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

Supplementary Material

Supplementary Overview

Supplementary material contains:

- **Experimental setup details (Section A)** — evaluation protocol (Section A.1), implementation specifics (Section A.2), competitor configurations (Section A.3), and details on offline baseline comparisons (Section A.4).
- **Additional experimental results (Section B)** — out-of-domain support (Section B.1), partial visual support results on finegrained datasets (Section B.2), backbone comparison (Section B.3), seen/unseen class analysis (Section B.4), a complete version of Table 2 in the main paper (Section B.5), runtime comparisons (Section B.6), and an additional ablation study (Section B.7).
- **Qualitative results (Section C)** — SAM Mask vs. patch-level prediction comparisons (Section C.1), qualitative comparisons on considered baselines (Section C.2 and Section C.3), and support sets for personalized segmentation results reported in the main paper (Section C.4).
- **Per dataset results** of Figures 3-4 of the main paper are reported in Figure X, Figure XI, Figure XII and Figure XIII.

A. Experimental setup details

A.1. Evaluation protocol

To construct our few-shot benchmark, we sample support images from the *training split* of each dataset. We create a random permutation of the classes. Then we iterate over classes and randomly sample B images per class. If an encountered class already appears in previously sampled images B times, we skip it. This procedure intentionally *over-represents frequent classes*, *e.g.* `road` in driving scenes, preserving a realistic long tail distribution. We create the support set with *four random seeds*, except for VOC and City that we use *eight random seeds*, and report average performance. For partial support experiments, we randomly drop visual or textual support for a fraction of classes. For partial textual support this corresponds to a fraction of test class names, while for partial visual support this corresponds to the annotated visual features of visually supported classes.

A.2. Implementation details

We use OpenCLIP ViT-B/16 [III] trained on LAION [XIV] and apply the MaskCLIP trick [XXVI]. For DINOv3 [XV], we use the public ViT-L/16 checkpoint adapted with dino.txt [VII] to derive text-aligned patch features, denoted by DINOv3.txt. For SigLIP2 [XVIII], we use the ViT-L/16 variant trained on images of resolution 512×512. For optional region proposals, we use SAM 2.1 [XII] Hiera-L with a 32×32 grid of points (one mask per point) and non-maximum suppression to ensure non-overlap. Pixels not belonging to any mask form an additional separate mask. For support images we extract dense patch features $X^i \in \mathbb{R}^{n \times d}$ using a sliding window of fixed crop size and stride, down-sample its mask Y^i to patch labels $P^i \in [0,1]^{n \times C}$ via label aggregation within each patch re-

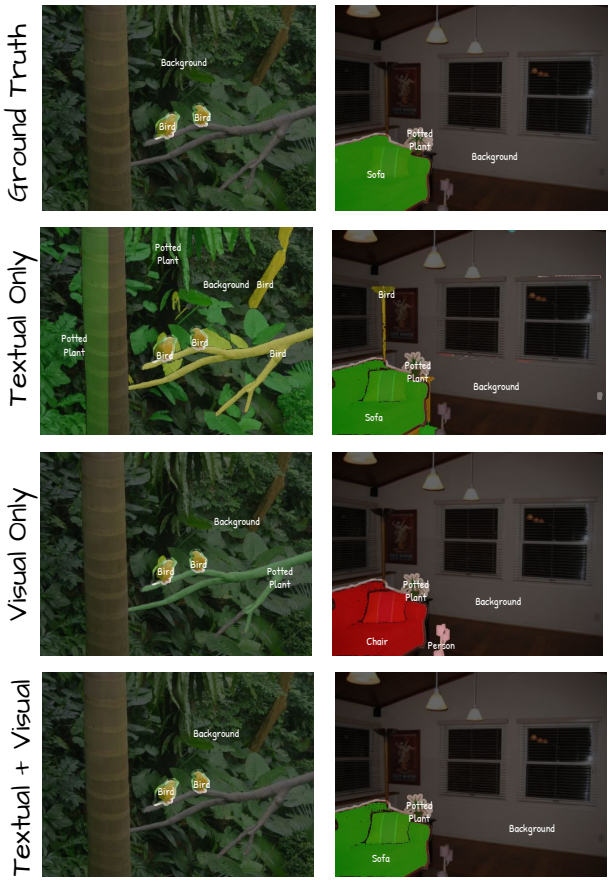

Figure I. **Open-vocabulary segmentation (OVS) results.** We compare three settings: (i) *textual-only* support (zero-shot OVS), (ii) a simplified version of RNS using *visual-only* support, and (iii) the full RNS combining *textual+visual* support. Text-only support leads to ambiguous predictions (tree branch as bird, and various background hallucinations). Visual-only support helps disambiguate some classes but still confuses contextually similar objects (sofa–chair, tree branch–potted plant). RNS effectively combines both modalities to achieve accurate segmentation.

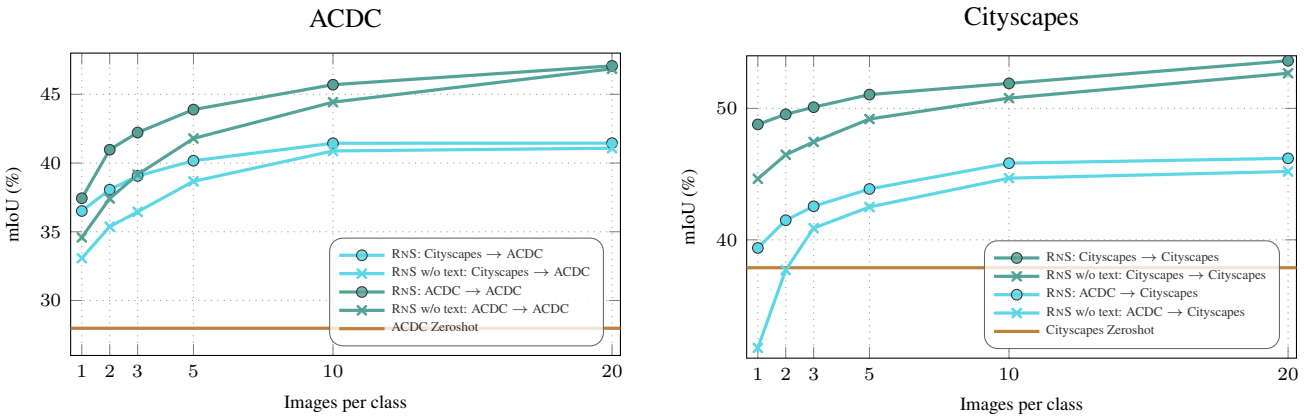

Figure II. **Out-of-domain vs. in-domain visual support between Cityscapes and ACDC.** The *left* plot reports performance on **ACDC** as we vary the number of visual support images per class from either Cityscapes (out-of-domain, Cityscapes → ACDC) or ACDC (in-domain, ACDC → ACDC). The *right* plot reports the analogous results when evaluating on **Cityscapes**, with support drawn from either ACDC (out-of-domain, ACDC→ Cityscapes) or Cityscapes (in-domain, Cityscapes → Cityscapes). In both cases we compare RNS with and without text, and include the corresponding zero-shot baseline.

gion. At inference, we resize images to a fixed shorter side size, preserving aspect ratio, and extract dense features using a sliding window with fixed crop size and stride. *Textual class features* $\mathbf{t}_c$ are obtained with standard CLIP ImageNet-1k templates, *e.g.*, `a photo of a {class}` [XI]. We average the text features across templates for each class. We do not expand class names beyond this. Regarding RNS, we fix hyperparameters across all experiments to k=4, τ=0.1, β_f=1.5, β_p=0.2, and $\Lambda = \{0.9, 0.8, 0.6, 0.4, 0.2, 0.0\}$. Our model is a linear classifier g_θ trained per test image. We train for 700 steps with a learning rate 0.02, optimizing using Adam [VIII] in full-batch mode, *i.e.* no mini-batch stochasticity is involved.

