

dinov3.seg: Open-Vocabulary Semantic Segmentation with DINOv3

Saikat Dutta^{1,2,3}, Biplab Banerjee², and Hamid Rezaatofighi³

¹ IITB-Monash Research Academy

² IIT Bombay

³ Monash University

Abstract. Open-Vocabulary Semantic Segmentation (OVSS) assigns pixel-level labels from an open set of text-defined categories, demanding reliable generalization to unseen classes at inference. Although modern vision–language models (VLMs) support strong open-vocabulary recognition, their representations learned through global contrastive objectives remain suboptimal for dense prediction, prompting many OVSS methods to depend on limited adaptation or refinement of image–text similarity maps. This, in turn, restricts spatial precision and robustness in complex, cluttered scenes. We introduce `dinov3.seg`, extending `dinov3.txt` into a dedicated framework for OVSS. Our contributions are four-fold. First, we design a task-specific architecture tailored to this backbone, systematically adapting established design principles from prior open-vocabulary segmentation work. Second, we jointly leverage text embeddings aligned with both the global [CLS] token and local patch-level visual features from ViT-based encoder, effectively combining semantic discrimination with fine-grained spatial locality. Third, unlike prior approaches that rely primarily on post hoc similarity refinement, we perform early refinement of visual representations prior to image–text interaction, followed by late refinement of the resulting image–text correlation features, enabling more accurate and robust dense predictions in cluttered scenes. Finally, we propose a high-resolution local–global inference strategy based on sliding-window aggregation, which preserves spatial detail while maintaining global context. We conduct extensive experiments on five widely adopted OVSS benchmarks to evaluate our approach. The results demonstrate its effectiveness and robustness, consistently outperforming current state-of-the-art methods.

1 Introduction

Open-Vocabulary Semantic Segmentation (OVSS) extends conventional semantic segmentation by allowing models to predict pixel-wise labels beyond a fixed, closed set of training categories. This flexibility is crucial for real-world deployments in robotics, autonomous driving, remote sensing, and medical imaging, where the label space is long-tailed, dynamic, and costly to annotate exhaustively. Despite rapid progress in vision–language models (VLMs) [12, 29] for open-vocabulary recognition, translating their global image–text alignment into

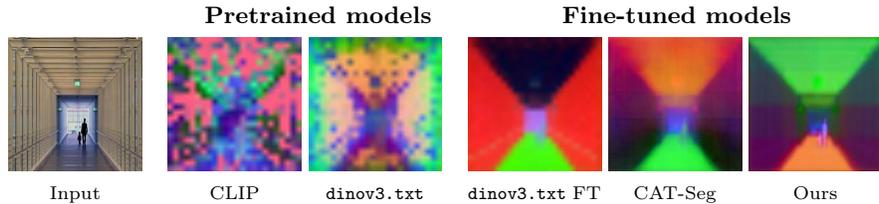


Fig. 1: Visualization of final visual features across different models. Pretrained models exhibit noisy features, while `dinov3.txt` produces comparatively cleaner features with more structural information than CLIP. Fine-tuned `dinov3.txt` features yield sharper boundaries, but fail to capture fine-grained details. CLIP-based OVSS approach CAT-Seg recovers more detail in the features, yet produces less well-defined boundaries. In contrast, our method produces features with sharp boundaries and rich fine-grained detail.

dense, fine-grained segmentation remains challenging. Representations learned primarily through global contrastive objectives often under-emphasize locality and spatial detail, which are essential for pixel-level discrimination—especially in cluttered scenes and high-resolution imagery.

Most OVSS pipelines [7, 13, 21, 22, 28, 40, 42, 46] build on CLIP-style VLMs [29, 36, 44] trained end-to-end via large-scale contrastive learning. While highly effective for image-level retrieval and classification, this training paradigm can bias features toward global semantics and weaken local cues, prompting downstream methods to rely on heuristic post-processing, prompt engineering, or late-stage refinement of similarity maps to recover spatial structure. In contrast, recently proposed `dino.txt` [14] offers a complementary route. It builds on self-supervised DINO visual foundation models [26, 33], which are renowned for producing high-quality, spatially rich visual features. To equip these models with open-vocabulary capabilities, it employs locked-image text tuning (LiT) [45], aligning a text encoder to a frozen DINO backbone and thereby preserving the strong local structure learned through self-supervision. To support both global and dense tasks, it combines the [CLS] token with pooled patch features and introduces lightweight transformer adapters that bridge visual pre-training with image-text data, delivering state-of-the-art zero-shot performance across open-vocabulary global and dense tasks.

However, despite improved performance on open-vocabulary dense tasks compared to CLIP-style VLMs, `dino.txt` remains a VLM trained under a largely global contrastive objective and is predominantly evaluated in a zero-shot setting. Even when fine-tuned for segmentation, it fails to fully adapt to the open-vocabulary setting. As visualized in Fig. 1, fine-tuned `dinov3.txt` features yield sharper boundaries but lose fine-grained detail, consistent with the limited performance on OVSS benchmarks observed in Sec. 4.4. Such adaptation lacks dedicated dense refinement, the utilization of complementary text representations for fine-grained vision-language correspondence and segmentation-aware optimization – all critical for precise open-vocabulary pixel classification. This motivates a central research question: *how can we explicitly optimize `dino.txt`-style VLMs for OVSS so that its dense features and image-text alignment become*

segmentation-aware, improving boundary fidelity and class separability while retaining open-vocabulary generalization?

We address this gap with `dinov3.seg`, the first OVSS framework explicitly built and trained on top of `dinov3.txt`, the DINOv3 instantiation of `dino.txt`. Our method is characterized by four synergistic design choices. First, we develop a task-specific segmentation architecture that moves beyond zero-shot reuse of `dinov3.txt`, drawing on common practices from prior literature. Second, we model each class using complementary textual views aligned with both the global [CLS] token and local patch-level representations, and ensemble their image–text correlations to strengthen semantic discrimination while preserving spatial locality. Third, we introduce a dual-stage refinement scheme: an early refinement module that enhances dense visual representations before image–text interaction, improving feature discriminability, followed by a SAM-guided late refinement stage that denoises and regularizes image–text correlation maps for sharper boundaries and improved class consistency in cluttered scenes. Fourth, we employ a high-resolution local–global inference strategy based on sliding-window aggregation, enabling fine-grained spatial detail while maintaining global semantic coherence across large images. These components are mutually reinforcing: complementary global–local textual modeling yields enriched image–text correlations, early refinement improves feature quality prior to image–text interaction, late refinement stabilizes image–text correlation features, and high-resolution inference consolidates locally consistent predictions. Together, these advances form a principled extension of `dinov3.txt` that directly addresses localization and optimization bottlenecks in OVSS. We summarize our contributions as follows:

- Extension of `dinov3.txt` into a dedicated dense OVSS framework.
- Integration of complementary global and local textual representations for stronger multimodal alignment.
- Dual-stage refinement of visual features and image–text correlations.
- High-resolution local–global inference via sliding-window aggregation.
- State-of-the-art performance across five challenging OVSS benchmarks.

2 Related Works

Open-Vocabulary Segmentation. OVSS assigns pixel-level labels from an open set of text-defined categories, and therefore requires simultaneously (i) reliable vision–language grounding and (ii) precise spatial localization for dense masks. Early *pre-VLM* approaches aligned dense CNN features with fixed semantic spaces: ZS3Net [1] synthesizes class-conditional features from Word2Vec guidance with DeepLab-v3+ supervision [4], and SPNet [39] projects dense features into shared visual–semantic spaces for text-driven labeling. However, the cross-modal coupling is relatively weak in these models, often yielding brittle transfer to novel concepts and coarse object boundaries. With the rise of VLMs, CLIP-based pipelines such as LSeg [17], OpenSeg [11], and ZegFormer [5, 8] considerably improve open-vocabulary recognition by strengthening image–text alignment. However, global contrastive pretraining in CLIP can bias representations toward image-level semantics and suppress fine-grained locality, compelling

many methods to rely on post hoc refinement of similarity maps to recover spatial detail. Transformer-centric designs, including SAN [42], FC-CLIP [43], and CAT-Seg [7], integrate VLM cues through language-aware queries and attention biases, frozen convolutional backbones, or dense image–text cost volumes constructed and refined before decoding. Nevertheless, these designs remain susceptible to noisy cross-modal correspondences when the underlying alignment is not segmentation-aware, a limitation that becomes especially pronounced under clutter, occlusion, and high-resolution settings. Diffusion-based methods (ODISE [41], DeDOS [18], DP-Seg [46]) leverage rich intermediate semantics from generative models, typically at the expense of higher compute and greater sensitivity to feature extraction, prompting, or multi-stage processing. Segment Anything Model (SAM)-based hybrids (EB-Seg [31], ESC-Net [16], USE [37]) incorporate strong mask priors or universal segment representations, yet their performance can be bounded by proposal quality and imperfect pixel-level alignment between segments and text labels. Recent efforts explore complementary directions such as parameter-efficient tuning, hyperbolic adaptation, multi-resolution training, or seen-class bias mitigation [22, 27, 28, 49]. Different from previous works, `dinov3.seg` upgrades the backbone signal itself by training an OVSS-specific architecture on top of `dinov3.txt`, combining textual semantic ensembling with early and late refinement schemes and an effective high-resolution local–global aggregation-based inference scheme, thereby improving boundary fidelity and spatial precision in complex scenes.

DINO Family of Visual Foundation Models. DINO models learn visual representations via self-distillation, producing object-centric attention and spatially coherent features that transfer effectively to dense prediction tasks. DINOv2 scales this paradigm to larger data and stronger architectures, while DINOv3 further improves locality and correspondence under massive training and refined objectives, making it a particularly strong backbone for segmentation and dense matching. Building on these visual foundations, `dino.txt` [14] extends DINO to the vision–language regime via locked-image text tuning (LiT), aligning a text encoder to frozen DINO features to preserve locality while enabling open-vocabulary recognition and zero-shot dense prediction. Our work complements `dino.txt` by extending it into a fully trainable OVSS framework: `dinov3.seg` capitalizes on DINO’s inherent spatial coherence through multi-granular image–text alignment and segmentation-aware optimization, yielding precise open-vocabulary masks that generalize robustly across seen and unseen classes.

