Retrieve and Segment: Are a Few Examples Enough to Bridge the Supervision Gap in Open-Vocabulary Segmentation?

Tilemachos Aravanis[1] Vladan Stojnić[1] Bill Psomas[1] Nikos Komodakis[2 3 4] Giorgos Tolias[1]
[1]VRG, FEE, Czech Technical University in Prague
[2]University of Crete [3]Archimedes, Athena RC [4]IACM-FORTH

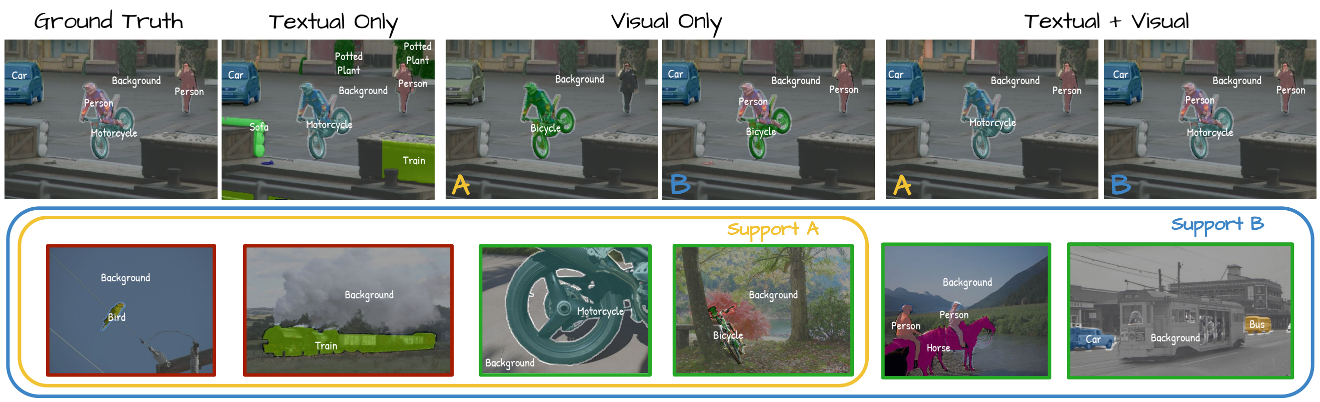

Figure 1. **Open-vocabulary segmentation (OVS) results.** We compare three settings: (i) *textual-only* support (zero-shot OVS), (ii) a simplified version of RnS using *visual-only* support, and (iii) the full RnS combining *textual+visual* support. Textual support: class name or description. Visual support: a small set of pixel-annotated images for some classes. Initially, visual support includes images in A and is later expanded to B, with $A \subseteq B$. Text-only support often yields ambiguous predictions (rider as motorcycle, background hallucinations). Visual-only support struggles when some classes lack support (person, car) and can confuse similar objects (motorcycle, bicycle) even when all classes have support. By *retrieving* information from images relevant to the test image and combining with the textual support, RnS is robust under missing visual support for some classes and achieves accurate *segmentation*.

Abstract

Open-vocabulary segmentation (OVS) extends the zero-shot recognition capabilities of vision–language models (VLMs) to pixel-level prediction, enabling segmentation of arbitrary categories specified by text prompts. Despite recent progress, OVS lags behind fully supervised approaches due to two challenges: the coarse image-level supervision used to train VLMs and the semantic ambiguity of natural language. We address these limitations by introducing a few-shot setting that augments textual prompts with a support set of pixel-annotated images. Building on this, we propose a retrieval-augmented test-time adapter that learns a lightweight, per-image classifier by fusing textual and visual support features. Unlike prior methods relying on late, hand-crafted fusion, our approach performs learned, per-query fusion, achieving stronger synergy between modalities. The method supports continually expanding support sets, and applies to fine-grained tasks such as personalized segmentation. Experiments show that we significantly narrow the gap between zero-shot and supervised segmentation while preserving open-vocabulary ability. Project Page

1. Introduction

Semantic segmentation traditionally relies on fully supervised models trained on dense pixel-level annotations within a fixed set of categories [9, 43, 85]. While this approach yields accurate, well-localized masks, it does not scale; collecting pixel-level annotations is costly, and models cannot recognize categories unseen during training.

Open-vocabulary segmentation (OVS) builds on the zero-shot recognition capabilities of contrastively trained vision–language models (VLMs) [24, 51, 82]. Trained on large image–text datasets [8, 56], VLMs learn a shared embedding space where images and text are directly comparable, enabling recognition of arbitrary categories specified at test time via text prompts or class names. Extending this ability from image-level to pixel-level prediction has driven rapid progress in OVS [32, 33, 63, 73]. However, a substantial gap remains to fully supervised models [9]; recent improvements show signs of plateauing [55].

Two key challenges underlie this gap: (i) the mismatch between image-level supervision used to train VLMs and the fine-grained predictions required for segmentation, and

(ii) the semantic ambiguity of natural language supervision. While language enables open-vocabulary recognition, it often lacks the precision needed for pixel-level tasks.

We address these challenges by introducing a few-shot setting that supplements the textual support of class names with a small support set of visual examples, *i.e*. images with pixel-level annotation. We aim to *bridge the gap* between zero-shot OVS and supervised segmentation, while preserving the open-vocabulary predictions. We propose a retrieval-augmented test-time adapter that trains a lightweight classifier per test image. Inspired by retrieval-based methods [29, 71], our approach retrieves relevant visual support examples and fuses them with textual support to construct test-time training data. Unlike previous work [2, 18] relying on hand-crafted fusion, our method performs a learned per-image fusion of textual and visual prototypes, enabling strong synergy between modalities, as shown in Figure 1. Importantly, we store only a compact set of visual prototypes from the support images, keeping the memory footprint minimal, while our test-time training requires less than a second on an NVIDIA $A100$ GPU[1].

This design is enabled by the strong, generalizable features of modern VLMs trained at large-scale [8, 56]. These rich features allow us to avoid retraining the backbone and instead steer predictions with a lightweight classifier trained on only a few visual examples. By leveraging these robust embeddings, our approach achieves efficient adaptation while preserving the open-vocabulary nature of the task.

Our method, called *Retrieve and Segment (*RNS*)*, can handle diverse real world settings, from textual and visual support for all classes, to partial support for some classes, where either a textual description is not straightforward to obtain or visual examples are not yet available. Moreover, it is compatible with a dynamic continually changing setting where new visual examples can be added to the support set at any time, without sacrificing the open-vocabulary nature. Due to its simplicity and its dynamic adaptability, our approach is easily applicable to segmentation of particular objects, *i.e*. the so called personalized segmentation [64, 84]. In summary, our contributions are:

- We investigate multiple few-shot settings for open-vocabulary segmentation, enriching textual prompts with pixel-annotated visual examples.
- We introduce RNS, a retrieval-augmented, test-time adapter that learns to fuse textual and visual support more effectively than prior approaches.
- RNS significantly reduces the performance gap between zero-shot and fully supervised segmentation, while maintaining open-vocabulary generalization.
- RNS supports dynamic support expansion in continually evolving environments, and adapts seamlessly to fine-grained tasks such as personalized segmentation.

[1] See supplementary material for runtime details.

