

# Diffusion Models for Open-Vocabulary Segmentation

Laurynas Karazija, Iro Laina, Andrea Vedaldi, and Christian Rupprecht

Visual Geometry Group, Department of Engineering Science, University of Oxford  
{laurynas,iro,vedaldi,chrisr}@robots.ox.ac.uk

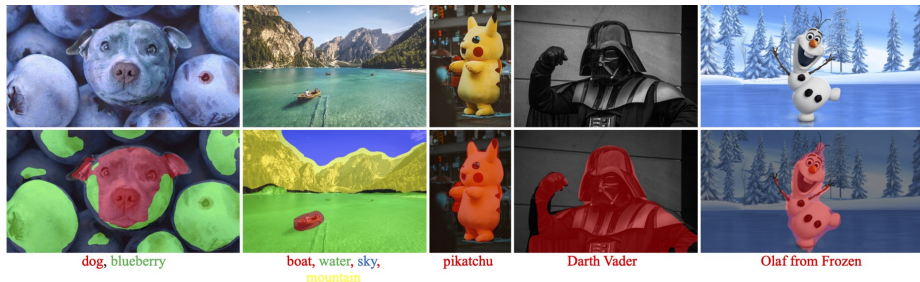
**Abstract.** Open-vocabulary segmentation is the task of segmenting anything that can be named in an image. Recently, large-scale vision-language modelling has led to significant advances in open-vocabulary segmentation, but at the cost of gargantuan and increasing training and annotation efforts. Hence, we ask if it is possible to use *existing* foundation models to synthesise on-demand efficient segmentation algorithms for specific class sets, making them applicable in an open-vocabulary setting without the need to collect further data, annotations or perform training. To that end, we present OVDiff, a novel method that leverages generative text-to-image diffusion models for unsupervised open-vocabulary segmentation. OVDiff synthesises support image sets for arbitrary textual categories, creating for each a set of prototypes representative of both the category and its surrounding context (background). It relies solely on pre-trained components and outputs the synthesised segmenter directly, without training. Our approach shows strong performance on a range of benchmarks, obtaining a lead of more than 5% over prior work on PASCAL VOC.

**Keywords:** Open-vocabulary Segmentation · Vision-language

## 1 Introduction

Open-vocabulary semantic segmentation is the task of segmenting images into regions matching several free-form textual categories. As the field of Computer Vision moves towards large-scale general-purpose models, open-vocabulary “foundation” models have similarly emerged. Yet, the development of ones suitable for dense localisation tasks such as semantic segmentation incurs both enormous training costs and requires expensive mask annotations. Instead, we show that the open-vocabulary segmentation task can be effectively tackled starting from a set of frozen foundation models, without requiring additional data or even fine-tuning.

In order to do so, we introduce OVDiff, a method that turns existing foundation models into a “factory” of image segmenters, *i.e.*, using foundation models to synthesise on-demand a segmenter for any new concepts specified in natural language. Thus, OVDiff can be used for open-vocabulary segmentation, where



**Fig. 1:** OVDiff is an open-vocabulary segmentation method that, given an image and a free-form set of class names, can segment any user-defined classes. It is fully automatic and does not require any further training.

it achieves state-of-the-art results in standard benchmarks. Moreover, once synthesised, the segmenters can be efficiently applied to any number of images and easily extended to new categories.

Specifically, segmenting an image using OVDiff can be done in three steps: *generation*, *representation*, and *matching*. Given a textual prompt, OVDiff uses an off-the-shelf text-to-image generator like StableDiffusion [47] to *generate* a support set of images. In the representation step, we use a feature extractor (that can be the same network as in the generation step) to extract feature prototypes that represent the textual category. Finally, we use simple nearest-neighbour *matching* scheme to segment the target image using the prototypes computed in the previous step.

This approach differs from prior work that largely approaches the problem in either of two ways. Starting from multi-modal representations (*e.g.*, CLIP [43]) to bridge vision and language, the first way relies on labelled data to fine-tune image-level representations for the segmentation task. Hence, in line with the zero-shot setting [9], these methods require costly dense annotations for some known categories while also extending the segmentation to unseen categories by incorporating language.

The second category of prior work [12, 34, 40, 46, 66, 68] observes that large-scale vision-language models such as CLIP have a limited understanding of the positioning of objects within an image and extend these models with additional grouping mechanisms for better localisation using only image-level captions, but no mask supervision. This, however, requires expensive additional contrastive training at scale. Additionally, most methods resort to heuristics to segment the background (*i.e.*, leave some pixels unlabelled), as it often cannot be described as a textual category. The usual approach is to threshold the similarities to all categories. Finding an appropriate threshold, however, can be challenging and may vary depending on the image, often resulting in imprecise object boundaries. Effectively handling the background remains an open issue.

Our three-step approach departs substantially from both of these schemes. We show that large-scale text-to-image generative models, such as StableDiffusion [47], can help bridge the vision-and-language gap without the need for annotations or costly training. Furthermore, diffusion models also produce latent spaces that are semantically meaningful and well-localised. This solves a second problem: multi-modal embeddings are difficult to learn and often suffer from ambiguities and differences in detail between modalities. Instead, our approach can use unimodal features for open-vocabulary segmentation, which offers several advantages. Firstly, as text-to-image generators encode a distribution of possible images, this offers a means to deal with intra-class variation and captures the ambiguity in textual descriptions. Secondly, the generative image models encode not only the visual appearance of objects but also provide contextual priors, which we use for direct background segmentation.

This work presents a simple framework that achieves state-of-the-art performance across open-vocabulary segmentation benchmarks. It combines several off-the-shelf pre-trained networks into a segmenter “factory” that segments images into arbitrary textual categories in three simple steps. OVDiff requires no additional data, mask supervision, nor fine-tuning. To summarise, we make the following core contributions: (1) We introduce a method to use pre-trained diffusion models for the task of open-vocabulary segmentation, that requires no additional data, mask supervision, or fine-tuning. (2) We propose a principled way to handle backgrounds by forming prototypes from contextual priors built into text-to-image generative models. (3) A set of additional techniques for further improving performance, such as multiple prototypes, category filtering and "stuff" filtering.

