PEARL: Geometry Aligns Semantics for Training-Free Open-Vocabulary Semantic Segmentation

Supplementary Material

Appendix

This appendix presents further model settings (§A), ablation studies (§B), quantitative (§C), and qualitative results (§D).

A. More Model Settings

Hyperparameter setting. For fair and consistent evaluation, we use a unified hyperparameter configuration for all datasets without dataset-specific tuning. The detailed settings are summarized in Table A1.

Table A1. **Fixed hyperparameter setting used for all datasets.**

Config	τ_s	β	ϵ	κ	λ	τ
Value	0.5	10	10^{-6}	5	1	1

B. More Ablation Studies

Alignment Objective. We further analyze the behavior of Procrustes Alignment (PA) on V21 [11]. PA is applied only to the *last* self-attention layer, where patch-text logits are formed, and aligns the key basis to the query basis ($\boldsymbol{K} \rightarrow \boldsymbol{Q}$) to correct token geometry with minimal disturbance to the original similarity structure. As an orthogonal Procrustes map, $\boldsymbol{R}^\star$ is the minimum-change, inner-product-preserving transformation under an orthogonality constraint, which explains why simpler variants such as centering only, whitening, or global rotation are consistently inferior. As shown in Fig. B1(a), PA makes the centered query/key clouds substantially better aligned in the projected space. Fig. B1(b) shows that the learned rotations have non-trivial yet well-behaved magnitudes across image-head pairs, while Fig. B1(c) indicates a positive correlation between alignment-error reduction and mIoU improvement. The component ablation in Fig. B1(d) further confirms that the full weighted formulation performs best, improving over its unweighted counterpart by 4.1 mIoU. Finally, Fig. B1(e)-(f), together with Tables B2 and B3, show that the iterative *Newton-Schulz* (N-S) solver is stable in practice: it matches SVD in accuracy while being substantially faster, and both the N-S iterations and the conjugate-gradient (CG) iterations exhibit clear performance plateaus, motivating the default settings used in all experiments.

Alignment Solver. Consistent with the trend in Fig. B1(e), Table B2 compares SVD and the N-S iterative solver inside our Procrustes alignment. Both solvers achieve the same average mIoU (**43.2**), and the per-dataset differences are within 0.3 points: SVD is slightly better on V21/PC60/City, while N-S is slightly better on Object/V20/ADE, indicating that the choice of solver has a negligible impact on accuracy. In terms of efficiency, Table B3 further shows that N-S consistently yields lower inference latency across all datasets (*e.g.*, **48.7** *vs.* 60.9 ms/img on V21 and **498.5** *vs.* 975.9 ms/img on City), reducing the average latency from 267.3 to **150.3** ms/img. This speedup is possible because our Procrustes module only requires the orthogonal factor of the SVD of a small $C \times C$ matrix: this factor coincides with the orthogonal polar factor $\boldsymbol{M}(\boldsymbol{M}^\top \boldsymbol{M})^{-1/2}$, where the inverse square root $(\boldsymbol{M}^\top \boldsymbol{M})^{-1/2}$ can be efficiently approximated by the N-S iteration (*cf*. §3.2) using only matrix multiplications on the GPU. Therefore, we adopt the SVD-free N-S variant as our default solver, which preserves accuracy while substantially improving efficiency.

Table B2. **Ablation analysis of different alignment solvers (*cf*. §3.2).** "N-S" denotes the Newton-Schulz iterative algorithm.

Solver	**with BG**			**without BG**					**Avg.**
	V21	PC60	Object	V20	PC59	Stuff	City	ADE	
SVD	**64.2**	**35.2**	37.2	86.7	**38.6**	**26.3**	**37.9**	19.2	**43.2**
N-S	64.1	35.1	**37.3**	**86.9**	**38.6**	**26.3**	37.6	**19.4**	**43.2**

Table B3. **Inference latency (ms/img) comparison of alignment solvers.** "N-S" denotes the Newton-Schulz iterative algorithm.

