# `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]    Patrick Pérez[2]
[1]Warsaw University of Technology, [2]Valeo.ai, [3]Meta AI, [4]Tooploox, [5]IDEAS NCBR

Figure 1. **Examples of open-vocabulary semantic segmentation results** obtained with our method `CLIP-DINOiser` on 'in-the-wild' images vs. those of MaskCLIP [75]. 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* [25] 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.

## 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 pri-ors 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`.

## 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) [27, 47, 71] is

---

arXiv:2312.12359v2 [cs.CV] 27 Mar 2024

1

stimulating a shift from a closed-vocabulary 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 [20, 38], but struggle to provide dense *open-vocabulary features* [21, 75]. Obtaining such an alignment between pixels and language can lead to open-vocabulary extensions for multiple other modalities, such as point clouds [9, 26, 42, 45], 3D scenes [60], 3D shapes [1], radiance fields [30], inter-modality alignment [22, 26], with multiple potential applications for which the construction of training datasets is even more challenging and where CLIP-derived models are showing 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, 41, 48, 68, 69] or finetuning on an annotated segmentation dataset with pre-defined classes [36, 75]. 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, 26, 30, 55, 56, 66], but are less practical due to the extensive computational overhead. MaskCLIP [75] 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. 3 middle column).

Meanwhile, recent self-supervised learning (SSL) approaches [4, 5, 10, 76] 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 [57, 58, 63, 64]. DINO features prove useful also for zero-shot semantic segmentation [28, 30, 66], but require expensive sliding window sampling [30, 66] or building concept-specific prototypes and ensemble strategies [28].

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 [58] 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 [14]. 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 [19], Cityscapes [12] and ADE20K [74].

## 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, 23, 24, 29, 33, 44, 67, 72]. 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 [39, 46]. 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 [25, 27, 47] 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 [21, 75]. 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, 41, 48, 68, 69] on datasets with annotations of varying granularity and quality: dense annotations [32, 34], class-agnostic object masks [16, 21, 49], coarse captions [6, 21, 34, 36, 37, 41, 48, 68, 69, 73] or pseudo-labels [75]. Recent works leverage image-level captions to align text to regions (obtained without supervision):

PACL [41] trains an embedder module to learn patch-to-text affinity, TCL [6] proposes a local constrative objective to align well-selected patches to the text and ViewCO [50] 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 [32, 34, 75], but alters the open-vocabulary properties of the features [26].

Most related to us is a line of works that investigate how to directly densify CLIP features [1, 26, 30, 66, 75] to obtain per-patch CLIP features. Such densification can be performed by aggregating features from multiple views [1, 30] or from sliding windows [26, 66] at the extra-cost of multiple forward passes. MaskCLIP [75] drops the global pooling layer of CLIP and matches the projected features directly to text via a $1 \times 1$ convolution layer. By doing so they achieve dense predictions, however noisy.

With a concept-driven perspective, some methods [28, 55, 56] build codebooks of visual prototypes per concept, including negative prototypes [28], and then perform co-segmentation [55]. 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, 10, 13, 76] have demonstrated features with good localization properties [57, 58, 63, 64]. Such features have also been exploited in the context of open-vocabulary segmentation methods, e.g. for pre-training for the visual backbone [8, 48, 69], co-segmentation [55], clustering patches into masks [51], representing object prototypes [28]. Related to us is the recent CLIP-DIY [66] which computes patch-level representations from CLIP features from different image crops with guidance from an unsupervised saliency segmenter [58] 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 [75] 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. 5). 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 [17] 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 [25] 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. [75] 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 $1 \times 1$ 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 [75].

**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. 3 (middle column).

---

[1] We adopt the taxonomy defined in the recent survey [65] and define our method as 'open-vocabulary', with capabilities to generalize to unseen datasets.

## 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, 43, 57–59, 64] .

**Extracting self-supervised correlation information.** Recent works [57, 64] 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 Sec. A.2). 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.

**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 $A^\xi$. 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 $A^\xi$, presented in Fig. 2. We zero-out $A^\xi$ correlations below a threshold $\gamma$, following [57, 64], and compute the new features for patch $p \in \{1, \dots, N\}$:

$$F_p^+ = \frac{1}{\sum_{q=1}^N A_{p,q}^\xi} \sum_{q=1}^N A_{p,q}^\xi \cdot F_p. \qquad (1)$$
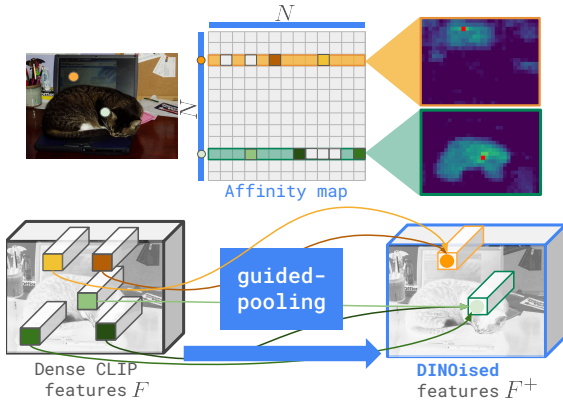


Figure 2. **Guided pooling** strategy defined in Eq. (1). The $N \times N$ affinity matrix is computed from patch features and is used to refine MaskCLIP features (bottom left).
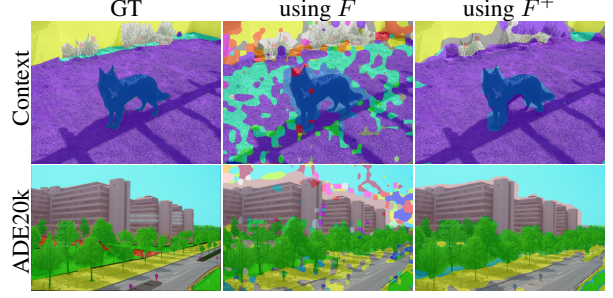


Figure 3. **Impact of the pooling.** We compare our results with $F^+$ (right) versus those obtained with MaskCLIP features (middle).