## 2 Related work

*Zero-shot open-vocabulary segmentation.* Open-vocabulary semantic segmentation is a relatively new problem and is typically approached in two ways. The first line of work poses the problem as “zero-shot”, *i.e.*, segmenting unseen classes after training on a set of observed classes with dense annotations. Early approaches [9, 14, 22, 30] explore generative networks to sample features using conditional language embeddings for classes. In [29, 65] image encoders are trained to output dense features that can be correlated with word2vec [38] and CLIP [43] text embeddings. Follow-up works [19, 21, 32, 69] approach the problem in two steps, predicting class-agnostic masks and aligning the embeddings of masks with language. IFSeg [70] generates synthetic feature maps by pasting CLIP text embeddings into a known spatial configuration to use as additional supervision. Different from our approach, all these works rely on mask supervision for a set of known classes.

The second line of work eliminates the need for mask annotations and instead aims to align image regions with language using only image-text pairs. This is largely enabled by recent advancements in large-scale vision-language models [43]. Some methods introduce internal grouping mechanisms such as

hierarchical grouping [46, 64, 66], slot-attention [68], or cross-attention to learn cluster centroids [33, 34]. Assignment to language queries is performed at group level. Another line of work [12, 40, 45, 75] aims to learn dense features that are better localised when correlated with language embeddings at pixel level. With the exception of [45, 64, 75], thresholding is often required to determine the background during inference. Alternatively, a curated list of background prompts can be used [45].

Our method falls into the second category. However, in contrast to prior work, we leverage a generative model to translate language queries to pre-trained image feature extractors without further training. We also segment the background directly, without relying on thresholding or curated list of background prompts. A closely related approach to ours is ReCO [51], where CLIP is used for image retrieval compiling a set of exemplar images from ImageNet for a given language query, which is then used for co-segmentation. In our method, the shortcoming of an image database is addressed by synthesising data on-demand. Furthermore, instead of co-segmentation, we leverage the cross-attention of the generator to extract objects. Instead of similarity of support images, we use diverse samples and both foreground and contextual backgrounds. Follow up works [3, 4] to OVDiff exchange contextual prior for backgrounds with compiling a database of prototypes.

*Diffusion models.* Diffusion models [26, 54, 55] are a class of generative methods that have seen tremendous success in text-to-image systems such as DALL-E [44], Imagen [48], and Stable Diffusion [47], trained on Internet-scale data such as LAION-5B [49]. The step-wise generative process and the language conditioning make pre-trained diffusion models attractive also for discriminative tasks. They have been recently used in few-shot classification [73], few-shot segmentation [2] and panoptic segmentation [67], and to generate pairs of images and segmentation masks [31]. However, these methods rely on dense manual annotations to associate diffusion features with the desired output.

Annotation-free discriminative approaches such as [17, 28, 57] use pre-trained diffusion models as zero-shot classifiers. DiffuMask [63] uses prompt engineering to synthesise a dataset of “known” and “unseen” categories and trains a closed-set segmenter with masks obtained from the cross-attention maps of the diffusion model. DiffusionSeg [35] uses DDIM inversion [55] to obtain feature maps and attention masks of object-centric images to perform unsupervised object discovery, but relies on ImageNet labels and is not open-vocabulary. Our approach also leverages the rich semantic information present in diffusion models for segmentation; unlike these methods, however, it is open-set and does not require further training.

*Unsupervised segmentation.* Our work is also related to unsupervised segmentation approaches. While early works relied on hand-crafted priors [15, 41, 62, 71, 72] later approaches leverage feature extractors such as DINO [11] and perform further analysis of these methods [23, 36, 50, 52, 53, 59–61]. Some approaches make use of generative methods, usually GANs, to separate images in foreground

and background layers [5–7, 13] or analyse latent structure to induce known foreground-background changes [37, 58] to synthesise a training dataset with labels. Some works explore interaction with different modalities such as optical flow [16, 27] or depth [8]. Largely focused on unsupervised saliency prediction, these methods are class-agnostic and do not incorporate language.

### 3 Method

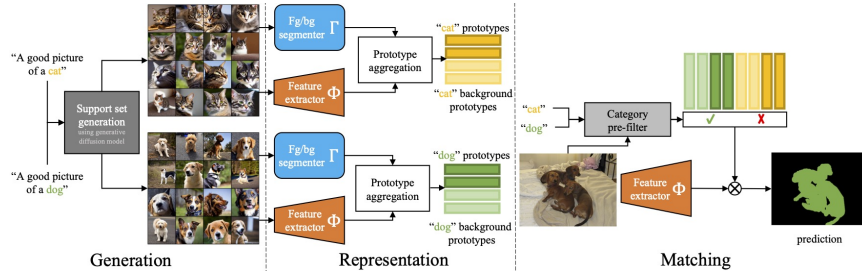
We present OVDiff, a method for open-vocabulary segmentation, *i.e.*, semantic segmentation of any category described in natural language. We achieve this goal in three steps: (1) we leverage text-to-image generative models to *generate* a set of images representative of the described category, (2) use these to ground *representations* from off-the-shelf pretrained feature extractors, and (3) *match* these against input image features to perform segmentation.

#### 3.1 OVDiff: Diffusion-based open-vocabulary segmentation

Our goal is to devise an algorithm which, given a new vocabulary of categories  $c_i \in \mathcal{C}$  formulated as natural language queries, can segment any image against it. Let  $I \in \mathbb{R}^{H \times W \times 3}$  be an image to be segmented. Let  $\Phi_v : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{H' \times W' \times D}$  be an off-the-shelf visual feature extractor and  $\Phi_t : \mathbb{R}^{d_t} \rightarrow \mathbb{R}^D$  a text encoder. Assuming that image and text encoders are aligned, one can achieve segmentation by simply computing a similarity function, for example, the cosine similarity  $s(\Phi_v(I), \Phi_t(c_i))$ , with  $s(x, y) = \frac{x^T y}{\|x\| \|y\|}$ , between the encoded image  $\Phi_v(I)$  and an encoding of a class label  $c_i$ . To meaningfully compare different modalities, image and text features must lie in a shared representation space, which is typically learned by jointly training  $\Phi_v$  and  $\Phi_t$  using image-text or image-label pairs [43].