A.3. Details on competitors

For **kNN-CLIP** [VI], we follow the official implementation. To ensure a fair comparison we fix the hyperparameters across datasets and few-shot settings. The values are chosen based on the average test set performance on the considered benchmarks. The w/o text variant of the method arises by applying T=1.0 to the confidence threshold they introduce on zero-shot predictions. Please refer to the original paper for additional information.

FREEDA [I] generates images and pseudo-labels using a diffusion model, to construct a set of weakly labeled synthetic prototypes similar to our per-image visual class features. In their setup, these synthetic features are indexed via descriptive text features. At test time, the text feature of each test class is used to retrieve the most similar text indexes and thus match that class with similar synthetic prototypes. In our case, each per-image visual class feature already has a ground-truth label, so the correspondence to test classes is known. We therefore remove the retrieval step and assume oracle access to this matching when adapting their method to real visual support.

We fix the hyperparameters of the method in the same way that we do for kNN-CLIP. The w/o text variant is obtained by applying β=1.0 which is the weight used to linearly combine local visual similarities with global textual similarities. Please refer to the original paper for additional information.

In both methods, in the partial text support setup, we replace missing textual class features with the average textual class feature across classes with an available class name.

A.4. Details on comparison to offline baselines

In Figure 6 of the main paper we compare RNS with baselines trained in an offline, closed-vocabulary way. To ensure a fair comparison we use OpenCLIP ViT-B/16 features and no mask proposer across all methods. For each method we tune learning rate, batch size, and number of training iterations on an 85−15% train–validation split of each support set. We use the Optuna library to get a per-support-set "optimal" hyperparameter set, based on validation performance, for all methods.

To motivate this tuning, Figure V compares RNS and the linear offline classifier on per-image visual class features under two fixed hyperparameter settings. The linear classifier is highly sensitive to this choice: performance can differ by more than 30 mIoU, and even collapses under mismatched settings. In contrast, RNS changes only modestly across the two configurations and stays close to its tuned "optimal" curve. This shows that offline baselines need careful hyperparameter tuning to stay competitive, largely because the effective training set size changes drastically between low-shot settings. In contrast, RNS is much more robust to fixed training settings: by retrieving only support features relevant to each test image, it keeps the effective training set size more stable.

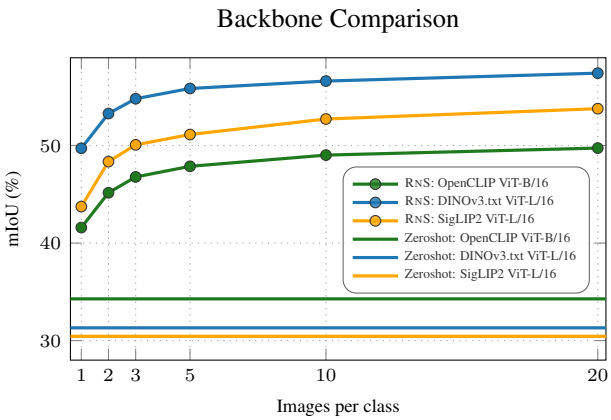

Figure III. **Backbone comparison.** We compare the performance of RNS using three different vision–language backbones (OpenCLIP ViT-B/16, DINOV3.txt ViT-L/16, and SigLIP2 ViT-L/16), along with the corresponding zero-shot baselines for each one of them.

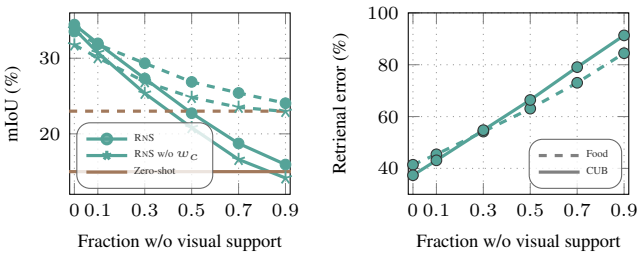

Figure IV. **Partial visual (left) and Retrieval quality (right).** OpenCLIP (ViT- B/16) + SAM 2.1 for region proposals are used.

B. Additional experimental results

B.1. Out-of-domain visual support

In Figure II, we analyze how RNS behaves when the visual support comes from out-of-domain images of the test classes. We use ACDC [XIII], which provides annotations for the Cityscapes [V] class set but is captured under adverse conditions (fog, snow, rain, night). Out-of-domain support is consistently weaker than in-domain support, yet it still yields substantial gains over zero-shot segmentation and continues to improve as we add more visual examples. Remarkably, when evaluating on Cityscapes, RNS improves over the zero-shot baseline even when using ACDC as visual support, despite the fact that many ACDC scenes have ambiguous semantics (*e.g.*, sidewalks completely covered by snow).

B.2. Partial visual support on fine-grained datasets.

In Figure IV left we present partial visual support performance in Food and CUB. On the right plot we present the retrieval error (percentage of retrieved instances from classes that are not present in each test image). Even for high retrieval errors, RNS effectively takes advantage of support examples and outperforms zero-shot. Moreover,

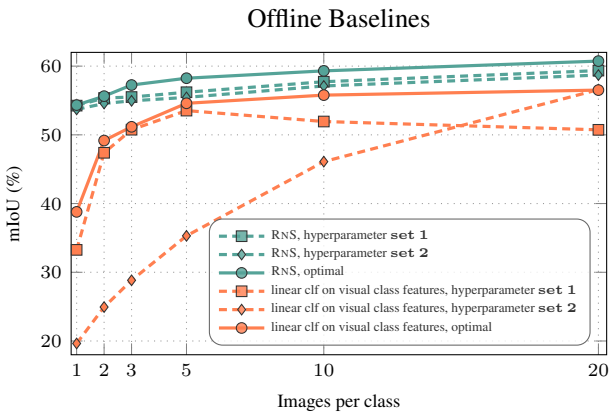

Figure V. **Comparison to offline baseline with and w/o hyperparameter tuning**. We compare RNS against the offline linear classifier trained on per image visual class features on VOC. We include the curves presented in Figure 6 that use hyperparameter tuning per B and support seed (noted as "optimal"). We report the performance of both methods on two hyperparameter configurations: *hyperparameter* **set 1** which corresponds to an optimal set of hyperparameters of the linear classifier for $B = 1$, and *hyperparameter* **set 2** which corresponds to an optimal set of hyperparameters of the linear classifier for $B = 20$.