Latency	**with BG**			**without BG**					**Avg.**
	V21	PC60	Object	V20	PC59	Stuff	City	ADE	
SVD	60.9	199.7	239.3	59.6	198.8	212.1	975.9	192.3	267.3
N-S	**48.7**	**111.7**	**149.4**	**47.1**	**111.2**	**120.7**	**498.5**	**115.4**	**150.3**

Key-Key Self-Correlation. As shown in Table B4, we evaluate the effect of adding a key-key self-correlation term to our Procrustes Alignment (*cf*. §3.2) and enabling this term (*w/*) improves results on all datasets. In the "with BG" group, it brings gains of +0.9, +0.4, and +0.5 mIoU on V21 [11], PC60 [19], and Object [18], respectively. In the "without BG" group, it yields +0.5, +0.3, +0.2, +3.0, and +0.7 mIoU on V20 [11], PC59 [19], Stuff [3], City [7], and ADE [26]. Overall, the average mIoU increases from 42.4 to **43.2** (**+0.8**), with no degradation on any dataset and a particularly notable boost on City (**+3.0**), where long-range dependencies and cluttered scenes are common. In our implementation, Procrustes Alignment first recenters queries and

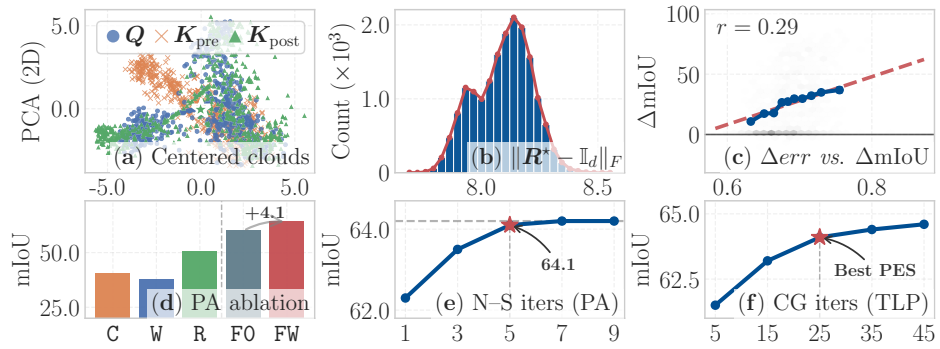

Figure B1. **Diagnostics of Procrustes Alignment on V21 [11].** (a) PCA projection of centered queries and keys before and after PA. (b) Distribution of the per-(image, head) rotation magnitude $\|\boldsymbol{R}^\star - \boldsymbol{I}_d\|_F$. (c) Correlation between alignment-error reduction Δerr and mIoU gain ΔmIoU. (d) Component ablation of centering (C), whitening (W), rotation (R), full PA without weights (FO), and full weighted PA (FW). (e) Stability of the Newton-Schulz iterations used in PA. (f) Stability of the conjugate-gradient iterations used in TLP.

keys using the token weights π_n in Eq. (5) and solves the orthogonal Procrustes problem in Eq. (6), either via SVD or via an SVD-free Newton-Schulz approximation of the polar factor, to obtain an orthogonal map $\boldsymbol{R}^\star$ that aligns the key basis to the query basis. Aligned attention scores are then computed as in Eq. (7), and we add a lightweight key-key term constructed from the centered keys, $\boldsymbol{K}_c\boldsymbol{K}_c^\top$, scaled by a factor $\alpha = d^{-1/2}$. Geometrically, the Procrustes term aligns the cross-covariances ($\boldsymbol{K}_c^\top\boldsymbol{Q}_c$) at the first order. Meanwhile, the key-key Gram matrix captures the self-correlation of $\boldsymbol{K}_c$, serving as a second-order regularizer for the attention kernel. Since $\boldsymbol{K}_c$ is already debiased by weighted centering (which suppresses dominant background and CLS tokens), this self-correlation term reinforces coherent foreground regions while dampening isolated noise. Consequently, the attention mechanism merges query alignment with the internal structure of the key space. This yields more stable token interactions and drives the segmentation improvements shown in Table B4.

