ALIGN-R: RETHINKING RESIDUALS IN OPEN-VOCABULARY SEMANTIC SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In open-vocabulary semantic segmentation (OVSS), the community commonly discards the final residual connections to bridge the adaptation gap between pretrained vision-language models and dense prediction, yielding better segmentation mIoU. We rethink this widely adopted heuristic and uncover an overlooked trade-off: improved segmentation performance comes at the cost of recognition recall for categories present in the image. Pretrained residuals therefore carry meaningful pre-learned signals rather than pure noise, but align poorly with dense prediction objectives. From this observation, we identify two core failure modes: *spatial misalignment*, where semantic responses are displaced from target regions; and *conceptual misalignment*, where residual features produce false activations for categories absent from the image. To address both misalignments, we present **Align-R**, which preserves residual information via *explicit alignment* instead of naive removal. Align-R performs spatial alignment through semantic-center-guided random exploration to recover fine-grained spatial cues, and concept alignment via reliable-anchor distillation to preserve correct category concepts and intra-class relationships. Evaluated across 8 OVSS benchmarks, Align-R yields consistent gains for both training-based and training-free pipelines. Our results show that Align-R effectively mitigates residual misalignment and achieves a more favorable trade-off between recognition recall and segmentation performance. The code will be made publicly available.

1 INTRODUCTION

Open-vocabulary semantic segmentation (OVSS) (Li et al., 2022a; Xu et al., 2022; Cho et al., 2024) aims to recognize and segment categories beyond predefined label sets, enabling dense visual understanding in open-world scenarios. Recent OVSS methods commonly leverage vision-language models (VLMs), particularly Contrastive Language-Image Pre-training (CLIP) (Radford et al., 2021), to transfer open-vocabulary recognition capabilities from image-text pretraining to dense prediction. However, the **mismatch** between image-level pretraining and pixel-level prediction often produces noisy and spatially inconsistent dense responses. A widely adopted heuristic is therefore to discard the final-layer residual connection in the visual encoder. Owing to its simplicity and effectiveness, this *residual-removal* strategy has become a common practice for improving segmentation mIoU.

Despite its empirical effectiveness, we rethink residual removal by asking whether the discarded final-layer residual is *harmful noise* or contains *useful information* for OVSS. As shown in Fig. 1-b, residual removal yields coherent predictions but misses subtle background categories (*e.g.*, Arrow area). In contrast, retaining the residual (Fig. 1-c) preserves broader category cues but produces fragmented responses. This contrast reveals an overlooked trade-off: residual removal improves spatial coherence and segmentation score (mIoU = 0.17) while reducing recognition score (Recall = 0.75) for present categories (see Appendix A.2 for dataset-level statistical analysis). Without these cues, downstream OVSS modules lack enough information to find and segment these targets. Thus, residuals contain meaningful pretrained signals but are misaligned with dense prediction. This motivates us to perform *explicit alignment* instead of *naive removal*.

To understand this misalignment, we identify two fundamental forms as illustrated in Fig. 1-c–e. ❶ *Spatial misalignment* displaces or fragments semantic responses across object regions, producing spatially incoherent predictions. ❷ *Conceptual misalignment* spuriously activates absent categories,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

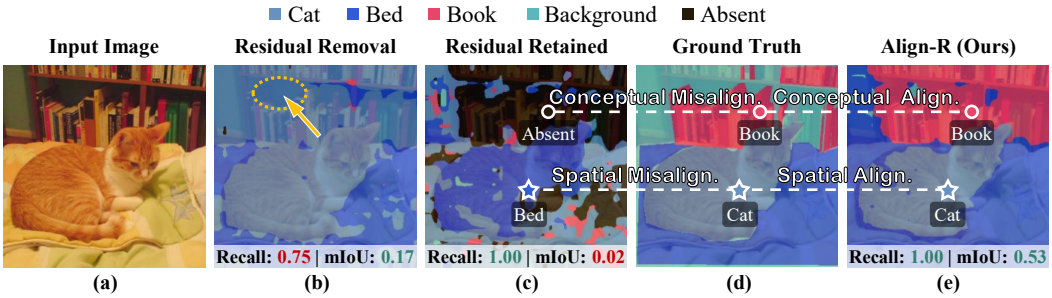


Figure 1: **Trade-off between recognition recall and segmentation mIoU in residuals.** (a) Input image. (b) Residual removal yields coherent predictions but misses a present category. (c) Retaining the residual improves recognition recall but introduces spatial and conceptual misalignment. (d) Ground truth. (e) Align-R addresses both forms of residual misalignment while preserving category cues. Recall measures recognition recall for present categories, while mIoU is computed on the displayed image. ■ Absent denotes activations assigned to absent categories.

impairing open-vocabulary discrimination. Taken together, this trade-off and these failure modes reveal that we cannot fully leverage residual features by simply keeping or discarding them. We therefore propose **Align-R**, whose two components explicitly address these failure modes. To address ❶, *Spatial Alignment* (SA) decouples global semantic calibration from local detail recovery. It uses semantic-center-guided random exploration to recover fine-grained spatial cues. To address ❷, *Concept Alignment* (CA) conducts concept distillation via reliable anchors at pointwise and relational levels. It preserves category directions and intra-class relations for residual alignment.

Built upon these two alignment modules, Align-R maintains a lightweight design with minimal computational cost. By keeping the entire CLIP model frozen and optimizing only a lightweight alignment module, Align-R enables efficient training with modest inference overhead. Across eight benchmarks, it consistently improves OVSS performance for CLIP models with different backbone scales and pretraining sources. Moreover, Align-R can be seamlessly integrated into both training-based and training-free OVSS methods without modifying their downstream pipelines. Together, these results establish Align-R as an efficient and broadly applicable plug-and-play framework that improves OVSS performance via explicit residual alignment.

The main contributions of this work are summarized as follows:

- We rethink the common practice of residual removal in OVSS and uncover an overlooked trade-off between recognition recall and segmentation performance. Discarding the final-layer residual boosts segmentation mIoU at the cost of lower recall for categories present in the image, showing that residuals carry meaningful pretrained semantics.
- We identify two fundamental forms of residual misalignment: spatial misalignment and conceptual misalignment. Accordingly, we propose **Align-R**, a lightweight framework that explicitly aligns residuals at both levels while retaining their useful semantic signals.
- Extensive evaluations across eight OVSS benchmarks yield consistent gains on both training-based and training-free pipelines over diverse visual backbones, verifying the effectiveness and generalizability of the proposed Align-R.

2 Related Work

VLMs. VLMs (Radford et al., 2021; Li et al., 2022b; Sun et al., 2023; Liu et al., 2023; Zhai et al., 2023; Tschannen et al., 2025; Ma et al., 2026) learn transferable visual-text representations from large-scale image-text data. Among them, CLIP (Radford et al., 2021) is widely adopted for its simple contrastive objective and strong zero-shot transfer. By aligning image and text representations, CLIP provides a scalable foundation for downstream open-vocabulary tasks, including object detection (Zhong et al., 2022; Minderer et al., 2023; Cheng et al., 2024), semantic segmentation (Xu et al., 2022; Cho et al., 2024), and other dense vision-language tasks (Zou et al., 2023a;b; Liu et al., 2026). However, such image-level pretraining encourages global semantic recognition but lacks

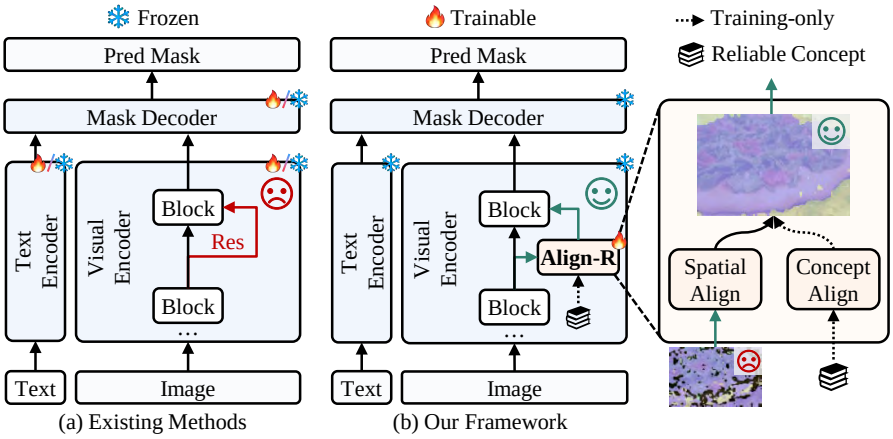


Figure 2: **Overview of Align-R for residual alignment in OVSS.** For unified illustration, we show residual branches in all paradigms. (a) Existing OVSS methods either discard or inherit misaligned residuals during dense transfer. (b) Align-R inserts a lightweight module to align residuals via spatial and concept alignment, preserving pretrained knowledge for dense segmentation.

explicit pixel-level localization, making VLM dense features suboptimal for segmentation. This supervision-granularity gap motivates representation alignment for OVSS.

OVSS. OVSS transfers the image-text alignment learned by pretrained VLMs to dense prediction. Existing methods generally follow training-based or training-free paradigms. Training-based methods adapt CLIP representations using task-specific decoders or trainable modules (Li et al., 2022a; Xu et al., 2022; 2023; Yu et al., 2023; Cho et al., 2024; Jiao et al., 2024; Xie et al., 2024; Liu et al., 2025; Li et al., 2025; Wang et al., 2025b), whereas training-free methods refine attention maps or token interactions to improve the spatial coherence of patch features (Zhou et al., 2022; Wang et al., 2024; Lan et al., 2024a; Jin et al., 2025; Yang et al., 2025; Hajimiri et al., 2025; Pei et al., 2026). Some approaches additionally leverage external foundation models to obtain stronger pixel-level spatial priors, although at the cost of increased inference overhead (Lan et al., 2024b; Karazija et al., 2024; Barsellotti et al., 2024; Kim et al., 2025; Shi et al., 2025; Zhu et al., 2026).

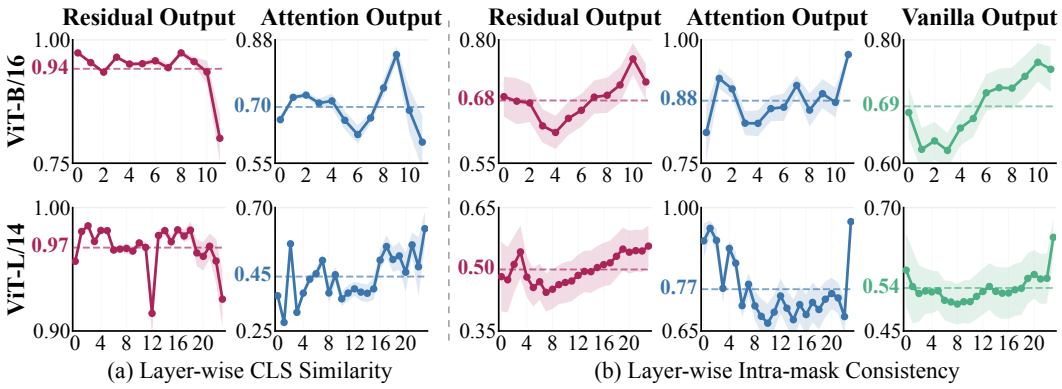
Despite their different designs, existing OVSS methods primarily bridge the gap between image-level pretraining and pixel-level prediction through downstream adaptation or inference-time modification, leaving residual representations largely underexplored. ClearCLIP (Lan et al., 2024a) shows that removing the final-layer residual suppresses noisy dense responses and improves segmentation, a strategy later adopted by several methods (Lan et al., 2024b; Hajimiri et al., 2025; Yang et al., 2025; Kim et al., 2025; Lee et al., 2026). As illustrated in Fig. 2, existing methods either discard residuals to obtain cleaner predictions or retain them without explicitly addressing their misalignment. Residual removal may discard useful pretrained semantics, whereas direct retention propagates misaligned responses into dense prediction. In contrast, Align-R explicitly aligns residual representations to preserve their pretrained semantics and adapt them for pixel-level prediction.