Method	10% unseen			50% unseen			90% unseen		
	S	U	A	S	U	A	S	U	A
Zero-shot	52.13	59.61	52.85	56.24	49.12	52.85	69.08	50.14	52.85
RNS *(Partial)*	72.55	62.80	71.63	76.39	50.01	63.83	73.73	49.54	52.04
RNS *(Full)*	71.83	77.89	72.41	77.42	66.89	72.41	88.98	69.65	72.41

(a) VOC

Method	10% unseen			50% unseen			90% unseen		
	S	U	A	S	U	A	S	U	A
Zero-shot	21.87	31.49	22.83	21.82	23.85	22.83	22.06	22.92	22.83
RNS *(Partial)*	34.30	30.19	33.89	32.09	22.49	27.29	27.10	22.56	23.01
RNS *(Full)*	34.95	36.05	35.06	35.47	34.65	35.06	36.18	34.94	35.06

(b) ADE20K

Table I. **Partial seen/unseen classes.** We report the performance of RNS under varying fractions of unseen classes (classes w/o visual support) on VOC (a) and ADE20K (b). Columns show mean IoU on seen (S), *i.e.* classes with visual support, unseen (U), *i.e.* class that lack visual support, and all (A), *i.e.* the entire class set. We report the zero-shot (no class visually supported) and RNS with full visual support (all classes visually supported) on the same class sets as a reference.

the class relevance weights defined in Eq. 8 of the main manuscript suppress the loss of retrieved instances irrelevant to the test image, resulting in a significant improvement in such scenarios (Figure IV left - curves w/o w_c).

B.3. Backbone comparison

Figure III compares RNS across three vision-language backbones. DINOv3.txt ViT-L/16 achieves the highest mIoU and benefits the most from additional visual support, while SigLIP2 ViT-L/16 follows closely. OpenCLIP ViT-B/16 performs consistently lower but still improves steadily

Method	Training set	Annot.	Backbones	VOC	City	ADE	C-59	Food	CUB	Avg.
ODISE [XXI]	COCO	118k	Stable Diffusion v1.3, CLIP (ViT-L/14)	84.6	–	29.9	57.3	–	–	–
OVSeg† [IX]	COCO	118k	CLIP (ViT-L/14), Swin-B	–	–	29.6	55.7	16.4	14.0	–
SAN† [XXII]	COCO	118k	CLIP (ViT-L/14)	–	–	32.1	57.7	24.5	19.3	–
CAT-Seg [IV]	COCO	118k	CLIP (ViT-L/14)	82.5	47.0*	37.9	63.3	33.3	22.9	47.8
LPOSS+ [XVI]	✗	✗	OpenCLIP (ViT-B/16), DINO (ViT-B/16)	62.4	37.9	22.3	38.6	26.1	12.0	33.2
CLIP-DINOiser [XX]	✗	✗	OpenCLIP (ViT-B/16), DINO (ViT-B/16)	62.1	31.7	20.0	35.9	–	–	–
CorrCLIP [XXIII]	✗	✗	OpenCLIP (ViT-H/14), DINO (ViT-B/8), SAM2.1 (Hiera-L)	76.4	49.9	28.8	47.9	–	–	–
MaskCLIP [XXVI] + SAM	✗	✗	OpenCLIP (ViT-B/16), SAM2.1 (Hiera-L)	54.4	37.1	22.9	38.0	23.0	15.0	31.7
+ RNS $B = 1$	Domain	66	OpenCLIP (ViT-B/16), SAM2.1 (Hiera-L)	68.6	46.1	28.6	45.9	28.1	24.9	40.4
+ RNS $B = 20$	Domain	964	OpenCLIP (ViT-B/16), SAM2.1 (Hiera-L)	76.2	52.5	38.5	54.4	37.2	50.6	51.6
DINOv3.txt [XV] + SAM	✗	✗	DINOv3 (ViT-L/16), SAM2.1 (Hiera-L)	31.3	39.3	27.7	36.3	27.2	5.8	27.9
+ RNS $B = 1$	Domain	66	DINOv3 (ViT-L/16), SAM2.1 (Hiera-L)	73.2	59.1	37.3	52.7	42.8	34.0	49.9
+ RNS $B = 20$	Domain	964	DINOv3 (ViT-L/16), SAM2.1 (Hiera-L)	82.1	61.7	47.8	62.5	**52.2**	65.2	61.9
InternImage [XIX]	Domain	–	InternImage-H	–	**87.0**	62.9	**70.3**	–	–	–
SETR [XXV]	Domain	–	SeTR-MLA	–	76.7	48.6	–	45.1	–	–
GFN [XXIV]	Domain	–	ResNet-101	–	–	–	–	–	**84.6**	–
Mask2Former [II]	Domain	–	DINOv3 (7B)	**90.4**	86.7	**63.0**	–	–	–	–
Fully Supervised	Domain	20k	Best of Above	**90.4**	**87.0**	**63.0**	**70.3**	45.1	**84.6**	**73.4**

Table II. **OVS *vs*. fully supervised segmentation.** *Fully Supervised*: best method picked per dataset. *: self-evaluated, †: from CAT-Seg. Mask2Former numbers from DINOv3 [XV] paper. *Domain*: using annotations from each training set. *Annot:* the number of pixel-level annotated images that each method uses. For *Domain* we report the ADE annotations.

with more support images. In all cases, RNS significantly surpasses the respective zero-shot baselines, showing that the method is effective across backbones of different capacity and training regimes.

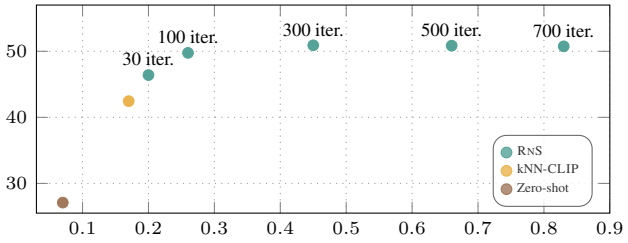

Figure VI. **Average performance (mIoU) *vs*. inference time (s).** DINOv3.txt, patch-level; B=5. Avg. on VOC, ADE, Stuff. A single NVIDIA $A100$ GPU is used.

B.4. Performance on seen/unseen classes

In Table I we report the performance of our partial visual support setup by separately measuring accuracy on seen classes (those with visual support) and unseen classes (those without support). The tables show that, when only a small fraction of classes is unseen, our partially supported variant performs on seen classes almost identically to RNS with full support. As the proportion of unseen classes increases, a widening gap appears. This is expected: as the number of unseen classes increases so does the number of false-positives for unseen classes, which impacts the performance on seen ones too.

At the same time, performance on unseen classes remains close to the zero-shot baseline across all settings. This confirms that RNS does not degrade zero-shot behavior for classes without visual support—when support is absent, the model naturally falls back to its zero-shot capabilities.