Impact of Conjugate-Gradient Iterations. Table B5 and Fig. B1(f) ablate the number of CG iterations used to solve Eq. (13) in our Text-aware Laplacian Propagation (TLP) on V21 [11]. To balance segmentation quality and efficiency, we define a *Precision-Efficiency Score* (PES) that averages normalized mIoU, pAcc, Latency and GPU Memory for each setting. For each metric $q \in \{\text{mIoU}, \text{pAcc}\}$ and $r \in \{\text{Latency}, \text{Memory}\}$, let $q^{\max} = \max_j q_j$, $q^{\min} = \min_j q_j$ and $r^{\max} = \max_j r_j$, $r^{\min} = \min_j r_j$. The normalized scores are represented as follows:

$$q_k^{\text{norm}} = \begin{cases} \dfrac{q_k - q^{\min}}{q^{\max} - q^{\min}}, & q^{\max} > q^{\min}, \\ 1, & \text{otherwise}, \end{cases} \tag{1}$$

$$r_k^{\text{norm}} = \begin{cases} \dfrac{r^{\max} - r_k}{r^{\max} - r^{\min}}, & r^{\max} > r^{\min}. \\ 1, & \text{otherwise}. \end{cases} \tag{2}$$

Here, mIoU and pAcc are "*the higher the better*", while Latency and GPU Memory are flipped so that lower cost yields higher normalized scores. For this ablation, GPU Memory is constant across k, so its normalized term is identical for all rows and does not affect the ranking. The overall *Precision–Efficiency Score* is:

$$\begin{aligned} \text{PES}_k = \frac{1}{4}\big(&\text{mIoU}_k^{\text{norm}} + \text{pAcc}_k^{\text{norm}} \\ &+ \text{Latency}_k^{\text{norm}} + \text{Memory}_k^{\text{norm}}\big). \end{aligned} \tag{3}$$

On V21, PES peaks at CG=25, yielding clearly higher mIoU and pAcc compared to 5 or 15 iterations. However, further increasing CG to 35 or 45 brings only marginal accuracy gains while incurring noticeably higher latency, thus reducing the overall PES. We therefore fix the number of CG iterations to 25 for all experiments, as this provides an optimal trade-off between precision and efficiency.

Table B4. **Ablation analysis of a key-key self-correlation term.**

key-key	with BG			without BG					Avg.
	V21	PC60	Object	V20	PC59	Stuff	City	ADE	
w/o	63.2	34.7	36.8	86.4	38.3	26.1	34.6	18.7	42.4
w/	**64.1**	**35.1**	**37.3**	**86.9**	**38.6**	**26.3**	**37.6**	**19.4**	**43.2**

Table B5. **Ablation analysis of conjugate-gradient iterations on V21 [11] (*cf*. §3.3).** "PES" denotes the precision-efficiency score.

Iteration	Precision		Efficiency		PES
	mIoU	pAcc	Latency (ms/img)	Memory (GB)	
5	61.5	87.4	**31.7**	1.32	0.50
15	63.2	88.2	40.5	1.32	0.73
25	64.1	88.5	48.7	1.32	**0.80**
35	64.4	**88.7**	58.5	1.32	0.79
45	**64.6**	**88.7**	66.5	1.32	0.75

C. More Quantitative Results

Table C6 presents a quantitative comparison with recent open-vocabulary semantic segmentation methods. For all

comparison baselines, we keep their default post-processing (*e.g.*, PAMR [1] or DenseCRF [16]), while PEARL is evaluated with CLIP ViT-B/16 only and *without* any mask refinement. Even in this setting, PEARL achieves the highest average mIoU of **43.2%** across the eight benchmarks. It surpasses the strongest baseline, CASS [15] with DINOv3 (42.6%), by **0.6** points and outperforms other training-free CLIP-based methods, such as NACLIP [12] (42.5%) and SFP [13] (41.6%). This demonstrates that our alignment and propagation modules provide clear improvements without relying on stronger backbones or extra training data.