3 Methodology

3.1 Preliminaries: `dino.txt`

`dino.txt` [14] is a vision–language alignment framework built on top of a frozen DINOv2 visual backbone. It follows LiT strategy [45], where the visual encoder remains fixed and only the text encoder is trained, preserving the spatial structure and locality of self-supervised visual features. Image representations are constructed by combining the global [CLS] token with average-pooled patch

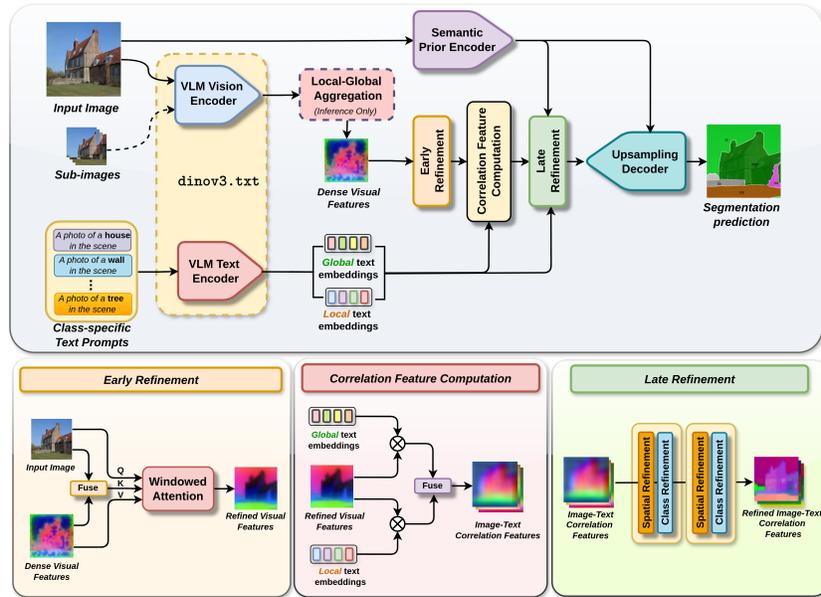


Fig. 2: Overall architecture of proposed `dinov3.seg`. Given an input image (and sub-images at inference time), the `dinov3.txt` backbone extracts dense visual features and multiple textual embeddings. The dense visual features are passed to the Early Refinement Module — via the Local-Global Aggregation module at inference time, or directly during training. The refined visual features interact with the textual embeddings to yield image–text correlation features, which are subsequently enriched with auxiliary semantic guidance from the Semantic Prior Encoder via the Late Refinement Module. The final segmentation map is generated by the Upsampling Decoder.

features to support both image-level and dense prediction tasks. To bridge the domain gap between visual pre-training and image–text data, `dino.txt` introduces lightweight transformer blocks that align visual and textual embeddings using a contrastive objective. Although `dino.txt` is originally developed for DINOv2, the same text-alignment formulation is also available for DINOv3; hereafter, we refer to the DINOv2 and DINOv3 variants as `dinov2.txt` and `dinov3.txt`, respectively.

3.2 Our proposal: `dinov3.seg`

Overview. Given an input image I , the objective of Open-Vocabulary Semantic Segmentation is to assign a semantic label to each pixel from a set of categories defined by textual names or descriptions. Under the open-vocabulary setting, the class set available during training, denoted as $\mathcal{C}_{\text{train}}$, is not necessarily identical to the class set encountered at inference time, $\mathcal{C}_{\text{test}}$, thereby requiring the model to generalize beyond the seen categories.

In this work, we propose a novel Open-Vocabulary Segmentation model, `dinov3.seg`, based on `dinov3.txt` backbone. Our framework consists of the

following components: (i) *dinov3.txt backbone* that extracts dense visual features from images and textual embeddings from category descriptions; (ii) an *Early Refinement Module* that enhances the dense visual representations; (iii) a *Semantic Prior Encoder* that provides auxiliary semantic guidance; (iv) a *Correlation Feature Computation Module* that models image–text interactions to produce correlation features; (v) a *Late Refinement Module* that further refines the correlation features using the semantic priors; and (vi) an *Upsampling Decoder* that progressively decodes the refined features to generate the final segmentation output. We describe the model architecture in detail in the following paragraphs. An overview of the proposed framework is illustrated in Fig. 2.

Visual and Textual Feature Extraction. Given an input image I , we extract dense visual features using the vision encoder of `dinov3.txt`, denoted by \mathcal{F}_v :

$$\phi_{\text{CLS}}, \phi_v = \mathcal{F}_v(I) \quad (1)$$

where, $\phi_{\text{CLS}} \in \mathbb{R}^C$ represent CLS token and $\phi_v \in \mathbb{R}^{C \times H \times W}$ represent final patch features from ViT-based vision encoder.

For each class $c \in \mathcal{C}_{\text{train}}$ (or $c \in \mathcal{C}_{\text{test}}$ during inference), we extract textual representations using the text encoder of `dinov3.txt`, denoted by \mathcal{F}_t . Specifically, we construct a text prompt of the form “A photo of a <class> in the scene” and encode it as,

$$\phi_t^g(c), \phi_t^l(c) = \mathcal{F}_t(c) \quad (2)$$

where $\phi_t^g(c) \in \mathbb{R}^C$ is aligned with the ϕ_{CLS} ; we refer to this as the *global text embedding*. $\phi_t^l(c) \in \mathbb{R}^C$ is aligned with the average of patch-wise visual features; we refer to this as the *local text embedding*.

While `dinov3.txt` [14] leverages only the local text embedding ϕ_t^l for open-vocabulary segmentation, we demonstrate that jointly utilizing both global and local text embeddings leads to more effective and robust segmentation performance in our framework (see Table 3). The complementary nature of these embeddings is further validated in our analysis in the supplementary material.

Semantic Prior Encoder (SPE). In addition to the vision–language backbone, we introduce a semantic prior encoder \mathcal{F}_{SPE} to provide complementary visual cues. Following [9], we adopt the ViT-L based image encoder of Segment Anything Model (SAM) [15] as our SPE. Experiments with alternative SPE choices are provided in the supplementary material.

We denote final-layer representation from \mathcal{F}_{SPE} as $F_g^{(L)}$, which encodes high-level semantic context and global structural information for segmentation guidance. Additionally, we also extract intermediate representations from the 7th and 15th transformer blocks, denoted as $F_g^{(7)}$ and $F_g^{(15)}$. Although all features share the same spatial resolution, they capture progressively richer abstractions, from structural cues to higher-level semantics. SPE is jointly fine-tuned with the rest of the architecture to ensure compatibility with the vision–language representations. Formally, given an input image I , semantic prior features are obtained as:

$$F_g^{(7)}, F_g^{(15)}, F_g^{(L)} = \mathcal{F}_{\text{SPE}}(I). \quad (3)$$

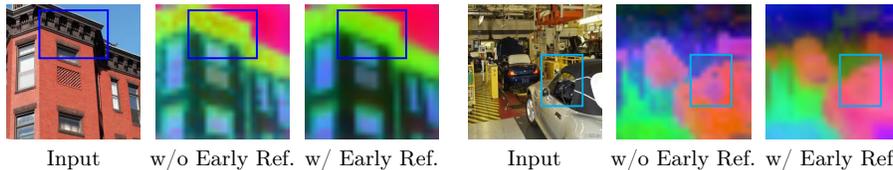


Fig. 3: Effect of Early Refinement on `dinov3.txt` visual features.

Early Refinement of VLM features. Like most Vision–language models, `dinov3.txt` is also optimized with a global contrastive objective, which often results in dense visual features ϕ_v that are noisy and lack the discriminative power required for fine-grained segmentation. To address this limitation, we introduce an early-stage refinement of visual features before any explicit image–text interaction. Crucially, this refinement must preserve the original image–text alignment so as not to disrupt compatibility with textual embeddings.

We adopt the transformer-based AnyUp [38] module to refine ϕ_v . Although originally designed for guided feature upsampling, AnyUp can be fine-tuned to effectively refine dense visual representations for open-vocabulary segmentation. Visualization of its effect on `dinov3.txt` features is shown in Fig 3.

Concretely, given an input image I and dense visual features ϕ_v , the image is first processed by a lightweight convolutional encoder to obtain intermediate features ψ_v . Rotary positional encodings (RoPE) [34] are added to ψ_v to preserve spatial structure. Refinement is performed via a window-based attention module [30], where queries are computed from ψ_v , keys from both ψ_v and ϕ_v , and values are the original dense features ϕ_v . Since values are drawn solely from ϕ_v , the output is a linear recombination of original patch-level representations, ensuring that the refined features ϕ_v remain anchored to the `dinov3.txt` feature space and retain image–text alignment: $\phi_v^{\text{ref}} = \text{EarlyRef}(\phi_v)$.

Image–Text Correlation Feature Computation. To explicitly model image–text correspondence, we compute cosine similarity between dense visual features and textual embeddings for each spatial location and each semantic class. Given refined dense visual features $\phi_v^{\text{ref}} \in \mathbb{R}^{C \times H \times W}$ and class-wise textual embeddings, we compute for every spatial location (h, w) and class c :

$$\begin{aligned} s^g(c, h, w) &= \cos(\phi_v^{\text{ref}}(h, w), \phi_t^g(c)), \\ s^l(c, h, w) &= \cos(\phi_v^{\text{ref}}(h, w), \phi_t^l(c)). \end{aligned} \quad (4)$$

where $\phi_t^g(c)$ and $\phi_t^l(c)$ denote the global and local textual embeddings of class c . The resulting similarity tensors $S^g, S^l \in \mathbb{R}^{N \times H \times W}$ encode class-wise semantic alignment at each spatial location, with N denoting the number of classes.

We concatenate the similarity tensors along the channel dimension and project them using a convolutional layer to obtain higher-level correlation features:

$$\phi_{\text{corr}} = \text{Conv}([S^g; S^l]), \quad \phi_{\text{corr}} \in \mathbb{R}^{N \times C_{\text{corr}} \times H \times W}. \quad (5)$$

where C_{corr} is the number of output channels. We refer to ϕ_{corr} as the *image–text correlation features*. These features are subsequently refined by Late Refinement module to improve semantic consistency and localization accuracy.