2. Related work

Open-vocabulary segmentation is performed by leveraging vision–language models (VLMs) that align images and text in a shared space [24, 51, 82]. The fixed classifier is replaced by a text encoder and matches patch features to text features, but vanilla VLMs pre-trained with image-level supervision struggle with dense localization [4, 88]. There are three lines of work: (i) *Training VLMs for segmentation*: either weakly supervised with masks without labels [13, 17] or image–caption pairs [7, 44, 47, 53, 76, 77], or fully supervised with pixel annotations [11, 25, 35, 37, 38, 75, 79, 86]. However, this approach hinders the open-vocabulary performance outside of the training domain [11, 63], as shown in datasets like MESS [3]. (ii) *Training-free VLM tweaks*: modified inference processes to boost spatial sensitivity, *e.g*. removing the final attention layer [88] or residual connections and feedforward networks [32], with further variants [4, 32, 67]. (iii) *VLM+VM hybrids*: combining VLM semantics with the localization of vision models (VMs) [27, 30, 33, 58, 63, 72, 73, 83]. Methods typically localize objects with DINO [6, 49] or SAM [31, 54], then classify the localized regions in an open-vocabulary manner using VLMs. Despite progress, OVS still trails task-specific, fully supervised models.

Few-shot segmentation learns on base classes and adapts to novel classes from few (support) labeled examples. Meta-learning and prototypical methods perform episodic $1-$ or $N-$way training, create per-class prototypes and classify via similarity [45, 57, 60, 65, 68], typically assuming a closed world. A recent generalized setting evaluates on both base and novel classes [19, 20, 41, 66], but still requires abundant pixel-level annotations for base classes and does not leverage VLMs. Close to our setting, a concurrent work Power-of-One [21] introduces one-shot per class fine-tuning of text embeddings and specific backbone layers. They require access to raw images, while we operate on pre-extracted features, and fine-tune internal VLM [36] layers for each new class set, which is not as light-weight as our test-time adapter. Beyond VLMs, CAT-SAM [74] explores few-shot adaptation of SAM [31] via conditional tuning with lightweight adapters, but does not combine visual and textual support. COSINE [42] unifies open-vocabulary (text-prompted) and in-context (image-prompted) segmentation by training a decoder on top of frozen foundation models to handle multi-modal prompts. For open vocabulary semantic segmentation with many categories the model is only evaluated with unimodal prompts though.

Retrieval augmentation in segmentation. Retrieval-augmented models for prediction and generation have shown strong performance by dynamically expanding their knowledge base [22, 40, 52, 80]. Following this paradigm in semantic segmentation, recent work enhances in-context scene understanding of vision encoders [1, 50, 62]. FreeDA [2] is conceptually related to RnS, but relies on generated visual examples. At test time, textual class features are expanded into visual counterparts via retrieval, forming a non-parametric visual classifier, which is then combined with a standard zero-shot textual classifier. For fair comparison, we adapt this method to use real support images instead of generated ones. Closely related to our work is kNN-CLIP [18], which enhances open-vocabulary semantic segmentation by leveraging a memory-efficient support set of class vectors derived from pixel-annotated images. At test time, it assigns labels to image regions based on the similarity to their k nearest neighbors in the support set. This approach outperforms continual learning baselines, demonstrating that a dynamic support set can incorporate visual examples of new classes without forgetting previously learned ones. However, while kNN-CLIP claims to expand the model's vocabulary to arbitrary class sets, its performance remains limited to classes for which annotated examples are available.

Test-time adaptation (TTA) for VLMs has grown rapidly, but many methods operate in batch/transductive or streaming modes [15, 28, 34] and carry unrealistic assumptions, such as class-complete batches or i.i.d. streams, which compromises zero-shot robustness [81]. In contrast, single-image TTA adapts per sample: TPT [59] and ZERO [16] perform optimization or predictions over augmentations. For segmentation, single-image TTA is explored with self-supervised objectives [23], and OVS-specific TTA is just emerging, proposing adaptation layers for VLM-based segmenters at test time [48].

3. Method

3.1. Task formulation

Given image $I \in \mathbb{R}^{H\times W\times 3}$ and set $\mathcal{C}$ of C semantic classes, we aim to assign each pixel of I to one of the classes. The class set is arbitrary and defined at test time. Therefore, the task is referred to as *open-vocabulary segmentation (OVS)*.

Each class is specified either by a textual example, *e.g.* a class name, or by a small set of visual examples in the form of raw images with pixel-level annotations, like in a few-shot setting. We refer to these as the *textual support set* and *visual support set*, respectively. Typically, a textual example is available for every class. However, visual examples may be missing initially, but their number and diversity can increase over inference time, reflecting a continually expanding open-world scenario. In rare cases, textual examples may also be absent, *e.g.* for novel categories with unknown names or in specialized domains such as medical imaging or remote sensing, where naming is non-trivial.

We consider the following settings: (i) *full-support*: every class in $\mathcal{C}$ has a class name and at least one annotated support image. (ii) *partial-visual-support*: some classes lack visual examples, but all have class names. (iii) *partial-textual-support*: some classes lack class names, but all have visual examples. (iv) *only-textual-support*: class names are available for all classes in $\mathcal{C}$, but no visual support images are provided, *i.e.* the commonly studied *zero-shot* segmentation setting.

In contrast to zero-shot segmentation, all other settings have visual support examples available, and our goal is to leverage them to improve segmentation, while preserving the ability to operate with an open-vocabulary.

We denote a support image by I^i, a test (query) image by I^q, while we drop the superscript when referring to images in general and we adopt consistent notation for other variables as introduced in the following sections.

3.2. Zero-shot segmentation with VLMs

Image $I \in \mathbb{R}^{H\times W\times 3}$, processed by the vision encoder of a VLM, is mapped to a *patch-level feature matrix* $X \in \mathbb{R}^{n\times d}$. Here $n = h \times w$ corresponds to the flattened $h \times w$ patch grid produced by a vision transformer (ViT), and d denotes the feature dimension. The *patch feature* at position $j \in 1, \dots, n$ is denoted by $\mathbf{x}_j \in \mathbb{R}^d$.

Each class name $c \in \mathcal{C}$, processed by the VLM's text encoder, is mapped to a *textual class feature* $\mathbf{t}_c \in \mathbb{R}^d$. Both textual and visual features are normalized to unit length.

The patch-level prediction map is denoted by $\hat{P} \in [0,1]^{n\times C}$ with elements $\hat{P}_{jc}$ denoting the probability of patch j for class c and computed by the dot product similarity between patch feature and textual class feature as

$$\hat{P}_{jc} = s_{\mathcal{C}}(\mathbf{x}_j^\top \mathbf{t}_c), \tag{1}$$

where $s_{\mathcal{C}}(\cdot)$ is the softmax over class set $\mathcal{C}$. Then, low-resolution prediction $\hat{P}$ is reshaped and upsampled to the full-resolution prediction $\hat{Y} \in [0,1]^{H\times W\times C}$, and segmentation is obtained by the $\arg\max$ of $\hat{Y}$ over classes.

3.3. Full textual and visual support

In the following, we describe how to process examples in the textual and visual support sets to construct two support features sets, namely the *visual support feature set* and the *fused support feature set* combining visual and textual information. During inference, the elements of the support feature sets that are the most relevant to a test image are used to train a lightweight patch-level linear classifier, which is then applied to the features of the same test image. Overview of this process is shown in Figure 2.