Method	$B = 1$	$B = 5$	$B = 10$
RNS ($K = 1$)	40.02 -1.57	46.83 -1.04	48.06 -0.96
RNS ($K = 4$)	41.59	47.87	49.02
RNS ($K = 16$)	41.97 +0.38	47.78 -0.09	48.96 -0.06
RNS ($K = 32$)	42.02 +0.43	47.59 -0.28	48.84 -0.18

Table III. **Ablations of the k-NN retrieval hyperparameter K in RNS.** We report average mIoU across the considered datasets for three different numbers of available support images per class ($B = 1$, $B = 5$, $B = 10$). The row with $K = 4$ (highlighted) corresponds to the configuration used in RNS, and blue numbers denote the difference to this row in the same column.

B.5. Full *OVS vs fully supervised* table

Table II is a complete version of Table 2, presented in the main paper. We augment the table with a column that mentions the backbones used within each method. We include additional OVS methods from the three categories mentioned in the main manuscript. We also include the individual *fully supervised* methods reported on each dataset.

B.6. Efficiency vs Runtime comparison.

In Figure VI we compare the performance and inference time of RNS, using different number of training iterations, with feedforward methods like kNN-CLIP and zero-shot baseline. We observe that the performance of RNS is robust under less iterations, while efficiency becomes comparable to feedforward kNN-CLIP. Thus, the overhead of RNS can be reduced while still achieving quite a performance boost.

B.7. Ablation on K in kNN retrieval

In Table III, we study the effect of the kNN retrieval hyperparameter K. Using a single neighbor ($K = 1$) per patch/region leads to clearly lower performance across all budgets B, confirming that aggregating information from multiple retrieved exemplars is beneficial. For $K \geq 4$, performance varies only marginally, with $K = 4$–16 achieving very similar mIoU, indicating that our method is robust to the exact choice of K once it is larger than 1. We therefore use $K = 4$ as our default setting.

C. Qualitative results

C.1. SAM mask vs patch-level predictions

In Figure VII we qualitatively compare two variants of RNS: one where the test-time linear layer is applied directly to patch-level features (RNS + Patch), and one where it operates on mask embeddings obtained by pooling features within SAM2.1 mask proposals (RNS + SAM). As expected, SAM2.1 proposals follow object boundaries much more closely than fixed patches, which yields noticeably sharper segmentation masks. Aggregating features within each mask also provides a form of spatial denoising, suppressing spurious patch-level predictions. SAM further exhibits a stronger notion of objectness, grouping together parts of the same object even under challenging appearance changes (*e.g.*, shadows in the second row).

At the same time, SAM does not always respect the semantic granularity required by the task and can over- or under-segment regions. In the bottom three rows, this leads to masks that merge distinct semantic regions or split single objects into multiple segments, introducing ambiguity despite the improved alignment with image structure.

C.2. Unimodal vs. multimodal support

In Figure I and Figure VIII we qualitatively compare methods that rely only on text (Zero-shot), only on visual support (RNS *w/o text*), or on both modalities (RNS). Using only class names often produces semantically ambiguous labels when several categories share similar appearance or context (*e.g.*, house vs. building in the first row, wall vs. building in the third row of Fig. VIII). Conversely, relying only on visual exemplars can confuse contextually similar objects (*e.g.*, train vs. bus in the fourth row). RNS leverages both textual and visual support: text provides a semantic prior that separates related categories, while visual examples anchor this prior to the image appearance, leading to more accurate segmentations across diverse scenes.

C.3. Comparisons with visually supported methods

In Figure VIII we additionally compare RNS to FREEDA and kNN-CLIP, which also use visual exemplars but rely on fixed, handcrafted fusion of text and visual cues. Such rigid fusion often struggles when the modalities conflict or when one is unreliable. In contrast, RNS learns how to fuse text and visual support, adapting to each image and yielding cleaner boundaries and fewer semantic confusions.

C.4. Personalized segmentation

In Figure IX we show the visual support sets used to perform personalized segmentation in Figure 7 of the main paper.