Late Refinement of Correlation Features. In addition to refining dense visual features from the vision–language model, we further refine the image–text correlation features in a late refinement stage. While early visual refinement improves feature quality prior to image–text interaction, refining the correlation features themselves is crucial for producing spatially accurate and class-consistent segmentation outputs. Following [7, 9], our late refinement module consists of two complementary components - *Spatial Refinement Block* and *Class Refinement Block*.

Spatial Refinement Block: To enhance the spatial coherence of the class-wise correlation features, we employ a Swin Transformer [23]-based refinement module. Refinement is performed independently for each class using two successive Swin Transformer blocks: the first applies Window Multi-head Self-Attention (W-MSA) to model local spatial interactions, followed by a Shifted Window Multi-head Self-Attention (SW-MSA) block to enable cross-window information exchange.

To further strengthen spatial modeling, we incorporate auxiliary guidance from SPE into the Spatial Refinement Block. Specifically, SPE feature $F_g^{(L)}$ is projected into the correlation feature space and used to guide spatial refinement. For each class c , the refined correlation feature $\phi'_{\text{corr}}(:, :, c)$ is computed as,

$$\phi'_{\text{corr}}(:, :, c) = \text{SpatialRef}\left(\phi_{\text{corr}}(:, :, c), \mathcal{M}_v\left(F_g^{(L)}\right)\right), \quad (6)$$

where $\text{SpatialRef}(\cdot)$ denotes the Spatial Refinement Block, $\phi_{\text{corr}}(:, :, c)$ represents the correlation feature corresponding to class c , and $\mathcal{M}_v(\cdot)$ is an MLP projection layer applied to the SPE feature.

Class Refinement Block. While the Spatial Refinement Block focuses on improving spatial coherence, the Class Refinement Block aims to enhance class-wise discrimination at each spatial location. We refine the correlation features by explicitly modeling semantic relationships across classes.

Specifically, we apply a transformer-based refinement module along the class dimension, where attention is computed across class channels for each spatial location. This allows the model to capture inter-class dependencies and suppress ambiguous class predictions. For each spatial location (h, w) , the refined correlation feature $\phi''_{\text{corr}}(h, w, :)$ is computed as,

$$\phi''_{\text{corr}}(h, w, :) = \text{ClassRef}\left(\phi'_{\text{corr}}(h, w, :), \mathcal{M}_t(\bar{\phi}_t)\right), \quad (7)$$

where $\text{ClassRef}(\cdot)$ denotes the Class Refinement Block, and $\phi'_{\text{corr}}(h, w, :)$ represents the class-wise correlation responses at spatial location (h, w) . $\mathcal{M}_t(\cdot)$ is an MLP projection layer applied to the textual guidance feature, $\bar{\phi}_t = (\phi_t^g + \phi_t^l)/2$.

In our framework, the combination of SpatialRef and ClassRef is applied twice in succession, enabling progressive refinement of correlation features by jointly improving spatial coherence and inter-class discrimination.

Upsampling Decoder. Since the refined correlation features ϕ''_{corr} are at $\frac{1}{16}$ of the input image resolution, we employ a lightweight convolutional upsampling decoder to progressively recover full spatial resolution. First, ϕ''_{corr} is upsampled

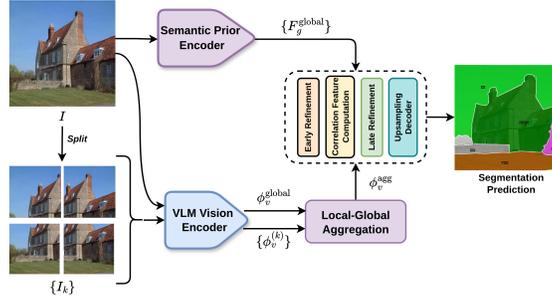


Fig. 4: Our proposed Local-Global Aggregation (LGA) Inference Strategy.

by a factor of 2 using a transposed convolution, yielding an intermediate feature map $\phi'_{\text{corr}}{}^{2\times}$. The upsampled correlation feature $\phi_{\text{corr}}{}^{2\times}$ is obtained by concatenating $\phi'_{\text{corr}}{}^{2\times}$ with upsampled SPE guidance features and applying a convolutional fusion:

$$\phi_{\text{corr}}{}^{2\times} = \text{conv}\left([\phi'_{\text{corr}}{}^{2\times}, \text{up}(F_g^{(7)}, 2)]\right). \quad (8)$$

This procedure is repeated with SPE guidance feature $F_g^{(15)}$ to yield $\phi_{\text{corr}}{}^{4\times}$, followed by a final convolution layer and bilinear upsampling to produce the full-resolution segmentation prediction \hat{y} .

Inference via Local-Global Aggregation (LGA). During training, our model operates on image crops of size 384×384 . To enable inference on high-resolution images, we design a sliding-window-based strategy inspired by [7, 32]. Specifically, the input image I is first resized to 640×640 and partitioned into overlapping sub-images $\{I_k\}$ of size 384×384 , with an overlap of 128×128 pixels. Each sub-image is processed independently by the `dinov3.txt` vision encoder. In parallel, the resized full image is encoded to obtain global VLM features:

$$\phi_v^{(k)} = \mathcal{F}_v(I_k), \quad \phi_v^{\text{global}} = \mathcal{F}_v(I). \quad (9)$$

The resulting sub-image features are merged by accounting for overlapping regions, producing locally aggregated VLM features, denoted as $\bar{\phi}_v$. The local and global VLM features are then fused via simple averaging:

$$\phi_v^{\text{agg}} = \frac{1}{2} (\bar{\phi}_v + \phi_v^{\text{global}}). \quad (10)$$

We refer to this strategy as *Local-Global Aggregation (LGA)*. For the SPE, we directly operate on resized full image to extract guidance features:

$$\{F_g^{\text{global}}\} = \mathcal{F}_{\text{SPE}}(I). \quad (11)$$

Finally, the aggregated VLM features ϕ_v^{agg} and the guidance features $\{F_g^{\text{global}}\}$ are fed into subsequent modules to produce the final segmentation map. Overall schematic of the proposed inference strategy is shown in Fig. 4. We validate the inference strategy design choices via an ablation study in Sec. 4.4.

3.3 Loss functions

The following loss functions are used to train our model.

Focal loss [20] is a variant of Binary Cross-Entropy loss that addresses the imbalance between easy and hard samples, a prevalent challenge in segmentation. Focal loss is given by,

$$\mathcal{L}_{\text{focal}}(\hat{y}, y) = -\frac{1}{HW} \sum_{i=1}^{HW} [(1 - \hat{y}_i)^\gamma y_i \log(\hat{y}_i) + \hat{y}_i^\gamma (1 - y_i) \log(1 - \hat{y}_i)] \quad (12)$$

where \hat{y} is the predicted probability, y is the ground truth label, and γ is the focusing parameter that controls the down-weighting of easy samples.

Dice loss [35] maximizes the Intersection over Union (IoU) between predicted map and ground truth segmentation mask. Dice loss is given by,

$$\mathcal{L}_{\text{dice}}(\hat{y}, y) = 1 - \frac{2 \sum_{i=1}^{HW} y_i \hat{y}_i}{\sum_{i=1}^{HW} y_i + \sum_{i=1}^{HW} \hat{y}_i} \quad (13)$$

We adopt a weighted combination of Focal loss and Dice loss [6, 48], given by $\mathcal{L} = \mathcal{L}_{\text{focal}} + \lambda \mathcal{L}_{\text{dice}}$. Our experiments in Sec. 4.4 demonstrate that this combination leads to more effective segmentation performance over the conventional Binary Cross-Entropy loss [7, 28, 40, 46].

4 Experiments

4.1 Dataset Description

We train our method on the COCO-Stuff [2] train split consisting of 118K images, and evaluate on ADE20K [47], Pascal Context [24], and Pascal VOC [10]. ADE20K is a diverse indoor-outdoor benchmark with 2K validation images, and we report results under both the 150-class (A-150) and 847-class (A-847) settings. Pascal Context contains 5K validation images covering a wide range of scenes, and we evaluate on both the 59-class (PC-59) and 459-class (PC-459) variants. Pascal VOC (PAS-20) includes 1.5K validation images with 20 classes.

4.2 Implementation details

We implement our method using PyTorch and the Detectron2 framework. The vision-language backbone, the semantic prior encoder and Early Refinement module are finetuned with an initial learning rate of 2×10^{-6} , while the remaining modules use a higher initial learning rate of 2×10^{-4} . All models are optimized using AdamW with a batch size of 4. We employ a cosine learning rate schedule and train the models for 80K iterations. The values of λ and γ are set to 0.05 and 2 empirically. Performance under varying λ and γ values is reported in Supplementary Material. All experiments are conducted on a server equipped with four NVIDIA A100 80 GB GPUs. We evaluate our models using the mean Intersection-over-Union (mIoU) metric.

Table 1: Quantitative comparison with state-of-the-art OVSS methods. Best and second-best results are highlighted in red and blue, respectively. Avg denotes the mean mIoU over A-847, PC-459, A-150, PC-59, and PAS-20.

Method	Venue	A-847	PC-459	A-150	PC-59	PAS-20	Avg.
OVSeg [19]	CVPR'23	9.0	12.4	29.6	55.7	94.5	40.24
SAN [42]	CVPR'23	12.4	15.7	32.1	57.7	94.6	42.50
ODISE [41]	CVPR'23	11.1	14.5	29.9	57.3	–	–
FC-CLIP [43]	NeurIPS'23	14.8	18.2	34.1	58.4	95.4	44.18
SCAN [21]	CVPR'24	14.0	16.7	33.5	59.3	97.2	44.14
SED [40]	CVPR'24	13.9	22.6	35.2	60.6	96.1	45.68
CAT-Seg [7]	CVPR'24	16.0	23.8	37.9	63.3	97.0	47.60
MAFT+ [13]	ECCV'24	15.1	21.6	36.1	59.4	96.5	45.74
EOV-Seg [25]	AAAI'25	12.8	16.8	32.1	56.9	94.8	42.68
H-CLIP [27]	CVPR'25	16.5	24.2	38.4	64.1	97.7	48.18
HyperCLIP [28]	CVPR'25	16.3	24.1	38.2	64.2	98.3	48.22
ESCNet [16]	CVPR'25	18.1	27.0	41.8	65.6	98.3	50.16
DPSeg [46]	CVPR'25	15.7	24.1	37.1	62.3	98.5	47.54
DEDOS [18]	ICCV'25	17.9	25.6	39.4	65.7	97.6	49.24
OVSNet [22]	ICCV'25	16.2	23.5	37.1	62.0	96.9	47.14
SAM3 [3]	ICLR'26	13.8	18.8	39.0	60.8	–	–
dinov3.seg (Ours)	–	20.09	27.80	42.19	64.27	97.86	50.44

4.3 Comparison with state-of-the-art

We compare `dinov3.seg` against a broad set of state-of-the-art open-vocabulary semantic segmentation methods, including OVSeg [19], SAN [42], ODISE [41], FC-CLIP [43], SCAN [21], SED [40], CAT-Seg [7], MAFT+ [13], EOV-Seg [25], H-CLIP [27], HyperCLIP [28], ESCNet [16], DPSeg [46], DEDOS [18], and OVSNet [22]. Additionally, we compare against SAM3 [3], a unified foundational segmentation model. For a fair comparison, we report results using the *large* variants of all competing models whenever available. Although `dinov3.txt` serves as our primary VLM backbone, our framework generalizes across alternative backbones with minimal adjustments while maintaining consistently strong results; full details are provided in the supplementary material.