D. More Qualitative Results

As illustrated in Figs. D2-D9, we provide additional qualitative results of our PEARL on all eight benchmarks: Pascal VOC 21 (**V21**) [11], Pascal Context 60 (**PC60**) [19], COCO-Object (**Object**) [18], Pascal VOC 20 (**V20**) [11], Pascal Context 59 (**PC59**) [19], COCO-Stuff (**Stuff**) [3], Cityscapes (**City**) [7], and ADE20K (**ADE**) [26]. These examples include both successes and typical failure cases.

For each example, we show the input (**Image**), our prediction (**PEARL**), and the ground-truth (**GT**) mask. All visualizations utilize CLIP ViT-B/16 as the vision backbone, and no post-processing (*e.g.*, PAMR [1] or DenseCRF [16]) is applied. Therefore, the masks directly reflect the behavior of our training-free pipeline. On V21/V20 [11] and PC60/PC59 [19], our PEARL produces accurate object extents and clean boundaries for a wide variety of categories, including animals, vehicles, and artificial objects. The method remains robust under large appearance changes (*e.g.*, illumination and pose) and complex foreground-background compositions, and it preserves small details such as thin structures and disconnected parts in many cases. On the Object [18] and Stuff [3] datasets, our PEARL can localize both foreground instances and amorphous "*stuff*" regions, showing that the proposed Procrustes alignment and text-aware propagation generalize well from object-centric images to more cluttered scenes.

For the more challenging City [7] and ADE [26] benchmarks, our PEARL still captures the dominant layouts and most large regions (*road, building, sky, vegetation, cars*), but some fine-grained structures and rare categories are not perfectly segmented. The failure cases in figures mainly fall into several patterns: boundary leakage between adjacent regions, missing or fragmented small objects, and confusion between semantically related classes under cluttered scenes or weak visual evidence. These failure modes highlight the remaining gap between current open-vocabulary semantic segmentation and fully supervised models on large-scale, high-resolution urban or scene parsing datasets: long-range context, small distant objects, and heavily overlapping classes remain difficult to resolve using frozen backbones and text prompts alone. We hope that these visualizations will motivate future work on stronger open-vocabulary priors and the better exploitation of geometric and contextual cues in complex, real-world scenes.

References

[1] Nikita Araslanov and Stefan Roth. Single-stage semantic segmentation from image labels. In *CVPR*, pages 4253–4262, 2020. 3, 4, 5, 6, 7

[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*, pages 3689–3698, 2024. 4

[3] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and stuff classes in context. In *CVPR*, pages 1209–1218, 2018. 1, 3, 6

[4] 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*, pages 11165–11174, 2023. 4

[5] 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*, pages 3558–3568, 2021. 4

[6] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 4

[7] 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*, pages 3213–3223, 2016. 1, 3, 7

[8] Timothee Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *ICLR*, 2024. 4

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 4

[10] Karan Desai, Gaurav Kaul, Zubin Aysola, and Justin Johnson. Redcaps: Web-curated image-text data created by the people, for the people. *arXiv preprint arXiv:2111.11431*, 2021. 4

[11] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015. 1, 2, 3, 4, 6

[12] Sina Hajimiri, Ismail Ben Ayed, and Jose Dolz. Pay attention to your neighbours: Training-free open-vocabulary semantic segmentation. In *WACV*, pages 5061–5071, 2025. 3, 4

[13] Shuo Jin, Siyue Yu, Bingfeng Zhang, Mingjie Sun, Yi Dong, and Jimin Xiao. Feature purification matters: Suppressing outlier propagation for training-free open-vocabulary semantic segmentation. In *ICCV*, pages 20291–20300, 2025. 3, 4

[14] Dahyun Kang and Minsu Cho. In defense of lazy visual grounding for open-vocabulary semantic segmentation. In *ECCV*, pages 143–164, 2024. 4

Table C6. **Quantitative results of open-vocabulary semantic segmentation.** "Extra data" denotes external datasets (*e.g.*, CC3M [21], CC12M [5], RedCaps [10], COCO Captions [6, 18], and ImageNet-1K [9]), and "Extra backbone" lists auxiliary models. "Training-free" indicates no extra training. We evaluate prior methods using their default post-processing (official or re-implemented), while reporting PEARL without any post-processing. All metrics are mIoU (%). Best results are highlighted with **bold**, and second best with underlined.