3 Preliminaries

3.1 CLIP for OVSS: Image-Level Alignment

Following common OVSS practice, we adopt CLIP with a ViT-based visual encoder. Given an image $I \in \mathbb{R}^{H\times W\times 3}$, the visual encoder divides it into M patches and prepends a class token, yielding $X_v^0 = [v_{\text{cls}}^0, v_1^0, \ldots, v_M^0]$. In the l-th block, we denote the residual output as $X_{\text{res}}^l = X_v^{l-1}$, while self-attention produces the attention output X_{attn}^l. Their sum is subsequently passed to the Feed-Forward Network (FFN) sublayer. After L blocks, the encoder produces $X_v^L = [v_{\text{cls}}^L, v_1^L, \ldots, v_M^L] \in \mathbb{R}^{(M+1)\times C}$, where v_{cls}^L is the image feature and $\{v_i^L\}_{i=1}^M$ are the patch features.

For N candidate categories, the text encoder produces embeddings $\{t_j\}_{j=1}^N$ from prompted class names, *e.g.*, "a photo of a [CLASS]." CLIP matches v_{cls}^L with these embeddings for image-level



(a) Layer-wise CLS Similarity (b) Layer-wise Intra-mask Consistency

Figure 3: **Layer-wise analysis of pretraining-oriented residual misalignment on COCO Object.** (a) CLS similarity measures the preservation of pretrained semantics. (b) Intra-mask consistency measures dense feature coherence within object regions. Shaded regions represent ± 1 standard deviation, while color-matched dashed lines indicate averages across layers.

recognition, whereas OVSS extends this matching to dense patch features:

$$\hat{Y} = \text{Upsample}(\text{Reshape}(\{\arg\max_j \cos(v_i^L, t_j)\}_{i=1}^M)). \quad (1)$$

Here, $\cos(\cdot,\cdot)$ denotes cosine similarity and $\hat{Y}$ is the predicted segmentation map. Since CLIP is optimized for image-level alignment, its dense features remain suboptimal for OVSS.

3.2 Residuals for OVSS: Pretraining-Oriented Misalignment

To understand how the image-level pretraining of CLIP affects dense prediction, we disentangle residual and attention outputs across ViT blocks and analyze their respective properties. We hypothesize that residuals retain pretrained semantic information but exhibit limited spatial coherence. We examine this hypothesis through class-token similarity and intra-mask feature consistency.

Residuals Preserve Pretraining-Oriented Semantics. The class token encodes image-level semantics learned during CLIP pretraining. We pass residual and attention outputs through the FFN and measure the cosine similarity, which quantifies their directional alignment, between their class tokens and that of the combined, vanilla output. As shown in Fig. 3-a, residual-derived class tokens exhibit higher layer-averaged similarity than attention-derived tokens, reaching 0.94 versus 0.70 for ViT-B/16 and 0.97 versus 0.45 for ViT-L/14. These results indicate that residual-derived class tokens remain directionally closer to the pretrained model's vanilla representations.

Residuals Lack Dense Semantic Consistency. We measure intra-mask consistency by averaging pairwise cosine similarities within each ground-truth mask. As shown in Fig. 3-b, residual outputs are less coherent than attention outputs, averaging 0.68 versus 0.88 for ViT-B/16 and 0.50 versus 0.77 for ViT-L/14. The vanilla output broadly follows the residual's layer-wise consistency trend, with similarly low averages of 0.69 and 0.54, respectively. This correspondence suggests that residual features substantially influence the within-mask coherence of the combined output, despite more coherent attention outputs. Together with Fig. 3-a, these results reveal high class-token similarity alongside limited spatial coherence, a contrast more pronounced in ViT-L/14.

Together, these results reveal pretraining-oriented residual misalignment: residuals preserve rich pretrained semantics that benefit the recognition of present categories but lack the spatial coherence required for dense segmentation. This finding motivates aligning rather than discarding residuals to effectively leverage their pretrained recognition information for pixel-level segmentation.

4 Methodology

This section presents the proposed Align-R framework. We first describe its architecture in Sec. 4.1, followed by *Spatial Alignment* in Sec. 4.2 and *Concept Alignment* in Sec. 4.3. Finally, Sec. 4.4 presents the overall training objective used to optimize Align-R.

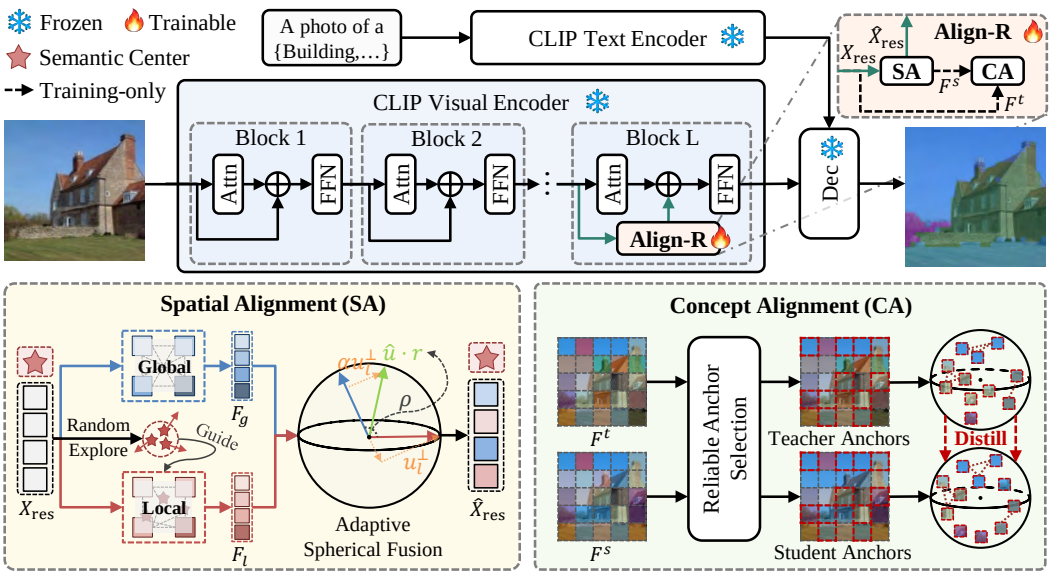


Figure 4: **Overall architecture of the proposed Align-R.** Align-R aligns the final-layer residuals of a frozen CLIP visual encoder through Spatial Alignment and Concept Alignment. For clarity, the remaining operations within each visual block are abstracted as an FFN block. Here, Dec denotes the simplest cosine-similarity decoder corresponding to Eq. 1.

4.1 Overall Architecture

As shown in Fig. 4, Align-R is inserted into the last visual block of a ViT-based CLIP encoder, while all CLIP components remain frozen. Let X_v^{L-1} denote the block input, which serves as the final-layer residual $X_{\text{res}}^L = X_v^{L-1}$. Align-R transforms this residual into an aligned representation:

$$\hat{X}_{\text{res}}^L = \text{Align-R}(X_{\text{res}}^L). \tag{2}$$

Afterward, X_{res}^L is replaced by $\hat{X}_{\text{res}}^L$ in the last ViT block and passed through subsequent frozen modules, producing residuals aligned for dense prediction without modifying the original pipeline.

4.2 Spatial Alignment

Spatial misalignment arises from the supervision granularity gap between pretraining and dense prediction, resulting in spatially displaced and fragmented semantic features. SA addresses this issue by decoupling global semantic calibration from local detail recovery. It constructs a stable global reference, recovers fine-grained spatial cues through Semantic-Center-Guided Random Exploration (SGRE), and combines the global and local responses using Adaptive Spherical Fusion (ASF).

For simplicity, we omit the superscript L in this subsection. Given $X_{\text{res}} = [x_{\text{cls}}, X_s] \in \mathbb{R}^{(M+1)\times C}$, SA computes $\bar{Q} = \phi_q(X_{\text{res}})$, $\bar{K} = \phi_k(X_{\text{res}})$, and $\bar{V} = \phi_v(X_{\text{res}})$ using lightweight projections $\phi_q, \phi_k, \phi_v : \mathbb{R}^C \to \mathbb{R}^D$. Their class components are denoted by $(\bar{q}_{\text{cls}}, \bar{k}_{\text{cls}}, \bar{v}_{\text{cls}})$ and spatial components by $(\bar{Q}_s, \bar{K}_s, \bar{V}_s)$. SA updates only the spatial tokens while keeping x_{cls} unchanged.

Global Semantic Reference. Although residuals preserve recognition-favorable semantics, they are spatially scattered and cannot directly support dense prediction. The global branch reorganizes them into a stable semantic reference for calibration. It uses spatial queries $\bar{Q}_s$ to attend to residual keys and values $\bar{K}, \bar{V}$, producing $F_g = \phi_g(\text{Softmax}(\bar{Q}_s \bar{K}^\top / \sqrt{D})\bar{V})$, where $\phi_g : \mathbb{R}^D \to \mathbb{R}^C$ restores the original dimension. This convex aggregation provides calibrated semantics for local alignment.

Semantic-Center-Guided Random Exploration. The global response F_g provides coarse semantic calibration but lacks locality for fine-grained spatial recovery. We therefore construct a complementary local response F_l from residual features. Specifically, the projected class token $\bar{v}_{\text{cls}}$, which encodes present semantics, serves as the global semantic center. We expand it into a 2D memory with randomly initialized learnable embeddings for semantic exploration. The resulting memory is

concatenated with spatial residual values and processed by deformable attention (Zhu et al., 2021):

$$\begin{aligned} \bar{V}_c &= \mathrm{Repeat}(\bar{v}_{\mathrm{cls}}) + E_c, \quad \bar{V}_e = [\bar{V}_c, \bar{V}_s], \\ F_l &= \phi_l(\mathrm{DAttn}(\bar{Q}_s, \bar{V}_e; \boldsymbol{p}_{\mathrm{ref}}, \mathcal{G})). \end{aligned} \tag{3}$$

Here, E_c denotes learnable embeddings, while $\boldsymbol{p}_{\mathrm{ref}}$ and $\mathcal{G} = \{(H_c, W_c), (H_s, W_s)\}$ specify reference points and memory shapes. Guided by the semantic center, deformable attention adaptively samples from the diversified semantic memory to produce F_l with fine-grained cues.

Adaptive Spherical Fusion. Directly adding the global response F_g and local response F_l may accumulate high-norm activations and disrupt feature propagation (Darcet et al., 2024). We therefore introduce ASF, which decouples direction and magnitude on the hypersphere. We normalize the responses as $u_g = F_g/\|F_g\|$ and $u_l = F_l/\|F_l\|$. The tangent component of u_l refines local details under the guidance of u_g, while their angular consistency determines the contribution of F_l to the fused magnitude. Specifically, we define $u_l^{\perp} = u_l - \langle u_l, u_g\rangle u_g$ and compute:

$$\begin{aligned} \rho &= \sigma(\exp(\gamma)\langle u_g, u_l\rangle + \beta), \\ \hat{u} &= \mathrm{Normalize}(u_g + \alpha u_l^{\perp}), \quad r = (1-\rho)\|F_g\| + \rho\|F_l\|. \end{aligned} \tag{4}$$

The fused response $F_o = \hat{u}r$ combines the rectified direction $\hat{u}$ with the consistency-aware magnitude r, where $\sigma(\cdot)$ denotes the sigmoid function, and learnable parameters α, γ, and β control the direction correction, consistency sensitivity, and magnitude weighting, respectively.

Residual Update. SA projects the fused response F_o through a lightweight FFN and adds it to the spatial residual tokens, while leaving the class token unchanged. This yields the aligned residual in Eq. 2, preserving image-level semantics while improving spatial consistency for dense prediction.