Table 1 reports quantitative results across benchmarks. `dinov3.seg` attains the best average mIoU across all datasets. In particular, it achieves the top performance on A-847, PC-459, and A-150, while remaining competitive on PC-59 and PAS-20. The substantial gains on A-847, PC-459, and A-150 highlight strong scalability to large and fine-grained category sets that demand robust generalization to numerous unseen classes. In contrast, PC-59 and PAS-20 exhibit significant overlap with the COCO-Stuff training categories, as noted in prior work [42] and further supported by our seen–unseen class partition (details in the supplementary material). Only 3 out of 59 classes in PC-59 and *none* of the 20 classes in PAS-20 qualify as unseen. Consequently, results on these benchmarks largely reflect seen-class optimization rather than open-vocabulary ability, and our method remains competitive even under these conditions.

Table 2: Seen and unseen class mIoU comparison across datasets. PAS-20 is omitted as all its categories are classified as seen under our similarity-based partition. Δ indicates absolute difference between our method and CAT-Seg.

Method	A-847		PC-459		A-150		PC-59	
	Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen
SED [40]	31.67	12.23	40.22	15.86	47.02	28.07	61.73	40.35
MAFT+ [13]	38.11	13.40	39.71	14.01	48.47	29.11	61.11	29.37
CAT-Seg [7]	38.67	13.80	42.83	16.40	48.92	31.17	64.68	37.90
dinov3_seg (Ours)	43.07	17.99	44.72	21.27	54.06	35.11	65.43	42.43
Δ	+4.39	+4.19	+1.89	+4.87	+5.14	+3.95	+0.75	+4.53

To further examine generalization, Table 2 reports mIoU on seen and unseen class subsets using the same similarity-based partition. `dinov3_seg` consistently improves both seen and unseen performance over state-of-the-art methods across all datasets. Notably, on PC-59 and PC-459, the gains on unseen classes are substantially larger than those on seen classes, indicating that the proposed design primarily enhances open-vocabulary generalization and recognition of novel categories.

Fig. 5 provides qualitative comparisons with state-of-the-art approaches. Our method produces more accurate segmentations for both seen categories (e.g., “chair” and “book”) and unseen categories (e.g., “column” and “fireplace”), further demonstrating improved generalization across diverse classes. Additional qualitative results are provided in the supplementary material.

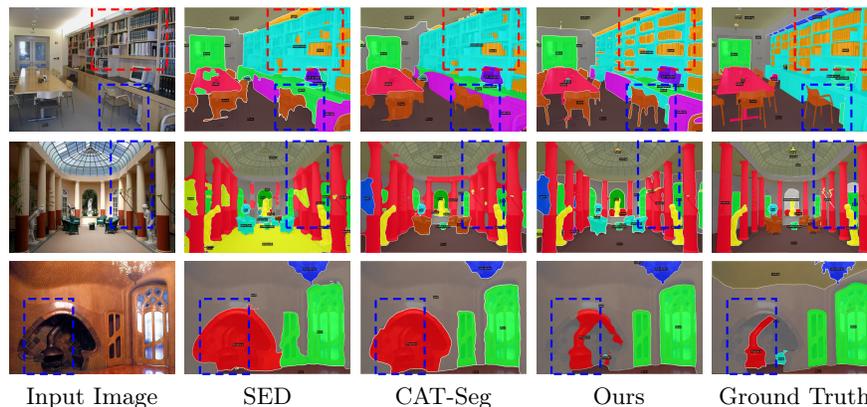


Fig. 5: Qualitative comparison with state-of-the-art. Highlighted regions in bounding boxes illustrate cases where our model achieves more accurate segmentation results.

4.4 Ablation Study

Ablation Study of Design Components. Table 3 presents a systematic evaluation of individual design components. We first report two reference baselines using `dinov3.txt` — a zero-shot variant (ZS) without any task-specific training, and a fine-tuned variant (FT) trained with Binary Cross-Entropy (BCE) loss — both following the inference protocol of [7] and relying solely on local text embeddings for correlation map computation. While the fine-tuned variant shows notable gains over its zero-shot counterpart, there remains significant room for improvement, which we address through the design choices explored in the following configurations.

Building on the fine-tuned baseline, Config I strengthens the setup by incorporating established practices in OVSS [7, 9], namely SAM-guided correlation feature refinement and an upsampling decoder. This modification consistently improves performance across all datasets, establishing a stronger reference for subsequent ablations.

Introducing a text embedding ensemble that combines global and local text representations (Config II) yields consistent gains on most benchmarks, particularly on A-847, PC-459, and PC-59, indicating improved class-level semantic alignment. Adding early VLM feature refinement (Config III) further improves performance on ADE benchmarks (A-847 and A-150), highlighting the benefit of refining dense visual features at earlier stages. Replacing BCE with a weighted focal-dice objective (Config IV) provides additional improvements, especially on A-847 and PC-459.

Finally, incorporating the proposed LGA-based inference strategy (Config V) improves results on ADE datasets and PC-459 while maintaining comparable performance on the remaining benchmarks, demonstrating the complementary effects of the proposed training and inference components.

Table 3: Ablation study on various design choices. Text Ens. = Text Embedding Ensemble; Early Ref. = Early Refinement; LGA Inf. = LGA-based Inference Strategy.

Config.						A-847	PC-459	A-150	PC-59	PAS-20
<i>Reference Baselines</i>										
dinov3.txt ZS						8.26	8.82	18.12	25.94	82.31
dinov3.txt FT						8.86	17.46	28.33	59.64	96.69
<i>Ablation Configurations</i>										
	<i>Text Ens.</i>	<i>Early Ref.</i>	<i>Focal-Dice</i>	<i>LGA Inf.</i>						
I	×	×	×	×	18.70	26.51	41.57	64.05	97.61	
II	✓	×	×	×	19.22	27.29	41.45	64.20	97.75	
III	✓	✓	×	×	19.44	27.23	41.84	64.22	97.64	
IV	✓	✓	✓	×	19.67	27.45	41.84	64.33	97.88	
V	✓	✓	✓	✓	20.09	27.80	42.19	64.27	97.86	

Fine-tuning Strategy Analysis. Following [7], we study different finetuning strategies for adapting the VLM backbone. Specifically, we evaluate: (i) **QV FT**, where only the query and value projection matrices of the transformer lay-

ers are updated; (ii) **QK FT**, where the query and key projection matrices are trained while keeping the remaining parameters frozen; and (iii) **Full FT**, which finetunes all backbone parameters end-to-end. All variants are evaluated using our Config I model. As shown in Table 4, full finetuning consistently outperforms partial finetuning strategies across most benchmarks.

Table 4: Comparison across finetuning strategies.

FT Strategy	A-847	PC-459	A-150	PC-59	PAS-20
QV FT	17.72	26.22	40.62	62.63	97.27
QK FT	17.92	26.61	40.26	62.95	97.32
Full FT (Ours)	18.70	26.51	41.57	64.05	97.61

Ablation on Local-Global Aggregation (LGA). We analyze the impact of Local-Global Aggregation (LGA) at inference time by selectively enabling it for the `dinov3.txt` vision encoder and the SAM semantic prior encoder. When LGA is disabled for a module, the resized high-resolution image is directly fed into the corresponding encoder. As shown in Table 5, enabling LGA for the `dinov3.txt` vision encoder consistently improves performance over the global-only baseline across most benchmarks. In contrast, applying LGA to the SAM encoder does not yield additional gains compared to global-only inference, suggesting that high-resolution global guidance features are already sufficient. Therefore, we adopt the configuration with LGA enabled for `dinov3.txt` and disabled for SAM as the final inference strategy, as it provides the best balance between accuracy and inference efficiency.

Table 5: Ablation on Local-Global Aggregation.

LGA		Performance					
<code>dinov3.txt</code>	SAM	A-847	PC-459	A-150	PC-59	PAS-20	Avg
×	×	20.06	27.43	42.14	63.94	98.34	50.38
×	✓	20.02	27.49	42.03	64.07	98.37	50.40
✓	×	20.09	27.80	42.19	64.27	97.86	50.44
✓	✓	19.95	27.83	42.10	64.32	97.94	50.43

5 Conclusion

In this paper, we introduced `dinov3.seg`, a dedicated OVSS framework built on top of `dinov3.txt` that explicitly optimizes vision-language representations for dense, segmentation-aware prediction. By integrating complementary global and local textual embeddings, dual-stage refinement of visual features and image-text correlations, and a high-resolution local-global inference strategy, our approach strengthens semantic alignment while preserving fine-grained spatial structure and boundary fidelity. Extensive experiments across five challenging OVSS benchmarks demonstrate consistent and state-of-the-art performance, highlighting the importance of segmentation-specific adaptation of DINO-based VLMs for robust open-vocabulary dense understanding.