Method	Pub. & Year	Extra data	Extra backbone	Training free	with background			without background					Avg.
					V21	PC60	Object	V20	PC59	Stuff	City	ADE	
w/ Mask Refinement													
GroupViT [25]	CVPR'22	CC12M+RedCaps	✗	✗	51.1	19.0	27.9	81.5	23.8	15.4	11.6	9.4	30.0
TCL [4]	CVPR'23	CC3M+CC12M	✗	✗	55.0	30.4	31.6	83.2	33.9	22.4	24.0	17.1	37.2
CLIP-DINOiser [24]	ECCV'24	ImageNet-1K	DINOv1 (ViT-B/16)	✗	**64.6**	33.5	36.1	81.5	37.1	25.3	31.5	20.6	41.3
ReCo [22]	NeurIPS'22	ImageNet-1K	✗	✓	27.2	21.9	17.3	62.4	24.7	16.3	22.8	12.4	25.6
FreeDA [2]	CVPR'24	COCO Captions	DINOv2 (ViT-B/14)	✓	52.0	**35.2**	25.8	79.5	**40.2**	**27.1**	34.4	**20.9**	39.4
LaVG [14]	ECCV'24	✗	DINOv1 (ViT-B/8)	✓	62.1	31.6	34.2	82.5	34.7	23.2	26.2	15.8	38.8
ProxyCLIP* [17]	ECCV'24	✗	DINOv2† (ViT-B/14)	✓	62.0	**35.2**	**38.7**	83.1	38.9	26.6	35.4	20.3	42.5
CASS* [15]	CVPR'25	✗	DINOv2† (ViT-B/14)	✓	58.7	32.6	33.3	86.4	35.9	24.0	34.2	17.9	40.4
CASS* [15]	CVPR'25	✗	DINOv3 (ViT-B/16)	✓	62.5	34.5	36.2	**87.1**	38.2	25.6	37.4	18.9	42.6
MaskCLIP [27]	ECCV'22	✗	✗	✓	37.2	22.6	18.9	72.1	25.3	15.1	11.2	9.0	26.4
SCLIP* [23]	ECCV'24	✗	✗	✓	61.7	31.5	32.1	83.5	36.1	23.9	34.1	17.8	40.1
NACLIP* [12]	WACV'25	✗	✗	✓	64.1	35.0	36.2	83.0	38.4	25.7	**38.3**	19.1	42.5
SFP* [13]	ICCV'25	✗	✗	✓	58.8	34.3	34.9	85.1	38.3	25.9	36.0	19.4	41.6
w/o Mask Refinement													
PEARL (Ours)		✗	✗	✓	64.1	35.1	37.3	86.9	38.6	26.3	**37.6**	19.4	**43.2**

Notes: "∗" denotes performance reproduced in this work. "†" indicates DINOv2 with registers [8].

Figure D2. **Qualitative results of open-vocabulary semantic segmentation.** Results are shown on the V21 [11] dataset. Our PEARL use CLIP ViT-B/16 [20], and no post-processing (*e.g.*, PAMR [1] or DenseCRF [16]) is applied for a fair comparison.

[15] 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*, pages 15033–15042, 2025. 3, 4

[16] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *NeurIPS*, pages 109–117, 2011. 3, 4, 5, 6, 7

[17] Mengcheng Lan, Chaofeng Chen, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. ProxyCLIP: Proxy attention improves CLIP for open-vocabulary segmentation. In *ECCV*, pages 70–88, 2024. 4

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755, 2014. 1, 3, 4, 5

[19] 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*, pages 891–898, 2014. 1, 3, 5, 6

[20] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry,

Figure D3. **Qualitative results of open-vocabulary semantic segmentation.** Results are shown on the PC60 [19] dataset. Our PEARL use CLIP ViT-B/16 [20], and no post-processing (*e.g.*, PAMR [1] or DenseCRF [16]) is applied for a fair comparison.