Visual support features. Given a support image I^i and its ground-truth segmentation labels $Y^i \in \{0,1\}^{H\times W\times C}$ at full resolution, its patch-level feature matrix $X^i \in \mathbb{R}^{n\times d}$ is extracted. Then, the ground-truth pixel-level labels Y^i are down-sampled and reshaped to obtain patch-level labels, which are not binary anymore due to interpolation and are subsequently L_1-normalized per class (column) to obtain

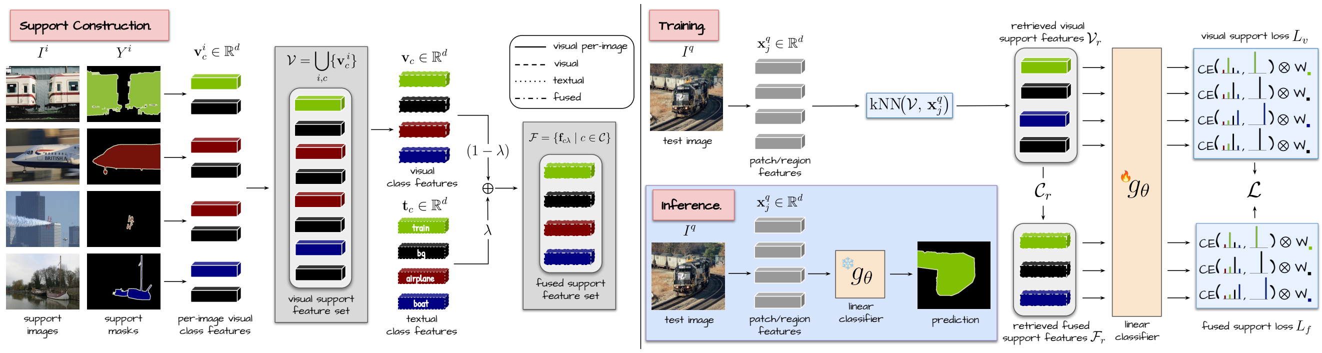

Figure 2. **Overview of RNS when *full textual and visual support* is available.** Having access to a set of pixel-level annotated images, *per-image visual class features* $\mathbf{v}_c^i$ are extracted. These features are then aggregated by class to form *visual class features* $\mathbf{v}_c$, which are combined with *textual class features* $\mathbf{t}_c$, through a mixing coefficient λ, to produce *fused class features* $\mathbf{f}_{c\lambda}$. During test-time training, a test-image-relevant subset of *visual support features* and *fused class features*, along with their class labels, are used to train a lightweight linear classifier g_θ using cross-entropy loss. Each training sample is weighted with a class relevance weight w_c (*e.g.* $w_{\blacksquare}$ for bg). At inference, this classifier, trained per test image, is applied to patch-level features $\mathbf{x}_j^q$ to generate segmentation predictions. When SAM is available, patch-level features are replaced by region-level features $\mathbf{x}_r^q$ for improved accuracy.

$P^i \in [0,1]^{n\times C}$. These labels are used to pool the patch features into *per-image visual class features* $\mathbf{v}_c^i$ for classes $\mathcal{C}_i$ present in image i by

$$\mathbf{v}_c^i = \sum_{j=1}^{n} P_{jc}^i \, \mathbf{x}_j^i. \tag{2}$$

Assuming M available support images, the union of per-image visual class features is given by

$$\mathcal{V} = \bigcup_{i=1}^{M} \{\mathbf{v}_c^i : c \in \mathcal{C}_i\}, \tag{3}$$

and referred to as *visual support feature set*, which is used to support the test-time adaptation process.

Fused support features. We aim to leverage the semantic information in the textual class features. However, due to the modality gap between visual and textual features in VLMs [39], and because our goal is to classify image patch-level features, combining them directly with visual support features does not perform well, as confirmed by our experiments. Thus, for class c, we create a *fused class feature*

$$\mathbf{f}_{c\lambda} = \lambda\, \mathbf{t}_c + (1-\lambda)\, \mathbf{v}_c, \qquad \lambda \in [0,1], \tag{4}$$

by combining textual class feature $\mathbf{t}_c$ and *visual class feature* $\mathbf{v}_c$ obtained by aggregating all per-image visual class features for class c across the visual support set by

$$\mathbf{v}_c = \sum_{i \in \mathcal{I}_c} \mathbf{v}_c^i, \tag{5}$$

where $\mathcal{I}_c$ is the set of support images that contain class c. The fusion process is performed for a set of mixing coefficients $\Lambda \subseteq [0,1]$ to capture diverse and complementary information from both modalities, yielding multiple fused class features per class. We denote the *fused support feature set* by $\mathcal{F} = \{\mathbf{f}_{c\lambda} | c \in \mathcal{C}, \lambda \in \Lambda\}$, which is used to support the test-time adaptation process.

Support feature set maintenance. For each support image, we extract its feature matrix and pool patch features into per-image visual class features (2). If a new support image arrives, we update the visual support feature set $\mathcal{V}$ (3), the visual class features (5), the fused class features (4), and consequently the fused support feature set $\mathcal{F}$. Therefore, our support sets are dynamically expandable in a straightforward and efficient manner, allowing to operate in a continually evolving open-world scenario.

Test-time adaptation. For test image I^q with feature matrix $X^q \in \mathbb{R}^{n\times d}$, and the feature of patch j denoted by $\mathbf{x}_j^q$, we train linear classifier $g_\theta : \mathbb{R}^d \to \mathbb{R}^C$, specifically for I^q, to project features to class probabilities. We leverage the relevant elements of the two support feature sets.

To this end, we retrieve the k nearest neighbors of each test image patch features from the visual support feature set and unite them into the *retrieved visual support feature set*

$$\mathcal{V}_r = \bigcup_{j=1}^{n} \mathrm{kNN}(\mathcal{V}, \mathbf{x}_j^q). \tag{6}$$

We define the *visual support loss* by

$$L_v = \sum_{\mathbf{v} \in \mathcal{V}_r} w_{l(\mathbf{v})} \; \mathrm{CE}\big(g_\theta(\mathbf{v}), \mathbf{1}_{l(\mathbf{v})}\big), \tag{7}$$

where CE is the cross entropy loss, $l(\mathbf{v})$ provides the label of feature $\mathbf{v}$ (a per-image visual class feature), $\mathbf{1}_c$ is the one-hot encoding of class c, and w_c is a *class relevance weight* for class c. This loss encourages the classifier to assign high probability to the correct class for each retrieved support feature. The retrieved set is expected to include features corresponding to classes present in the test image, particularly those originating from visually similar images.