We propose two modifications to this approach. First, we observe that it is better to compare representations of the *same* modality than across vision and language modalities. We thus replace  $\Phi_t(c_i)$  with a  $D$ -dimensional *visual* representation  $\bar{P}$  of class  $c_i$ , which we refer to as a *prototype*. In this case, the same feature extractor can be used for both prototypes and target images; thus, their comparison becomes straightforward and does not necessitate further training. Second, we propose utilising *multiple* prototypes per category instead of a single class embedding. This enables us to accommodate intra-class variations in appearance, and, as we explain later, it also allows us to exploit contextual priors, which in turn help to segment the background.

Our approach, thus, proceeds in three steps: (1) a set of support images is sampled based on vocabulary  $\mathcal{C}$ , (2) a set of prototypes  $\mathcal{P}$  is calculated, and (3) a set of images  $\{I_1, I_2 \dots\}$  is segmented against these prototypes. We observe that in practical applications, whole image collections are processed using the same vocabulary, as altering the set of target classes for individual images in an informed way would already require some knowledge of their contents. Steps (1) and (2) are, thus, performed very infrequently, and their cost is heavily amortised. Next, we detail each step.



**Fig. 2:** OVDiff overview. Prototype sampling: text queries are used to sample a set of support images which are further processed by a feature extractor and a segmenter forming positive and negative (background) prototypes. Segmentation: image features are compared against prototypes. The CLIP filter removes irrelevant prototypes based on global image contents.

### 3.2 Support set generation

To construct a set of prototypes, the first step of our approach is to sample a support set of images representative of each category  $c_i$ . This can be accomplished by leveraging pretrained text-conditional generative models. Sampling images from a generative model, as opposed to a curated dataset of real images, aligns well with the goals of open-vocabulary segmentation as it enables the construction of prototypes for *any* user-specified category or description, even those for which a manually labelled set may not be readily available (*e.g.*,  $c_i = \text{“donut with chocolate glaze”}$ ).

Specifically, for each query  $c_i$ , we define a prompt “A good picture of a  $\langle c_i \rangle$ ” and generate a small batch of  $N$  support images  $\mathcal{S} = \{S_1, S_2, \dots, S_N \mid S_n \in \mathbb{R}^{h \times w \times 3}\}$  of height  $h$  and width  $w$  using Stable Diffusion [47].

### 3.3 Representing categories

Naïvely, prototypes  $\bar{P}_{c_i}$  could be constructed by averaging all features across all images for class  $c_i$ . This is unlikely to result in good prototypes because not all pixels in the sampled images correspond to the class specified by  $c_i$ . Instead, we propose to extract the class prototypes as follows.

*Class prototypes.* Our approach generates two sets of prototypes, positive and negative, for each class. Positive prototypes are extracted from image regions that are associated with  $\langle c_i \rangle$ , while negative prototypes represent “background” regions. Thus, to obtain prototypes, the first step is segmenting the sampled images into foreground and background. To identify regions most associated with  $c_i$ , we use the fact that the layout of a generated image is largely dependent on the cross-attention maps of the diffusion model [25], *i.e.*, pixels attend more strongly to words that describe them. For a given word or description (in our case  $c_i$ ), one can generate a set of attribution maps  $\mathcal{A} = \{A_1, A_2, \dots, A_N \mid A_n \in \mathbb{R}^{h \times w}\}$ ,

corresponding to the support set  $\mathcal{S}$ , by summing the cross-attention maps across all layers, heads, and denoising steps of the network [56].

Yet, thresholding these attribution maps may not be optimal for segmenting foreground/background, as they are often coarse or incomplete, and sometimes only parts of objects receive high activation. To improve segmentation quality, we propose to optionally leverage an unsupervised instance segmentation method  $\Gamma$ . Unsupervised segmenters are not vocabulary-aware and may produce multiple binary object proposals. We denote these as  $\mathcal{M}_n = \{M_{nr} \mid M_{nr} \in \{0, 1\}^{hw}\}$ , where  $n$  indexes the support images and  $r$  indexes the object masks (including a mask for the background). We thus construct a promptable extension of  $\Gamma$  segmenter to select appropriate proposals for foreground and background: for each image, we select from  $\mathcal{M}_n$  the mask with the highest (lowest) average attribution as the foreground (background):

$$M_n^{\text{fg}} = \arg \max_{M \in \mathcal{M}_n} \frac{M^\top A_n}{M^\top M}, \quad M_n^{\text{bg}} = \arg \min_{M \in \mathcal{M}_n} \frac{M^\top A_n}{M^\top M}. \quad (1)$$

*Prototype aggregation.* We can compute prototypes  $P_n^g$  for foreground and background regions ( $g \in \{\text{fg}, \text{bg}\}$ ) as

$$P_n^g = \frac{(\hat{M}_n^g)^\top \Phi_v(S_n)}{m_n^g} \in \mathbb{R}^D, \quad (2)$$

where  $\hat{M}_n^g$  denotes a resized version of  $M_n^g$  that matches the spatial dimensions of  $\Phi_v(S_n)$ , and  $m_n^g = (\hat{M}_n^g)^\top \hat{M}_n^g$  counts the number of pixels within each mask. In other words, prototypes are obtained by means of an off-the-shelf pretrained feature extractor and computed as the average feature within each mask.

We refer to these as *instance* prototypes because they are computed from each image individually, and each image in the support set can be viewed as an instance of class  $c_i$ .

In addition to instance prototypes, we found it helpful to also compute *class-level* prototypes  $\bar{P}^g$  by averaging the instance prototypes weighted by their mask sizes as  $\bar{P}^g = \sum_{n=1}^N m_n^g P_n^g / \sum_{n=1}^N m_n^g$ .