References

1. Bucher, M., Vu, T.H., Cord, M., Pérez, P.: Zero-shot semantic segmentation. *Advances in Neural Information Processing Systems* **32** (2019)
2. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 1209–1218 (2018)
3. Carion, N., Gustafson, L., Hu, Y.T., Debnath, S., Hu, R., Suris, D., Ryali, C., Alwala, K.V., Khedr, H., Huang, A., Lei, J., Ma, T., Guo, B., Kalla, A., Marks, M., Greer, J., Wang, M., Sun, P., Rädle, R., Afouras, T., Mavrouti, E., Xu, K., Wu, T.H., Zhou, Y., Momeni, L., Hazra, R., Ding, S., Vaze, S., Porcher, F., Li, F., Li, S., Kamath, A., Cheng, H.K., Dollár, P., Ravi, N., Saenko, K., Zhang, P., Feichtenhofer, C.: Sam 3: Segment anything with concepts (2025), <https://arxiv.org/abs/2511.16719>
4. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: *Proceedings of the European conference on computer vision (ECCV)*. pp. 801–818 (2018)
5. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 1290–1299 (2022)
6. Cheng, B., Schwing, A., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems* **34**, 17864–17875 (2021)
7. Cho, S., Shin, H., Hong, S., Arnab, A., Seo, P.H., Kim, S.: Cat-seg: Cost aggregation for open-vocabulary semantic segmentation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 4113–4123 (2024)
8. Ding, J., Xue, N., Xia, G.S., Dai, D.: Decoupling zero-shot semantic segmentation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 11583–11592 (2022)
9. Dutta, S., Vasim, A., Gole, S., Rezatofghi, H., Banerjee, B.: Aeroseg: Harnessing sam for open-vocabulary segmentation in remote sensing images. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 2254–2264 (2025)
10. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. *International journal of computer vision* **88**, 303–308 (2009)
11. Ghiasi, G., Gu, X., Cui, Y., Lin, T.Y.: Scaling open-vocabulary image segmentation with image-level labels. In: *European Conference on Computer Vision*. pp. 540–557. Springer (2022)
12. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: *International conference on machine learning*. pp. 4904–4916. PMLR (2021)
13. Jiao, S., Zhu, H., Huang, J., Zhao, Y., Wei, Y., Humphrey, S.: Collaborative vision-text representation optimizing for open-vocabulary segmentation. In: *European Conference on Computer Vision* (2024)
14. Jose, C., Moutakanni, T., Kang, D., Baldassarre, F., Darcet, T., Xu, H., Li, D., Szafranec, M., Ramamonjisoa, M., Oquab, M., et al.: Dinov2 meets text: A unified framework for image-and pixel-level vision-language alignment. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 24905–24916 (2025)

15. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4015–4026 (2023)
16. Lee, M., Cho, S., Lee, J., Yang, S., Choi, H., Kim, I.J., Lee, S.: Effective sam combination for open-vocabulary semantic segmentation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 26081–26090 (2025)
17. Li, B., Weinberger, K.Q., Belongie, S., Koltun, V., Ranftl, R.: Language-driven semantic segmentation. In: International Conference on Learning Representations (2022)
18. Li, F., Wang, X., Wang, X., Zhang, Z., Xu, Y.: Images as noisy labels: Unleashing the potential of the diffusion model for open-vocabulary semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 24255–24265 (2025)
19. Liang, F., Wu, B., Dai, X., Li, K., Zhao, Y., Zhang, H., Zhang, P., Vajda, P., Marculescu, D.: Open-vocabulary semantic segmentation with mask-adapted clip. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7061–7070 (2023)
20. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
21. Liu, Y., Bai, S., Li, G., Wang, Y., Tang, Y.: Open-vocabulary segmentation with semantic-assisted calibration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3491–3500 (2024)
22. Liu, Y., Wu, S.L., Bai, S., Wang, J., Wang, Y., Tang, Y.: Stepping out of similar semantic space for open-vocabulary segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22664–22674 (2025)
23. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)
24. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 891–898 (2014)
25. Niu, H., Hu, J., Lin, J., Jiang, G., Zhang, S.: Eov-seg: Efficient open-vocabulary panoptic segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 6254–6262 (2025)
26. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
27. Peng, Z., Xu, Z., Zeng, Z., Huang, Y., Wang, Y., Shen, W.: Parameter-efficient fine-tuning in hyperspherical space for open-vocabulary semantic segmentation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 15009–15020 (2025)
28. Peng, Z., Xu, Z., Zeng, Z., Wen, C., Huang, Y., Yang, M., Tang, F., Shen, W.: Understanding fine-tuning clip for open-vocabulary semantic segmentation in hyperbolic space. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 4562–4572 (2025)

29. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
30. Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., Shlens, J.: Stand-alone self-attention in vision models. *Advances in neural information processing systems* **32** (2019)
31. Shan, X., Wu, D., Zhu, G., Shao, Y., Sang, N., Gao, C.: Open-vocabulary semantic segmentation with image embedding balancing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 28412–28421 (2024)
32. Shi, Y., Dong, M., Xu, C.: Harnessing vision foundation models for high-performance, training-free open vocabulary segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 23487–23497 (2025)
33. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., et al.: Dinov3. arXiv preprint arXiv:2508.10104 (2025)
34. Su, J., Lu, Y., Pan, S., Wen, B., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. arXiv preprint arXiv:2104.09864 (2021)
35. Sudre, C.H., Li, W., Vercauteren, T., Ourselin, S., Jorge Cardoso, M.: Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3. pp. 240–248. Springer (2017)
36. Sun, Q., Fang, Y., Wu, L., Wang, X., Cao, Y.: Eva-clip: Improved training techniques for clip at scale. arXiv preprint arXiv:2303.15389 (2023)
37. Wang, X., He, W., Xuan, X., Sebastian, C., Ono, J.P., Li, X., Behpour, S., Doan, T., Gou, L., Shen, H.W., et al.: Use: Universal segment embeddings for open-vocabulary image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4187–4196 (2024)
38. Wimmer, T., Truong, P., Rakotosaona, M.J., Oechsle, M., Tombari, F., Schiele, B., Lenssen, J.E.: Anyup: Universal feature upsampling. arXiv preprint arXiv:2510.12764 (2025)
39. Xian, Y., Choudhury, S., He, Y., Schiele, B., Akata, Z.: Semantic projection network for zero-and few-label semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8256–8265 (2019)
40. Xie, B., Cao, J., Xie, J., Khan, F.S., Pang, Y.: Sed: A simple encoder-decoder for open-vocabulary semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3426–3436 (2024)
41. Xu, J., Liu, S., Vahdat, A., Byeon, W., Wang, X., De Mello, S.: Open-vocabulary panoptic segmentation with text-to-image diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2955–2966 (2023)
42. Xu, M., Zhang, Z., Wei, F., Hu, H., Bai, X.: Side adapter network for open-vocabulary semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2945–2954 (2023)
43. Yu, Q., He, J., Deng, X., Shen, X., Chen, L.C.: Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. *Advances in Neural Information Processing Systems* **36** (2023)

44. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 11975–11986 (2023)
45. Zhai, X., Wang, X., Mustafa, B., Steiner, A., Keysers, D., Kolesnikov, A., Beyer, L.: Lit: Zero-shot transfer with locked-image text tuning. 2022 ieee. In: CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 18102–18112 (2021)
46. Zhao, Z., Li, X., Shi, L., Imanpour, N., Wang, S.: Dpseg: Dual-prompt cost volume learning for open-vocabulary semantic segmentation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 25346–25356 (2025)
47. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision* **127**, 302–321 (2019)
48. Zhou, Z., Lei, Y., Zhang, B., Liu, L., Liu, Y.: Zegclip: Towards adapting clip for zero-shot semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11175–11185 (2023)
49. Zhu, Y., Zhu, B., Chen, Y., Niu, Y., Tang, M., Wang, J.: Mrovseg: Breaking the resolution curse of vision-language models in open-vocabulary image segmentation. arXiv preprint arXiv:2408.14776 (2024)

1 Outline of Supplementary Material

This supplementary material offers further insights and expanded analysis to complement the main paper. The key sections are outlined below:

- **(Sec. 2) Analysis of Local and Global Textual Embeddings:** Provides quantitative and qualitative analysis on the complementary nature of local and global textual embeddings.
- **(Sec. 3) Seen and Unseen Class Performance Evaluation:** Details how the seen-unseen split is constructed for benchmark datasets, along with corresponding results.
- **(Sec. 4) Additional Ablation Studies:** We present ablation analysis on: (i) the number of late refinement blocks used, (ii) different SPE choices, (iii) different VLM choices, and (iv) loss function hyperparameters, (v) qualitative analysis of LGA-based inference strategy.
- **(Sec. 5) Comparison with CAT-Seg equipped with dinov3.txt:** Compares our approach against CAT-Seg with `dinov3.txt` VLM backbone.
- **(Sec. 6) Complexity Analysis:** Examines the computational complexity of the proposed method and provides comparisons with various state-of-the-art models.
- **(Sec. 7) Additional Qualitative Results:** Presents further qualitative examples to complement the results reported in the main paper.

2 Analysis on Local and Global textual embeddings

2.1 Quantitative Analysis: Text-to-Visual Prototype Predictability

To quantitatively assess whether Global Text Embeddings and Local Text Embeddings contribute complementary information with respect to the visual space, we train a Ridge regression model to predict the visual prototype embeddings from the text embeddings and measure the coefficient of determination (R^2) via 5-fold cross-validation. We extract visual prototype embeddings for each class by computing the class-wise average of mask-pooled pretrained DINOv3 features. We consider three configurations: Global Text Embeddings alone, Local Text Embeddings alone, and the concatenation of both. All embeddings are L2-normalized prior to regression. Results are reported in Table 6.

Table 6: R^2 scores for predicting DINOv3 visual prototypes from text embeddings using Ridge regression.

Configuration	R^2
Global Text Embeddings only	0.0987
Local Text Embeddings only	0.1376
Both (concatenated)	0.1718

Both Global Text Embeddings and Local Text Embeddings individually explain a meaningful portion of the visual prototype space, achieving R^2 scores of 0.0987 and 0.1376 respectively. More importantly, combining both yields an R^2 of 0.1718, which is substantially higher than either embedding alone. This improvement demonstrates that Global Text Embeddings and Local Text Embeddings capture complementary visual-semantic correspondences, and that both are necessary to better account for the structure of the visual space.