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. 3, 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

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 $A^\xi$ from CLIP with a single convolutional layer.
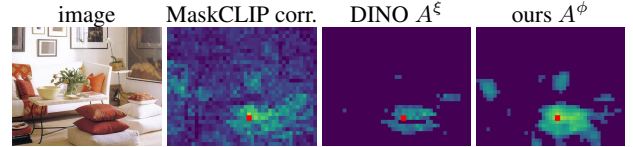


Figure 4. **Comparison of the affinity maps** between a *seed* (on a 'pillow') and the other patch features when using features of MaskCLIP, DINO and ours after training.

In order to predict the DINO affinity map $A^\xi$ from CLIP features, we train a *single $3 \times 3$ 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 $A^\phi \in [-1, 1]^{N \times N}$:

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

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

$$\mathcal{L}^c = \sum_{p=1}^N \left[ D_p \log A_p^\phi + (1 - D_p) \log(1 - A_p^\phi) \right]. \qquad (3)$$
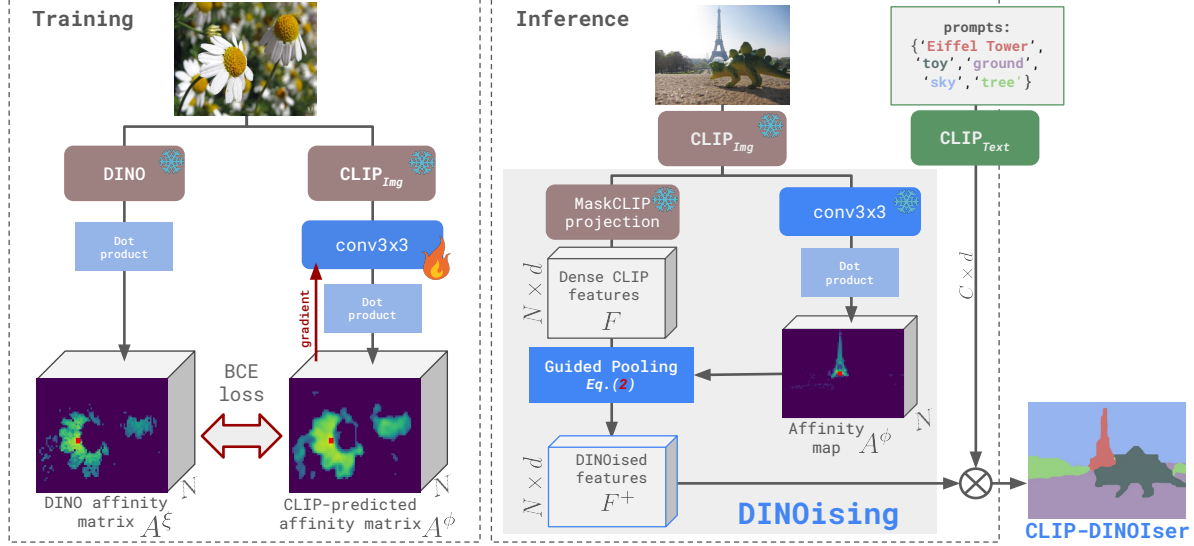
Figure 5. **Overview of CLIP-DINOiser** which leverages the quality of self-supervised features to improve the notoriously noisy MaskCLIP feature maps. We use DINO as a teacher which 'teaches' CLIP how to extract localization information. We train (left) a conv3 × 3 layer to reproduce the patch correlations 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 present our layer training in Fig. 5 (left part) and observe the quality of CLIP-predicted affinity matrix $A^\phi$. We also show in Fig. 4 another example of obtained $A^\phi$ and observe their similarity to DINO-based correlations. We use the CLIP-produced correlations $A^\phi$ to replace $A^\xi$ 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. 5 (*inference* step), our method runs with a single forward pass of CLIP model and a small extra layer.

### 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 [58]. We argue that relying solely on the textual prompt 'background' to catch all non-salient patches is underperforming and, similarly to [66], we propose to use a very light-weight *unsupervised* foreground/background segmentation method, namely FOUND [58] 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. In order 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 $M$.



Figure 6. **Comparison of *objectness* mask** generated by FOUND [58] (left) and with our layer using CLIP features (right).

**Learning FOUND objectness.** Moreover, we are also able to learn the predictions of FOUND [58] directly from CLIP features. To do so, we train a *single* $1 \times 1$ 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. 6 and observe their very high similarity to those pro-
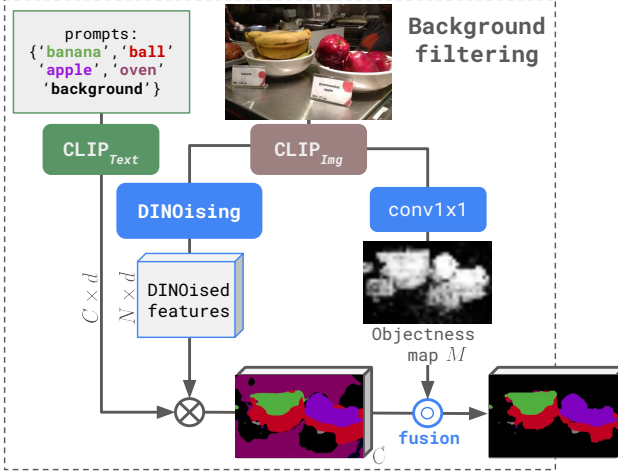
Figure 7. **Overview of our *background filtering*** applied when a 'background' prompt is provided to help reduce hallucinations.

duced 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 provide an example of the *background filtering* with trained objectness in Fig. 7.