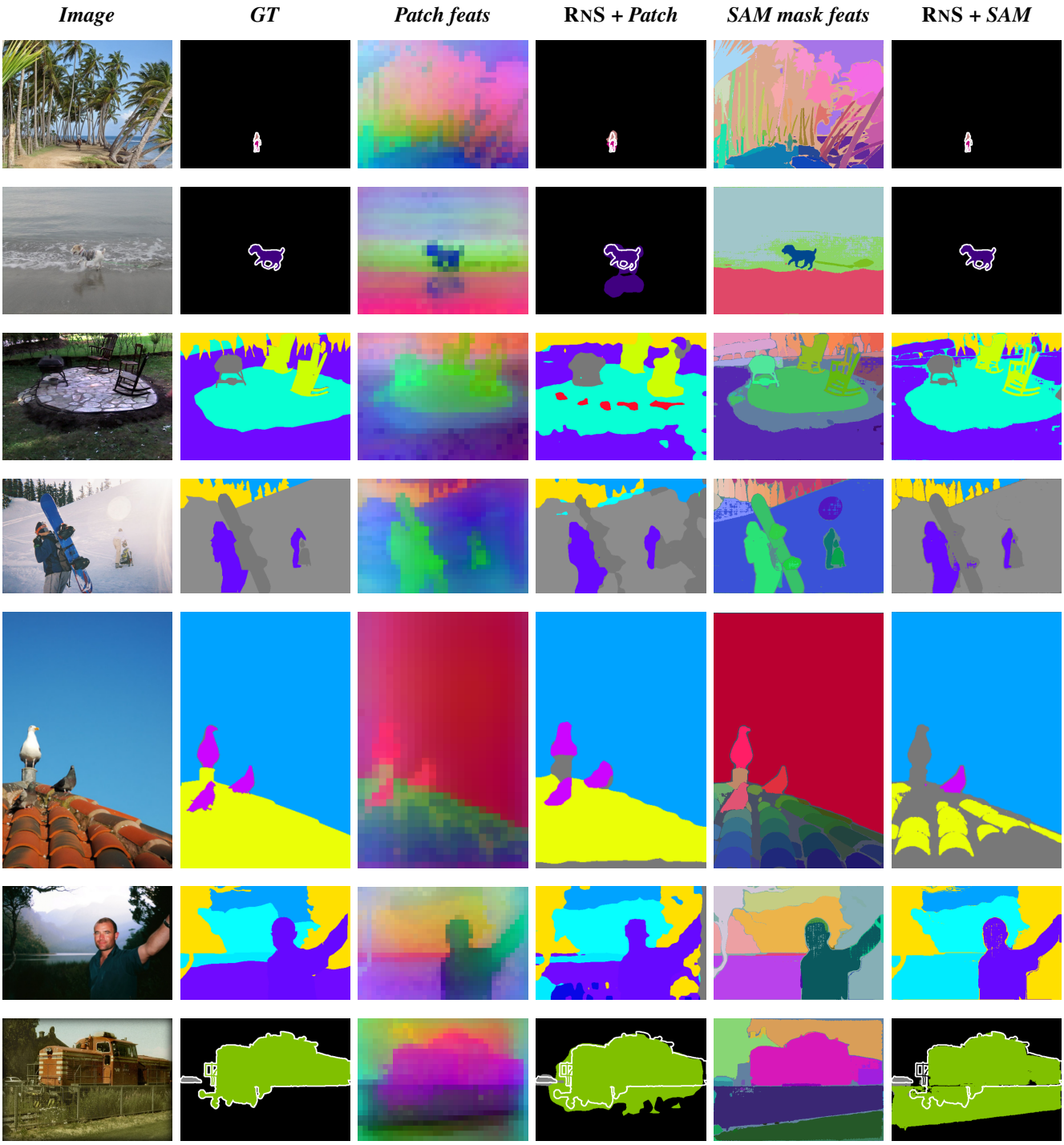

Figure VII. **Qualitative comparison of patch-level vs. mask-level segmentation.** Each row shows the input image, ground-truth mask, the PCA projection of either patch features or SAM mask features (average VLM patch features on top of SAM mask), and the corresponding predictions of RNS when applied on patches or on region proposals. Both variants use DINOv3.txt features [XV], and SAM 2.1 is used as the mask proposal generator [XII].

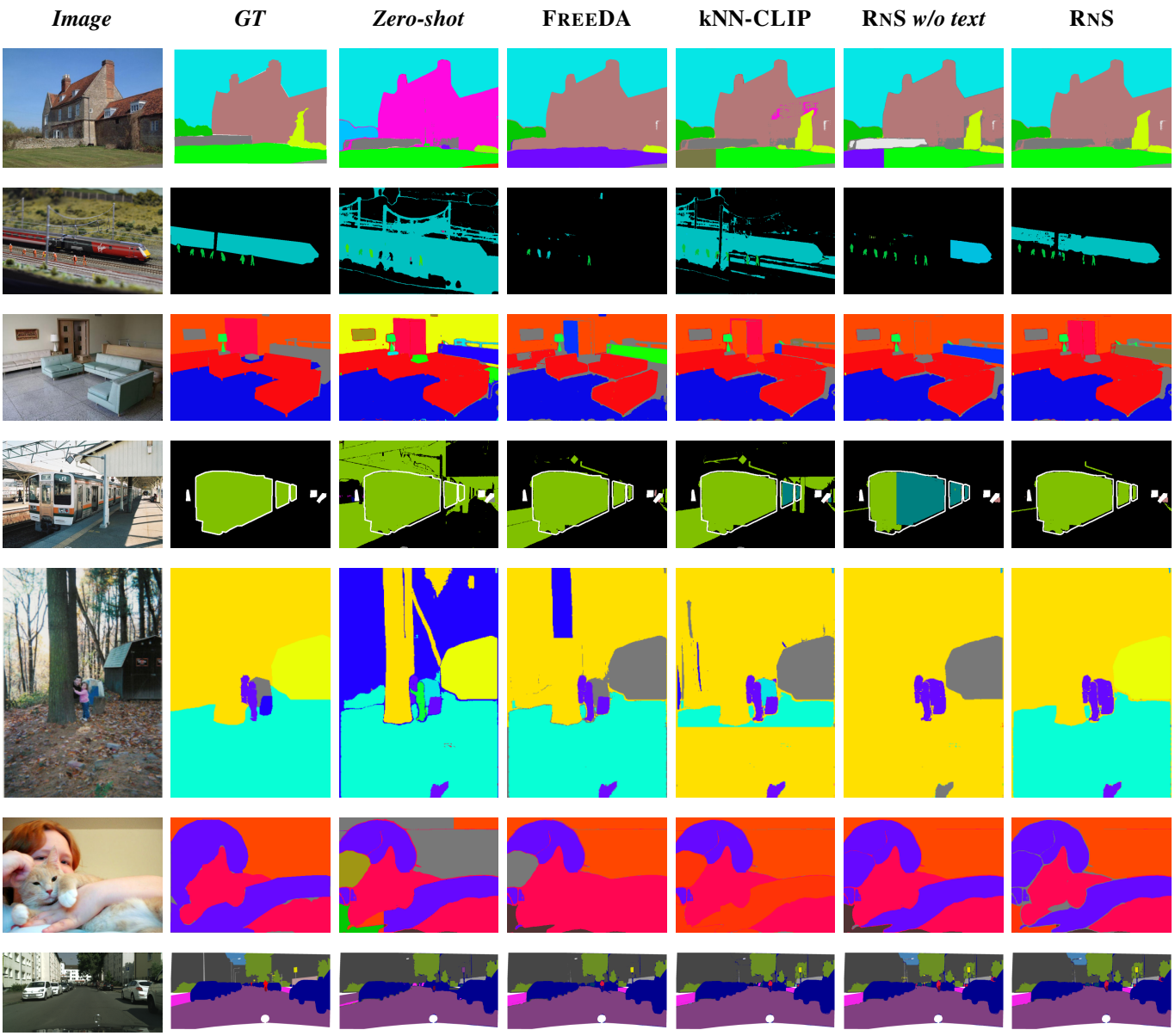

Figure VIII. **Qualitative comparisons** between zero-shot baseline, FREEDA, kNN-CLIP and RNS with and without class name information. All visually supported methods use one support image per class (B=1). All methods use OpenCLIP ViT-B/16 as VLM features [XV] and SAM 2.1 as region proposal generator [XII].