2.2 Qualitative Analysis: Nearest-Neighbor Complementarity

We inspect the nearest neighbors of individual classes in the Global Text Embedding space, Local Text Embedding space, and the Visual prototype space extracted using DINOv3, and present two representative examples in Table 7.

Table 7: Top-5 nearest neighbors in each embedding space for two representative classes in the A-150 dataset. Neighbors matching the visual space are **bolded**.

Class	Embedding Space	Top-5 Nearest Neighbors
<i>Stairway</i>	Visual Prototypes	stairs, bannister, railing, step, grandstand
	Global Text Embeddings	stairs, bannister , path, step, railing
	Local Text Embeddings	stairs , escalator, bannister , path, railing
<i>Minibike</i>	Visual Prototypes	bicycle, car, van, truck, ashcan
	Global Text Embeddings	bicycle, car , road, path, person
	Local Text Embeddings	bicycle , chair, truck, car , bench

For *stairway*, Global Text Embeddings retrieve more neighbors consistent with the visual space by capturing structural parts such as *bannister*, *railing*, and *step*, while Local Text Embeddings introduce *escalator* — an object that shares functional purpose but differs visually. For *minibike*, Local Text Embeddings recover three visually consistent object-level neighbors, whereas Global Text Embeddings retrieves scene-level elements (*road*, *path*) that reflect the typical environment of a minibike rather than its appearance. In both cases, the two embedding spaces make different errors and different correct retrievals, suggesting that Global Text Embeddings and Local Text Embeddings encode complementary semantic information that together better covers the structure of the visual embedding space.

3 Seen and Unseen Class Performance Evaluation

Since the benchmark datasets used in this work do not provide an explicit seen–unseen class split, we approximate this partition using two complementary similarity measures: (i) visual feature similarity, and (ii) textual embedding similarity.

3.1 Visual Feature Similarity-Based Seen–Unseen Class Partition

We extract a visual prototype for each training and test class using pretrained DINOv3, as described in Sec. 2.1. A test class is designated as *seen* if its cosine similarity to at least one training-class visual prototype exceeds a threshold of 0.9; otherwise, it is considered *unseen*. The resulting class counts under this partition are reported in Table 8, and the corresponding mIoU scores on these subsets are presented in Table 2 of the main paper. Figure 6 shows a t-SNE visualization of visual prototypes for A-150 dataset.

Table 8: Seen and unseen class splits for different datasets based on visual similarity.

Dataset	# Seen	# Unseen
A-847	72	775
PC-459 ⁴	96	249
A-150	56	94
PC-59	56	3
PAS-20	20	0

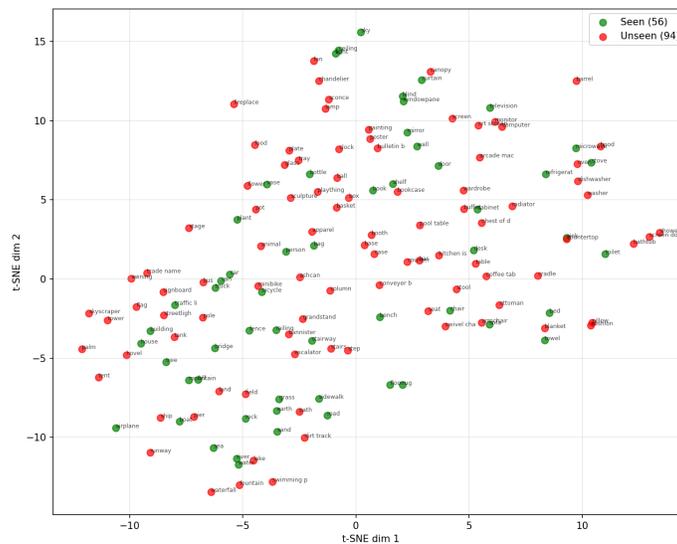


Fig. 6: t-SNE visualization of visual prototypes for A-150, with seen classes (green) and unseen classes (red).

3.2 Textual Embeddings Similarity-based Seen-Unseen Class Partition.

To construct the seen–unseen partition based on textual similarity, we extract global and local text embeddings from pretrained `dinov3.txt` text encoder and

⁴ PC-459 has 114 absent classes which are not considered in our analysis.

concatenate them to form a class-level representation for each training and test class. A test class is designated as *seen* if its cosine similarity to at least one training-class textual representation exceeds a threshold of 0.9; otherwise, it is considered *unseen*. The resulting class counts under this partition are reported in Table 9. Figure 7 presents a t-SNE visualization of textual embeddings on the A-150 dataset.

Table 9: Seen and unseen class splits for different datasets based on textual similarity.

Dataset	# Seen	# Unseen
A-847	196	651
PC-459	119	226
A-150	88	62
PC-59	56	3
PAS-20	20	0

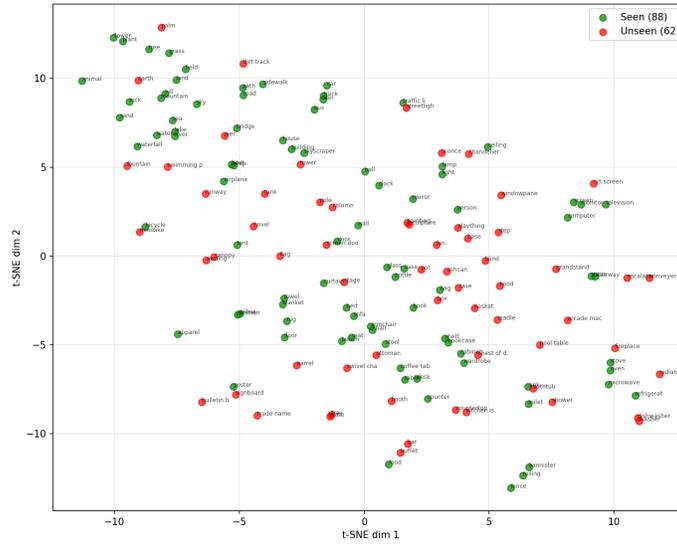


Fig. 7: t-SNE visualization of textual embeddings for A-150, with seen classes (green) and unseen classes (red).

We present seen and unseen mIoU scores based on this split in Table 10. Our method achieves the highest mIoU on both seen and unseen classes across A-847, PC-459, and A-150, with gains most pronounced on unseen categories — up to +5.90 mIoU over CAT-Seg on A-150 — demonstrating strong generalization to novel classes. On PC-59, our method improves seen class performance but shows a marginal drop of 0.28 mIoU on unseen classes, possibly owing to the very few unseen classes (only 3) in that split.

Method	A-847		PC-459		A-150		PC-59	
	Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen
SED [40]	18.98	12.35	35.27	15.99	40.30	27.83	61.43	45.97
MAFT+ [13]	21.59	13.66	34.80	13.98	41.93	28.39	61.26	26.52
CAT-Seg [7]	22.60	13.90	38.99	15.73	42.22	31.51	61.43	45.97
dinov3.seg (Ours)	25.57	18.49	40.63	21.04	45.55	37.42	65.26	45.69
Δ	+2.97	+4.58	+1.64	+5.31	+3.33	+5.90	+3.83	-0.28

Table 10: Seen and unseen class (based on textual similarity) mIoU comparison across datasets. Δ indicates absolute improvement over CAT-Seg.

4 Additional ablation studies

4.1 Ablation on number of late refinement blocks

Table 11 reports the effect of the number of late refinement blocks (N_L) in our framework, where each late refinement block consists of a spatial refinement block and a class refinement block. Using a single refinement block ($N_L=1$) leads to noticeably weaker performance on A-150 and A-847, while results on PAS-20 remain high, indicating limited sensitivity on simpler datasets. Increasing N_L to 2 yields consistent improvements across most datasets, giving the best overall performance. Further increasing N_L to 3 results in a slight drop on most datasets, suggesting over-refinement. As shown in Fig. 8, $N_L=2$ strikes the best balance between inference time and segmentation performance, and is therefore chosen as the default setting throughout the paper.

Table 11: Effect of number of late refinement blocks.

N_L	A-847	PC-459	A-150	PC-59	PAS-20	Avg.
1	18.09	26.52	40.19	63.85	97.81	49.29
2	18.70	26.51	41.57	64.05	97.61	49.69
3	18.43	26.28	41.34	64.09	97.75	49.58

4.2 Exploration with different SPEs

Table 12 reports the effect of different Semantic Prior Encoder (SPE) choices on segmentation performance, with all experiments conducted using our Config I model. In the *none* setting, DINOv3 features from the `dinov3.txt` backbone are used directly as semantic prior features. Incorporating a dedicated SPE consistently improves over this baseline, confirming the benefit of explicit semantic priors. Among all variants, SAM ViT-L achieves the best average mIoU of 49.69 with top scores on three out of five benchmarks, and is therefore set as our default SPE throughout the paper. While SAM3 PE-L+ and SAM-2.1 Hiera-B+

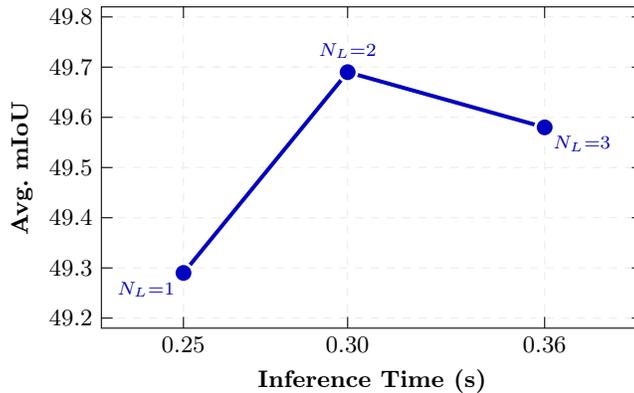


Fig. 8: Accuracy–runtime trade-off for varying numbers of late refinement blocks N_L . $N_L=2$ achieves the best average mIoU (49.69) with a moderate inference cost of 0.30 s.

are competitive on some datasets, neither matches the overall consistency of SAM ViT-L.

Table 12: Comparison of different Semantic Prior Encoders (SPE).