4.3 Concept Alignment

Although SA improves spatial consistency, conceptual misalignment may still persist during alignment training as spurious activations of absent categories. CA mitigates this by performing feature distillation over reliable anchors selected from the frozen teacher to preserve category directions and intra-class relations. Since value-side features provide rich patch-level cues (Liu et al., 2026), we use them as the teacher F^t and the SA output as the student F^s. Their logits P^t and P^s yield predictions $\hat{y}_i^t = \arg\max_j P_{i,j}^t$ and $\hat{y}_i^s = \arg\max_j P_{i,j}^s$, where Y_i is the patch label.

Reliable Anchor Selection. CA identifies reliable anchors through prediction and ground-truth agreement. The teacher-reliable and mutual-reliable anchor sets are defined as:

$$\mathcal{A}_t = \{i \mid \hat{y}_i^t = Y_i,\ Y_i \neq y_{\mathrm{ignore}}\}, \quad \mathcal{A}_m = \{i \mid \hat{y}_i^t = Y_i,\ \hat{y}_i^s = Y_i,\ Y_i \neq y_{\mathrm{ignore}}\}. \tag{5}$$

$\mathcal{A}_t$ provides trustworthy targets for pointwise direction alignment, whereas $\mathcal{A}_m$ limits relation distillation to mutually correct predictions, avoiding unreliable constraints.

Reliable Anchor Self-Distillation. To prevent spatial refinement from distorting pretrained category concepts, $\mathcal{L}_{\mathrm{rasd}}$ aligns student features with reliable teacher directions at anchors in $\mathcal{A}_t$ via cosine distance, thereby preserving category semantics without over-constraining spatial refinement:

$$\mathcal{L}_{\mathrm{rasd}} = \frac{1}{|\mathcal{A}_t|} \sum_{i \in \mathcal{A}_t} (1 - \cos(F_i^s, F_i^t)). \tag{6}$$

Reliable Anchor Relation Distillation. Beyond pointwise alignment, $\mathcal{L}_{\mathrm{rard}}$ distills intra-class relations among mutually reliable anchors. For $i, k \in \mathcal{A}_m$, we compute the teacher and student pairwise similarities as $S_{i,k}^t = \cos(F_i^t, F_k^t)$ and $S_{i,k}^s = \cos(F_i^s, F_k^s)$. Let $\tilde{S}^t$ and $\tilde{S}^s$ denote the corresponding similarities over same-class non-self anchor pairs in $\mathcal{A}_m$. The relation loss is:

$$\mathcal{L}_{\mathrm{rard}} = \mathrm{KL}\Big(\mathrm{Softmax}(\tilde{S}^t/\tau) \,\Big\|\, \mathrm{Softmax}(\tilde{S}^s/\tau)\Big), \tag{7}$$

where τ is a temperature that controls the sharpness of the distilled distribution.

4.4 Training Objective

The overall objective combines pixel-wise supervision with the two Concept Alignment losses:

$$\mathcal{L} = \mathcal{L}_{\mathrm{seg}} + \lambda_1 \mathcal{L}_{\mathrm{rasd}} + \lambda_2 \mathcal{L}_{\mathrm{rard}}, \tag{8}$$

where $\mathcal{L}_{\mathrm{seg}}$ denotes the pixel-wise cross-entropy loss, with λ_1 and λ_2 balancing the two distillation losses. Unless otherwise specified, we set $\lambda_1 = 1$ and $\lambda_2 = 0.5$.

5 EXPERIMENTS

5.1 EXPERIMENT SETTINGS

Datasets & Metric. We evaluate Align-R on eight benchmarks: PASCAL VOC (VOC21) (Everingham et al., 2010), PASCAL Context (PC60) (Mottaghi et al., 2014), and COCO Object (Object) (Caesar et al., 2018) with background categories; and PASCAL VOC20 (VOC20) (Everingham et al., 2010), PASCAL Context59 (PC59) (Mottaghi et al., 2014), COCO-Stuff (Stuff) (Caesar et al., 2018), Cityscapes (City) (Cordts et al., 2016), and ADE20K (ADE) (Zhou et al., 2017) without background categories. Mean Intersection-over-Union (mIoU) is used as the evaluation metric.

Implementation Details. Align-R is trained on COCO-Stuff and then plugged into various OVSS frameworks for evaluation on all eight datasets. Following CAT-Seg (Cho et al., 2024) and DeCLIP (Wang et al., 2025a), we train with an input resolution of 384×384 and the same augmentation and cropping strategies. The model is optimized with AdamW (Loshchilov & Hutter, 2019) for 80K iterations, using a batch size of 16 and a learning rate of 4×10^{-4} on two NVIDIA RTX 3090 GPUs. The entire CLIP model remains frozen throughout training, and only Align-R is optimized. Further implementation details are provided in Appendix A.4.

5.2 MAIN RESULTS

Table 1 reports the main comparison on eight OVSS datasets under a unified setting. We evaluate Align-R on both training-based and training-free frameworks to examine its compatibility with different paradigms. For training-based methods, Align-R is integrated into representative frameworks, including SAN, CAT-Seg, and DeCLIP, and consistently improves them without architectural modifications. This suggests that residual alignment complements existing segmentation architectures.

Table 1: **Effectiveness of Align-R on various OVSS frameworks across eight datasets.** For fair comparison, all methods are evaluated under the same setting. † denotes results reproduced from official implementations, and Avg. is computed over the reported datasets.

Methods	Backbone	With background			Without background					Avg.
		VOC21	PC60	Object	VOC20	PC59	Stuff	City	ADE	
Training-based Methods										
SAN (Xu et al., 2023)†	ViT-B/16	-	-	-	93.0	52.6	40.5	-	27.0	53.3
+ Align-R	ViT-B/16	-	-	-	93.7 +0.7	55.5 +2.9	43.9 +3.4	-	27.7 +0.7	55.2 +1.9
CAT-Seg (Cho et al., 2024)†	ViT-B/16	75.9	-	-	94.7	56.3	45.7	-	31.1	60.7
+ Align-R	ViT-B/16	77.6 +1.7	-	-	94.9 +0.2	57.7 +1.4	47.6 +1.9	-	31.8 +0.7	61.9 +1.2
DeCLIP (Wang et al., 2025a)†	ViT-B/16	80.9	-	-	96.5	59.7	48.6	-	34.9	64.1
+ Align-R	ViT-B/16	81.3 +0.4	-	-	96.7 +0.2	60.2 +0.5	49.4 +0.8	-	35.3 +0.4	64.6 +0.5
Training-free Methods										
ClearCLIP (Lan et al., 2024a)	ViT-B/16	51.8	32.6	33.0	80.9	35.9	23.9	30.0	16.7	38.1
+ Align-R	ViT-B/16	66.1 +14.3	44.0 +11.4	48.4 +15.4	90.3 +9.4	49.6 +13.7	39.0 +15.1	39.4 +9.4	22.9 +6.2	50.0 +11.9
ProxyCLIP (Lan et al., 2024b)	ViT-B/16	61.3	35.3	37.5	80.3	39.1	26.5	38.1	20.2	42.3
+ Align-R	ViT-B/16	74.1 +12.8	45.4 +10.1	52.1 +14.6	91.2 +10.9	50.7 +11.6	40.4 +13.9	43.0 +4.9	24.8 +4.6	52.7 +10.4
CASS (Kim et al., 2025)	ViT-B/16	65.8	36.7	37.8	87.8	40.2	26.7	39.4	20.4	44.4
+ Align-R	ViT-B/16	71.4 +5.6	44.0 +7.3	48.4 +10.6	92.3 +4.5	49.2 +9.0	38.7 +12.0	44.3 +4.9	23.9 +3.5	51.5 +7.1
PEARL (Pei et al., 2026)	ViT-B/16	64.1	35.1	37.3	86.9	38.6	26.3	37.6	19.4	43.2
+ Align-R	ViT-B/16	72.8 +8.7	44.7 +9.6	51.4 +14.1	91.6 +4.7	50.3 +11.7	40.2 +13.9	43.8 +6.2	24.7 +5.3	52.4 +9.2
GLA-CLIP (Lee et al., 2026)	ViT-B/16	66.5	35.9	37.7	84.2	39.8	27.0	39.9	20.0	43.9
+ Align-R	ViT-B/16	72.7 +6.2	44.3 +8.4	50.3 +12.6	90.1 +5.9	50.7 +10.9	40.3 +13.3	45.1 +5.2	24.7 +4.7	52.3 +8.4
ClearCLIP (Lan et al., 2024a)	ViT-L/14	46.1	26.7	30.1	80.0	29.6	19.9	27.9	15.0	34.4
+ Align-R	ViT-L/14	65.7 +19.6	45.6 +18.9	50.4 +20.3	92.8 +12.8	52.1 +22.5	41.2 +21.3	40.2 +12.3	24.3 +9.3	51.5 +17.1
ProxyCLIP (Lan et al., 2024b)	ViT-L/14	60.6	34.5	39.2	83.2	37.7	25.6	40.1	22.6	42.9
+ Align-R	ViT-L/14	74.9 +14.3	47.3 +12.8	55.2 +16.0	93.3 +10.1	53.4 +15.7	42.3 +16.7	44.4 +4.3	26.8 +4.2	54.7 +11.8
CASS (Kim et al., 2025)	ViT-L/14	61.2	34.8	35.5	89.3	38.5	25.8	40.1	20.8	43.3
+ Align-R	ViT-L/14	69.1 +7.9	45.1 +10.3	49.0 +13.5	93.5 +4.2	51.4 +12.9	40.5 +14.7	46.1 +6.0	24.9 +4.1	52.5 +9.2
PEARL (Pei et al., 2026)	ViT-L/14	53.9	31.4	33.9	85.7	34.6	23.7	36.6	19.6	39.9
+ Align-R	ViT-L/14	70.5 +16.6	47.3 +15.9	54.2 +20.3	93.0 +7.3	54.2 +19.6	43.1 +19.4	46.0 +9.4	26.5 +6.9	54.4 +14.5
GLA-CLIP (Lee et al., 2026)	ViT-L/14	63.5	35.0	35.8	87.2	38.9	26.6	38.6	21.9	43.4
+ Align-R	ViT-L/14	73.6 +10.1	46.6 +11.6	52.7 +16.9	92.8 +5.6	53.9 +15.0	42.8 +16.2	46.6 +8.0	26.4 +4.5	54.4 +11.0

For training-free methods, Align-R brings larger gains because these methods operate more directly on the original CLIP feature space and therefore benefit more from residual alignment. Across different training-free frameworks, Align-R achieves an average improvement of about 10 mIoU, with larger gains on ViT-L/14 than ViT-B/16. The gains are consistent across datasets with and without background categories, showing that Align-R improves residual alignment rather than overfitting to dataset-specific characteristics. Additional evaluations across different CLIP variants and backbone scales are provided in Appendix A.5. Overall, these results validate Align-R as an effective plug-and-play framework with strong generality across diverse OVSS methods.

5.3 Ablation Studies

We conduct ablation studies to evaluate the effectiveness of each proposed component in Align-R. Unless otherwise specified, all experiments use CLIP ViT-B/16 as the backbone. Additional experiments and analyses are provided in Appendix A.5 due to space constraints.

Effectiveness of Each Component. We ablate the proposed SA and the two CA objectives, $\mathcal{L}_{\text{rasd}}$ and $\mathcal{L}_{\text{rard}}$. As shown in Table 2, SA improves the average mIoU from 12.5 to 44.5, confirming the importance of residual alignment. Adding $\mathcal{L}_{\text{rasd}}$ or $\mathcal{L}_{\text{rard}}$ individually brings gains of 2.3 and 1.3 points by preserving reliable category directions and intra-class relations, respectively. Their benefits are particularly evident on Object, City, and ADE, where the complete CA improves SA by 5.0, 4.4, and 4.4 points. Combining both objectives achieves the best average mIoU of 47.2, outperforming SA by 2.7 points and demonstrating their complementarity.