## 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 [25]. 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 [61]). The first layer $g(\cdot)$ has a kernel $3 \times 3$ and output dimension $d_g = 256$ and $h(\cdot)$ a kernel $1 \times 1$ 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 [58]–which also uses DINO model. We train both layers with a binary cross-entropy loss on *only 1k raw images* randomly sampled from ImageNet [14] dataset *without any annotation*. We report average scores over 3 runs with different sampling seeds and provide standard deviations in appendix (Sec. A.1). We follow [64] 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 [31], which takes approxi-

mately 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 [19] (noted 'VOC'), PASCAL Context [40] (noted 'Context'), and COCO Object [3] (noted 'Object') and the second without: PASCAL VOC20 [19] (noted 'VOC20'), PASCAL Context59 [40] (noted 'C59'), COCO-Stuff [3] (noted 'Stuff'), Cityscapes [12] (noted 'City'), and ADE20K [74] (noted 'ADE'). We evaluate results with the standard mIoU metric. We also follow the evaluation protocol of [6], use the implementations provided by MMSegmentation [11], 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 [25, 68, 75].

**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 [65] 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 [56], MaskCLIP+ [75]); (2) *construct prototypes*: ReCO [55], OVDiff [28]; (3) *train with text supervision* including GroupViT [68], ZeroSeg [51], SegCLIP [37], TCL [6], CLIPpy [48], OVSegmentor [69], 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 [66] and MaskCLIP [75], 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.

### 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.

| Methods | Concept spec. | Frozen backbone | Extra data | Backbone at inference | No background prompt | | | | | W/ bkg prompt | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | VOC20 | C59 | Stuff | City | ADE | Context | Object | VOC |
| **Dataset specific** | | | | | | | | | | | | |
| MaskCLIP+ [75] | ✓ | ✗ | Target dataset | DeepLabv2 | - | - | 18.0 | - | - | <u>31.1</u> | - | - |
| NamedMask [56] | ✓ | ✗ | IN(1.2M)+Target | DeepLabv3+ | - | - | - | - | - | - | 27.7 | 59.2 |
| **Build prototypes per visual concept** | | | | | | | | | | | | |
| ReCo [55] | ✓ | ✓ | IN(1.2M) | CLIP | 57.8 | 22.3 | 14.8 | 21.1 | 11.2 | 19.9 | 15.7 | 25.1 |
| OVDiff [28] | ✓ | ✓ | ✗ | CLIP+DINO+SD | **81.7** | <u>33.7</u> | - | - | <u>14.9</u> | 30.1 | **34.8** | **67.1** |
| **Text/image alignment training with captions** | | | | | | | | | | | | |
| GroupViT [68] | ✗ | ✗ | CC12M [7]+RedCaps [15] | CLIP | 79.7 | 23.4 | 15.3 | 11.1 | 9.2 | 18.7 | 27.5 | 50.4 |
| ZeroSeg [8] | ✗ | ✗ | IN(1.2M)+CC12M [7] | CLIP | - | - | - | - | - | 21.8 | 22.1 | 42.9 |
| SegCLIP [37] | ✗ | ✗ | CC3M [53]+COCO(400k) | CLIP | - | - | - | 11.0 | 8.7 | 24.7 | 26.5 | 52.6 |
| TCL [6] | ✗ | ✗ | CC12M [7]+CC3M [53] | CLIP | 77.5 | 30.3 | <u>19.6</u> | 23.1 | <u>14.9</u> | 24.3 | 30.4 | 51.2 |
| CLIPpy [48] | ✗ | ✗ | HQITP-134M [48] | CLIP | - | - | - | - | 13.5 | - | 32.0 | 52.2 |
| OVSegmentor [69] | ✗ | ✗ | CC4M [69] | CLIP | - | - | - | - | 5.6 | 20.4 | 25.1 | 53.8 |
| **Frozen CLIP** | | | | | | | | | | | | |
| CLIP-DIY [66]* | ✗ | ✓ | ✗ | CLIP+DINO | 79.7 | 19.8 | 13.3 | 11.6 | 9.9 | 19.7 | 31.0 | 59.9 |
| MaskCLIP [75] [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 | <u>25.0</u> | 14.3 | 22.9 | 16.4 | 32.9 |
| MaskCLIP* † | ✗ | ✓ | ✗ | CLIP | 71.9 | 27.4 | 18.6 | 23.0 | <u>14.9</u> | 24.0 | 21.6 | 41.3 |
| `CLIP-DINOiser` | ✗ | ✓ | IN (random 1k im.) | CLIP | <u>80.9</u> | **35.9** | **24.6** | **31.7** | **20.0** | **32.4** | **34.8** | <u>62.1</u> |

Table 1. **Open-vocabulary semantic segmentation quantitative comparison** using the mIoU metric. We separate the datasets used for evaluation into two columns: 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 [25] and mark with † the use of MaskCLIP refinement. The first and second best methods are respectively **bold** and <u>underlined</u>. We specify if a method assumes prior access to names of concepts ('Concept spec.') and what additional data is used at training ('Extra data'). 'IN' stands for ImageNet [14] and 'SD' for Stable Diffusion [52]. We refer to Sec. 4.1 for more details on baselines.

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. 8 `CLIP-DINOiser` with high-performing TCL [6], CLIP-DIY [66] (two recent methods which provide code) and our baseline method MaskCLIP [75] 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'.