The class relevance weights are used to suppress the impact of retrieved features irrelevant to the test image. Weight w_c is estimated via the similarity between the image-level feature $\mathbf{x}^q \in \mathbb{R}^d$ and the corresponding textual class feature, followed by softmax:

$$w_c = s_{\mathcal{C}}\left((\mathbf{x}^q)^\top \mathbf{t}_c \right), \tag{8}$$

with $\mathbf{x}^q$ given by global average pooling

$$\mathbf{x}^q = \frac{1}{n} \sum_{j=1}^{n} \mathbf{x}_j^q. \tag{9}$$

We utilize textual support, via the fused support set, by training the classifier on the fused class features (denoted by $\mathcal{F}_r$ and referred to as *retrieved fused support feature set*) of the classes (denoted by $\mathcal{C}_r$) that appear in retrieved visual support feature set $\mathcal{V}_r$. This is performed via the *fused support loss*:

$$L_f = \sum_{c \in \mathcal{C}_r} w_c \sum_{\lambda \in \Lambda} \mathrm{CE}\big(g_\theta(\mathbf{f}_{c\lambda}), \mathbf{1}_c\big). \tag{10}$$

We observe that using multiple mixing coefficients Λ improves performance compared to using a single coefficient. The total loss is given by $\mathcal{L} = L_v + \beta_f L_f$. After training, classifier g_θ is applied to test image patch features to obtain patch-level predictions. Then, this low resolution prediction map is upsampled to the original image resolution to obtain the final segmentation map.

3.4. Partial visual support

We present an additional loss, while other components remain as in Section 3.3.

Classes without visual support. The set of classes supported by their class name but with no image examples are denoted as $\mathcal{C}_d$. Given the absence of visual support for these classes, we cannot compute their visual class feature $\mathbf{v}_c$ (5), and consequently, their fused class feature $\mathbf{f}_{c\lambda}$ (4).

To circumvent this, we exploit the test image to identify whether any of the classes in $\mathcal{C}_d$ are present in it via zero-shot prediction $\hat{P}^q$ (1). Predictions in $\hat{P}^q$ are converted to one-hot vectors by assigning each patch to its most probable class, and the result is L_1-normalized per class (column) to obtain $\tilde{P}^q \in [0,1]^{n \times C}$. We denote by $\mathcal{C}_q \subseteq \mathcal{C}$ the set of classes assigned to at least one patch according to $\tilde{P}^q$. Then, we define the visual class feature by pooling the patch features according to these pseudo-labels:

$$\mathbf{v}_c = \sum_{j=1}^{n} \tilde{P}_{jc}^q \mathbf{x}_j^q, \quad c \in \mathcal{C}_d \cap \mathcal{C}_q. \tag{11}$$

Note[2] that we perform this only for classes in $\mathcal{C}_q \cap \mathcal{C}_d$ whose visual class feature cannot be obtained by (5). This allows us to perform the modality fusion by (4) for all classes.

Fused feature pseudo-labeling. Fused class feature $\mathbf{f}_{c\lambda}$ is derived through visual class features that use pseudo-labeling. Therefore, its association with class c is uncertain. To circumvent this, we pseudo-label those fused class features. The predicted probability distribution for $\mathbf{f}_{c\lambda}$ denoted by $\hat{\mathbf{p}}_{c\lambda} \in [0,1]^C$ has its c' element, associated with class c', estimated by $\mathbf{f}_{c\lambda}^\top \mathbf{t}_{c'}$ followed by softmax over all classes.

Extended test-time adaptation loss. We introduce a *pseudo-label loss* term exploiting such pseudo-labels:

$$L_p = \sum_{c \in \mathcal{C}_d \cap \mathcal{C}_q} w_c \sum_{\lambda \in \Lambda} \mathrm{KL}\big(\hat{\mathbf{p}}_{c\lambda} \,\|\, g_\theta(\mathbf{f}_{c\lambda})\big), \tag{12}$$

where KL is the Kullback-Leibler divergence.

The total loss is given by $\mathcal{L} = L_v + \beta_f L_f + \beta_p L_p$. Note that classes with visual support are handled in the second loss term, while the rest are either handled in the third loss term ($\mathcal{C}_r \cap \mathcal{C}_d = \emptyset$) or ignored if they are not predicted to be present in the test image.

3.5. Partial textual support

We modify *fused support features* of classes without textual support, while other components remain as in Section 3.3.

Classes without textual support. When class names are absent, we cannot compute textual class features $\mathbf{t}_c$ or their fused counterparts $\mathbf{f}_{c\lambda}$ (4). Excluding these classes from the loss introduces bias toward classes with both supports. Instead, we replace missing textual class features with the average textual class feature across classes with an available class name. This provides a neutral semantic prior, ensuring that all classes equivalently participate in the loss.

If textual support is missing for all classes, no textual class features can be formed. In this case, we simply set $\Lambda = \{0\}$ and class relevance weights w_c are set equal to 1 for all classes, effectively reducing to our *w/o text* baseline.

3.6. Region-proposal predictions

Our method so far assumes patch-level feature extraction and predictions, *i.e.* $\mathbf{x}_j^q$ for patch j. When region proposals $S \in \{0,1\}^{H \times W \times R}$ for R binary masks are available for the test image, *e.g.* from SAM [31, 54], we proceed as follows. We downsample each mask to patch resolution and apply L_1 normalization per region, yielding $\bar{S} \in [0,1]^{n \times R}$.

We then pool patch features into region-level features by

$$\mathbf{x}_r^q = \sum_{j=1}^{n} \bar{S}_{jr} \mathbf{x}_j^q, \quad r = 1, \ldots, R, \tag{13}$$

where r denotes the region index. Finally, we assign labels at the region level and map them to the corresponding mask regions at the full image resolution to obtain the final segmentation map.

[2] Notation $\mathbf{v}_c$ used for both types of visual class features; (5) used for classes with visual-textual support, while (11) for those with only textual.

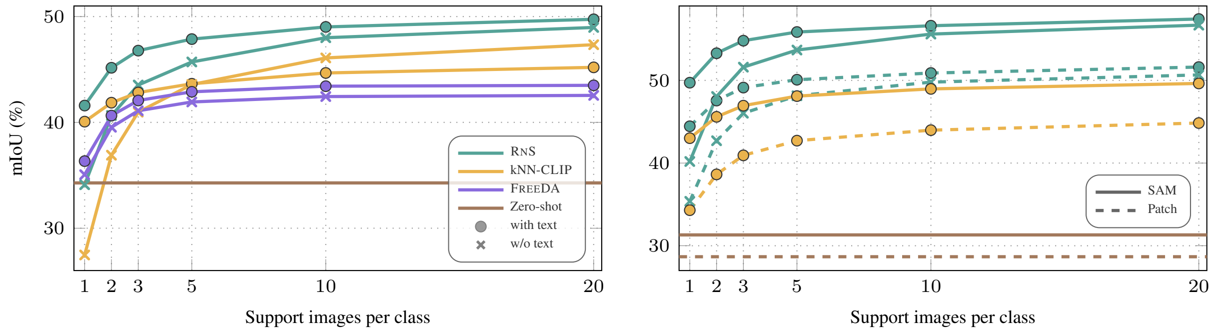

Figure 3. **Full textual and visual support.** We compare zero-shot, RNS, kNN-CLIP and FREEDA and their variants without class name information (w/o text) for increasing number of support images per class. SAM 2.1 is used for region proposals. *Left:* OpenCLIP (ViT-B/16) for region-level predictions. *Right:* DINOv3.txt (ViT-L/16) for patch-level and region-level predictions.