Finally, we propose to augment the set of class and instance prototypes using  $K$ -Means clustering of the masked features to obtain *part-level* prototypes. We perform spatial clustering separately on foreground and background regions and take each cluster centroid as a prototype  $P_k^g$  with  $1 \leq k \leq K$ . The intuition behind this is to enable segmentation at the level of parts, support greater intra-class variability, and a wider range of feature extractors that might not be scale invariant.

We consider the union of all these feature prototypes:

$$\mathcal{P}^g = \bar{P}^g \cup \{P_n^g \mid 1 \leq n \leq N\} \cup \{P_k^g \mid 1 \leq k \leq K\} \quad (3)$$

for  $g \in \{\text{fg}, \text{bg}\}$ , and associate them with a single category.

We note that this process is repeated for each  $c_i \in \mathcal{C}$  and we hereby refer to  $\mathcal{P}^{\text{fg}}$  (and  $\mathcal{P}^{\text{bg}}$ ) as  $\mathcal{P}_{c_i}^{\text{fg}}$  ( $\mathcal{P}_{c_i}^{\text{bg}}$ ), *i.e.*, as the foreground (background) prototypes of class  $c_i$ .

Since  $\mathcal{P}_{c_i}^{\text{fg}}$  ( $\mathcal{P}_{c_i}^{\text{bg}}$ ) depend only on class  $c_i$ , they can be precomputed, and the set of classes can be dynamically expanded without the need to adapt existing prototypes.

### 3.4 Segmentation via prototype matching

To perform segmentation of any target image  $I$  given a vocabulary  $\mathcal{C}$ , we first extract image features using the same visual encoder  $\Phi_v$  used for the prototypes. The vocabulary is expanded with an additional background class  $\hat{\mathcal{C}} = \{c_{\text{bg}}\} \cup \mathcal{C}$ , for which the positive (*foreground*) prototype is the union of all *background* prototypes in the vocabulary:  $\mathcal{P}_{c_{\text{bg}}}^{\text{fg}} = \bigcup_{c_i \in \mathcal{C}} \mathcal{P}_{c_i}^{\text{bg}}$ . Then, a segmentation map can simply be obtained by matching dense image features to prototypes using cosine similarity. A class with the highest similarity in its prototype set is chosen:

$$M = \arg \max_{c \in \hat{\mathcal{C}}} \max_{P \in \mathcal{P}_c^{\text{fg}}} s(\Phi_v(I), P). \quad (4)$$

*Category pre-filtering.* To limit the impact of spurious correlations that might exist in the feature space of the visual encoder, we introduce a pre-filtering process for the target vocabulary given image  $I$ . Specifically, we leverage CLIP [43] as a strong open-vocabulary classifier but propose to apply it in a multi-label fashion to constrain the segmentation to the subset of categories  $\mathcal{C}' \subseteq \mathcal{C}$  that appear in the target image. First, we encode the target image and each category using CLIP. Any categories that do not score higher than  $1/|\mathcal{C}|$  are removed from consideration, that is we keep the subset  $\{P_{c'}^g \mid c' \in \mathcal{C}'\}$ ,  $g \in \{\text{fg}, \text{bg}\}$ . If more than  $\eta$  categories are present, then the top- $\eta$  are selected. We then form “multi-label” prompts as “ $\langle c_a \rangle$  and  $\langle c_b \rangle$  and . . .” where the categories are selected among the top scoring ones taking into account all  $2^\eta$  combinations. The best-scoring multi-label prompt determines the final list of categories to be used in Equation (4).

*“Stuff” filtering.* Occasionally,  $c_i$  might not describe a countable object category but an identifiable region in the image, *e.g.*, **sky**, often referred to as a “stuff” class. “Stuff” classes warrant additional consideration as they might appear as background in images of other categories, *e.g.*, **boat** images might often contain regions of **water** and **sky**. As a result, the process outlined above might sample background prototypes for one class that coincide with the foreground prototypes of another. To mitigate this issue, we introduce an additional filtering step to detect and reject such prototypes, when the full vocabulary, *i.e.*, the set of classes under consideration, is known. First, we only consider foreground prototypes for “stuff” classes. Additionally, any negative prototypes of “thing” classes with high cosine similarity with any of the “stuff” class prototypes are simply removed. In our experiments, we use ChatGPT [42] to automatically categorise a set of classes as “thing” or “stuff”.

## 4 Experiments

We evaluate OVDiff on the open-vocabulary semantic segmentation task. First, we consider different feature extractors and investigate how they can be grounded

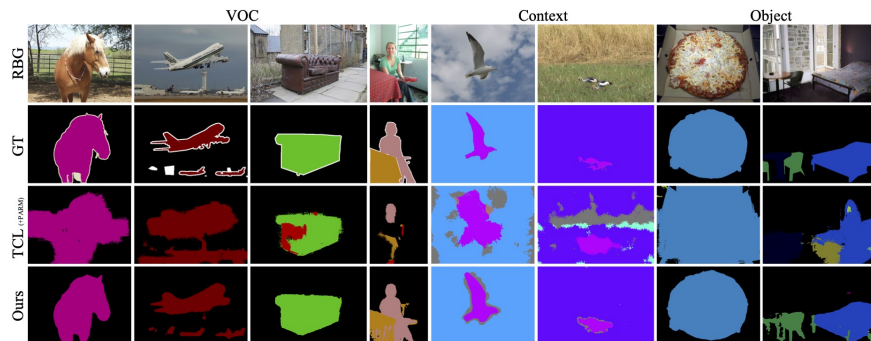


**Table 1:** Open-vocabulary segmentation. Comparison of our approach, OVDiff, to the state of the art (under the mIoU metric). Our results are an average of 5 seeds  $\pm\sigma$ . \*results from [12].