### 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.
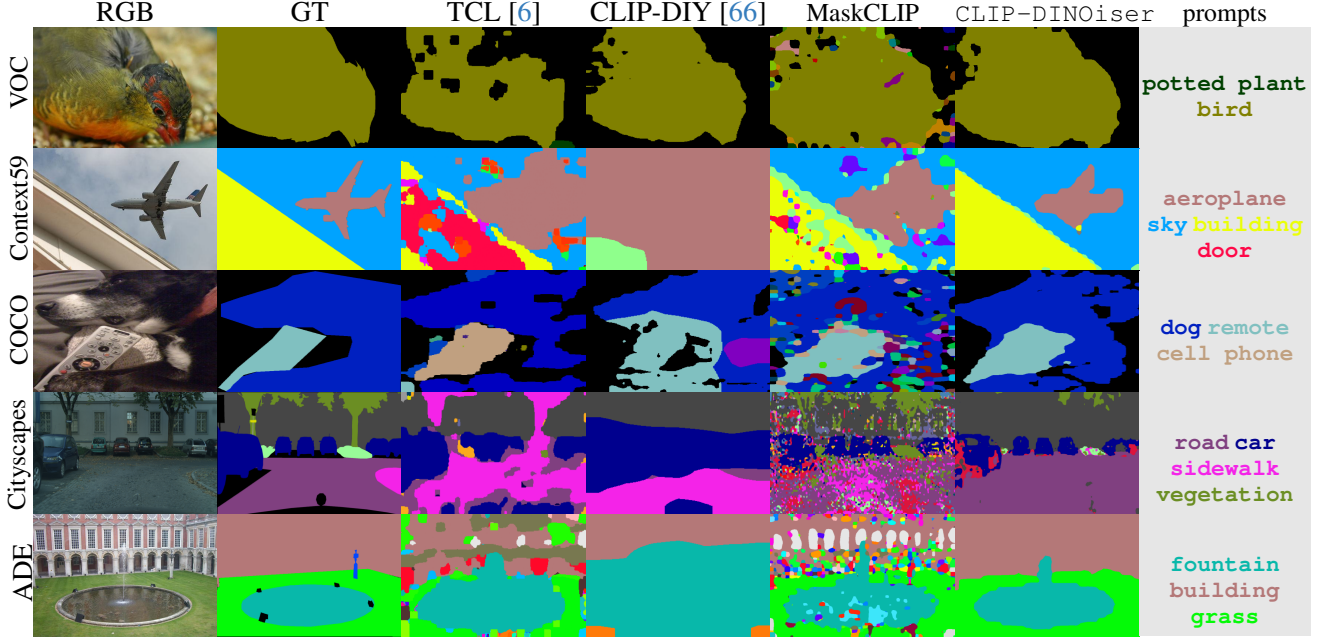
Figure 8. **Qualitative open-vocabulary segmentation results**. We compare ours against CLIP-DIY [66], TCL [6] and MaskCLIP [75]. For a fair comparison, we do not apply post-processing. All pixels annotated in black are from the background class.

**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 [75], 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 $A^\xi$ and further improve when using trained CLIP-based weights $A^\phi$.

| Pooling strategy | VOC | VOC20 | C59 | Stuff | ADE |
|---|---|---|---|---|---|
| MaskCLIP [75] - *baseline* | 32.9 | 61.8 | 25.6 | 17.6 | 14.3 |
| CLIP keys (preds.) [75] | 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 $A^\xi$ | 53.7 | 79.1 | 35.5 | 24.7 | 20.4 |
| ours w. trained $A^\phi$ | 54.0 | 80.9 | 35.9 | 24.6 | 20.0 |

(a) **Pooling strategy**

| Pooling | Bkg det. | Object | VOC |
|---|---|---|---|
| MaskCLIP [75] - *baseline* | | 16.4 | 32.9 |
| ours w. DINO $A^\xi$ | | 29.9 | 53.7 |
| ours w. DINO $A^\xi$ | FOUND | 32.1 | 60.1 |
| ours w. DINO $A^\xi$ | ours w. $M$ | 34.1 | 62.1 |
| ours w. DINO $A^\xi$ | ours w. $M^\phi$ | 34.2 | 61.9 |
| ours w. trained $A^\phi$ | ours w. $M^\phi$ | 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 $A^\xi$' or those extracted from CLIP, noted 'w. trained $A^\phi$'). When using solely 'FOUND' for background detection, as in [66], 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 $M^\phi$ both with DINO-based $A^\xi$ and trained CLIP-based $A^\phi$ correlations, although we observe that best results are overall obtained with trained $A^\phi$. We visualize CLIP-based mask $M^\phi$ in Fig. 6 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

## References

[1] Ahmed Abdelreheem, Ivan Skorokhodov, Maks Ovsjanikov, and Peter Wonka. Satr: Zero-shot semantic segmentation of 3d shapes. In *ICCV*, 2023. 2, 3

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

[3] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, 2018. 2, 6

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 2, 3

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 1, 2, 3, 4, 6, 12

[6] Junbum Cha, Jonghwan Mun, and Byungseok Roh. Learning to generate text-grounded mask for open-world semantic segmentation from only image-text pairs. In *CVPR*, 2023. 2, 3, 6, 7, 8

[7] 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. 7

[8] Jun Chen, Deyao Zhu, Guocheng Qian, Bernard Ghanem, Zhicheng Yan, Chenchen Zhu, Fanyi Xiao, Sean Chang Culatana, and Mohamed Elhoseiny. Exploring open-vocabulary semantic segmentation from clip vision encoder distillation only. In *ICCV*, 2023. 3, 7

[9] Runnan Chen, Youquan Liu, Lingdong Kong, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, Yu Qiao, and Wenping Wang. Clip2scene: Towards label-efficient 3d scene understanding by clip. In *CVPR*, 2023. 2

[10] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2, 3

[11] MMSegmentation Contributors. MMSegmentation: Open-mmlab semantic segmentation toolbox and benchmark, 2020. 6