SPE	A-847	PC-459	A-150	PC-59	PAS-20	Avg.
<i>none</i>	18.23	26.15	40.30	63.04	97.76	49.10
SAM-2.1 Hiera-B+	18.33	26.86	41.04	63.77	97.70	49.54
SAM-2.1 Hiera-L	18.39	25.89	40.55	62.84	97.60	49.05
SAM3 PE-L+	18.51	26.60	41.16	63.72	97.84	49.57
SAM ViT-L	18.70	26.51	41.57	64.05	97.61	49.69

4.3 Exploration with different VLMs

Apart from `dinov3.txt`, we have experimented with two additional VLM backbones: CLIP (ViT-L) [29] and `dinov2.txt` [14]. Since CLIP only offers *global* text embeddings, we only utilize *global* embeddings instead of using text ensemble. QV finetuning is used for CLIP since it produces optimal results as shown in [7]. Table-13 compares performance across different VLM backbones. `dinov3.txt` consistently achieves the best performance on most datasets, with notable gains on A-847, PC-459, and A-150. `dinov2.txt` remains competitive on PC59 and PAS-20, while CLIP lags behind across all benchmarks. Notably, our method with CLIP (ViT-L) backbone outperforms multiple state-of-the-art methods [7, 27, 28] with same VLM backbone.

Table 13: Ablation on VLM backbone choice and comparison with state-of-the-art CLIP (ViT-L) based OVSS methods.

Method	A-847	PC-459	A-150	PC-59	PAS-20	Avg.
<i>Ours with Different VLM Backbones</i>						
Ours (CLIP ViT-L)	17.82	26.55	38.48	64.08	97.24	48.83
Ours (dinov2.txt)	17.25	26.07	39.61	64.46	97.74	49.03
Ours (dinov3.txt)	20.09	27.80	42.19	64.27	97.86	50.44
<i>Existing CLIP (ViT-L) based Methods</i>						
CAT-Seg [7]	16.0	23.8	37.9	63.3	97.0	47.60
H-CLIP [27]	16.5	24.2	38.4	64.1	97.7	48.18
HyperCLIP [28]	16.3	24.1	38.2	64.2	98.3	48.22

4.4 Ablation on loss function hyperparameters

We evaluate different combinations of λ and γ , as summarized in Table 14. Overall, $\lambda = 0.05$ and $\gamma = 2$ achieve the best average performance and consistently score best or near-best across most datasets compared to other settings. Notably, variations in average performance across different hyperparameter choices are relatively small.

Table 14: Loss function hyperparameter analysis.

λ	γ	A-847	PC-459	A-150	PC-59	PAS-20	Avg.
0.01	2	19.65	28.51	41.82	64.26	97.85	50.42
0.1	2	20.15	27.84	42.10	64.08	97.86	50.40
0.05	2	20.09	27.80	42.19	64.27	97.86	50.44
0.05	1	19.89	28.03	41.89	64.42	97.91	50.43
0.05	3	19.93	27.96	42.10	64.06	97.86	50.38

4.5 Qualitative analysis of Proposed LGA-based inference strategy.

Figure 9 presents a qualitative comparison between the CAT-Seg inference strategy and our proposed LGA-based inference strategy applied to our model. Unlike CAT-Seg, our approach extracts features from both the full image and its sub-images, which are then aggregated before being passed to subsequent modules. This allows our model to capture finer-grained visual details, resulting in sharper segmentation boundaries and more accurate classification of thin and complex object structures, as illustrated in the figure.

5 Comparison with CAT-Seg equipped with dinov3.txt

To demonstrate that the performance gains of our framework are not simply attributable to the stronger `dinov3.txt` backbone, we retrain CAT-Seg [7] us-

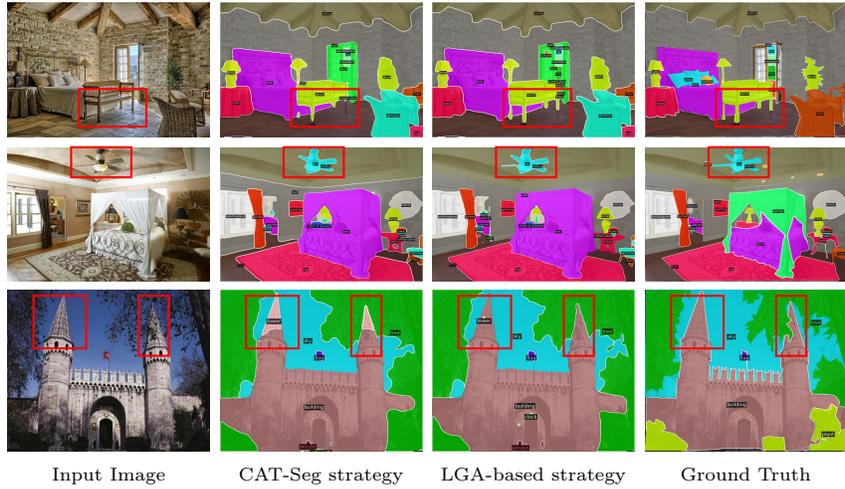


Fig. 9: Qualitative results on Inference Strategy. Highlighted regions in boxes show effectiveness of proposed LGA-based strategy over CAT-Seg inference strategy.

ing the same `dinov3.txt` backbone and report results in Table 15. As shown, `dinov3.seg` consistently outperforms this baseline across all benchmarks, with more pronounced improvements on the larger-vocabulary datasets A-847, PC-459, and A-150. This confirms that the gains stem from our proposed segmentation framework rather than the backbone alone.

Table 15: Comparison w.r.t CAT-Seg with `dinov3.txt`.

Method	A-847	PC-459	A-150	PC-59	PAS-20	Avg.
CAT-Seg w/ <code>dinov3.txt</code>	18.93	26.24	41.17	63.45	97.79	49.51
<code>dinov3.seg</code>	20.09	27.80	42.19	64.27	97.86	50.44

6 Complexity Analysis

Table 16 compares the computational complexity of `dinov3.seg` against existing OVSS methods. `dinov3.seg` carries a larger parameter count owing to `dinov3.txt` vision-language backbone, which accounts for 866.6M of the 1,178.2M total parameters, with the remaining 311.6M attributed to the Semantic Prior Encoder and auxiliary components. Despite this, the inference time of 0.37s remains well below that of OVSeg and SCAN (1.31s and 1.09s respectively). Notably, `dinov3.seg` achieves significantly lower GFLOPs (4,500.4) compared to both OVSeg (9,810.1) and SCAN (13,502.9), demonstrating that the increased parameter count does not translate to a proportional increase in computational cost at inference. While `dinov3.seg` is moderately heavier than CAT-Seg in terms of parameters and inference time, this overhead is modest relative to the consistent performance gains observed across all benchmarks. In future work, knowledge distillation from the Semantic Prior Encoder could be explored as a promising direction to reduce the overall computational cost of the framework.

Table 16: Model complexity comparison. Inference time and GFLOPs are measured on an A100 GPU at 640×640 resolution. GFLOPs are computed using the `fvcore` library.

Model	Parameters (M)			Inf. Time (s)	Inf. GFLOPs
	Total	VLM	Other		
OVSeg [19]	532.6	427.9	104.7	1.31	9,810.1
SCAN [21]	890.3	427.6	462.7	1.09	13,502.9
CAT-Seg [7]	433.7	427.9	5.8	0.22	1,963.5
<code>dinov3.seg</code>	1,178.2	866.6	311.6	0.37	4,500.4

7 Additional Qualitative Results

We show qualitative comparison with CAT-Seg on A-847, PC-459, A-150 and PC-59 datasets in Fig. 10, 11, 12, 13 respectively.