Method	Support Set	Further Training	VOC	Context	Object
ReCo* [51]	Real	✗	25.1	19.9	15.7
ViL-Seg [33]	✗	✓	37.3	18.9	-
MaskCLIP* [75]	✗	✗	38.8	23.6	20.6
TCL [12]	✗	✓	51.2	24.3	30.4
CLIPpy [45]	✗	✓	52.2	-	<u>32.0</u>
GroupViT [66]	✗	✓	52.3	22.4	-
ViewCo [46]	✗	✓	52.4	23.0	23.5
SegCLIP [34]	✗	✓	52.6	<u>24.7</u>	26.5
OVSegmentor [68]	✗	✓	53.8	20.4	25.1
CLIP-DIY [64]	✗	✗	<u>59.9</u>	-	31.0
<b>OVDiff</b> (-CutLER)	Synth.	✗	62.8	28.6	34.9
<b>OVDiff</b>	Synth.	✗	<b>66.3 <math>\pm</math> 0.2</b>	<b>29.7 <math>\pm</math> 0.3</b>	<b>34.6 <math>\pm</math> 0.3</b>
TCL [12] (+PAMR)	✗	✓	<u>55.0</u>	<u>30.4</u>	<u>31.6</u>
<b>OVDiff</b> (+PAMR)	Synth.	✗	<b>68.4 <math>\pm</math> 0.2</b>	<b>31.2 <math>\pm</math> 0.4</b>	<b>36.2 <math>\pm</math> 0.4</b>

by leveraging our approach. We then turn to comparisons of our method with prior work. We ablate the components of OVDiff, visualize the prototypes, and conclude with a qualitative comparison with prior works on in-the-wild images.

*Datasets and implementation details.* As the approach does not require further training of components, we only consider data for evaluation. Following prior work [66], to assess the segmentation performance, we report mean Intersection-over-Union (mIoU) on validation splits of PASCAL VOC (VOC) [20], PASCAL Context (Context) [39] and COCO-Object (Object) [10] datasets, with 20, 59, and 80 foreground classes, respectively. These datasets include a background class to reflect a realistic setting of non-exhaustive vocabularies. Context also contains both “things” and “stuff” classes. We also evaluate without background on VOC, Context, ADE20K [74], COCO-Stuff [10] and Cityscapes [18], with 20, 59, 150, 171, and 19 classes, respectively, but do not consider this a realistic setting as it relies on knowing which pixels cannot be described by a set of categories. Similar to [12, 66, 68], we employ a sliding window approach. We use two scales to aid with the limited resolution of off-the-shelf feature extractors with square window sizes of 448 and 336 and a stride of 224 pixels. We set the size of the support set to  $N = 32$ . For the diffusion model, we use Stable Diffusion v1.5; for unsupervised segmenter  $\Gamma$ , we employ CutLER [60].



**Fig. 3:** Qualitative results. OVDiff in comparison to TCL (+ PAMR). OVDiff provides more accurate segmentations across a range objects and stuff classes with well defined object boundaries that separate from the background well.

#### 4.1 Grounding feature extractors

Our method can be combined with *any* pretrained visual feature extractor for constructing prototypes and extracting image features. To verify this quantitatively, we experiment with various self-supervised ViT feature extractors (Tab. 2): DINO [11], MAE [24], and CLIP [43]. We also use SD as a feature extractor.

We find that SD performs the best, though CLIP and DINO also show strong performance based on our experiments on VOC. MAE shows the weakest performance, which may be attributed to its lack of semanticity [24]; yet it is still competitive with the majority of purposefully trained networks when employed as part of our approach. We find that taking *keys* of the second to last layer in CLIP yields better results than using patch tokens (CLIP token). As feature extractors have different training objectives, we hypothesise that their feature spaces might be complementary. Thus, we also consider an ensemble approach. In this case, the cosine distances formed between features of different extractors and respective prototypes are averaged. The combination of SD, DINO, and CLIP performs the best. We adopt this formulation for the main set of experiments.

#### 4.2 Comparison to existing methods

In Tab. 1, we compare our method with prior work that does not rely on manual mask annotation on three datasets: VOC, Context, Object. We include a brief overview of the methods in the supplement. We find that our method compares favourably, outperforming other methods in all settings. In particular, results on VOC show the largest margin, with more than 5% improvement over prior work.

We also consider a version of our method, OVDiff (-CutLER), that does not rely on an additional unsupervised segmenter  $\Gamma$ . Instead, the attention masks are thresholded. We observe that such a version of OVDiff has strong performance, outperforming prior work as well. CutLER is helpful, but not a critical component, and OVDiff performs strongly without it.

**Table 2:** Performance of OVDiff based on different feature extractors.

Feature Extractor	VOC
MAE	54.9
DINO	59.1
CLIP (tokens)	51.4
CLIP (keys)	61.8
SD	64.4
SD+CLIP+DINO	66.4

**Table 3:** Ablation of different components. Each component is removed in isolation, measuring the drop ( $\Delta$ ) in mIoU on VOC and Context datasets. Using SD features.

Configuration	VOC	$\Delta$	Context	$\Delta$
Full	64.4		29.4	
w/o bg prototypes	53.2	-11.2	28.9	-0.5
w/o category filter	54.4	-10.0	25.2	-4.2
w/o “stuff” filter	n/a		26.9	-2.5
w/o CutLER	60.4	-4.0	27.6	-1.8
w/o sliding window	62.2	-2.2	28.6	-0.8
only average $\bar{P}$	62.5	-1.9	28.4	-1.0

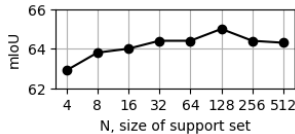
In the same table, we also combine our method with PAMR [1], the post-processing approach employed by TCL. We find that it improves results for our method, though improvements are less drastic since our method already yields better segmentation and boundaries.

Qualitative results are shown in Fig. 3. This figure highlights a key benefit of our approach: the ability to exploit contextual priors through the use of background prototypes, which in turn allows for the direct assignment of pixels to a background class. This improves segmentation quality because it makes it easier to differentiate objects from the background and to delineate their boundaries. In comparison, TCL predictions are very coarse and contain more noise.