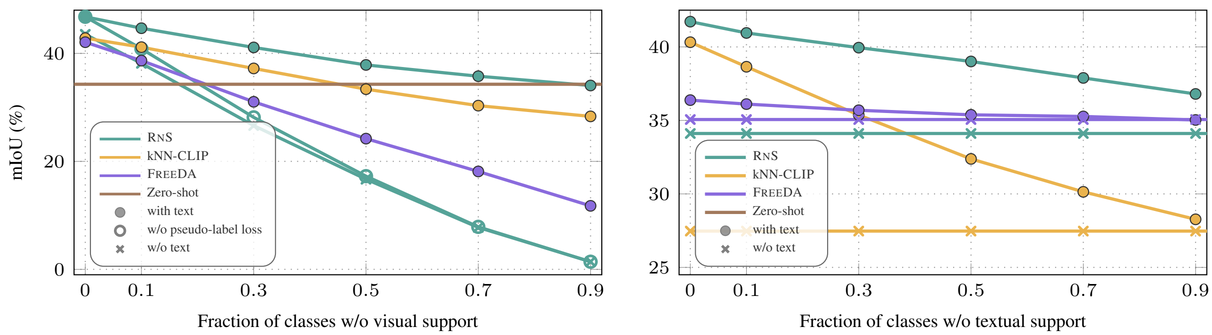

Figure 4. **Partial visual (left) and textual (right) support settings.** Results of zero-shot, RNS, kNN-CLIP, FREEDA and their variants without class name information (w/o text). RNS evaluated w/o the pseudo-label loss in (12). OpenCLIP ViT-B/16 and SAM 2.1 are used. Left: a fraction of classes lack visual examples, while $B = 3$ for the rest. Right: a fraction of classes lack textual class names, and $B = 1$.

4. Experiments

4.1. Experimental setup

Datasets and evaluation. We evaluate on the validation split of six OVS benchmarks: PASCAL VOC [14] (VOC), PASCAL Context [46] (Context), COCO Object (Object), COCO-Stuff [5] (Stuff), Cityscapes [12] (City), and ADE20K [87] (ADE). We report the *average mIoU* over all datasets unless otherwise noted. Per dataset results are presented in the supplementary. We also evaluate on PASCAL Context-59 (C-59) [46], FoodSeg103 (Food) [70], and CUB [69] to compare OVS and fully supervised methods. Details about sampling the B support images per class for the visual support set, and the construction of the few-shot benchmark are shown in the supplementary.

Implementation details. We use OpenCLIP ViT-B/16 [10] trained on LAION [56] and apply the MaskCLIP trick [88]. For DINOv3 [61], we use the public ViT-L/16 checkpoint adapted with dino.txt [26] to derive text-aligned patch features, denoted by DINOv3.txt. For region proposals, we run SAM 2.1 [54] Hiera-L on each query image with a 32×32 grid of points (one mask per point) and non-maximum suppression to ensure non-overlap. More implementation details are shown in the supplementary.

Competitors. We compare to zero-shot prediction (Section 3.2), and to two retrieval-based OVS methods that leverage both visual and textual support sets: kNN-CLIP [18] , which we re-implement, and FREEDA [2], where we adapt the official code to use a real support set, *i.e.* pixel-annotated images, rather than synthetic-only prototypes. Please refer to the supplementary material for details in the implementation of the aforementioned competitors. We also report performance for training a linear classifier or the full network "offline" on the entire support set in a closed-set manner. These variants lose the open-vocabulary ability as they cannot provide predictions for unseen classes, but provide useful reference baselines. Specifically, we consider the following three offline training methods: (i) Linear classifier on per-image visual class features $\mathbf{v}_c^i$ and their corresponding labels. (ii) Offline linear classifier on patch-level features $\mathbf{x}_j^i$ and pixel-level annotations. Predictions are upsampled to apply the loss at the full image resolution. (iii) Offline finetuning on images and pixel-level annotations. This configuration builds upon the previous setup but, in addition to training the linear classifier, it also finetunes the parameters of the vision encoder. We also compare to SOTA fully supervised and different kinds of OVS methods in Table 2.

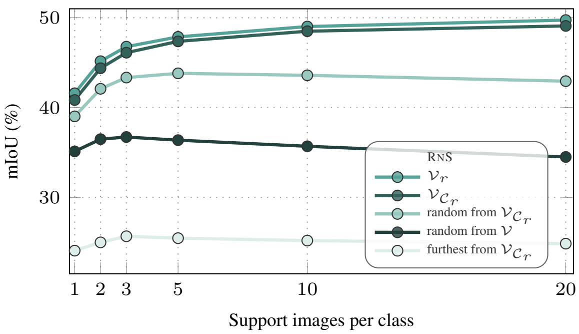

Figure 5. **Impact of retrieval on RNS.** We replace the retrieved visual support feature set $\mathcal{V}_r$ of RNS with a random subset of the visual support feature set $\mathcal{V}$, or different variants of visual support features from the retrieved classes $\mathcal{V}_{\mathcal{C}_r}$.

Method	$B = 1$	$B = 5$	$B = 10$
RNS	41.59	47.87	49.02
RNS w/o w_c	41.20 -0.39	47.43 -0.44	48.54 -0.48
RNS w/o w_c $\Lambda = \{0.8\}$	36.40 -5.19	46.55 -1.32	48.38 -0.64
RNS w/o text	34.11 -7.48	45.71 -2.16	48.00 -1.02

Table 1. **Ablations of RNS.** We report average mIoU across the considered datasets for three different numbers of available support images per class ($B = 1$, $B = 5$, $B = 10$). Blue numbers denote difference to the number in the same column but first row.

4.2. Experimental results

Full textual and visual support. Figure 3 reports mIoU as we vary the number of support images B per class. RNS consistently outperforms all competitors on every B, both backbones, and input feature granularity, with large gains. Moreover, it yields significant improvement over zero-shot segmentation: $+7.3\%$ on OpenCLIP and $+18.4\%$ on DINOv3.txt with just one image per class. RNS effectively leverages text, achieving a performance gain at $B = 1$ over the w/o-text variant while performing on par in the $B = 20$ case, indicating that textual priors are valuable when support is sparse, while visuals dominate as support densifies. Interestingly, kNN-CLIP 's fusion heuristics help at $B = 1$, but hinder after $B = 5$, suggesting sensitivity to hand-tuned fusion as support grows. The variant of kNN-CLIP that discards text is competitive when enough support images are available but scores low in the few-shot regime, where the diversity and quality of the support are minimal. FREEDA does not benefit enough from combining textual with visual support; low gains with respect to w/o-text baseline. Moreover, on newer backbones with stronger localization, e.g., DINOv3, the gap between RNS and kNN-CLIP widens, indicating that RNS scales better with better visual representations. Region-proposals consistently boost performance compared to patch-level predictions, by leveraging the large-scale training of SAM, at the expense of higher computational cost at inference time.