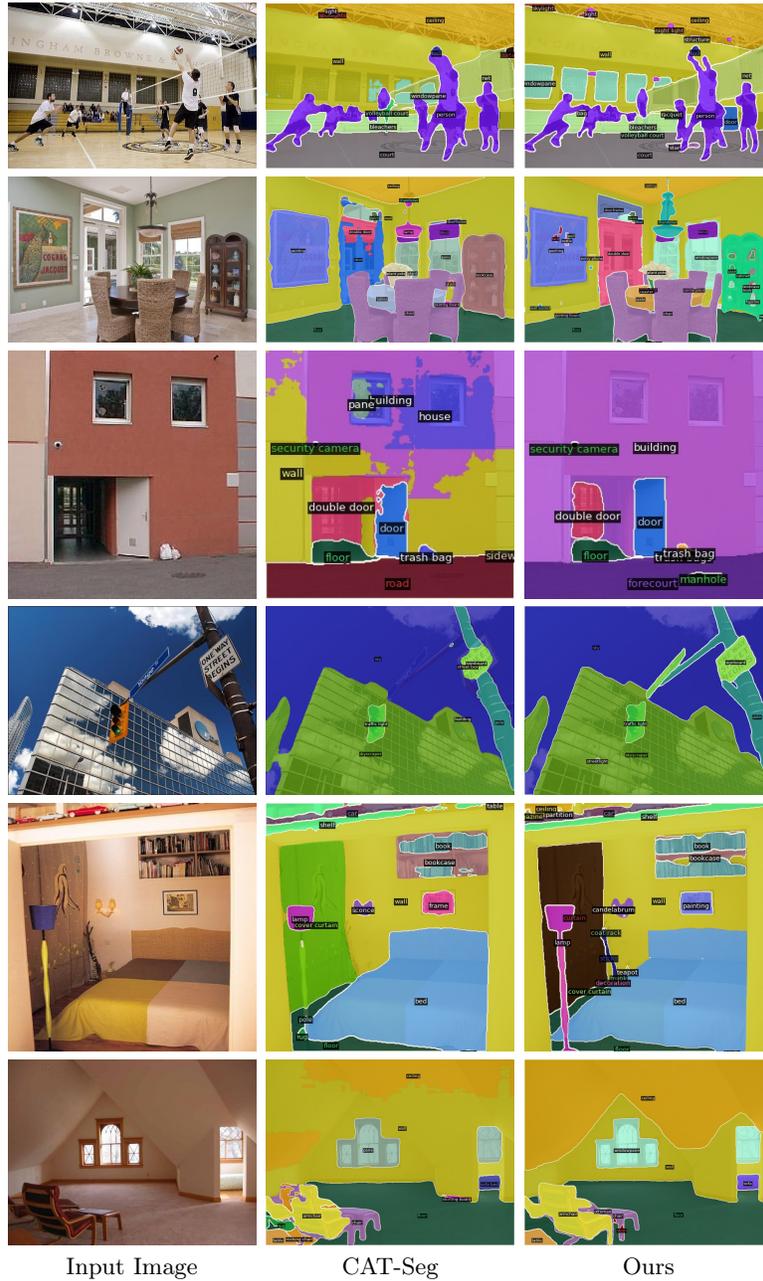
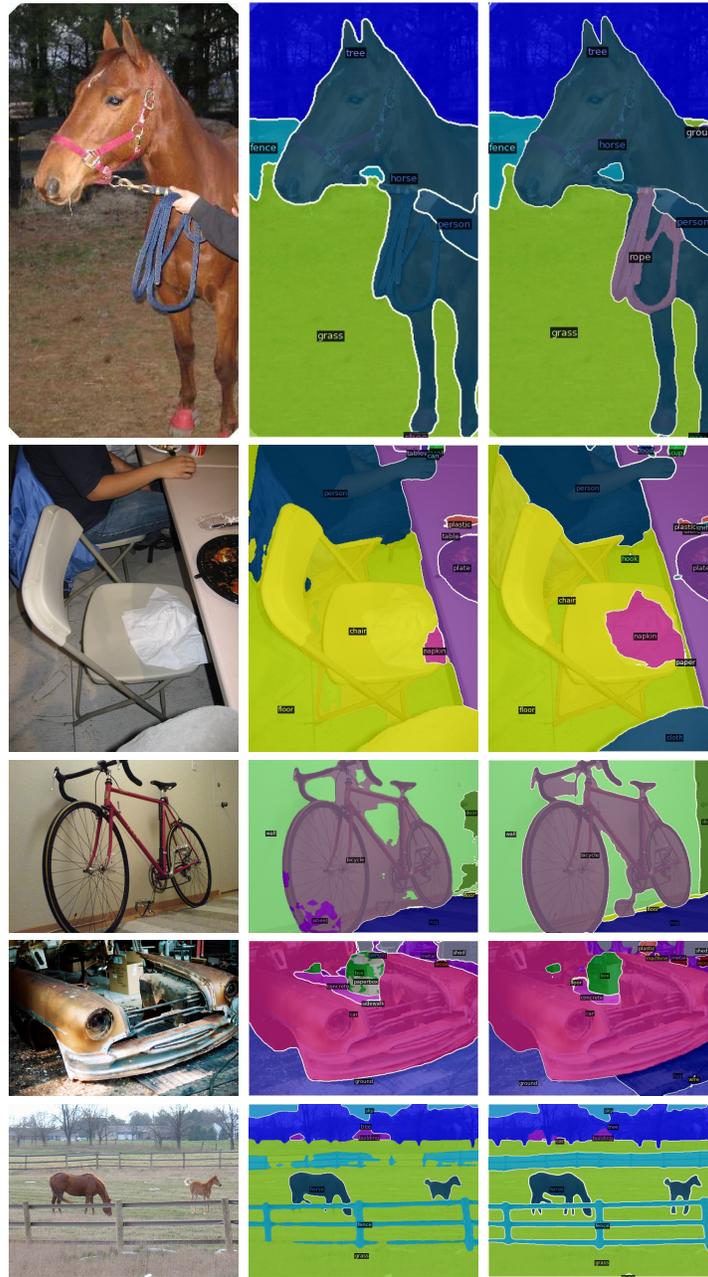


Fig. 10: Qualitative results on A-847 dataset.



Input Image CAT-Seg Ours

Fig. 11: Qualitative results on PC-459 dataset.

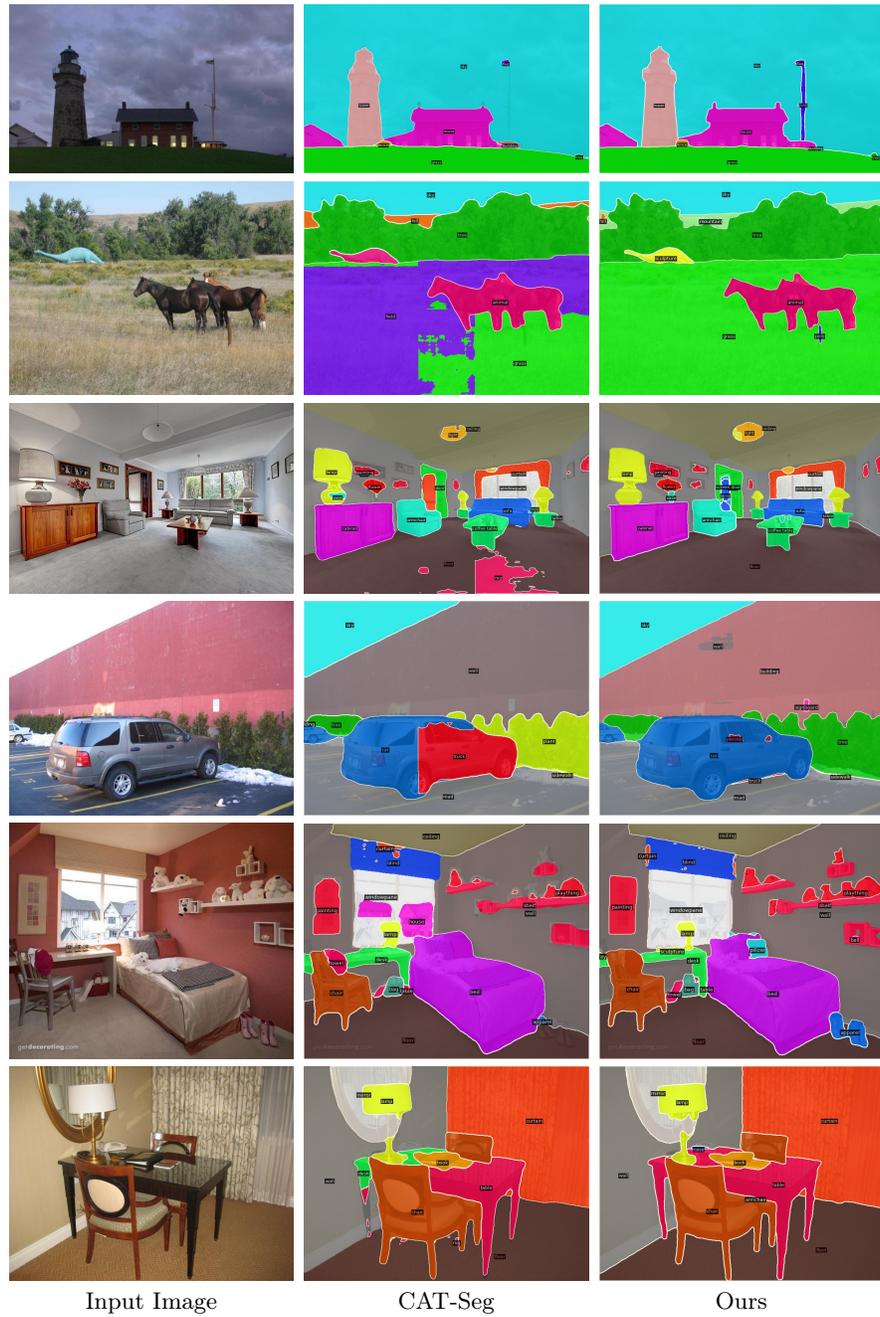


Fig. 12: Qualitative results on A-150 dataset.

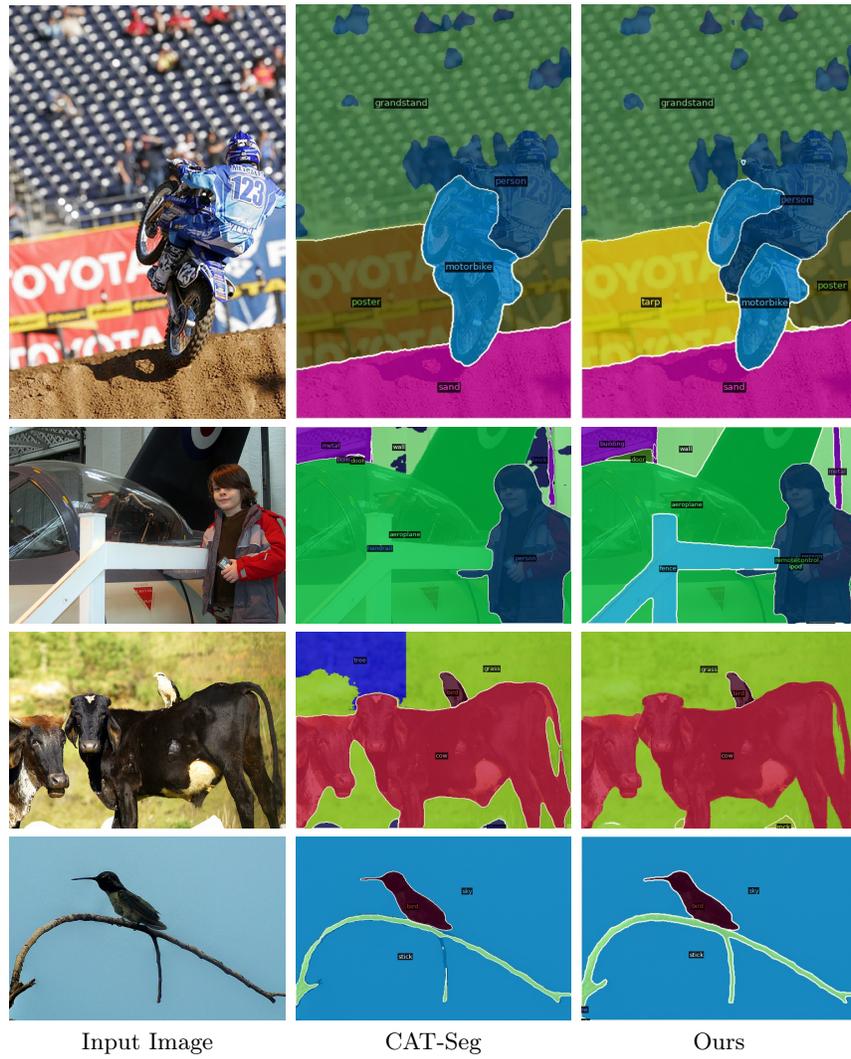


Fig. 13: Qualitative results on PC-59 dataset.