### 4.3 Ablations

Next, we ablate the components of OVDiff on VOC and Context datasets. For these experiments, only SD is employed as a feature extractor. We remove individual components and measure the change in segmentation performance, summarising the results in Tab. 3. Our first observation is that background prototypes have a major impact on performance. When removing them from consideration, we instead threshold the similarity scores of the images with the foreground prototypes (set to 0.72, determined via grid search); in this case, the performance drops significantly, which again highlights the importance of leveraging contextual priors. On Context, the impact is less significant, likely due to the fact that the dataset contains “stuff” categories. Removing the *instance-* and *part-level* prototypes also negatively affects performance. Additionally, removing the category pre-filtering has a major impact. We hypothesize that this introduces spurious correlations between prototypes of different classes. On Context, “stuff” filtering is also important.

We again consider the importance of using an unsupervised segmenter, CutLER, for prototype mask extractions, using thresholding instead. We find this

**Fig. 4:** PascalVOC results with increasing support size  $N$ .

**Table 4:** Comparison with methods when background is excluded (decided by ground truth). OVDiff shows comparable performance to prior works despite only relying on pretrained feature extractors. \* result from [12].

Method	VOC-20	Context-59	ADE	Stuff	Cityscapes
CLIPpy	–	–	13.5	–	–
OVSegmentor	–	–	5.6	–	–
GroupViT*	<u>79.7</u>	23.4	9.2	15.3	11.1
MaskCLIP*	74.9	26.4	9.8	16.4	12.6
ReCo*	57.5	22.3	11.2	14.8	21.1
TCL	77.5	30.3	<b>14.9</b>	19.6	23.1
<b>OVDiff</b>	<b>80.9</b>	<b>32.9</b>	<u>14.1</u>	<b>20.3</b>	<b>23.4</b>

slightly reduces performance in this setting as well. Overall, background prototypes and pre-filtering contribute the most.

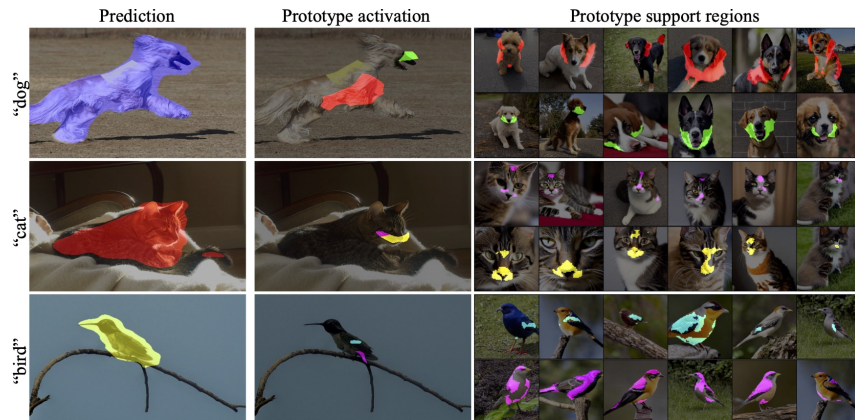
Finally, we measure the effect of varying the size of the support set  $N$  in Fig. 4. We find that OVDiff already shows strong performance even at a low number of samples for each query. With increasing the number of samples, the performance improves, saturating at around  $N = 32$ , which we use in our main experiments.

#### 4.4 Evaluation without background

One of the notable advantages of our approach is the ability to represent background regions via (negative) prototypes, leading to improved segmentation performance. Nevertheless, we hereby also evaluate our method under a different evaluation protocol adopted in prior work, which excludes the *background* class from the evaluation. We note that prior work often requires additional considerations to handle background, such as thresholding. In this setting, however, the background class is *not* predicted, and the set of categories, thus, must be exhaustive. As in practice, this is not the case, and datasets contain unlabelled pixels (or simply a background label), such image areas are removed from consideration. Consequently, less emphasis is placed on object boundaries in this setting. As in this setting the background prediction is invalid, we do not consider negative prototypes. This setting tests the ability of various methods to discriminate between different classes, which for OVDiff is inherent to the choice of feature extractors. Despite this, our method shows competitive performance across wide range of benchmarks Tab. 4.

#### 4.5 Explaining segmentations

We inspect how our method segments certain regions by considering which prototype from  $\mathcal{P}_c^{\text{fg}}$  was used to assign a class  $c$  to a pixel. Prototypes map to regions in the support set from where they were aggregated, *e.g.*, instances prototypes are associated with foreground masks  $M_n^{\text{fg}}$  and part prototypes with



**Fig. 5:** Analysis of the segmentation output by linking regions to samples in the support set. Left: our results for different classes. Middle: select color-coded regions “activated” by different prototypes for the class. Right: regions in the support set images corresponding to these (part-level) prototypes.

centroids/clusters. By following these mappings, a set of support image regions can be retrieved for each segmentation decision, providing a degree of explainability. Fig. 5 illustrates this for examples of `dog`, `cat`, and `bird` classes. For visualisation purposes, selected prototypes and corresponding regions are shown. On the left, we show the full segmentation result of each image. In the middle, we select regions that correlate best with certain class prototypes. On the right, we retrieve images from the support set and highlight where each prototype emerged. We find that meaningful part segmentation merges due to clustering the support image features, and similar regions are segmented by corresponding prototypes. However, sometimes region covered in the input image will not fully align with the whole prototype (*e.g.* `cat`’s face around the eyes or lower belly/tail of `bird`). Each segmentation is explained by precise regions in a small support set.

#### 4.6 In-the-wild

In Fig. 6, we investigate OVDiff on challenging in-the-wild images with simple and complex backgrounds. We compare with TCL+PAMR. In the first three images, both methods correctly detect the objects identified by the queries. OVDiff has small false positive “corgi” patches. TCL however misses large parts of the objects, such as most of the person, and parts of animal bodies. The distinction between the house and the bridge in the second image is also better with OVDiff. We also note that our segmentations sometimes have halos around objects. This is caused by upscaling the low-resolution feature extractor (SD in this case). The last two images contain challenging scenarios where both approaches struggle. The fourth image only contains similar objects of the same type. Both methods incorrectly identify plain donuts as either of the specified queries. OVDiff however correctly



**Fig. 6:** Qualitative comparison on challenging in-the-wild images with TCL, which struggles with object boundaries, missing parts of objects, or including surroundings. Our method has more appropriate boundaries and makes fewer errors overall, but does produce a small halo effect around objects due to the upscaling of feature extractors.

identifies chocolate donuts with varied sprinkles and separates all donuts from the background. In the final picture, the query “red car” is added, although no such object is present. The extra query causes TCL to incorrectly identify parts of the red bus as a car. Both methods incorrectly segment the gray car in the distance. However, overall, our method is more robust and delineates objects better despite the lack of specialized training or post-processing.