Partial visual support. In Figure 4 (left), we vary the fraction of classes without visual examples, while keeping text descriptions for all. RNS degrades smoothly, and

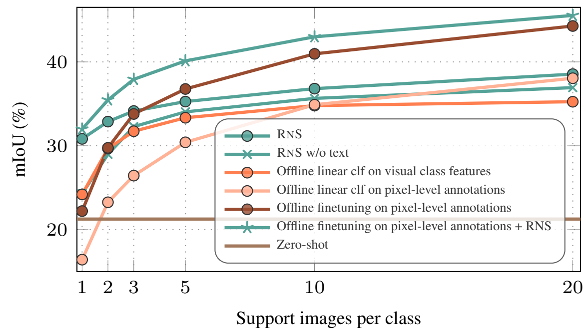

Figure 6. **Comparison in a closed vocabulary setting.** We compare RNS to the offline baseline competitors. To ensure a fair comparison we tune the learning rate, batch size, and number of iterations using a train-validation split from the available support images. No mask proposals are used. We report average performance on VOC, ADE, and Stuff.

manages to benefit from visual support even for small fractions of supported classes. Removing pseudo-label loss (12) leads to a steep drop in performance; this is our mechanism to compensate for the missing visual support. Further removing textual support (w/o text) degrades performance more, as the model cannot leverage neither fusion nor class relevance weights. kNN-CLIP and FREEDA drop significantly, and soon perform below zero-shot, as they do not account for missing classes in their support set. Note that missing visual support for some classes is inevitable in an open-world and dynamically expanding environment.

Partial textual support. In Figure 4 (right), we vary the fraction of classes without text descriptions, while keeping the visual support set intact. RNS remains the best across the entire range, dropping only mildly as text becomes scarce. Results suggest that training jointly on visual and textual support is superior to heuristic late fusion of independent predictions; kNN-CLIP degrades steeply, while FREEDA barely benefits even when text is available. RNS uses text as an auxiliary signal, used for fusion and relevance weighting, yielding a consistent boost when available but without making the model fragile when it disappears.

Ablations are presented in Table 1. Removing class relevance weights (w_c) yields a reduction across all shots, confirming their role in suppressing irrelevant retrieved classes. Additionally, creating fused features (4) with a single $\lambda = 0.8$ harms especially the low-shot regime, where effective use of text fusion is more important. Overall, components of RNS provide complementary benefits, with lower but considerable impact as visual support increases.

In Figure 5, we analyze the impact of retrieval mechanism on the performance of RNS. When replacing retrieved visual support feature set $\mathcal{V}_r$ (6) with a random subset of $\mathcal{V}$ performance degrades a lot, confirming that using similar image regions is crucial. Then, we perform the default retrieval (6) but instead of keeping $\mathcal{V}_r$, we focus on retrieved classes $\mathcal{C}_r$, and form a set of visual support features from the retrieved classes $\mathcal{V}_{\mathcal{C}_r} = \bigcup_{i=1}^{M} \{\mathbf{v}_c^i : c \in \mathcal{C}_r\}$. We perform the test-time adaptation using different subsets of this set. Us-

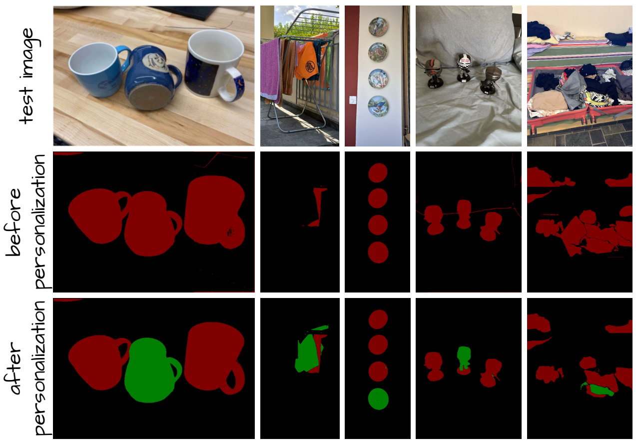

Figure 7. **RNS for personalized segmentation.** We append examples of a specific instance to the support. **Green:** personalized instance, **red:** generic class, and **black:** background. Support sets shown in the supplementary.

ing the full set $\mathcal{V}_{\mathcal{C}_r}$ yields slightly lower performance than the default approach, suggesting that $\mathcal{V}_{\mathcal{C}_r}$ includes irrelevant examples that are filtered out by RNS. Using a random subset of $\mathcal{V}_{\mathcal{C}_r}$ performs quite better than a random subset of $\mathcal{V}$, indicating that restricting adaptation to semantically relevant classes is beneficial. In contrast, selecting the furthest per query feature examples from $\mathcal{V}_{\mathcal{C}_r}$ results in the worst performance, even below random sampling from $\mathcal{V}$. These demonstrate that retrieval in RNS identifies relevant support features, which improves performance significantly.

Closed-set comparisons. We also include a comparison of RNS with fully supervised offline baselines in Figure 6. The offline classifier trained on visual class features performs comparably to RNS w/o text when $B = 1-3$, but its performance deteriorates as the number of images per class increases. This highlights the advantage of test-time retrieval, which identifies the most relevant visual class features from the support set for each test image. Offline methods trained on pixel-level annotations perform poorly in low-shot settings, indicating that such methods require larger amounts of data to avoid overfitting. Freezing the backbone and training only a linear classifier with pixel-level cross entropy loss underperforms RNS in all shots, suggesting that our method and objective are more effective in few-shot support utilization. On the other hand, training the backbone as well results in stronger performance when sufficient support is available, at the cost of training computational complexity. The best performance is obtained when combining the pretrained backbone weights from the latter experiment with RNS, by adapting the linear classifier during test-time. This suggests: (i) the complementary gains of visual and textual supervision with our method (ii) the efficacy of our test-time training on test-image relevant support, compared to offline training on the entire support.

Personalized segmentation. In Figure 7, we qualitatively demonstrate that the dynamic expandability of the support set in RNS enables personalized segmentation of particular object instances within a class. Starting from a support set that allows RNS to segment pixels of a class, *e.g.*, `plate`, appending only a few examples of a particular instance, *e.g.*, `plate with a kingfisher / my plate`, enables the model to distinguish that instance from the broader class. We observe that RNS accurately segments partially occluded personalized objects, such as the `skirt with a tropical pattern`. Failure cases remain; *e.g.*, RNS incorrectly segments part of an `orange towel` as a `swimsuit`, likely due to insufficient contextual information and reliance on color cues. Nevertheless, results illustrate that RNS is effectively employed for personalized segmentation without any modifications.

Method	Training set	Annot.	VOC	City	ADE	C-59	Food	CUB	Avg.
SAN† [78]	COCO	118k	–	–	32.1	57.7	24.5	19.3	–
CAT-Seg [11]	COCO	118k	82.5	47.0*	37.9	63.3	33.3	22.9	47.8
LPOSS+ [63]	✗	✗	62.4	37.9	22.3	38.6	26.1	12.0	33.2
CorrCLIP [83]	✗	✗	76.4	49.9	28.8	47.9	–	–	–
DINOv3.txt [61] + SAM	✗	✗	31.3	39.3	27.7	36.3	27.2	5.8	27.9
+ RNS $B=1$	Domain	66	73.2	59.1	37.3	52.7	42.8	34.0	49.9
+ RNS $B=20$	Domain	964	82.1	61.7	47.8	62.5	**52.2**	65.2	61.9
Fully Supervised	Domain	20k	**90.4**	**87.0**	**63.0**	**70.3**	45.1	**84.6**	**73.4**

Table 2. **OVS *vs.* fully supervised segmentation.** *Fully supervised*: best method picked per dataset. *: self-evaluated, †: from CAT-Seg. *Domain*: using annotations from each training set. *Annot:* the number of pixel-level semantic segmentation annotations that each method uses. For *domain* we report the ADE annotations. Full table and details in the supplementary.