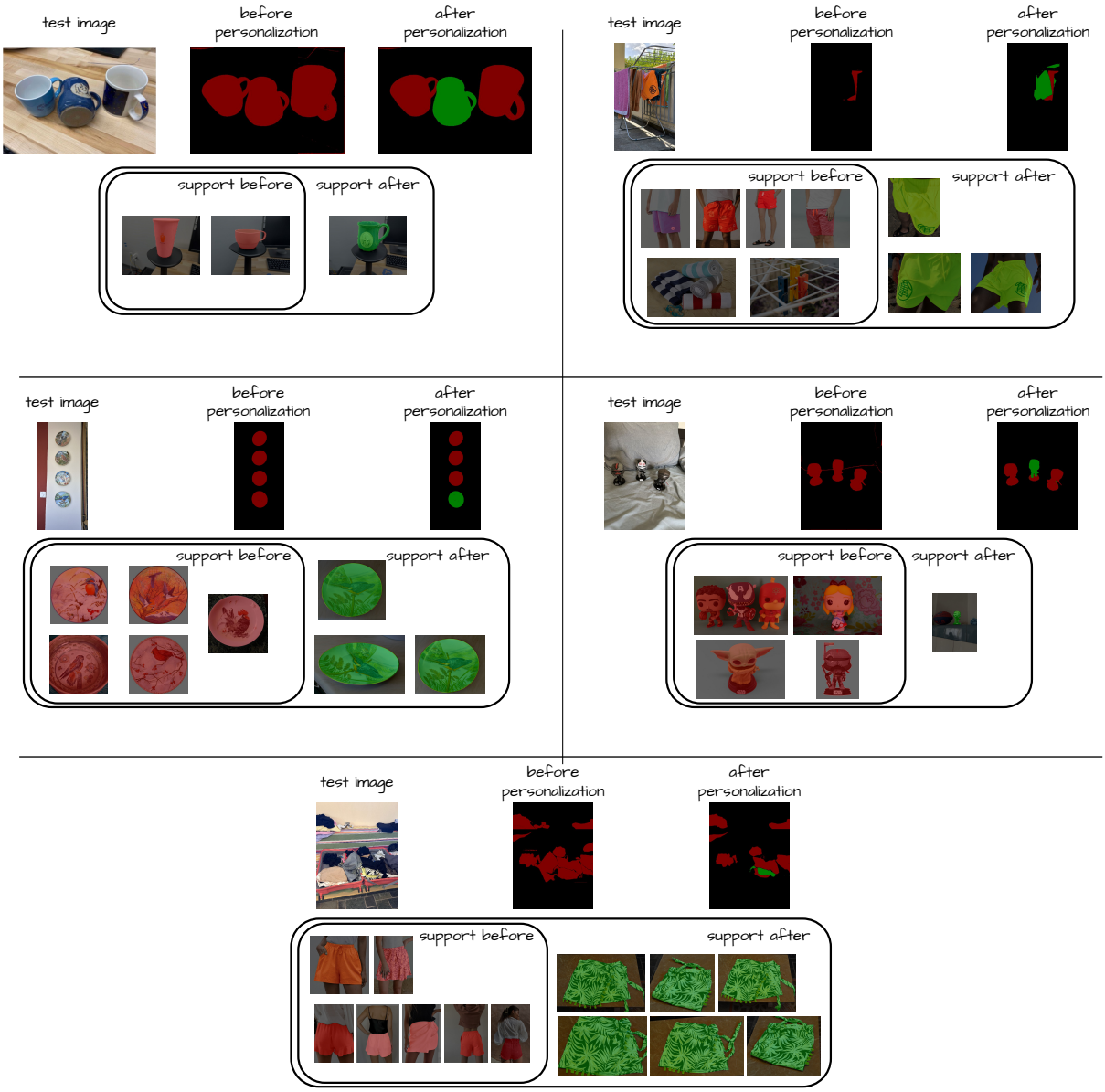

Figure IX. **Visual support sets for personalized segmentation.** RNS used for personalized segmentation using OpenCLIP ViT-B/16 features and SAM 2.1 as region proposer. Initially, visual support includeds images in *support before* and is later expanded to *support after*, with *support before* $\subset$ *support after*. **Green:** personalized instance, **red:** generic class, and **black:** background. Images from PODS [XVII], i-CIR [X], and self-collected.

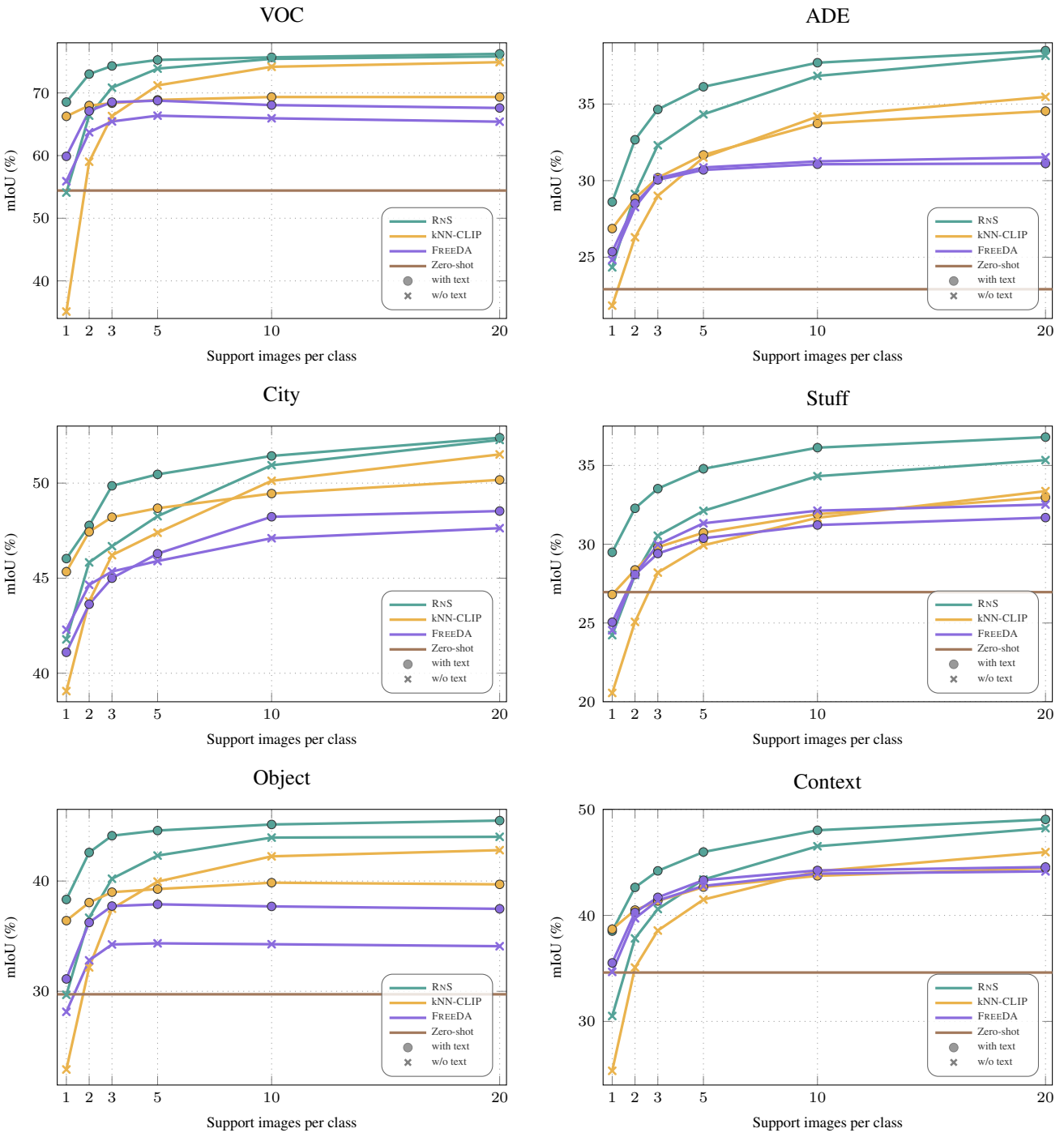

Figure X. **Full textual and visual support per dataset (OpenCLIP, ViT-B/16).** We compare zero-shot, RNS, kNN-CLIP and FREEDA and their variants without class name information (w/o text) for increasing number of support images per class. SAM 2.1 is used for region proposals.