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2, 6

[13] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023. 3

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2, 6, 7

[15] Karan Desai, Gaurav Kaul, Zubin Aysola, and Justin Johnson. RedCaps: Web-curated image-text data created by the people, for the people. In *NeurIPS Datasets and Benchmarks*, 2021. 7

[16] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *CVPR*, 2022. 2

[17] 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. In *ICLR*, 2021. 3

[18] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results, . 13

[19] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results, . 2, 6, 12, 13

[20] Alex Fang, Gabriel Ilharco, Mitchell Wortsman, Yuhao Wan, Vaishaal Shankar, Achal Dave, and Ludwig Schmidt. Data determines distributional robustness in contrastive language image pre-training (CLIP). In *ICML*, 2022. 2

[21] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *ECCV*, 2022. 2

[22] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *CVPR*, 2023. 2

[23] Zhangxuan Gu, Siyuan Zhou, Li Niu, Zihan Zhao, and Liqing Zhang. Context-aware feature generation for zero-shot semantic segmentation. In *ACM MM*, 2020. 2

[24] Ping Hu, Stan Sclaroff, and Kate Saenko. Uncertainty-aware learning for zero-shot semantic segmentation. In *NeurIPS*, 2020. 2

[25] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. 1, 2, 3, 6, 7

[26] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, et al. Conceptfusion: Open-set multimodal 3d mapping. In *RSS*, 2023. 2, 3

[27] 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 *ICML*, 2021. 1, 2

[28] Laurynas Karazija, Iro Laina, Andrea Vedaldi, and Christian Rupprecht. Diffusion models for zero-shot open-vocabulary segmentation. *arXiv preprint arXiv:2306.09316*, 2023. 2, 3, 6, 7

[29] Naoki Kato, Toshihiko Yamasaki, and Kiyoharu Aizawa. Zero-shot semantic segmentation via variational mapping. In *ICCVW*, 2019. 2

[30] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *ICCV*, 2023. 2, 3

[31] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[32] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *ICLR*, 2022. 2, 3

[33] Peike Li, Yunchao Wei, and Yi Yang. Consistent structural relation learning for zero-shot segmentation. *NeurIPS*, 2020. 2

[34] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *CVPR*, 2023. 2, 3

[35] 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*, 2014. 13

[36] Quande Liu, Youpeng Wen, Jianhua Han, Chunjing Xu, Hang Xu, and Xiaodan Liang. Open-world semantic segmentation via contrasting and clustering vision-language embedding. In *ECCV*, 2022. 2

[37] Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li. SegCLIP: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. In *ICML*, 2023. 2, 6, 7

[38] Prasanna Mayilvahanan, Thaddäus Wiedemer, Evgenia Rusak, Matthias Bethge, and Wieland Brendel. Does CLIP's generalization performance mainly stem from high train-test similarity? *arXiv preprint arXiv:2310.09562*, 2023. 2

[39] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013. 2

[40] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014. 6

[41] 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 *CVPR*, 2023. 2, 3

[42] Mahyar Najibi, Jingwei Ji, Yin Zhou, Charles R Qi, Xinchen Yan, Scott Ettinger, and Dragomir Anguelov. Unsupervised 3d perception with 2d vision-language distillation for autonomous driving. In *ICCV*, 2023. 2

[43] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024. 4

[44] Giuseppe Pastore, Fabio Cermelli, Yongqin Xian, Massimiliano Mancini, Zeynep Akata, and Barbara Caputo. A closer look at self-training for zero-label semantic segmentation. In *CVPR*, 2021. 2

[45] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *CVPR*, 2023. 2

[46] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 2

[47] 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 *ICML*, 2021. 1, 2

[48] Kanchana Ranasinghe, Brandon McKinzie, Sachin Ravi, Yinfei Yang, Alexander Toshev, and Jonathon Shlens. Perceptual grouping in contrastive vision-language models. In *ICCV*, 2023. 2, 3, 6, 7

[49] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *CVPR*, 2022. 2

[50] Pengzhen Ren, Changlin Li, Hang Xu, Yi Zhu, Guangrun Wang, Jianzhuang Liu, Xiaojun Chang, and Xiaodan Liang. Viewco: Discovering text-supervised segmentation masks via multi-view semantic consistency. In *ICLR*, 2023. 3

[51] Pitchaporn Rewatbowornwong, Nattanat Chatthee, Ekapol Chuangsuwanich, and Supasorn Suwajanakorn. Zero-guidance segmentation using zero segment labels. In *ICCV*, 2023. 3, 6

[52] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 7

[53] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. 2018. 7

[54] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. *T-PAMI*, 2015. 13

[55] Gyungin Shin, Weidi Xie, and Samuel Albanie. Reco: Retrieve and co-segment for zero-shot transfer. In *NeurIPS*, 2022. 2, 3, 6, 7

[56] Gyungin Shin, Weidi Xie, and Samuel Albanie. Namedmask: Distilling segmenters from complementary foundation models. In *CVPRW*, 2023. 2, 3, 6, 7

[57] Oriane Siméoni, Gilles Puy, Huy V. Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. In *BMVC*, 2021. 2, 3, 4, 12

[58] Oriane Siméoni, Chloé Sekkat, Gilles Puy, Antonín Vobecký, Éloi Zablocki, and Patrick Pérez. Unsupervised object localization: Observing the background to discover objects. In *CVPR*, 2023. 2, 3, 5, 6, 12, 13

[59] Oriane Siméoni, Éloi Zablocki, Spyros Gidaris, Gilles Puy, and Patrick Pérez. Unsupervised object localization in the era of self-supervised vits: A survey. *arXiv preprint arXiv:2310.12904*, 2023. 4