Bridging the gap. In Table 2, we compare different OVS methods, including RNS, and fully supervised ones (closed vocabulary). We consider methods from four categories (i) SAN and CAT-Seg as OVS methods that train on COCO-Stuff, (ii) LPOSS and CorrCLIP as training free OVS methods, (iii) RNS as an OVS method that uses support sets constructed from annotated images, (iv) fully supervised methods pre-trained offline on a closed set of classes. RNS with $B = 20$, narrows down the gap to fully supervised segmentation to 11.5 on average, improving the zero-shot baseline by 34. RNS also surpasses the second best OVS method, CAT-Seg, by 14.1, using much less pixel-level annotations. The improvement is more evident on fine-grained datasets like CUB and Food that are far from the domain of COCO-Stuff, which is commonly used as a training set in OVS.

5. Conclusion

We introduce RNS, a retrieval–augmented test-time adapter for open-vocabulary segmentation that learns a lightweight per-image linear classifier on frozen VLM features by fusing textual class features with retrieved visual support features. The method operates on patches or region proposals, handles full and partial support with a single objective, and avoids any type of hand-crafted solutions. Across six benchmarks and two backbones, RNS consistently outperforms all baselines and competitors. Finally, the dynamic support mechanism makes RNS naturally suited for open-world fine-grained tasks, *i.e.* personalized segmentation.

Acknowledgments

This work was supported by the Czech Technical University in Prague grant No. SGS23/173/OHK3/3T/13, the EU Horizon Europe programme MSCA PF RAVIOLI (No. 101205297), and the Junior Star GACR GM 21-28830M. We acknowledge VSB–Technical University of Ostrava, IT4Innovations National Supercomputing Center, Czech Republic, for awarding this project (OPEN-33-67) access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CSC (Finland) and the LUMI consortium, through the Ministry of Education, Youth and Sports of the Czech Republic via the e-INFRA CZ project (ID: 90254). The access to the computational infrastructure of the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics" is also gratefully acknowledged.

References

[1] Ivana Balažević, David Steiner, Nikhil Parthasarathy, Relja Arandjelović, and Olivier J. Hénaff. Towards in-context scene understanding. In *NeurIPS*, 2023. 2

[2] Luca Barsellotti, Roberto Amoroso, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Training-free open-vocabulary segmentation with offline diffusion-augmented prototype generation. In *CVPR*, 2024. 2, 3, 6

[3] Benedikt Blumenstiel, Johannes Jakubik, Hilde Kühne, and Michael Vössing. What a mess: Multi-domain evaluation of zero-shot semantic segmentation. In *NeurIPS*, 2023. 2

[4] Walid Bousselham, Felix Petersen, Vittorio Ferrari, and Hilde Kuehne. Grounding everything: Emerging localization properties in vision-language transformers. In *CVPR*, 2024. 2

[5] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and stuff classes in context. In *CVPR*, 2018. 6

[6] 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. 2

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

[8] 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. 1, 2

[9] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 1

[10] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *CVPR*, 2023. 6

[11] Seokju Cho, Heeseong Shin, Sunghwan Hong, Anurag Arnab, Paul Hongsuck Seo, and Seungryong Kim. Cat-seg: Cost aggregation for open-vocabulary semantic segmentation. In *CVPR*, 2024. 2, 8

[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. 6

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

[14] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 6

[15] Xinqi Fan, Xueli Chen, Luoxiao Yang, Chuin Hong Yap, Rizwan Qureshi, Qi Dou, Moi Hoon Yap, and Mubarak Shah. Test-time retrieval-augmented adaptation for vision-language models. In *ICCV*, 2025. 3

[16] Matteo Farina, Gianni Franchi, Giovanni Iacca, Massimiliano Mancini, and Elisa Ricci. Frustratingly easy test-time adaptation of vision-language models. In *NeurIPS*, 2024. 3

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

[18] Zhongrui Gui, Shuyang Sun, Runjia Li, Jianhao Yuan, Zhaochong An, Karsten Roth, Ameya Prabhu, and Philip Torr. knn-clip: Retrieval enables training-free segmentation on continually expanding large vocabularies. *TMLR*, 2024. 2, 3, 6

[19] Sina Hajimiri, Malik Boudiaf, Ismail Ben Ayed, and Jose Dolz. A strong baseline for generalized few-shot semantic segmentation. In *CVPR*, 2023. 2

[20] Mir Rayat Imtiaz Hossain, Mennatullah Siam, Leonid Sigal, and James J Little. Visual prompting for generalized few-shot segmentation: A multi-scale approach. In *CVPR*, 2024. 2

[21] Mir Rayat Imtiaz Hossain, Mennatullah Siam, Leonid Sigal, and James J Little. The power of one: A single example is all it takes for segmentation in vlms. *arXiv preprint arXiv:2503.10779*, 2025. 2

[22] Ahmet Iscen, Mathilde Caron, Alireza Fathi, and Cordelia Schmid. Retrieval-enhanced contrastive vision-text models. In *ICLR*, 2024. 2

[23] Klara Janouskova, Tamir Shor, Chaim Baskin, and Jiri Matas. Single image test-time adaptation for segmentation. *TMLR*, 2024. 3

[24] Hao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. 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

[25] Siyu Jiao, Hongguang Zhu, Jiannan Huang, Yao Zhao, Yunchao Wei, and Humphrey Shi. Collaborative vision-text representation optimizing for open-vocabulary segmentation. In *ECCV*, 2024. 2

[26] Cijo Jose, Théo Moutakanni, Dahyun Kang, Federico Baldassarre, Timothée Darcet, Hu Xu, Daniel Li, Marc

Szafraniec, Michaël Ramamonjisoa, Maxime Oquab, Oriane Siméoni, Huy V. Vo, Patrick Labatut, and Piotr Bojanowski. Dinov2 meets text: A unified framework for image- and pixel-level vision-language alignment. In *CVPR*, 2025. 6

[27] Dahyun Kang and Minsu Cho. In defense of lazy visual grounding for open-vocabulary semantic segmentation. In *ECCV*, 2024. 2

[28] Adilbek Karmanov, Dayan Guan, Shijian Lu, Abdulmotaleb El Saddik, and Eric Xing. Efficient test-time adaptation of vision-language models. In *CVPR*, 2024. 3

[29] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *ICLR*, 2020. 2

[30] Chanyoung Kim, Dayun Ju, Woojung Han, Ming-Hsuan Yang, and Seong Jae Hwang. Distilling spectral graph for object-context aware open-vocabulary semantic segmentation. In *CVPR*, 2025. 2

[31] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 2, 5

[32] Mengcheng Lan, Chaofeng Chen, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. Clearclip: Decomposing clip representations for dense vision-language inference. In *ECCV*, 2024. 1, 2

[33] Mengcheng Lan, Chaofeng Chen, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. Proxyclip: Proxy attention improves clip for open-vocabulary segmentation. In *ECCV*, 2024. 1, 2

[34] Youngjun Lee, Doyoung Kim, Junhyeok Kang, Jihwan Bang, Hwanjun Song, and Jae-Gil Lee. RA-TTA: Retrieval-augmented test-time adaptation for vision-language models. In *ICLR*, 2025. 3