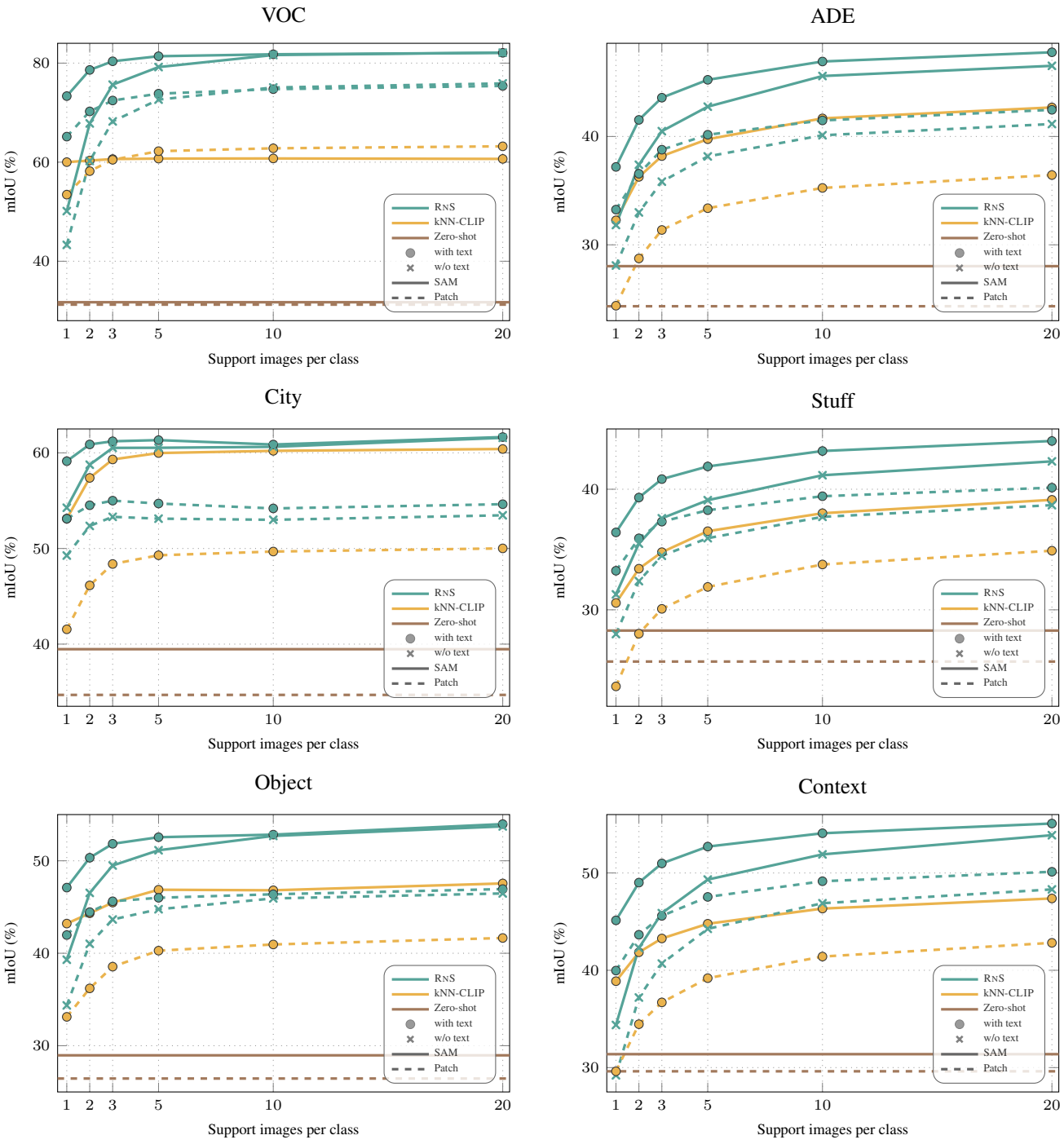

Figure XI. **Full textual and visual support per dataset (DINOv3.txt, ViT-L/16).** We compare zero-shot, RNS, kNN-CLIP and RNS without class name information (w/o text) for increasing number of support images per class. Prediction at the patch level or SAM 2.1 is used for region proposals.

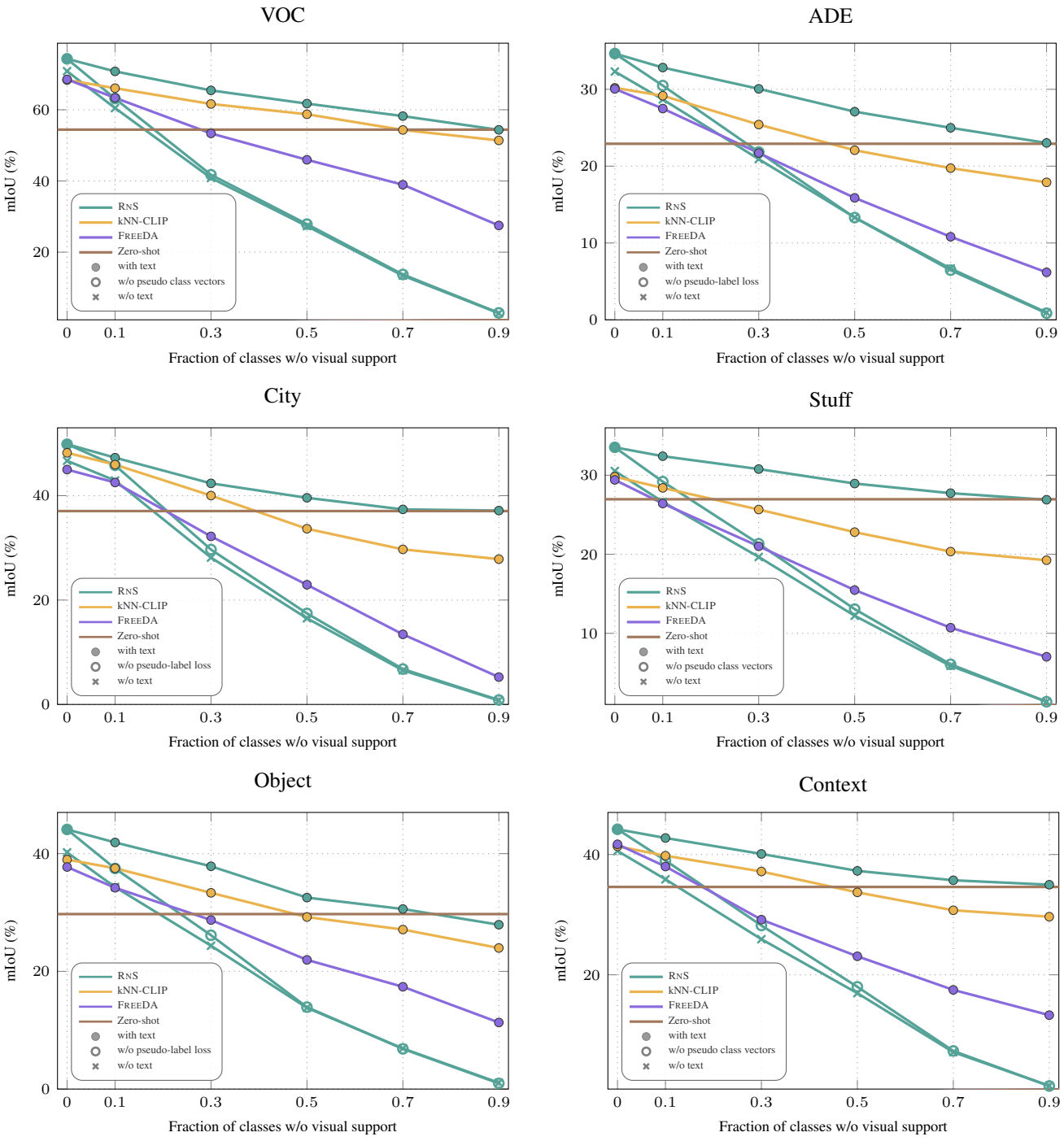

Figure XII. **Partial visual support setting per dataset.** Results of zero-shot, RNS, kNN-CLIP, and FREEDA, along with ablations of RNS without text and without the pseudo-label loss. OpenCLIP ViT-B/16 and SAM 2.1 are used. A fraction of classes lack visual examples, while $B = 3$ for the remaining classes.