[60] Antonín Vobecký, Oriane Siméoni, David Hurych, Spyros Gidaris, Andrei Bursuc, Patrick Perez, and Josef Sivic. Pop-3d: Open-vocabulary 3d occupancy prediction from images. In *NeurIPS*, 2023. 2

[61] Matthew Walmer, Saksham Suri, Kamal Gupta, and Abhinav Shrivastava. Teaching matters: Investigating the role of supervision in vision transformers. In *CVPR*, 2023. 6

[62] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *CVPR*, 2017. 12, 13

[63] Xudong Wang, Rohit Girdhar, Stella X Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation. In *CVPR*, 2023. 2, 3, 12

[64] 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 *CVPR*, 2022. 2, 3, 4, 6, 12

[65] Jianzong Wu, Xiangtai Li, Shilin Xu, Haobo Yuan, Henghui Ding, Yibo Yang, Xia Li, Jiangning Zhang, Yunhai Tong, Xudong Jiang, et al. Towards open vocabulary learning: A survey. *T-PAMI*, 2024. 3, 6

[66] Monika Wysoczanska, Michael Ramamonjisoa, Tomasz Trzcinski, and Oriane Simeoni. Clip-diy: Clip dense inference yields open-vocabulary semantic segmentation for-free. In *WACV*, 2024. 2, 3, 5, 6, 7, 8

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

[68] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. *arXiv preprint arXiv:2202.11094*, 2022. 2, 6, 7

[69] Jilan Xu, Junlin Hou, Yuejie Zhang, Rui Feng, Yi Wang, Yu Qiao, and Weidi Xie. Learning open-vocabulary semantic segmentation models from natural language supervision. In *CVPR*, 2023. 2, 3, 6, 7

[70] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, 2013. 12, 13

[71] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. 1

[72] Hang Zhao, Xavier Puig, Bolei Zhou, Sanja Fidler, and Antonio Torralba. Open vocabulary scene parsing. In *ICCV*, 2017. 2

[73] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *CVPR*, 2022. 2

[74] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 2019. 2, 6

[75] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *ECCV*, 2022. 1, 2, 3, 6, 7, 8

[76] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT: Image bert pre-training with online tokenizer. In *ICLR*, 2022. 2, 3

## A. More experimental results

### A.1. The impact of the training dataset

**Training stability.** We report in the main paper the final results averaged over three different randomly sampled subsets of ImageNet used for the training. In the first row of Tab. 3 we report the corresponding standard deviation. We observe that in all cases the standard deviation equals 0.1 mIoU or lower, therefore showing the stability of our training.

**Training with different datasets.** Our method `CLIP-DINOiser` does not require any labels to be trained. We investigate here the impact of training on the datasets used to train self-supervised DINO [5] and FOUND [58], namely Imagenet and DUTS-TR [62]. We report scores in Tab. 3. We also provide results when increasing the dataset size to 10k on ImageNet. In all cases, we observe no significant difference when using one dataset or another, and the size of the dataset does not seem to impact results positively.

| Train. dataset | C59 | V20 | Stuff | City | ADE |
|---|---|---|---|---|---|
| IN-1k | $35.9_{\pm 0.1}$ | $80.9_{\pm 0.0}$ | $24.6_{\pm 0.1}$ | $31.7_{\pm 0.1}$ | $20.0_{\pm 0.0}$ |
| IN-10k | $35.9_{\pm 0.0}$ | $80.3_{\pm 0.1}$ | $24.7_{\pm 0.0}$ | $31.9_{\pm 0.1}$ | $20.1_{\pm 0.0}$ |
| DUTS-TR [62] | 35.9 | 80.5 | 24.6 | 31.3 | 19.9 |

(a) Benchmark **without** 'background' prompt

| Train. dataset | VOC | Con. | Obj |
|---|---|---|---|
| IN-1k | $62.1_{\pm 0.0}$ | $32.4_{\pm 0.1}$ | $34.8_{\pm 0.1}$ |
| IN-10k | $61.9_{\pm 0.0}$ | $32.4_{\pm 0.0}$ | $34.6_{\pm 0.1}$ |
| DUTS-TR [62] | 62.0 | 32.4 | 34.8 |

(b) Benchmark **with** 'background' prompt

Table 3. **Performance with different training datasets.** When using random splits extracted from ImageNet (noted 'IN'), we report the average score and standard deviation computed over training with three random splits (of 1k or 10k) extracted in ImageNet. In (a) we report the scores on the datasets without 'background' class and in (b) with.

### A.2. Self-supervised features discussion

We present in Fig. 8 visualizations of correlation obtained using different DINO embeddings extracted from DINO's last attention layer, namely 'query', 'key' and 'value'. Most unsupervised localization methods [57, 58, 63, 64] use the 'key' embeddings which allow the easy separation of *foreground* from *background*. However, we observed in this work that using instead the *value* features allows us to separate better elements in the background, as visible in the figure. Patches in the background correlate to fewer background patches and regions are therefore better separated.
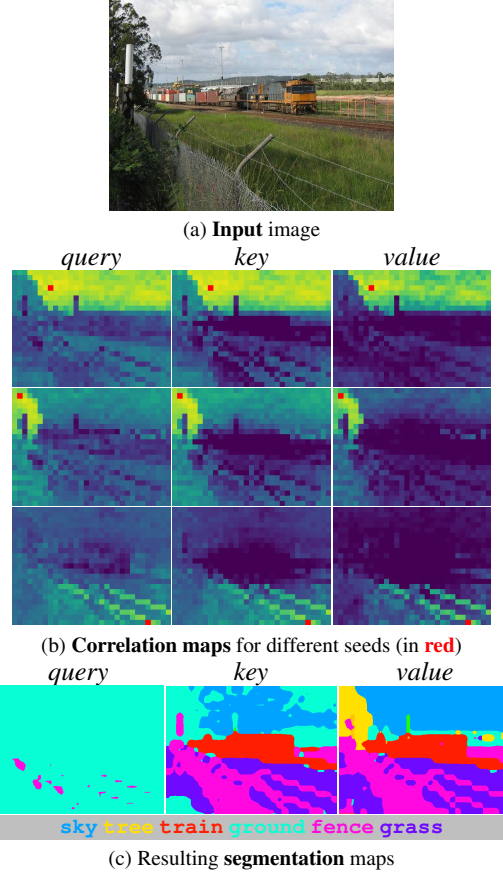