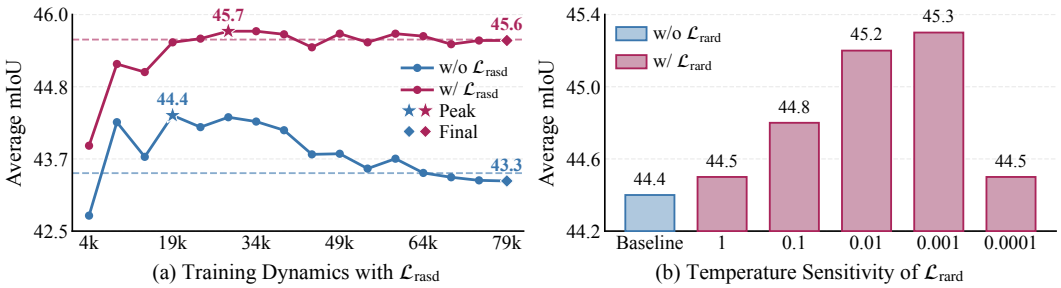
Table 2: Effectiveness of each component.

SA	CA		PC60	Object	VOC20	City	ADE	Avg.
	$\mathcal{L}_{\text{rasd}}$	$\mathcal{L}_{\text{rard}}$						
			7.7	5.5	41.8	5.5	2.1	12.5
✓			43.1	37.0	**91.7**	32.2	18.4	44.5
✓	✓		**43.9**	39.9	91.2	**36.8**	22.2	46.8
✓		✓	42.6	39.1	91.5	34.3	21.4	45.8
✓	✓	✓	43.4	**42.0**	91.3	36.6	**22.8**	**47.2**

Effectiveness of Spatial Alignment. Table 3 compares SA with alternative architectures and ablates its components. All alignment modules substantially improve over the baseline, highlighting the importance of residual alignment for dense prediction. Full SA achieves the highest average mIoU of 44.5, outperforming MLP and a ViT block by 1.5 and 1.1 points, respectively. These gains support the proposed design beyond generic trainable adaptation. Within SA, the global response F_g achieves 43.3 mIoU. Adding the local response F_l and subsequently ASF improves performance to 43.8 and 44.2, respectively, supporting the benefits of local cues and adaptive fusion. Finally, SGRE raises performance to 44.5. Together, these components improve over the global-only variant by 1.2 points, supporting their complementary contributions to residual alignment.

Table 3: Effectiveness of Spatial Alignment.

Method	PC60	Object	VOC20	City	ADE	Avg.
Baseline	7.7	5.5	41.8	5.5	2.1	12.5
MLP	40.6	**39.2**	88.8	29.8	16.7	43.0
ViT Block	42.1	36.2	91.6	30.2	16.8	43.4
F_g	41.7	35.5	90.1	**32.3**	17.0	43.3
F_g + F_l	42.2	36.8	90.5	31.8	17.7	43.8
F_g + F_l + ASF	42.7	37.0	91.5	31.7	18.2	44.2
Full SA	**43.1**	37.0	**91.7**	32.2	**18.4**	**44.5**



(a) Training Dynamics with $\mathcal{L}_{\text{rasd}}$ (b) Temperature Sensitivity of $\mathcal{L}_{\text{rard}}$

Figure 5: **Analysis of Concept Alignment.** (a) Training dynamics with and without $\mathcal{L}_{\text{rasd}}$. (b) Temperature sensitivity of $\mathcal{L}_{\text{rard}}$. Results are averaged across eight datasets.

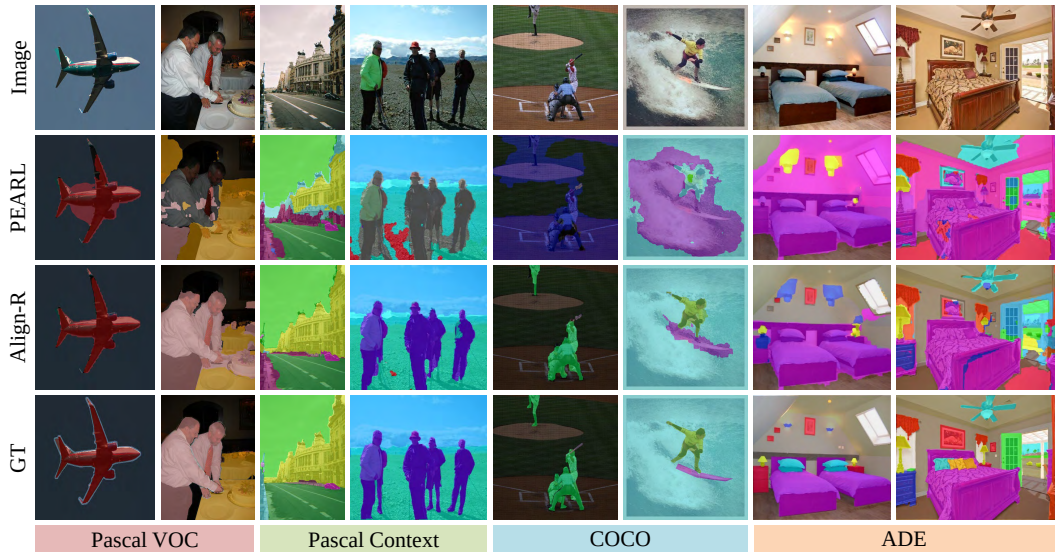


Figure 6: **Qualitative effects of Align-R across different OVSS datasets.** Examples are drawn from PASCAL VOC, PASCAL Context, COCO, and ADE20K. Predictions are shown before and after integrating Align-R into PEARL (Pei et al., 2026). Best viewed in color.

Effectiveness of Concept Alignment. Fig. 5 analyzes CA through the training dynamics of $\mathcal{L}_{\text{rasd}}$ and temperature sensitivity of $\mathcal{L}_{\text{rard}}$. Without $\mathcal{L}_{\text{rasd}}$, performance peaks at 44.4 mIoU but gradually declines to 43.3, indicating overfitting during later optimization. Introducing $\mathcal{L}_{\text{rasd}}$ raises the peak to 45.7 mIoU and maintains 45.6 at the final iteration, improving the final result by 2.3 points and stabilizing training. As shown in Fig. 5-b, $\mathcal{L}_{\text{rard}}$ improves over the 44.4 baseline across all evaluated temperatures, with $\tau = 0.001$ achieving the best result of 45.3 mIoU. The stronger results at moderate temperatures indicate that appropriately sharpened relation distributions provide more informative supervision for preserving intra-class structure. Together, these results validate the complementary roles of direction and relation distillation during residual alignment.

5.4 Visualization

Fig. 6 compares predictions from PEARL (Pei et al., 2026) before and after integrating Align-R. Align-R produces cleaner boundaries and more complete object regions on PASCAL VOC and PASCAL Context, while reducing fragmented responses in the more complex COCO scenes. It also better preserves fine structures and small objects that PEARL tends to miss or fragment. On ADE20K, Align-R improves mask coherence and category consistency across indoor regions. These results indicate that Align-R not only refines local boundaries but also corrects region-level semantic inconsistencies, demonstrating improvements in spatial localization and semantic discrimination. Additional qualitative comparisons across eight OVSS benchmarks are provided in Appendix A.5.2.

6 Conclusion

In this paper, we rethink the role of final-layer residuals in open-vocabulary semantic segmentation. We show that residuals are not merely harmful noise: they retain useful pretrained semantics that improve recognition recall for present categories but remain poorly aligned with OVSS. We attribute the resulting trade-off between recognition recall and segmentation performance to two forms of residual misalignment: spatial misalignment and conceptual misalignment. Motivated by these findings, we propose Align-R, a lightweight framework that aligns rather than removes residuals while keeping the entire CLIP model frozen. Align-R combines spatial alignment to recover fine-grained spatial cues with concept alignment to preserve reliable category directions and intra-class relations. Experiments across eight OVSS benchmarks demonstrate consistent gains for both training-based and training-free frameworks, validating residual alignment as an effective and generalizable strategy for harnessing pretrained residual semantics in OVSS.

AI USE STATEMENT

In this work, we used generative AI tools solely for editorial assistance, including polishing prose, correcting grammar and typography, tightening phrasing, and standardizing terminology. We did not use generative AI tools to generate research ideas, develop methods, conduct analyses, create figures or tables, produce or modify experimental results, write code, or modify data. All AI-assisted text was reviewed and verified by the authors, who take full responsibility for the final content of this work, including all claims, analyses, and artifacts.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed approach, experimental settings, datasets, baselines, and evaluation protocols in the main paper. The appendix provides additional implementation details, experimental results, analyses, and hyperparameter settings. We will release the source code and configuration files upon acceptance to facilitate reproduction of our results.

REFERENCES

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.

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

Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *CVPR*, 2024.

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.

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.

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.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *ICLR*, 2024.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

Sina Hajimiri, Ismail Ben Ayed, and Jose Dolz. Pay attention to your neighbours: Training-free open-vocabulary semantic segmentation. In *WACV*, 2025.

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.

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*, 2025.

Laurynas Karazija, Iro Laina, Andrea Vedaldi, and Christian Rupprecht. Diffusion models for open-vocabulary segmentation. In *ECCV*, 2024.

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.

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

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

ByeongCheol Lee, Hyun Seok Seong, Sangeek Hyun, Gilhan Park, WonJun Moon, and Jae-Pil Heo. Looking beyond the window: Global-local aligned clip for training-free open-vocabulary semantic segmentation. In *CVPR*, 2026.

Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. In *ICLR*, 2022a.

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

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.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

Yajie Liu, Jinjin Zhang, Qingjie Liu, and Di Huang. Reconstructing clip for open-vocabulary dense perception. In *CVPR*, 2026.

Yong Liu, Song-Li Wu, Sule Bai, Jiahao Wang, Yitong Wang, and Yansong Tang. Stepping out of similar semantic space for open-vocabulary segmentation. In *ICCV*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Qi Ma, Chen-Yang Wang, Dehong Gao, and Deng-Ping Fan. Alignednorm: Prompting vision–language models via coupled prompt field. In *ICML*, 2026.

Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. In *NeurIPS*, 2023.

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.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

Gensheng Pei, Xiruo Jiang, Xinhao Cai, Tao Chen, Yazhou Yao, and Byeungwoo Jeon. Pearl: Geometry aligns semantics for training-free open-vocabulary semantic segmentation. In *CVPR*, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

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

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

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

Junjie Wang, Bin Chen, Yulin Li, Bin Kang, Yichi Chen, and Zhuotao Tian. Declip: Decoupled learning for open-vocabulary dense perception. In *CVPR*, 2025a.

Xuehui Wang, Chongjie Si, Xue Yang, Yuzhi Zhao, Wenhai Wang, Xiaokang Yang, and Wei Shen. Opmapper: Enhancing open-vocabulary semantic segmentation with multi-guidance information. In *NeurIPS*, 2025b.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

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.

Hu Xu, Saining Xie, Xiaoqing Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data. In *ICLR*, 2024.

Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In *ECCV*, 2022.

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

Yuhang Yang, Jinhong Deng, Wen Li, and Lixin Duan. Resclip: Residual attention for training-free dense vision-language inference. In *CVPR*, 2025.

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.

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

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

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017.

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

Hao Zhu, Shuo Jin, Wenbin Liao, Jiayu Xiao, Yan Zhu, Siyue Yu, and Feng Dai. Vip: Visual-guided prompt evolution for efficient dense vision-language inference. In *ICML*, 2026.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021.

Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. In *CVPR*, 2023a.

Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. In *NeurIPS*, 2023b.