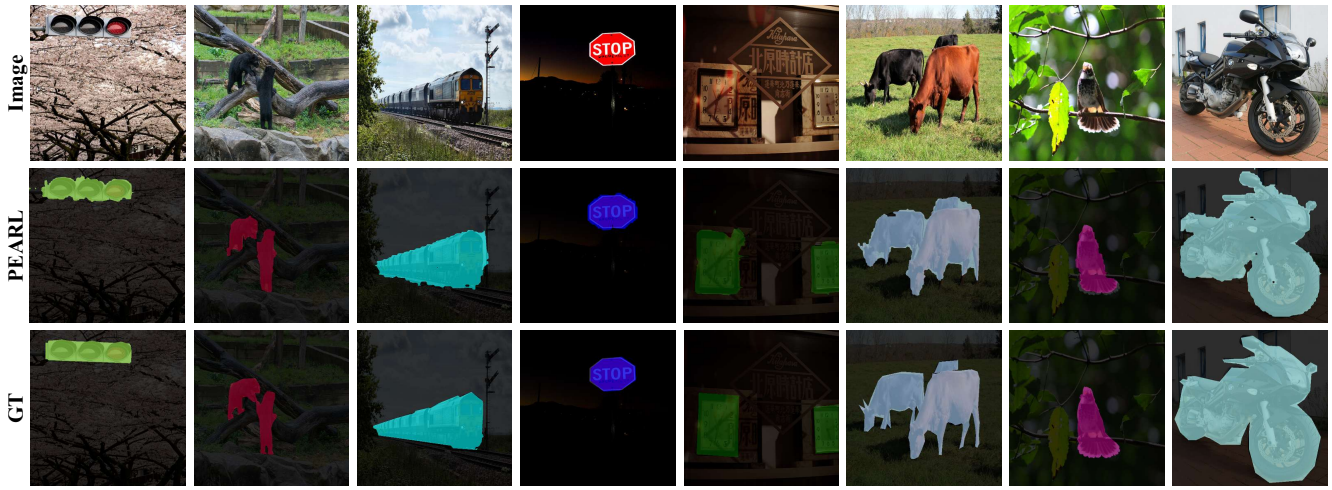

Figure D4. **Qualitative results of open-vocabulary semantic segmentation.** Results are shown on the Object [18] dataset. Our PEARL use CLIP ViT-B/16 [20], and no post-processing (*e.g.*, PAMR [1] or DenseCRF [16]) is applied for a fair comparison.

Amanda Askell, Pamela Mishkin, and Jack Clark. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021. 4, 5, 6, 7

[21] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, pages 2556–2565, 2018. 4

[22] Gyungin Shin, Weidi Xie, and Samuel Albanie. ReCo: Retrieve and co-segment for zero-shot transfer. In *NeurIPS*, pages 33754–33767, 2022. 4

[23] Feng Wang, Jieru Mei, and Alan Yuille. SCLIP: Rethinking self-attention for dense vision-language inference. In *ECCV*, pages 315–332, 2024. 4

[24] Monika Wysoczanska, Oriane Simeoni, Michaël Ramamonjisoa, Andrei Bursuc, Tomasz Trzcinski, and Patrick Perez. CLIP-DINOiser: Teaching CLIP a few DINO tricks for open-vocabulary semantic segmentation. In *ECCV*, pages 320–337, 2024. 4

[25] 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*, pages 18134–18144, 2022. 4

[26] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *CVPR*, pages 633–641, 2017. 1, 3, 7

[27] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from CLIP. In *ECCV*, pages 350–368, 2022. 4

Figure D5. **Qualitative results of open-vocabulary semantic segmentation.** Results are shown on the V20 [11] dataset. Our PEARL use CLIP ViT-B/16 [20], and no post-processing (*e.g.*, PAMR [1] or DenseCRF [16]) is applied for a fair comparison.