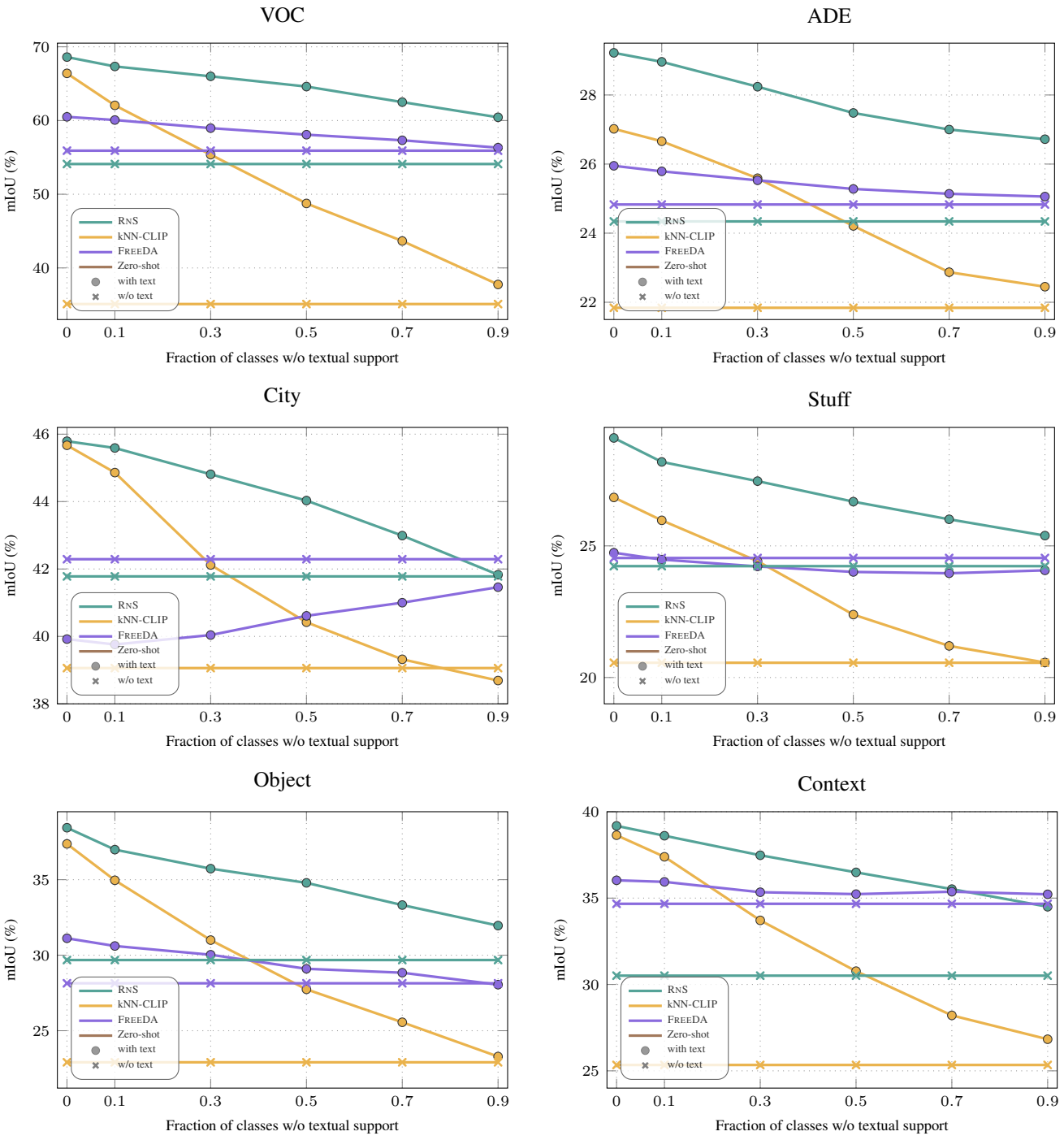

Figure XIII. **Partial textual support setting per dataset.** Results of Zero-shot, RNS, kNN-CLIP, and FREEDA, together with their variants without class name information (w/o text). OpenCLIP ViT-B/16 and SAM 2.1 are used. A fraction of classes lack textual class names, and $B = 1$ for all classes.

References

[I] 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

[II] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 4

[III] 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. 1

[IV] 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. 4

[V] 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. 3

[VI] 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

[VII] 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. 1

[VIII] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 2

[IX] 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. 4

[X] Bill Psomas, George Retsinas, Nikos Efthymiadis, Panagiotis Filntisis, Yannis Avrithis, Petros Maragos, Ondrej Chum, and Giorgos Tolias. Instance-level composed image retrieval. In *NeurIPS*, 2025. 8

[XI] 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. 2

[XII] 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. 1, 6, 7

[XIII] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *ICCV*, 2021. 3

[XIV] 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

[XV] 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. 1, 4, 6, 7

[XVI] Vladan Stojnić, Yannis Kalantidis, Jiří Matas, and Giorgos Tolias. LPOSS: Label propagation over patches and pixels for open-vocabulary semantic segmentation. In *CVPR*, 2025. 4

[XVII] Shobhita Sundaram, Julia Chae, Yonglong Tian, Sara Beery, and Phillip Isola. Personalized representation from personalized generation. In *ICLR*, 2025. 8

[XVIII] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 1

[XIX] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhen Huang, Zijian Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongyang Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *CVPR*, 2023. 4

[XX] 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. 4

[XXI] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *CVPR*, 2023. 4

[XXII] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *CVPR*, 2023. 4

[XXIII] Dengke Zhang, Fagui Liu, and Quan Tang. Corrclip: Reconstructing patch correlations in clip for open-vocabulary semantic segmentation. In *ICCV*, 2025. 4

[XXIV] X. Zhang, W. Zhao, W. Zhang, J. Peng, and J. Fan. Guided filter network for semantic image segmentation. *IEEE Transactions on Image Processing*, 2022. 4

[XXV] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng

Feng, Tao Xiang, Philip H. S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021. 4

[XXVI] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *ECCV*, 2022. 1, 4