A Appendix

A.1 Limitations

Our analysis and Align-R are evaluated on ViT-based CLIP-style VLMs. Their applicability to CNN-based architectures remains unexplored, as convolutional blocks differ substantially from ViT blocks in their internal operations and residual pathways. Following the focus on final-layer residuals in prior work (Lan et al., 2024a), we restrict alignment to the last visual block to maintain a lightweight design. Extending residual alignment to multiple layers and assessing the trade-off between performance and efficiency remain directions for future work.

A.2 Statistical Analysis of the Overlooked Trade-off in Residuals

We quantitatively examine the trade-off between recognition recall and segmentation performance induced by final-layer residuals across eight OVSS benchmarks. Taking residual removal as the baseline, we measure the changes in recognition recall and segmentation mIoU after retaining the original residual or applying Align-R. Recognition is evaluated using class-level recall and precision:

$$\text{Recall} = \frac{|\mathcal{Y}_{\text{gt}} \cap \mathcal{Y}_{\text{pred}}|}{|\mathcal{Y}_{\text{gt}}|}, \qquad \text{Precision} = \frac{|\mathcal{Y}_{\text{gt}} \cap \mathcal{Y}_{\text{pred}}|}{|\mathcal{Y}_{\text{pred}}|}, \tag{9}$$

where

$$\mathcal{Y}_{\text{pred}} = \text{Unique}\left(\left\{\arg\max_j \cos(v_i, t_j)\right\}_{i=1}^{M}\right). \tag{10}$$

Here, $\mathcal{Y}_{\text{gt}}$ and $\mathcal{Y}_{\text{pred}}$ denote the ground-truth present categories and the predicted categories, respectively. v_i is the i-th visual patch feature, t_j is the text embedding of category j, and $\cos(v_i, t_j)$ denotes their cosine similarity. $\text{Unique}(\cdot)$ collects the distinct categories predicted over all M patches. Recall measures the coverage of present categories, whereas precision reflects the reliability of this coverage by penalizing predictions of absent categories.

As shown in Fig. 7, retaining the original residual improves recognition recall across all datasets by an average of 36.4 percentage points. This result confirms that residuals preserve category cues that are suppressed by residual removal. However, the corresponding segmentation mIoU decreases by an average of 11.1 points, showing that these cues cannot be directly transferred to dense prediction. Residual removal therefore improves spatial coherence at the cost of discarding information needed to recognize and segment less salient present categories.

The recognition recall gain alone does not indicate whether the additional category predictions are reliable. Because recall measures only present-category coverage, predicting a sufficiently broad category set can yield full recall even when many absent categories are activated. Table 4 reports the complementary precision results. Retaining the original residual yields a precision of only 20.3%, substantially below the 57.6% achieved by residual removal. The higher recall with residual retention is therefore accompanied by frequent predictions of absent categories, providing quantitative evidence of the conceptual misalignment targeted by Concept Alignment during residual adaptation for dense prediction.

Table 4: **Average recognition precision across eight OVSS datasets.**

Output	Precision (%)
Residual Removal	**57.6**
Residual Retained	20.3
Align-R (Ours)	45.4

Align-R achieves a more favorable trade-off by improving both recognition and segmentation performance. Relative to residual removal, it achieves average gains of 40.9 percentage points in recognition recall and 23.3 points in mIoU. Meanwhile, Align-R raises precision from 20.3% with the original residual to 45.4%, indicating that it retains broad present-category coverage while suppressing many absent-category predictions. Although residual removal attains higher precision by predicting a narrower category set, its substantially lower recall indicates that this reliability comes at the cost of missing present categories. Together, these results show that residual semantics are useful but unreliable when directly incorporated, whereas Align-R achieves a better balance between recognition and segmentation through spatial and concept alignment.

Representative Cases. Fig. 8 provides representative examples that intuitively explain the observed trade-off. The residual-removal output tends to focus on a few salient categories present in the

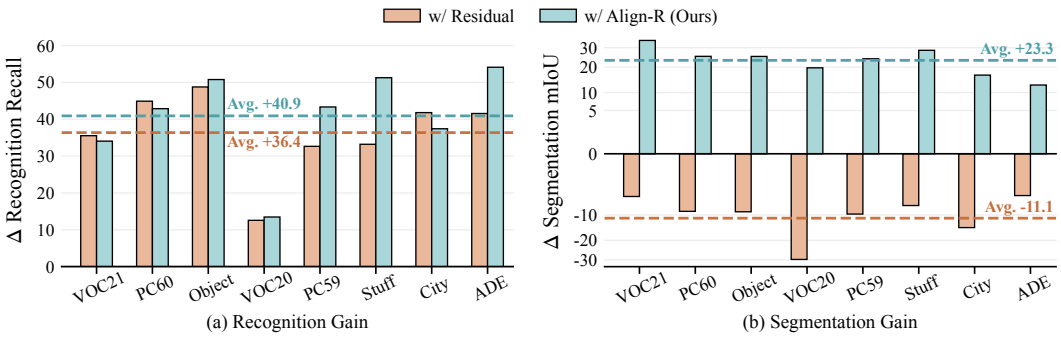


Figure 7: **Dataset-level analysis of the trade-off between recognition recall and segmentation mIoU in residuals.** Changes after retaining the original residual or applying Align-R are reported relative to residual removal across eight OVSS benchmarks. Color-matched dashed lines denote the average changes across datasets for the corresponding variants.

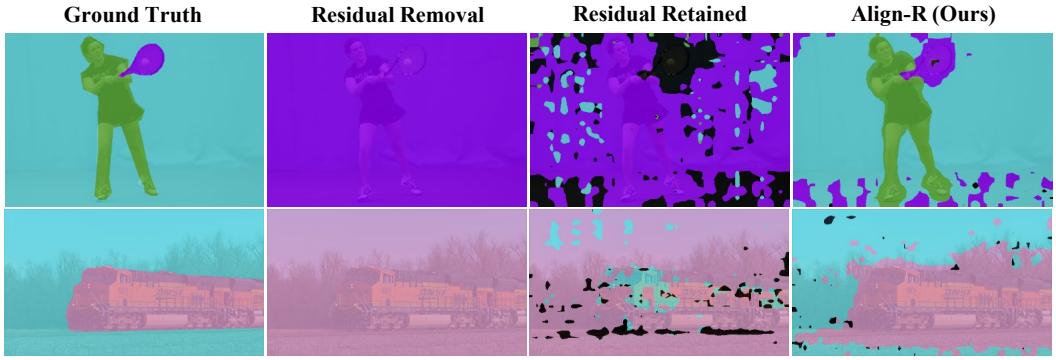


Figure 8: **Representative cases of the residual trade-off.** Residual removal produces spatially coherent predictions dominated by salient categories, whereas retaining the residual preserves broader cues for present categories but produces fragmented responses. Align-R preserves these recognition cues while improving their spatial coherence.

image, consistent with the image-level objective of vision-language pretraining, which encourages global representations dominated by semantics relevant to the paired text. When transferred to OVSS, this bias assigns large regions to these salient categories, producing spatially coherent masks. This tendency also helps explain the high recognition precision of residual removal: restricting predictions to a few salient present categories reduces absent-category predictions, although less salient categories may be missed. However, these overlooked categories may remain encoded in the residual branch. These residual cues provide broader category evidence but are spatially fragmented and thus unsuitable for dense prediction. Existing residual-removal strategies avoid these misaligned responses by discarding the residual, but simultaneously remove its useful semantics. In contrast, Align-R explicitly aligns these residual cues at the spatial and conceptual levels, converting them into coherent predictions while preserving evidence for less salient categories.

A.3 Statistical Analysis of Spatial and Conceptual Misalignment

To quantify the extent of spatial and conceptual misalignment across datasets and assess the effect of Align-R, we partition all valid pixels into three mutually exclusive sets: correct predictions, spatial misalignment, and conceptual misalignment. Let Y_i and $\hat{Y}_i$ denote the ground-truth and predicted labels at pixel i, respectively, and let $\mathcal{Y}_{\text{gt}}$ denote the set of ground-truth classes present in the image. Given the valid pixel set Ω, which excludes pixels assigned the ignore label, we define:

$$\begin{aligned}\Omega_{\text{cor}} &= \{i \in \Omega \mid \hat{Y}_i = Y_i\},\\ \Omega_{\text{spa}} &= \{i \in \Omega \mid \hat{Y}_i \in \mathcal{Y}_{\text{gt}},\ \hat{Y}_i \neq Y_i\},\\ \Omega_{\text{con}} &= \{i \in \Omega \mid \hat{Y}_i \notin \mathcal{Y}_{\text{gt}}\}.\end{aligned} \tag{11}$$

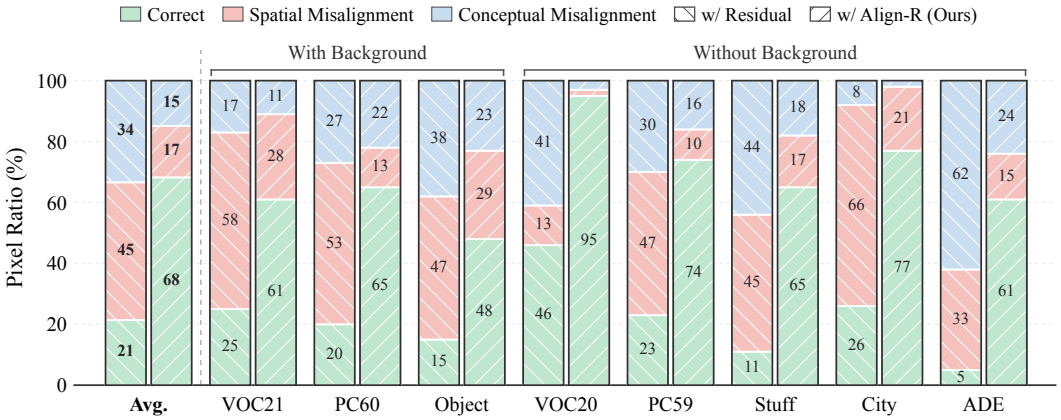


Figure 9: **Pixel-wise analysis of spatial and conceptual misalignment.** We report the proportions of correct, spatially misaligned, and conceptually misaligned pixels among all valid pixels for predictions obtained with the original residual and with Align-R. Pixel ratios are computed per image and averaged within each dataset. Avg. denotes the mean across the eight OVSS benchmarks.

Here, Ω_{spa} contains pixels assigned to present categories at incorrect spatial locations, whereas Ω_{con} contains predictions of classes absent from the image. Their pixel ratios are computed as:

$$r_k = \frac{|\Omega_k|}{|\Omega|}, \qquad k \in \{\text{cor}, \text{spa}, \text{con}\}, \tag{12}$$

where $r_{\text{cor}} + r_{\text{spa}} + r_{\text{con}} = 1$. We compute these ratios for each image and report their dataset-level averages for predictions obtained with the original residual and with Align-R.

Fig. 9 presents the pixel-wise distributions of correct predictions and the two forms of misalignment across eight benchmarks. The output with the original residual exhibits substantial spatial misalignment, with many pixels assigned to present categories at incorrect locations, confirming that residuals retain useful semantics but lack sufficient spatial consistency. It also produces predictions of absent categories, revealing conceptual deviations beyond inaccurate localization.

After alignment, the average proportion of correctly predicted pixels increases from 21% to 68%, a substantial gain of 47 percentage points. Meanwhile, spatial and conceptual misalignment decrease from 45% to 17% and from 34% to 15%, respectively. These improvements are consistent across datasets with and without background categories, demonstrating the robustness of Align-R under different label settings. The reduction in spatial misalignment indicates that scattered present semantics are reorganized into more spatially coherent predictions, while the decrease in conceptual misalignment reflects fewer false activations of absent categories. Their simultaneous reduction confirms that Align-R improves both spatial localization and semantic reliability without trading one form of misalignment for the other. Overall, these results demonstrate that Align-R effectively alleviates residual misalignment while retaining useful semantics for OVSS.