(a) **Input** image



(b) **Correlation maps** for different seeds (in red)



(c) Resulting **segmentation** maps

Figure 8. **Visualization of correlation and segmentation** obtained with different embeddings of DINO: *query*, *key* and *value*.

| Method | Single obj. discovery | | | Unsup. saliency detection | | |
|---|---|---|---|---|---|---|
| | VOC7 | VOC12 | C20k | DUT-O. | DUTS-T. | ECSSD |
| FOUND [58] | 72.5 | **76.1** | 62.9 | **60.8** | 65.4 | 80.5 |
| ours | **73.1** | 75.9 | **64.4** | 60.6 | **66.6** | **81.3** |

Table 4. Results of **single object discovery and unsupervised saliency detection** obtained when following FOUND [58] protocol. We compute the single object discovery scores on classic VOC benchmarks [19] and 20k images of COCO (noted 'C20k') following [58] and use the CorLoc metric. We report the mIoU metric for unsupervised saliency detection and provide all results with the post-processing bilateral solver. We note 'DUT-O.' DUT-OMRON [70] and 'DUTS-T.' stands for DUTS-TEST [62].

We also depict the final segmentation when using each type of feature, and observe the best result with 'value'. We observe that more objects in the background are well-segmented and labeled, e.g., 'tree' and 'sky'.

### A.3. Background evaluation with FOUND

We now evaluate the quality of our background filtering using the class-agnostic foreground/background protocol defined in [58]. We report in Tab. 4 the scores on the
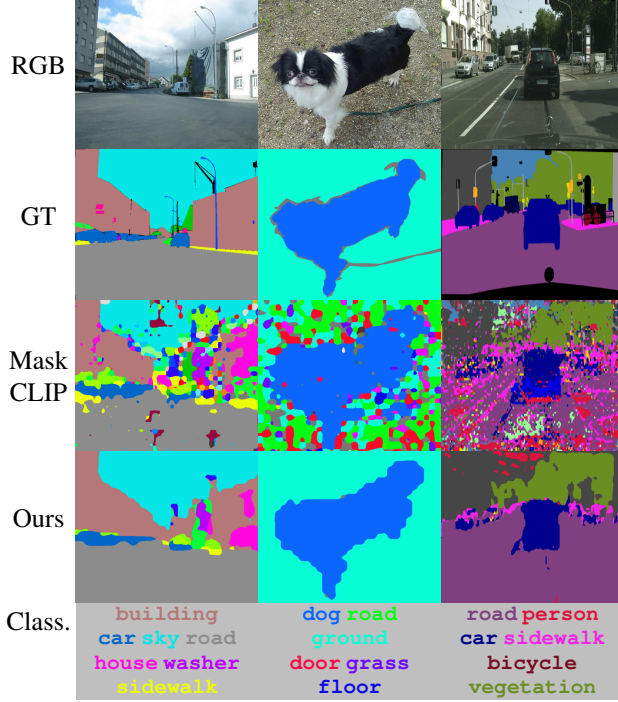
Figure 9. **Visual ablations of the impact of our pooling method**. Examples from ADE20K (left), PASCAL Context (middle), and Cityscapes (right) datasets.
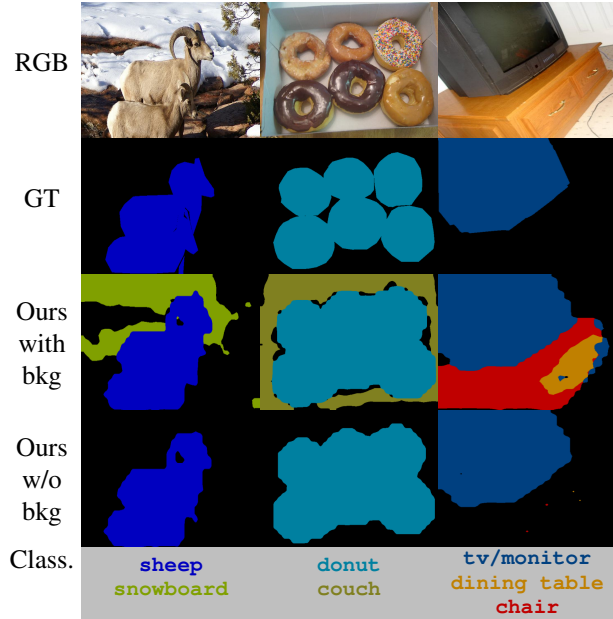


Figure 10. **Visual ablations of the impact of background detection**. We show examples from COCO Object (left, middle) and PASCAL VOC (right). We note 'bkg' our background refinement Sec. 3.5).

task of unsupervised object discovery (on VOC07 [18], VOC12 [19] and COCO20k [35] datasets with CorLoc metric) and unsupervised saliency detection in the 'multi' setup of [58] (all results are provided when using post-processing bilateral solver on the classic DUT-OMRON [70], DUTS-TEST [62] and ECSSD [54] datasets, with the mIoU metric). For more details on the evaluation setup, we refer to [58]. On both tasks, we observe on par or even better results than [58], therefore showing the quality of our foreground predictions learnt from CLIP.

## B. More qualitative results

In this section, we illustrate the benefits of our method through additional comparative qualitative results.