## 5 Conclusion

We introduce OVDiff, an open-vocabulary segmentation method that operates in two stages. First, given queries, support images are sampled and their features are extracted to create class prototypes. These prototypes are then compared to features from an inference image. This approach offers multiple advantages: diverse prototypes accommodating various visual appearances and negative prototypes for background localisation. OVDiff outperforms prior work on benchmarks, exhibiting fewer errors, effectively separating objects from background, and providing explainability through segmentation mapping to support set regions.

## Acknowledgements

Laurynas Karazija is supported by is supported by AIMS CDT EP/S024050/1. Iro Laina, Andrea Vedaldi, and Christian Rupprecht are supported by ERC-CoG UNION 101001212 and VisualAI EP/T028572/1.

*Ethics.* For further details on ethics, data protection, and copyright please see <https://www.robots.ox.ac.uk/~vedaldi/research/union/ethics.html>.

## References

1. Araslanov, N., Roth, S.: Single-stage semantic segmentation from image labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4253–4262 (2020)
2. Baranchuk, D., Voynov, A., Rubachev, I., Khrukov, V., Babenko, A.: Label-efficient semantic segmentation with diffusion models. In: International Conference on Learning Representations (2022)
3. Barsellotti, L., Amoroso, R., Baraldi, L., Cucchiara, R.: Fossil: Free open-vocabulary semantic segmentation through synthetic references retrieval. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1464–1473 (2024)
4. Barsellotti, L., Amoroso, R., Cornia, M., Baraldi, L., Cucchiara, R.: Training-free open-vocabulary segmentation with offline diffusion-augmented prototype generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3689–3698 (2024)
5. Benny, Y., Wolf, L.: Onegan: Simultaneous unsupervised learning of conditional image generation, foreground segmentation, and fine-grained clustering. In: European Conference on Computer Vision. pp. 514–530 (2020)
6. Bielski, A., Favaro, P.: Emergence of object segmentation in perturbed generative models. *Advances in Neural Information Processing Systems* (2019)
7. Bielski, A., Favaro, P.: Move: Unsupervised movable object segmentation and detection. In: *Advances in Neural Information Processing Systems* (2022)
8. Bowen, R.S., Tucker, R., Zabih, R., Snavely, N.: Dimensions of motion: Monocular prediction through flow subspaces. In: Proceedings of the International Conference on 3D Vision (3DV) (2022)
9. Bucher, M., Vu, T.H., Cord, M., Pérez, P.: Zero-shot semantic segmentation. In: *Advances in Neural Information Processing Systems* (2019)
10. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: *Computer vision and pattern recognition (CVPR), 2018 IEEE conference on. IEEE* (2018)
11. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9650–9660 (2021)
12. Cha, J., Mun, J., Roh, B.: Learning to generate text-grounded mask for open-world semantic segmentation from only image-text pairs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11165–11174 (2023)
13. Chen, M., Artières, T., Denoyer, L.: Unsupervised object segmentation by redrawing. *Advances in neural information processing systems* (2019)
14. Cheng, J., Nandi, S., Natarajan, P., Abd-Almageed, W.: Sign: Spatial-information incorporated generative network for generalized zero-shot semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9556–9566 (2021)
15. Cheng, M.M., Mitra, N.J., Huang, X., Torr, P.H.S., Hu, S.M.: Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **37**(3), 569–582 (2015)
16. Choudhury, S., Karazija, L., Laina, I., Vedaldi, A., Ruppel, C.: Guess What Moves: Unsupervised Video and Image Segmentation by Anticipating Motion. In: *British Machine Vision Conference (BMVC)* (2022)

17. Clark, K., Jaini, P.: Text-to-image diffusion models are zero shot classifiers. *Advances in Neural Information Processing Systems* (2024)
18. 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: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016)
19. Ding, J., Xue, N., Xia, G.S., Dai, D.: Decoupling zero-shot semantic segmentation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 11583–11592 (2022)
20. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html> (2012)
21. Ghiasi, G., Gu, X., Cui, Y., Lin, T.Y.: Scaling open-vocabulary image segmentation with image-level labels. In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVI*. pp. 540–557. Springer (2022)
22. Gu, Z., Zhou, S., Niu, L., Zhao, Z., Zhang, L.: Context-aware feature generation for zero-shot semantic segmentation. In: *Proceedings of the 28th ACM International Conference on Multimedia*. pp. 1921–1929 (2020)
23. Hamilton, M., Zhang, Z., Hariharan, B., Snavely, N., Freeman, W.T.: Unsupervised semantic segmentation by distilling feature correspondences. In: *International Conference on Learning Representations* (2022)
24. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 16000–16009 (2022)
25. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. *The Eleventh International Conference on Learning Representations* (2023)
26. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems* pp. 6840–6851 (2020)
27. Karazija, L., Choudhury, S., Laina, I., Rupperecht, C., Vedaldi, A.: Unsupervised Multi-object Segmentation by Predicting Probable Motion Patterns. In: *Advances in Neural Information Processing Systems* (2022)
28. Li, A.C., Prabhudesai, M., Duggal, S., Brown, E., Pathak, D.: Your diffusion model is secretly a zero-shot classifier. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 2206–2217 (2023)
29. Li, B., Weinberger, K.Q., Belongie, S., Koltun, V., Ranftl, R.: Language-driven semantic segmentation. In: *International Conference on Learning Representations* (2021)
30. Li, P., Wei, Y., Yang, Y.: Consistent structural relation learning for zero-shot segmentation. In: *Advances in Neural Information Processing Systems* (2020)
31. Li, Z., Zhou, Q., Zhang, X., Zhang, Y., Wang, Y., Xie, W.: Open-vocabulary object segmentation with diffusion models. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 7667–7676 (2023)
32. 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: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 7061–7070 (2023)
33. Liu, Q., Wen, Y., Han, J., Xu, C., Xu, H., Liang, X.: Open-world semantic segmentation via contrasting and clustering vision-language embedding. In: *Computer*



- Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XX. pp. 275–292. Springer (2022)
34. Luo, H., Bao, J., Wu, Y., He, X., Li, T.: SegCLIP: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. In: International Conference on Machine Learning. pp. 23033–23044. PMLR (2023)
  35. Ma, C., Yang, Y., Ju, C., Zhang, F., Liu, J., Wang, Y., Zhang, Y., Wang, Y.: Diffusionseg: Adapting diffusion towards unsupervised object discovery. arXiv preprint arXiv:2303.09813 (2023)
  36. Melas-Kyriazi, L., Rupprecht, C., Laina, I., Vedaldi, A.: Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8364–8375 (June 2022)
  37. Melas-Kyriazi, L., Rupprecht, C., Laina, I., Vedaldi, A.: Finding an unsupervised image segmenter in each of your deep generative models. In: International Conference on Learning Representations (2022)
  38. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* (2013)
  39. 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: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 891–898 (2014)
  40. 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: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19413–19423 (2023)
  41. Nguyen, T., Dax, M., Mummadi, C.K., Ngo, N., Nguyen, T.H.P., Lou, Z., Brox, T.: Deepusps: Deep robust unsupervised saliency prediction via self-supervision. In: *Advances in Neural Information Processing Systems* (2019)
  42. OpenAI: Introducing chatgpt. <https://openai.com/blog/chatgpt> (2023)
  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: International conference on machine learning. pp. 8748–8763 (2021)
  44. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 (2022)
  45. Ranasinghe, K., McKinzie, B., Ravi, S., Yang, Y., Toshev, A., Shlens, J.: Perceptual grouping in contrastive vision-language models. in 2023 ieee. In: *CVF International Conference on Computer Vision (ICCV)*. vol. 1, p. 3 (2023)
  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. *The Eleventh International Conference on Learning Representations* (2023)
  47. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10684–10695 (2022)
  48. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems* (2022)

49. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems* (2022)
50. Shin, G., Albanie, S., Xie, W.: Unsupervised salient object detection with spectral cluster voting. In: *CVPRW* (2022)
51. Shin, G., Xie, W., Albanie, S.: Reco: Retrieve and co-segment for zero-shot transfer. In: *Advances in Neural Information Processing Systems* (2022)
52. 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. *Proceedings of the British Machine Vision Conference (BMVC)* (November 2021)
53. Siméoni, O., Sekkat, C., Puy, G., Vobecký, A., Zablocki, É., Pérez, P.: Unsupervised object localization: Observing the background to discover objects. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 3176–3186 (2023)
54. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: *International Conference on Machine Learning*. pp. 2256–2265 (2015)
55. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: *International Conference on Learning Representations* (2021)
56. Tang, R., Liu, L., Pandey, A., Jiang, Z., Yang, G., Kumar, K., Stenetorp, P., Lin, J., Ture, F.: What the DAAM: Interpreting stable diffusion using cross attention. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2023)
57. Udandarao, V., Gupta, A., Albanie, S.: Sus-x: Training-free name-only transfer of vision-language models. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 2725–2736 (2023)
58. Voynov, A., Morozov, S., Babenko, A.: Object segmentation without labels with large-scale generative models. In: *International Conference on Machine Learning*. pp. 10596–10606 (2021)
59. Wang, X., Yu, Z., De Mello, S., Kautz, J., Anandkumar, A., Shen, C., Alvarez, J.M.: Freesolo: Learning to segment objects without annotations. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 14176–14186 (2022)
60. Wang, X., Girdhar, R., Yu, S.X., Misra, I.: Cut and learn for unsupervised object detection and instance segmentation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 3124–3134 (2023)
61. 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: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 14543–14553 (June 2022)
62. Wei, Y., Wen, F., Zhu, W., Sun, J.: Geodesic saliency using background priors. In: *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part III 12* (2012)
63. Wu, W., Zhao, Y., Shou, M.Z., Zhou, H., Shen, C.: Diffumask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 1206–1217 (2023)

64. Wysoczańska, M., Ramamonjisoa, M., Trzciński, T., Siméoni, O.: Clip-diy: Clip dense inference yields open-vocabulary semantic segmentation for-free. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1403–1413 (2024)
65. Xian, Y., Choudhury, S., He, Y., Schiele, B., Akata, Z.: Semantic projection network for zero-and few-label semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8256–8265 (2019)
66. Xu, J., De Mello, S., Liu, S., Byeon, W., Breuel, T., Kautz, J., Wang, X.: Groupvit: Semantic segmentation emerges from text supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18134–18144 (2022)
67. Xu, J., Liu, S., Vahdat, A., Byeon, W., Wang, X., De Mello, S.: Open-vocabulary panoptic segmentation with text-to-image diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2955–2966 (2023)
68. 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: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2935–2944 (2023)
69. Xu, M., Zhang, Z., Wei, F., Lin, Y., Cao, Y., Hu, H., Bai, X.: A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In: European Conference on Computer Vision. pp. 736–753 (2022)
70. Yun, S., Park, S.H., Seo, P.H., Shin, J.: Ifseg: Image-free semantic segmentation via vision-language model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2967–2977 (2023)
71. Zeng, Y., Zhuge, Y., Lu, H., Zhang, L., Qian, M., Yu, Y.: Multi-source weak supervision for saliency detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
72. Zhang, J., Zhang, T., Dai, Y., Harandi, M., Hartley, R.I.: Deep unsupervised saliency detection: A multiple noisy labeling perspective. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 9029–9038 (2018)
73. Zhang, R., Hu, X., Li, B., Huang, S., Deng, H., Qiao, Y., Gao, P., Li, H.: Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15211–15222 (2023)
74. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralla, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 633–641 (2017)
75. Zhou, C., Loy, C.C., Dai, B.: Extract free dense labels from clip. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII. pp. 696–712. Springer (2022)