A.4 Additional Implementation Details

We implement Align-R using PyTorch (Paszke et al., 2019) and Detectron2 (Wu et al., 2019). Specifically, we build upon the CAT-Seg framework (Cho et al., 2024), replacing its cost volume decoder with CLIP's original cosine-similarity decoder to train the alignment module on COCO-Stuff. The entire CLIP model remains frozen, and only the alignment module is optimized. Once trained, Align-R is directly integrated into the official implementations of different OVSS frameworks, with their inference settings unchanged. Training configurations are provided in the main paper, and unspecified training settings follow CAT-Seg. The shared hidden dimension of Align-R is set to 192 across ViT-B, ViT-L, and ViT-H backbones and different CLIP pretraining sources. Its internal FFNs use an expansion ratio of 2, resulting in an intermediate dimension of 384. This compact configuration limits additional parameters and computation. Align-R uses 12, 16, and 16 attention heads for ViT-B, ViT-L, and ViT-H, respectively, matching the corresponding CLIP visual encoders. Deformable attention operates over two memory levels, corresponding to the expanded semantic-

center memory and spatial residual features, with four sampling points for each attention head at each memory level. For the MLP and ViT block baselines in the architecture ablation, we retain the original backbone feature dimension without channel compression and use an FFN expansion ratio of 4. These alternatives introduce more parameters and computation than SA but achieve lower average mIoU under the same training supervision, supporting the effectiveness and efficiency of the proposed alignment design beyond simply increasing trainable capacity.

A.5 Additional Experiments

This section presents additional quantitative and qualitative evaluations of Align-R across different OVSS frameworks and benchmark datasets, covering comparisons with related methods, backbone generalization, computational cost, and component ablations. Qualitative examples further illustrate its effects on spatial coherence and category prediction. Unless otherwise specified, all experiments use CLIP ViT-B/16 and follow the implementation details and evaluation protocols described above.

A.5.1 Additional Quantitative Results

Comparison with OPMapper. Among existing methods, OPMapper (Wang et al., 2025b) is closely related to Align-R because both adopt lightweight, plug-and-play designs. OPMapper trains a lightweight mapper with a standard CLIP backbone and subsequently integrates it into both training-based and training-free OVSS methods to assess cross-method generalization. It refines attention maps by promoting local compactness and global connectivity, thereby improving spatial coherence. Align-R instead targets residual representations and explicitly alleviates their spatial and conceptual misalignment. We evaluate Align-R on the same OVSS frameworks and backbone configurations reported for OPMapper, without modifying their downstream pipelines. Since OPMapper's official code is unavailable, we report its results directly from the original paper.

As shown in Table 5, OPMapper improves the corresponding baselines by 1.0 to 2.8 points in average mIoU, with a mean gain of 1.7 points across the four configurations. Align-R achieves substantially larger gains ranging from 7.1 to 17.1 points, with an average improvement of 11.6 points. Across individual benchmarks, Align-R consistently improves every evaluated baseline, whereas OPMapper slightly reduces performance on City with ProxyCLIP and PC59 with CASS. Under ClearCLIP, the gain of Align-R increases from 11.9 points with ViT-B/16 to 17.1 points with ViT-L/14, showing that its benefits extend across backbone scales. Overall, Align-R achieves larger and more consistent gains than the reported OPMapper results across the evaluated frameworks and backbone configurations, supporting its applicability to different OVSS pipelines.

Table 5: **Plug-and-play comparison with OPMapper.** Colored values denote absolute mIoU changes over the corresponding baseline, and Avg. is computed across eight OVSS benchmarks. The results of OPMapper are taken from the original paper.

Methods	Backbone	With background			Without background					Avg.
		VOC21	PC60	Object	VOC20	PC59	Stuff	City	ADE	
ClearCLIP (Lan et al., 2024a)	ViT-B/16	51.8	32.6	33.0	80.9	35.9	23.9	30.0	16.7	38.1
+ OPMapper	ViT-B/16	55.6 +3.8	34.8 +2.2	36.5 +3.5	84.7 +3.8	38.6 +2.7	25.9 +2.0	32.9 +2.9	18.1 +1.4	40.9 +2.8
+ Align-R (Ours)	ViT-B/16	66.1 +14.3	44.0 +11.4	48.4 +15.4	90.3 +9.4	49.6 +13.7	39.0 +15.1	39.4 +9.4	22.9 +6.2	50.0 +11.9
ProxyCLIP (Lan et al., 2024b)	ViT-B/16	61.3	35.3	37.5	80.3	39.1	26.5	38.1	20.2	42.3
+ OPMapper	ViT-B/16	62.8 +1.5	36.0 +0.7	38.7 +1.2	84.3 +4.0	40.1 +1.0	27.1 +0.6	37.2 -0.9	20.3 +0.1	43.3 +1.0
+ Align-R (Ours)	ViT-B/16	74.1 +12.8	45.4 +10.1	52.1 +14.6	91.2 +10.9	50.7 +11.6	40.4 +13.9	43.0 +4.9	24.8 +4.6	52.7 +10.4
CASS (Kim et al., 2025)	ViT-B/16	65.8	36.7	37.8	87.8	40.2	26.7	39.4	20.4	44.4
+ OPMapper	ViT-B/16	67.0 +1.2	38.2 +1.5	38.8 +1.0	90.0 +2.2	40.1 -0.1	27.1 +0.4	41.0 +1.6	20.8 +0.4	45.4 +1.0
+ Align-R (Ours)	ViT-B/16	71.4 +5.6	44.0 +7.3	48.4 +10.6	92.3 +4.5	49.2 +9.0	38.7 +12.0	44.3 +4.9	23.9 +3.5	51.5 +7.1
ClearCLIP (Lan et al., 2024a)	ViT-L/14	46.1	26.7	30.1	80.0	29.6	19.9	27.9	15.0	34.4
+ OPMapper	ViT-L/14	49.2 +3.1	28.7 +2.0	33.1 +3.0	81.5 +1.5	31.7 +2.1	21.3 +1.4	29.8 +1.9	16.1 +1.1	36.4 +2.0
+ Align-R (Ours)	ViT-L/14	65.7 +19.6	45.6 +18.9	50.4 +20.3	92.8 +12.8	52.1 +22.5	41.2 +21.3	40.2 +12.3	24.3 +9.3	51.5 +17.1

Generalization across CLIP Backbones and Pretraining Sources. Table 6 evaluates Align-R across three CLIP pretraining sources, namely CLIP (Radford et al., 2021), OpenCLIP (Cherti et al., 2023), and MetaCLIP (Xu et al., 2024), covering nine backbone configurations on eight OVSS benchmarks. Align-R improves every evaluated configuration on all eight datasets, with average mIoU gains ranging from 29.4 to 44.8 points. Beyond these gains, the resulting performance also

improves with backbone scale: average mIoU increases from 42.1 to 46.5 for OpenCLIP and from 44.5 to 48.3 for MetaCLIP when moving from ViT-B/16 to ViT-H/14. For CLIP, Align-R achieves 45.9 and 46.8 average mIoU with ViT-B/16 and ViT-L/14, respectively, and reaches 47.9 with ViT-L/14@336px. These results show that its benefits extend across pretraining sources, model scales, and resolution configurations. Notably, all variants share the same hidden dimension of 192 and FFN expansion ratio of 2, demonstrating that the compact alignment design remains effective across different backbone configurations. Together, these findings support the broad applicability of Align-R beyond a single CLIP model or pretraining source.

Table 6: **Generalization across CLIP backbones and pretraining sources.** Performance before and after incorporating Align-R is reported on eight OVSS benchmarks.

Method	Backbone	With background			Without background					Avg.
		VOC21	PC60	Object	VOC20	PC59	Stuff	City	ADE	
CLIP	ViT-B/16	16.2	7.7	5.5	41.8	9.2	4.4	5.5	2.1	11.6
+ Align-R	ViT-B/16	41.2+25.0	43.4+35.7	42.0+36.5	91.3+49.5	50.0+40.8	40.1+35.7	36.6+31.1	22.8+20.7	45.9+34.3
CLIP	ViT-L/14	8.7	3.7	2.6	15.8	4.5	2.4	2.9	1.3	5.2
+ Align-R	ViT-L/14	38.7+30.0	44.7+41.0	39.9+37.3	93.2+77.4	52.3+47.8	42.0+39.6	39.4+36.5	24.1+22.8	46.8+41.6
CLIP	ViT-L/14@336px	8.0	2.4	1.5	14.6	4.1	2.1	0.9	1.3	4.4
+ Align-R	ViT-L/14@336px	40.6+32.6	45.5+43.1	43.2+41.7	93.7+79.1	53.3+49.2	42.3+40.2	39.7+38.8	24.8+23.5	47.9+43.5
OpenCLIP	ViT-B/16	15.8	8.3	8.6	47.2	9.0	5.0	5.1	2.9	12.7
+ Align-R	ViT-B/16	30.3+14.5	40.9+32.6	29.7+21.1	90.7+43.5	47.5+38.5	38.9+33.9	35.6+30.5	23.5+20.6	42.1+29.4
OpenCLIP	ViT-L/14	13.7	7.1	7.7	42.9	7.8	4.7	1.9	2.8	11.1
+ Align-R	ViT-L/14	31.8+18.1	42.6+35.5	30.9+23.2	92.3+49.4	49.7+41.9	41.1+36.4	39.6+37.7	26.2+23.4	44.3+33.2
OpenCLIP	ViT-H/14	10.2	6.0	6.4	26.4	6.7	3.9	1.9	2.8	8.0
+ Align-R	ViT-H/14	36.0+25.8	45.2+39.2	33.4+27.0	93.6+67.2	53.0+46.3	42.1+38.2	41.4+39.5	27.6+24.8	46.5+38.5
MetaCLIP	ViT-B/16	14.8	7.1	3.9	38.0	8.9	4.9	2.5	2.8	10.4
+ Align-R	ViT-B/16	36.5+21.7	43.4+36.3	39.5+35.6	89.7+51.7	50.0+41.1	39.2+34.3	32.4+29.9	25.0+22.2	44.5+34.1
MetaCLIP	ViT-L/14	10.2	5.6	3.9	26.5	7.3	4.4	2.5	2.9	7.9
+ Align-R	ViT-L/14	38.6+28.4	45.1+39.5	41.4+37.5	93.0+66.5	52.9+45.6	43.0+38.6	40.8+38.3	27.5+24.6	47.8+39.9
MetaCLIP	ViT-H/14	6.2	1.6	1.3	11.2	3.3	1.4	1.2	1.5	3.5
+ Align-R	ViT-H/14	37.9+31.7	46.0+44.4	42.0+40.7	93.6+82.4	53.6+50.3	42.5+41.1	42.8+41.6	27.9+26.4	48.3+44.8

Computational Cost of Align-R. Table 7 summarizes the computational overhead of Align-R within the inference pipeline. Align-R adopts a shared 192-dimensional hidden space across backbones, balancing representation capacity and computational cost. The Concept Alignment losses are used only during training and introduce no inference overhead. Moreover, the CLIP backbone remains frozen, so only the lightweight alignment module requires parameter updates and optimizer states during training. Specifically, Align-R adds 2.0M and 3.2M parameters, 0.8G and 1.7G FLOPs, and 0.6 ms and 0.7 ms latency to ViT-B/16 and ViT-L/14, respectively, while improving average mIoU by 34.3 and 41.6 points. These additions correspond to parameter increases of approximately 1.3% and 0.7% and FLOPs increases of 2.3% and 1.0%, respectively. The relative parameter and FLOPs overhead therefore decreases for the larger backbone, while the additional inference latency remains below 1 ms under the measured setting. These results support the scalability of the compact alignment design across backbone sizes. Overall, Align-R substantially improves dense prediction performance with modest additional parameters and computational overhead.

Table 7: **Computational cost of Align-R.** Total parameters, FLOPs, and inference latency are reported for the complete inference pipeline. FLOPs are computed at a 224×224 input resolution with batch size 1. Latency is measured under the same setting on a single NVIDIA RTX 3090.