Figure D6. **Qualitative results of open-vocabulary semantic segmentation.** Results are shown on the PC59 [19] dataset. Our PEARL use CLIP ViT-B/16 [20], and no post-processing (*e.g.*, PAMR [1] or DenseCRF [16]) is applied for a fair comparison.

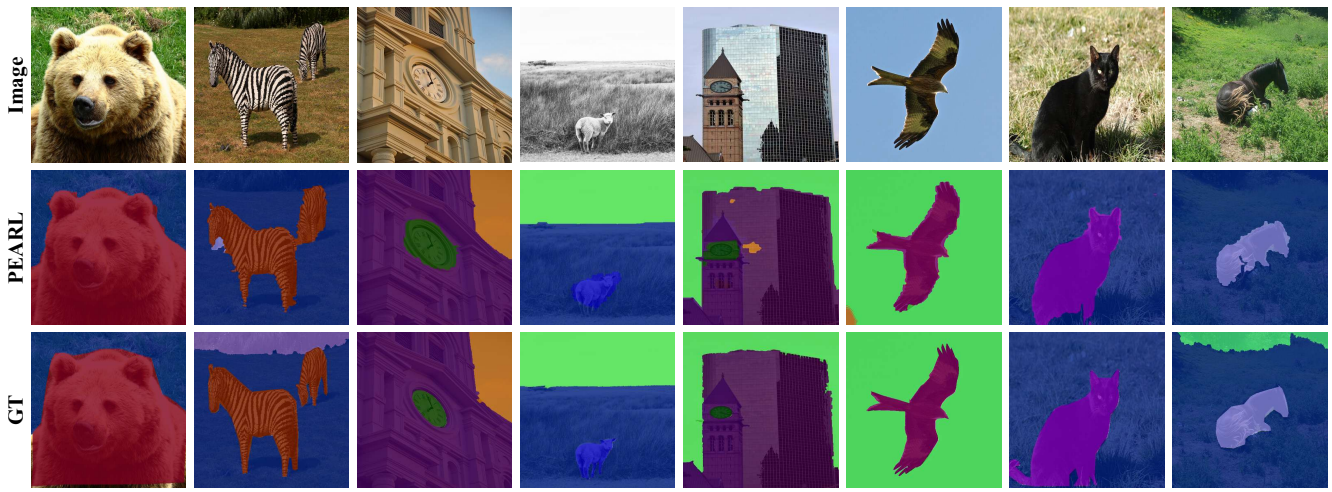

Figure D7. **Qualitative results of open-vocabulary semantic segmentation.** Results are shown on the Stuff [3] dataset. Our PEARL use CLIP ViT-B/16 [20], and no post-processing (*e.g.*, PAMR [1] or DenseCRF [16]) is applied for a fair comparison.

