic segmentation. In: CVPR (2022)
15. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
16. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results
17. Fang, A., Ilharco, G., Wortsman, M., Wan, Y., Shankar, V., Dave, A., Schmidt, L.: Data determines distributional robustness in contrastive language image pre-training (CLIP). In: ICML (2022)
18. Ghiasi, G., Gu, X., Cui, Y., Lin, T.Y.: Scaling open-vocabulary image segmentation with image-level labels. In: ECCV (2022)

19. Girdhar, R., El-Nouby, A., Liu, Z., Singh, M., Alwala, K.V., Joulin, A., Misra, I.: Imagebind: One embedding space to bind them all. In: CVPR (2023)
20. Gu, Z., Zhou, S., Niu, L., Zhao, Z., Zhang, L.: Context-aware feature generation for zero-shot semantic segmentation. In: ACM MM (2020)
21. Hu, P., Sclaroff, S., Saenko, K.: Uncertainty-aware learning for zero-shot semantic segmentation. In: NeurIPS (2020)
22. Ilharco, G., Wortsman, M., Wightman, R., Gordon, C., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., Schmidt, L.: Openclip (Jul 2021). https://doi.org/10.5281/zenodo.5143773, https://doi.org/10.5281/zenodo.5143773
23. Jatavallabhula, K.M., Kuwajerwala, A., Gu, Q., Omama, M., Chen, T., Li, S., Iyer, G., Saryazdi, S., Keetha, N., Tewari, A., et al.: Conceptfusion: Open-set multimodal 3d mapping. In: RSS (2023)
24. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: ICML (2021)
25. Karazija, L., Laina, I., Vedaldi, A., Rupprecht, C.: Diffusion models for zero-shot open-vocabulary segmentation. arXiv preprint arXiv:2306.09316 (2023)
26. Kato, N., Yamasaki, T., Aizawa, K.: Zero-shot semantic segmentation via variational mapping. In: ICCVW (2019)
27. Kerr, J., Kim, C.M., Goldberg, K., Kanazawa, A., Tancik, M.: Lerf: Language embedded radiance fields. In: ICCV (2023)
28. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
29. Li, B., Weinberger, K.Q., Belongie, S., Koltun, V., Ranftl, R.: Language-driven semantic segmentation. In: ICLR (2022)
30. Li, P., Wei, Y., Yang, Y.: Consistent structural relation learning for zero-shot segmentation. NeurIPS (2020)
31. Liang, F., Wu, B., Dai, X., Li, K., Zhao, Y., Zhang, H., Zhang, P., Vajda, P., Marculescu, D.: Open-vocabulary semantic segmentation with mask-adapted clip. In: CVPR (2023)
32. Liu, Q., Wen, Y., Han, J., Xu, C., Xu, H., Liang, X.: Open-world semantic segmentation via contrasting and clustering vision-language embedding. In: ECCV (2022)
33. Luo, H., Bao, J., Wu, Y., He, X., Li, T.: SegCLIP: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. In: ICML (2023)
34. Mayilvahanan, P., Wiedemer, T., Rusak, E., Bethge, M., Brendel, W.: Does CLIP's generalization performance mainly stem from high train-test similarity? arXiv preprint arXiv:2310.09562 (2023)
35. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: ICLR (2013)
36. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: CVPR (2014)
37. Mukhoti, J., Lin, T.Y., Poursaeed, O., Wang, R., Shah, A., Torr, P.H., Lim, S.N.: Open vocabulary semantic segmentation with patch aligned contrastive learning. In: CVPR (2023)
38. Najibi, M., Ji, J., Zhou, Y., Qi, C.R., Yan, X., Ettinger, S., Anguelov, D.: Unsupervised 3d perception with 2d vision-language distillation for autonomous driving. In: ICCV (2023)

39. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., HAZIZA, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y., Li, S.W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: DINOv2: Learning robust visual features without supervision. TMLR (2024)
40. Pastore, G., Cermelli, F., Xian, Y., Mancini, M., Akata, Z., Caputo, B.: A closer look at self-training for zero-label semantic segmentation. In: CVPR (2021)
41. Peng, S., Genova, K., Jiang, C., Tagliasacchi, A., Pollefeys, M., Funkhouser, T., et al.: Openscene: 3d scene understanding with open vocabularies. In: CVPR (2023)
42. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: EMNLP (2014)
43. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
44. Ranasinghe, K., McKinzie, B., Ravi, S., Yang, Y., Toshev, A., Shlens, J.: Perceptual grouping in contrastive vision-language models. In: ICCV (2023)
45. Rao, Y., Zhao, W., Chen, G., Tang, Y., Zhu, Z., Huang, G., Zhou, J., Lu, J.: Denseclip: Language-guided dense prediction with context-aware prompting. In: CVPR (2022)
46. Ren, P., Li, C., Xu, H., Zhu, Y., Wang, G., Liu, J., Chang, X., Liang, X.: Viewco: Discovering text-supervised segmentation masks via multi-view semantic consistency. In: ICLR (2023)
47. Rewatbowornwong, P., Chatthee, N., Chuangsuwanich, E., Suwajanakorn, S.: Zero-guidance segmentation using zero segment labels. In: ICCV (2023)
48. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)
49. Shin, G., Xie, W., Albanie, S.: Reco: Retrieve and co-segment for zero-shot transfer. In: NeurIPS (2022)
50. Shin, G., Xie, W., Albanie, S.: Namedmask: Distilling segmenters from complementary foundation models. In: CVPRW (2023)
51. Siméoni, O., Puy, G., Vo, H.V., Roburin, S., Gidaris, S., Bursuc, A., Pérez, P., Marlet, R., Ponce, J.: Localizing objects with self-supervised transformers and no labels. In: BMVC (2021)
52. Siméoni, O., Sekkat, C., Puy, G., Vobeckỳ, A., Zablocki, É., Pérez, P.: Unsupervised object localization: Observing the background to discover objects. In: CVPR (2023)
53. Siméoni, O., Zablocki, É., Gidaris, S., Puy, G., Pérez, P.: Unsupervised object localization in the era of self-supervised vits: A survey. arXiv preprint arXiv:2310.12904 (2023)
54. Vobeckỳ, A., Siméoni, O., Hurych, D., Gidaris, S., Bursuc, A., Perez, P., Sivic, J.: Pop-3d: Open-vocabulary 3d occupancy prediction from images. In: NeurIPS (2023)
55. Walmer, M., Suri, S., Gupta, K., Shrivastava, A.: Teaching matters: Investigating the role of supervision in vision transformers. In: CVPR (2023)
56. Wang, X., Girdhar, R., Yu, S.X., Misra, I.: Cut and learn for unsupervised object detection and instance segmentation. In: CVPR (2023)
57. Wang, Y., Shen, X., Hu, S.X., Yuan, Y., Crowley, J.L., Vaufreydaz, D.: Self-supervised transformers for unsupervised object discovery using normalized cut. In: CVPR (2022)
58. Wu, J., Li, X., Xu, S., Yuan, H., Ding, H., Yang, Y., Li, X., Zhang, J., Tong, Y., Jiang, X., et al.: Towards open vocabulary learning: A survey. T-PAMI (2024)

59. Wysoczanska, M., Ramamonjisoa, M., Trzcinski, T., Simeoni, O.: Clip-diy: Clip dense inference yields open-vocabulary semantic segmentation for-free. In: WACV (2024)
60. Xian, Y., Choudhury, S., He, Y., Schiele, B., Akata, Z.: Semantic projection network for zero-and few-label semantic segmentation. In: CVPR (2019)
61. Xu, J., De Mello, S., Liu, S., Byeon, W., Breuel, T., Kautz, J., Wang, X.: Groupvit: Semantic segmentation emerges from text supervision. arXiv preprint arXiv:2202.11094 (2022)
62. Xu, J., Hou, J., Zhang, Y., Feng, R., Wang, Y., Qiao, Y., Xie, W.: Learning open-vocabulary semantic segmentation models from natural language supervision. In: CVPR (2023)
63. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: ICCV (2023)
64. Zhao, H., Puig, X., Zhou, B., Fidler, S., Torralba, A.: Open vocabulary scene parsing. In: ICCV (2017)
65. Zhong, Y., Yang, J., Zhang, P., Li, C., Codella, N., Li, L.H., Zhou, L., Dai, X., Yuan, L., Li, Y., et al.: Regionclip: Region-based language-image pretraining. In: CVPR (2022)
66. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. IJCV (2019)
67. Zhou, C., Loy, C.C., Dai, B.: Extract free dense labels from clip. In: ECCV (2022)
68. Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., Kong, T.: iBOT: Image bert pre-training with online tokenizer. In: ICLR (2022)