Method	Backbone	Params (M) ↓	FLOPs (G) ↓	Latency (ms) ↓	mIoU ↑
CLIP	ViT-B/16	149.6	35.2	5.3	11.6
+ Align-R	ViT-B/16	151.6	36.0	5.9	45.9+34.3
CLIP	ViT-L/14	427.6	162.1	16.9	5.2
+ Align-R	ViT-L/14	430.8	163.8	17.6	46.8+41.6

Effectiveness of Residual Semantics. The substantial gains of Align-R may arise from its training supervision rather than its use of residual information. We therefore examine whether residuals provide useful recognition information learned during pretraining. To this end, we replace the residual input with the attention output while keeping the architecture, losses, and training settings unchanged. As shown in Table 8, the attention-input improves the average mIoU from 11.6 to 40.0, demonstrating that supervised alignment also benefits attention features. Using residual representations further raises the average mIoU to 45.9, outperforming the attention-input by 5.9 points under identical supervision. This advantage holds across all eight benchmarks, with particularly large gains of 15.1 points on PC60, 7.3 on PC59, and 6.4 on City. Improvements occur on datasets both with and without background categories, indicating that the advantage is not confined to a particular label setting. These results show that the gains of Align-R cannot be attributed to supervision alone. Under the same alignment architecture and training protocol, residual representations provide a more effective source of information for dense prediction than attention outputs.

Table 8: **Comparison of different inputs to Align-R.** The two Align-R variants use identical supervision and training settings for a fair comparison.

Method	With background			Without background					Avg.
	VOC21	PC60	Object	VOC20	PC59	Stuff	City	ADE	
CLIP	16.2	7.7	5.5	41.8	9.2	4.4	5.5	2.1	11.6
Align-R w/ Attn	37.3+21.1	28.3+20.6	40.4+34.9	88.8+47.0	42.7+33.5	34.4+30.0	30.2+24.7	18.0+15.9	40.0+28.4
Align-R w/ Res	41.2+25.0	43.4+35.7	42.0+36.5	91.3+49.5	50.0+40.8	40.1+35.7	36.6+31.1	22.8+20.7	45.9+34.3

Effectiveness in Addressing Spatial and Conceptual Misalignment. Table 9 reports pixel ratios averaged across eight OVSS benchmarks, following the definitions in Sec. A.3. Adding SA reduces both error types, with spatial misalignment decreasing from 45.3% to 18.2% and conceptual misalignment from 33.5% to 18.6%. Adding CA further reduces these ratios to 16.8% and 14.9%, respectively. SA yields a larger absolute reduction in spatial errors, whereas the additional gains from CA are greater for conceptual errors. The correct prediction ratio increases from 21.2% to 63.2% with SA and reaches 68.3% with CA. These results support their complementary roles while showing that both components benefit the two misalignment types.

Table 9: **Effects of SA and CA on residual misalignment.** Pixel ratios (%) are averaged across eight OVSS benchmarks. Spatial and Conceptual denote the two misalignment types. Colored values indicate changes in percentage points relative to the preceding row.

Method	Correct (%) ↑	Spatial (%) ↓	Conceptual (%) ↓
CLIP	21.2	45.3	33.5
CLIP + SA	63.2+42.0	18.2-27.1	18.6-14.9
CLIP + SA + CA	68.3+5.1	16.8-1.4	14.9-3.7

Ablation on Adaptive Spherical Fusion. Table 10 evaluates different strategies for fusing the global response F_g and local response F_l in Spatial Alignment. Starting from the residual baseline, using F_g alone substantially improves the average mIoU from 12.5 to 43.3. Directly adding F_l further improves the performance to 43.8, indicating that local cues provide complementary fine-grained information. However, direct addition couples the direction and magnitude of the fused response, which may lead to suboptimal fusion when the two responses differ in scale or direction. ASF therefore decouples directional and magnitude information and fuses them separately, enabling more flexible integration of global semantics and local cues.

We first compare the directional fusion in ASF with two representative alternatives: normalized addition and spherical linear interpolation. As a simple direction-only baseline, normalized addition equally combines and renormalizes the unit directions of the two responses:

$$\hat{u}_{\text{norm}} = \frac{u_g + u_l}{\|u_g + u_l\|}, \qquad u_g = \frac{F_g}{\|F_g\|}, \quad u_l = \frac{F_l}{\|F_l\|}. \tag{13}$$

Spherical linear interpolation (Slerp) instead follows the geodesic connecting the two unit directions:

$$\hat{u}_{\text{slerp}} = \frac{\sin((1-\eta)\theta)}{\sin\theta} u_g + \frac{\sin(\eta\theta)}{\sin\theta} u_l, \qquad \theta = \arccos(u_g^\top u_l), \tag{14}$$

where η controls their relative contribution and θ denotes their angular distance. By comparison, ASF-Dir. projects the local direction onto the tangent space of the global direction, using the orthogonal component to refine local details while preserving global semantic guidance.

Normalized addition improves the average mIoU from 43.8 to 43.9, showing the benefit of directional decoupling. Although Slerp performs slightly better (44.1 mIoU v.s 43.9 mIoU), it requires complex trigonometric operations and incurs higher computational overhead, whereas ASF-Dir. achieves comparable performance using only algebraic operations. Full ASF further incorporates consistency-aware magnitude fusion and reaches the best average mIoU of 44.2, improving its direction-only variant by 0.2 mIoU. These results support the effectiveness of decoupling direction and magnitude and the complementary benefit of adaptive magnitude fusion.

Table 10: **Ablation of global-local fusion strategies in Spatial Alignment.** Dir. and Mag. denote direction and magnitude decoupling, respectively.

Method	Decoupling		PC60	Object	VOC20	City	ADE	Avg.
	Dir.	Mag.						
Baseline	–	–	7.7	5.5	41.8	5.5	2.1	12.5
F_g	–	–	41.7	35.5	90.1	32.3	17.0	43.3
$F_g + F_l$ (Addition)	×	×	42.2	36.8	90.5	31.8	17.7	43.8
$F_g + F_l$ (Norm.)	✓	×	42.6	36.0	90.7	**32.4**	17.6	43.9
$F_g + F_l$ (Slerp)	✓	×	42.6	36.6	91.7	31.9	17.6	44.1
$F_g + F_l$ (ASF-Dir.)	✓	×	**42.7**	36.6	**91.9**	31.3	17.7	44.0
$F_g + F_l$ (Full ASF)	✓	✓	**42.7**	**37.0**	91.5	31.7	**18.2**	**44.2**

Ablation on Semantic-Center Expansion Size. Table 11 evaluates different semantic-center expansion sizes in SGRE. Compared with SA without SGRE, all tested sizes improve the average mIoU from 44.2 to at least 44.4. The 3×3 expansion achieves the highest average mIoU of 44.5, while both 2×2 and 4×4 achieve 44.4. Performance varies only slightly across these settings, and increasing the size beyond 3×3 provides no further gain in average mIoU. With a hidden dimension of 192, the 2×2, 3×3, and 4×4 memories introduce approximately 0.8K, 1.7K, and 3.1K learnable embedding parameters, respectively. The default 3×3 setting adds a semantic memory level containing only nine tokens, requiring little additional computation for memory processing and deformable sampling. We therefore adopt 3×3 as the default setting, achieving the best observed average performance with few additional parameters and low computational overhead.

Table 11: **Ablation of semantic-center expansion sizes in SGRE.**

Method	PC60	Object	VOC20	City	ADE	Avg.
SA w/o SGRE	42.7	37.0	91.5	31.7	18.2	44.2
SA w/ SGRE-2×2	43.0	36.9	91.5	31.9	**18.5**	44.4
SA w/ SGRE-3×3	**43.1**	37.0	**91.7**	**32.2**	18.4	**44.5**
SA w/ SGRE-4×4	43.0	**37.1**	**91.7**	31.9	**18.5**	44.4

Ablation on the Weight of Relation Distillation. Table 12 evaluates the two distillation losses individually with unit weights and varies λ_2 while fixing $\lambda_1 = 1$ when both losses are used. Adding $\mathcal{L}_{\text{rasd}}$ or $\mathcal{L}_{\text{rard}}$ alone improves the average mIoU from 44.5 to 46.8 and 45.8, respectively. Combining both losses achieves at least 47.1 across the tested weights, supporting their complementary contributions. The average mIoU varies only from 47.1 to 47.2, suggesting limited sensitivity to λ_2 within the evaluated range. With $\lambda_2 = 0.5$, the model matches the best average mIoU of 47.2 and achieves the highest ADE score of 22.8. We therefore adopt $\lambda_2 = 0.5$ unless otherwise specified.

Table 12: **Ablation on the relation distillation weight.** Each loss uses a unit weight when evaluated alone. When both losses are used, $\lambda_1 = 1$ and λ_2 varies.

Method	PC60	Object	VOC20	City	ADE	Avg.
SA	43.1	37.0	**91.7**	32.2	18.4	44.5
SA $+\mathcal{L}_{\text{rasd}}$ ($\lambda_1 = 1.0$)	**43.9**	39.9	91.2	**36.8**	22.2	46.8
SA $+\mathcal{L}_{\text{rard}}$ ($\lambda_2 = 1.0$)	42.6	39.1	91.5	34.3	21.4	45.8
Both ($\lambda_2 = 1.0$)	43.3	**42.1**	91.1	36.7	22.6	**47.2**
Both ($\lambda_2 = 0.5$)	43.4	42.0	91.3	36.6	**22.8**	**47.2**
Both ($\lambda_2 = 0.1$)	43.7	41.0	91.5	36.6	22.6	47.1

A.5.2 Additional Qualitative Results

Figs. 10–17 provide additional qualitative comparisons on PASCAL VOC, PASCAL Context, COCO, Cityscapes, and ADE20K. We integrate Align-R into PEARL (Pei et al., 2026) and compare its predictions with the original framework across diverse OVSS scenarios. In the displayed examples, Align-R produces sharper object boundaries and better preserves fine-grained structures, particularly in regions with complex shapes or ambiguous contours. It also reduces fragmented patch responses and spurious activations, yielding more complete object regions and more accurate category assignments. Beyond local improvements, the resulting masks exhibit greater spatial coherence and more consistent category predictions within object regions. The examples span object-centric images, complex indoor scenes, and urban environments, illustrating the effects of residual alignment across different scene types. Although some boundaries and category assignments remain imperfect, the qualitative improvements complement the quantitative gains and illustrate the benefits of Align-R for spatial localization and semantic discrimination.

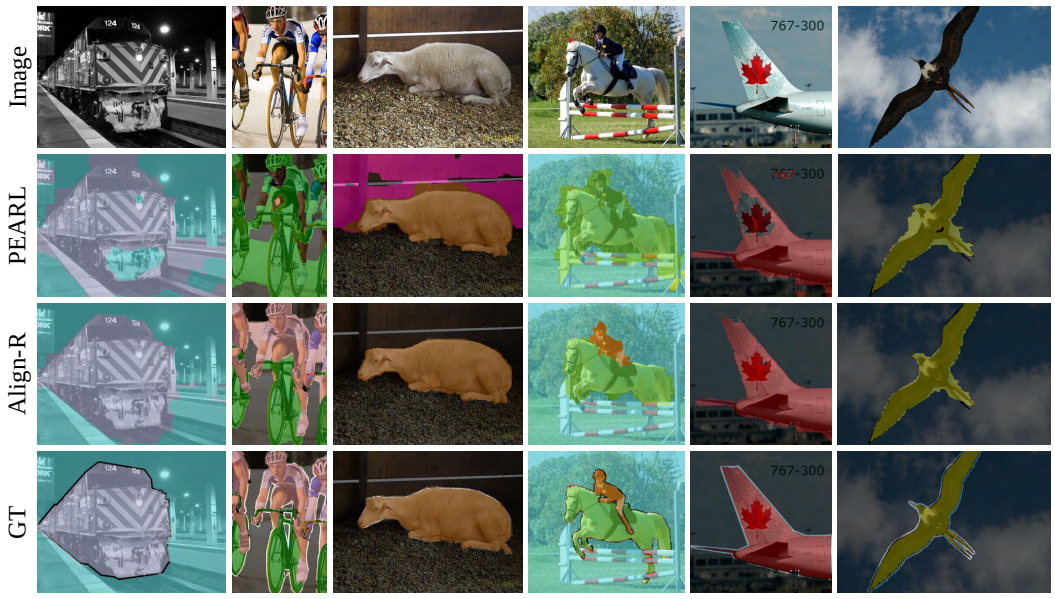


Figure 10: **Additional qualitative results on VOC21** (Everingham et al., 2010). We compare PEARL (Pei et al., 2026) with PEARL equipped with Align-R using the CLIP ViT-B/16 backbone (Radford et al., 2021).