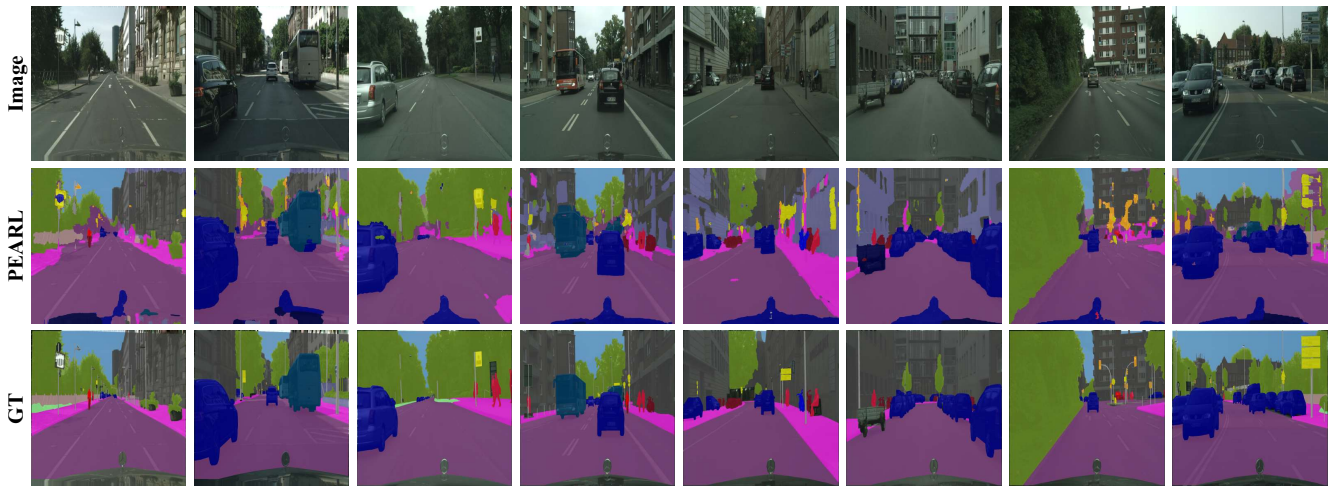

Figure D8. **Qualitative results of open-vocabulary semantic segmentation.** Results are shown on the City [7] dataset. Our PEARL use CLIP ViT-B/16 [20], and no post-processing (*e.g.*, PAMR [1] or DenseCRF [16]) is applied for a fair comparison.

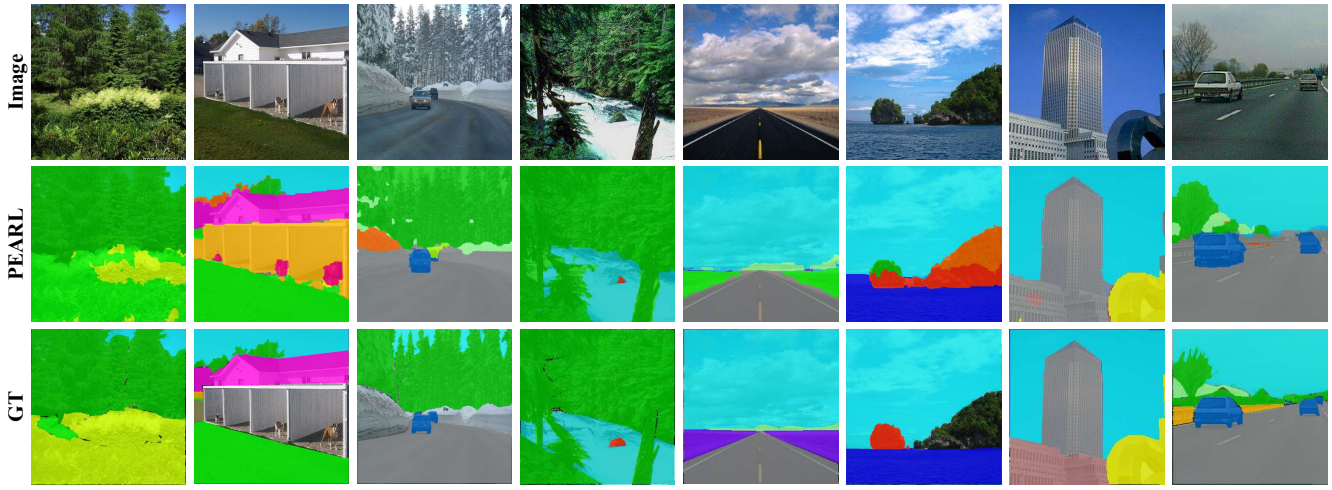

Figure D9. **Qualitative results of open-vocabulary semantic segmentation.** Results are shown on the ADE [26] dataset. Our PEARL use CLIP ViT-B/16 [20], and no post-processing (*e.g.*, PAMR [1] or DenseCRF [16]) is applied for a fair comparison.