### B.1. Visual ablations

**Our spatial pooling.** We show more examples of the application of our method CLIP-DINOiser and compare it to MaskCLIP results in Fig. 9. We observe that in all cases, our pooling reduces the noise in the predictions and helps produce good-quality segmentation.

**Our background filtering.** By visualizing more results with and without the background refinement step in Fig. 10, we observe that the background refinement step helps remove uncertain segmentation such as the snow area (which was classified as 'snowboard') in the left image, or on the cabinet, which is not annotated in VOC (right image).

### B.2. In-the-wild examples

We show more in-the-wild examples in Fig. 11, where we compare CLIP-DINOiser against MaskCLIP. MaskCLIP produces very noisy masks, especially when multiple *false positive* text queries are considered (we define such false positive queries as prompt queries that appear in the final segmentation but are not depicted in the image). Instead, CLIP-DINOiser eliminates such false positive predictions and produces less noisy segmentation.

### B.3. Limitations

We discuss here the known failure modes of our method CLIP-DINOiser and visualize some in Fig. 12.

We first observe some of the biases of CLIP, which for instance produces similar features for 'train' and 'train tracks' (left image), likely due to their frequent co-occurrence across images. We have observed other instances of this bias, e.g., for 'boat' and 'sea' queries. Second, although CLIP-DINOiser can produce rather fine-grained segmentation (in terms of object sizes and classes), it can miss small or far-away objects as in Cityscapes (middle image). Finally, as with other open-vocabulary semantic segmentation methods, CLIP-DINOiser is not robust to the ambiguities of the text queries. The example from
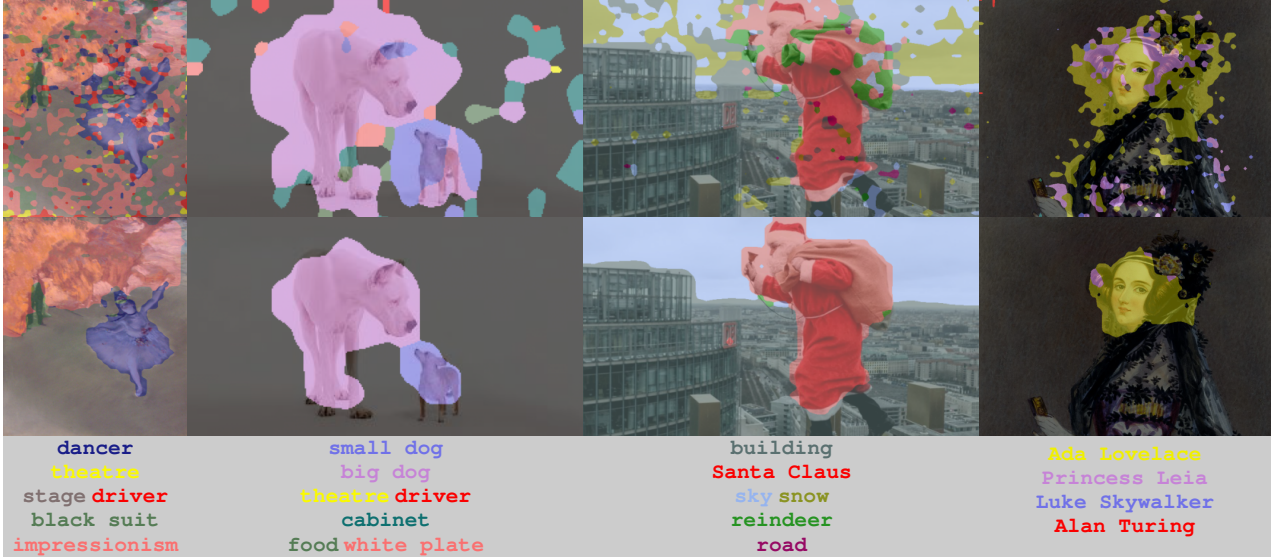
Figure 11. **In the wild comparative examples** between MaskCLIP (top) and CLIP-DINOiser (bottom). While MaskCLIP generates noisy masks when prompted with *false positive* classes our method is robust and produces cleaner masks.
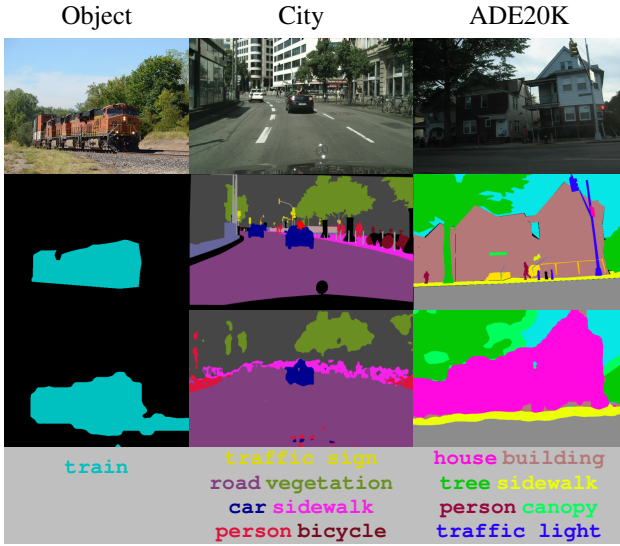


Figure 12. **Failure cases** of our method. From top to bottom: input RGB image, ground truth (GT) masks, masks predicted by CLIP-DINOiser, text prompts. We discuss these failure cases in Sec. B.3.

ADE20K (right image) is such a case, where 'house' is mistaken for 'building'. In our experiments, we observed multiple segmentation ambiguities and we believe that the redefinition of evaluation metrics could help address the issue. We stress that the current evaluation setup, which is taken directly from fully supervised settings, might be limiting in an open-vocabulary paradigm.