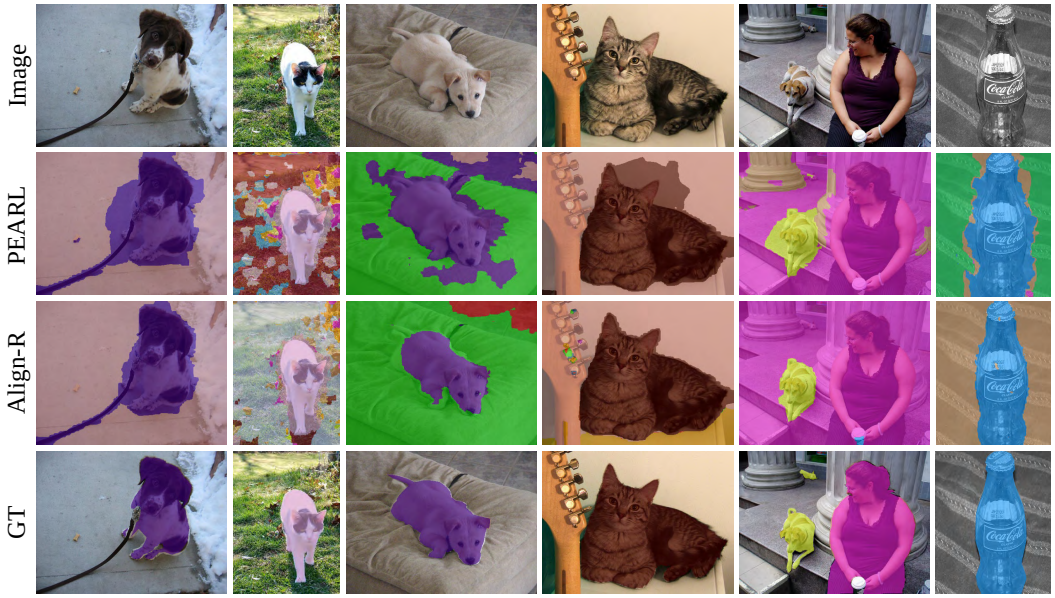


Figure 11: **Additional qualitative results on VOC20** (Everingham et al., 2010). We compare PEARL (Pei et al., 2026) with PEARL equipped with Align-R using the CLIP ViT-B/16 backbone (Radford et al., 2021).

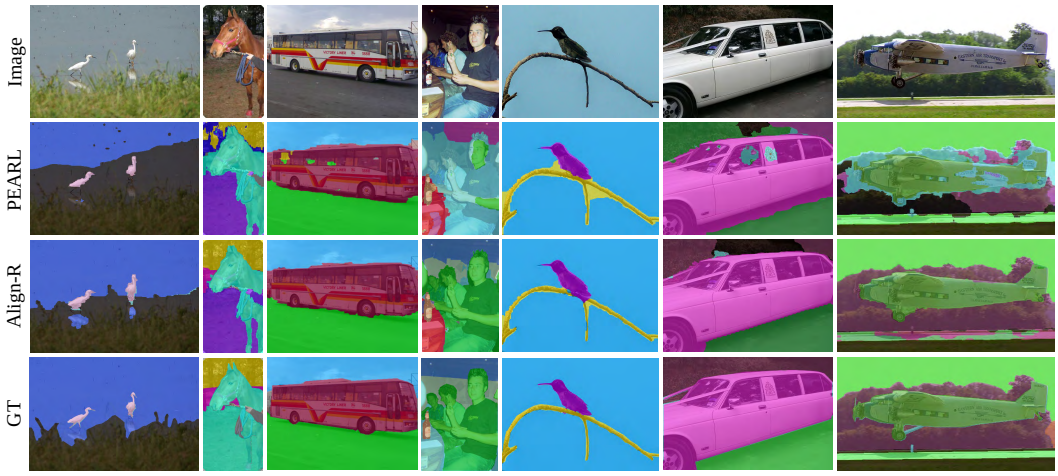


Figure 12: **Additional qualitative results on PC60** (Mottaghi et al., 2014). We compare PEARL (Pei et al., 2026) with PEARL equipped with Align-R using the CLIP ViT-B/16 backbone (Radford et al., 2021).

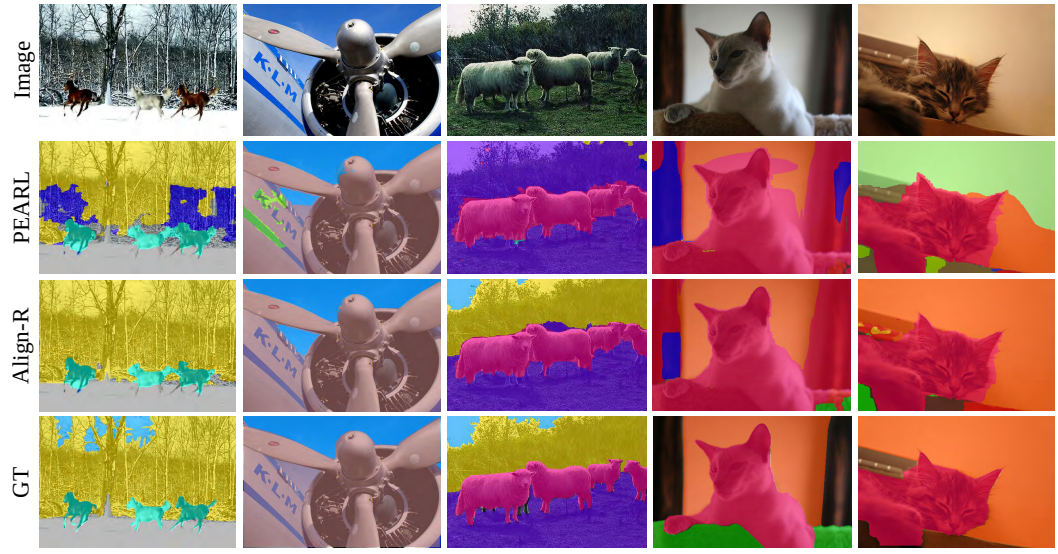


Figure 13: **Additional qualitative results on PC59** (Mottaghi et al., 2014). We compare PEARL (Pei et al., 2026) with PEARL equipped with Align-R using the CLIP ViT-B/16 backbone (Radford et al., 2021).



Figure 14: **Additional qualitative results on COCO Object** (Caesar et al., 2018). We compare PEARL (Pei et al., 2026) with PEARL equipped with Align-R using the CLIP ViT-B/16 backbone (Radford et al., 2021).

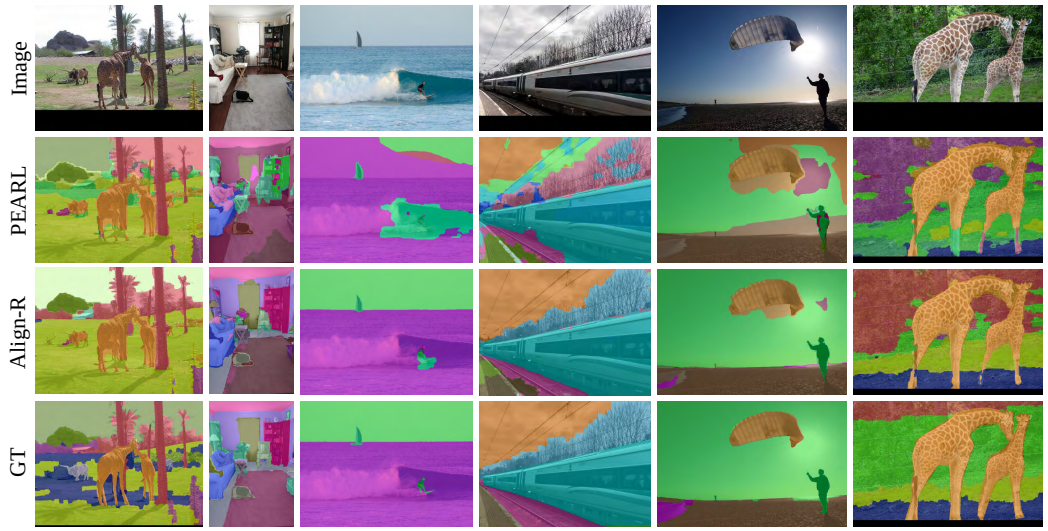


Figure 15: **Additional qualitative results on COCO-Stuff** (Caesar et al., 2018). We compare PEARL (Pei et al., 2026) with PEARL equipped with Align-R using the CLIP ViT-B/16 backbone (Radford et al., 2021).

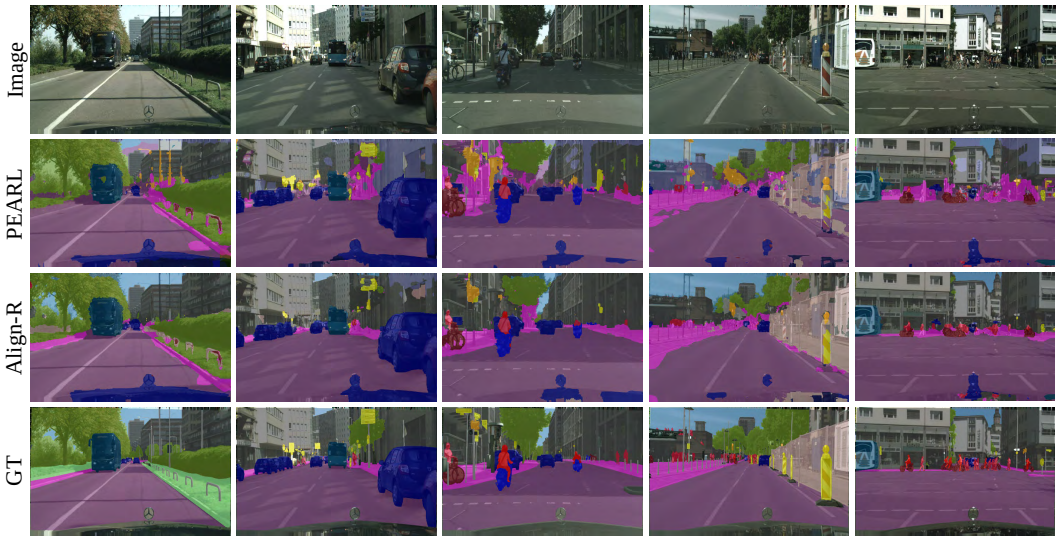


Figure 16: **Additional qualitative results on Cityscapes** (Cordts et al., 2016). We compare PEARL (Pei et al., 2026) with PEARL equipped with Align-R using the CLIP ViT-B/16 backbone (Radford et al., 2021).



Figure 17: **Additional qualitative results on ADE20K** (Zhou et al., 2017). We compare PEARL (Pei et al., 2026) with PEARL equipped with Align-R using the CLIP ViT-B/16 backbone (Radford et al., 2021).