[35] Fan Li, Xuanbin Wang, Xuan Wang, Zhaoxiang Zhang, and Yuelei Xu. Images as noisy labels: Unleashing the potential of the diffusion model for open-vocabulary semantic segmentation. In *CVPR*, 2025. 2

[36] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 2

[37] Yongkang Li, Tianheng Cheng, Bin Feng, Wenyu Liu, and Xinggang Wang. Mask-adapter: The devil is in the masks for open-vocabulary segmentation. In *CVPR*, 2025. 2

[38] 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

[39] Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y. Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. In *NeurIPS*, 2022. 4

[40] Haotian Liu, Kilho Son, Jianwei Yang, Ce Liu, Jianfeng Gao, Yong Jae Lee, and Chunyuan Li. Learning customized visual models with retrieval-augmented knowledge. In *CVPR*, 2023. 2

[41] Sun-Ao Liu, Yiheng Zhang, Zhaofan Qiu, Hongtao Xie, Yongdong Zhang, and Ting Yao. Learning orthogonal prototypes for generalized few-shot semantic segmentation. In *CVPR*, 2023. 2

[42] Yang Liu, Yufei Yin, Chenchen Jing, Muzhi Zhu, Hao Chen, Yuling Xi, Bo Feng, Hao Wang, Shiyu Li, and Chunhua Shen. Unified open-world segmentation with multi-modal prompts. In *ICCV*, 2025. 2

[43] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[44] 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

[45] Juhong Min, Dahyun Kang, and Minsu Cho. Hypercorrelation squeeze for few-shot segmentation. In *ICCV*, 2021. 2

[46] 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

[47] 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

[48] Mehrdad Noori, David Osowiechi, Gustavo Adolfo Vargas Hakim, Ali Bahri, Moslem Yazdanpanah, Sahar Dastani, Farzad Beizaee, Ismail Ben Ayed, and Christian Desrosiers. Test-time adaptation of vision-language models for open-vocabulary semantic segmentation. In *NeurIPS*, 2025. 3

[49] 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, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *TMLR*, 2024. 2

[50] Valentinos Pariza, Mohammadreza Salehi, Gertjan J. Burghouts, Francesco Locatello, and Yuki M. Asano. Near, far: Patch-ordering enhances vision foundation models' scene understanding. In *ICLR*, 2025. 2

[51] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2

[52] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 2023. 2

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

[54] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. In *ICLR*, 2025. 2, 5, 6

[55] Josip Šarić, Ivan Martinović, Matej Kristan, and Siniša Šegvić. What holds back open-vocabulary segmentation? In *ICCVW*, 2025. 1

[56] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022. 1, 2, 6

[57] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. In *BMVC*, 2017. 2

[58] Yuheng Shi, Minjing Dong, and Chang Xu. Harnessing vision foundation models for high-performance, training-free open vocabulary segmentation. In *ICCV*, 2025. 2

[59] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. In *NeurIPS*, 2022. 3

[60] Mennatullah Siam, Boris N Oreshkin, and Martin Jagersand. Amp: Adaptive masked proxies for few-shot segmentation. In *ICCV*, 2019. 2

[61] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. 6, 8

[62] Sophia Sirko-Galouchenko, Spyros Gidaris, Antonin Vobecky, Andrei Bursuc, and Nicolas Thome. Dip: Unsupervised dense in-context post-training of visual representations. In *ICCV*, 2025. 3

[63] Vladan Stojnić, Yannis Kalantidis, Jiří Matas, and Giorgos Tolias. LPOSS: Label propagation over patches and pixels for open-vocabulary semantic segmentation. In *CVPR*, 2025. 1, 2, 8

[64] Shobhita Sundaram, Julia Chae, Yonglong Tian, Sara Beery, and Phillip Isola. Personalized representation from personalized generation. In *ICLR*, 2025. 2

[65] Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia. Prior guided feature enrichment network for few-shot segmentation. *IEEE TPAMI*, 2020. 2

[66] Zhuotao Tian, Xin Lai, Li Jiang, Shu Liu, Michelle Shu, Hengshuang Zhao, and Jiaya Jia. Generalized few-shot semantic segmentation. In *CVPR*, 2022. 2

[67] Feng Wang, Jieru Mei, and Alan Yuille. Sclip: Rethinking self-attention for dense vision-language inference. In *ECCV*, 2024. 2

[68] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *ICCV*, 2019. 2

[69] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010. 6

[70] Xiongwei Wu, Xin Fu, Ying Liu, Ee-Peng Lim, Steven C. H. Hoi, and Qianru Sun. A large-scale benchmark for food image segmentation. In *ACM MM*, 2021. 6

[71] Yuhuai Wu, Markus Norman Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. In *ICLR*, 2022. 2

[72] Monika Wysoczańska, Michaël Ramamonjisoa, Tomasz Trzciński, and Oriane Siméoni. Clip-diy: Clip dense inference yields open-vocabulary semantic segmentation for-free. In *WACV*, 2024. 2

[73] Monika Wysoczańska, Oriane Siméoni, Michaël Ramamonjisoa, Andrei Bursuc, Tomasz Trzciński, and Patrick Pérez. Clip-dinoiser: Teaching clip a few dino tricks for open-vocabulary semantic segmentation. In *ECCV*, 2024. 1, 2

[74] Aoran Xiao, Weihao Xuan, Heli Qi, Yun Xing, Ruijie Ren, Xiaoqin Zhang, Ling Shao, and Shijian Lu. Cat-sam: Conditional tuning for few-shot adaptation of segment anything model. In *ECCV*, 2024. 2

[75] Bin Xie, Jiale Cao, Jin Xie, Fahad Shahbaz Khan, and Yanwei Pang. SED: A simple encoder-decoder for open-vocabulary semantic segmentation. In *CVPR*, 2024. 2

[76] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *CVPR*, 2022. 2

[77] 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

[78] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *CVPR*, 2023. 8

[79] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional CLIP. In *NeurIPS*, 2023. 2

[80] Jianhao Yuan, Shuyang Sun, Daniel Omeiza, Bo Zhao, Paul Newman, Lars Kunze, and Matthew Gadd. Rag-driver: Generalisable driving explanations with retrieval-augmented in-context learning in multi-modal large language model. In *Proceedings of the Robotics: Science and Systems (RSS) 2024*, 2024. 2

[81] Maxime Zanella, Benoît Gérin, and Ismail Ben Ayed. Boosting vision-language models with transduction. *NeurIPS*, 2024. 3

[82] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. 1, 2

[83] Dengke Zhang, Fagui Liu, and Quan Tang. Corrclip: Reconstructing patch correlations in clip for open-vocabulary semantic segmentation. In *ICCV*, 2025. 2, 8

[84] Renrui Zhang, Zhengkai Jiang, Ziyu Guo, Shilin Yan, Junting Pan, Hao Dong, Yu Qiao, Peng Gao, and Hongsheng Li. Personalize segment anything model with one shot. In *ICLR*, 2024. 2

[85] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1

[86] Ziyu Zhao, Xiaoguang Li, Lingjia Shi, Nasrin Imanpour, and Song Wang. Dpseg: Dual-prompt cost volume learning for open-vocabulary semantic segmentation. In *CVPR*, 2025. 2

[87] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *CVPR*, 2017. 6

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