Open-Vocabulary Part Segmentation via Progressive and Boundary-Aware Strategy

Xinlong Li[1]* **Di Lin**[1]* **Shaoyiyi Gao**[1] **Jiaxin Li**[2] **Ruonan Liu**[3] **Qing Guo**[4]†

[1]Tianjin University, China [2]Southwest University, China
[3]Shanghai Jiao Tong University, China [4]VCIP, CS, Nankai University, China
lxl_zs@tju.edu.cn

Abstract

Open-vocabulary part segmentation (OVPS) struggles with structurally connected boundaries due to the inherent conflict between continuous image features and discrete classification mechanism. To address this, we propose PBAPS, a novel training-free framework specifically designed for OVPS. PBAPS leverages structural knowledge of object-part relationships to guide a progressive segmentation from objects to fine-grained parts. To further improve accuracy at challenging boundaries, we introduce a Boundary-Aware Refinement (BAR) module that identifies ambiguous boundary regions by quantifying classification uncertainty, enhances the discriminative features of these ambiguous regions using high-confidence context, and adaptively refines part prototypes to better align with the specific image. Experiments on Pascal-Part-116, ADE20K-Part-234, PartImageNet demonstrate that PBAPS significantly outperforms state-of-the-art methods, achieving 46.35% mIoU and 34.46% bIoU on Pascal-Part-116. Our code is available at https://github.com/TJU-IDVLab/PBAPS.

1 Introduction

Semantic segmentation aims to assign each pixel in an image to a predefined class. Traditional methods [1, 2, 3, 4, 5, 6] are based on supervised learning with labeled data [7, 8, 9] and have achieved significant progress in closed-set but exhibit limited zero-shot generalizability in open-world scenarios. Open-vocabulary semantic segmentation (OVSS) [10, 11, 12] addresses this limitation by leveraging pre-trained vision-language models (VLM) [13, 14, 15] to enable segmentation of unseen classes. Existing OVSS methods can be broadly categorized into two paradigms: (1) direct matching via cross-modal similarity [16, 12, 17] and (2) mask classification frameworks [18, 19, 20] based on feature clustering. Although effective for object-level segmentation, their part-level performance degrades significantly [21]. Open-vocabulary part segmentation (OVPS) [22, 17, 23] faces the additional challenge of ambiguous part boundaries.

We subdivide boundaries into three types: object boundary, structurally connected part boundary, and non-structurally connected part boundary. Structurally connected parts refer to anatomically adjacent components with direct physical connections (e.g., cat head and neck). Existing OVPS methods, including multigranularity segmentation based on object-part modeling [24, 25, 26] and fine-grained feature enhancement approaches [27, 28], commonly exhibit inaccuracies in segmenting intra-object structurally connected part boundaries (Figure 1). These approaches neglect the key distinction between OVSS and OVPS: object-level segmentation depends on distinct feature variations across boundaries, while part-level segmentation encounters smooth and continuous feature transitions (e.g.,

*Co-first authors.
†Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

fur texture/color changes) at structurally connected boundaries, in contrast to the moderate local feature differences at non-structurally connected part boundaries.

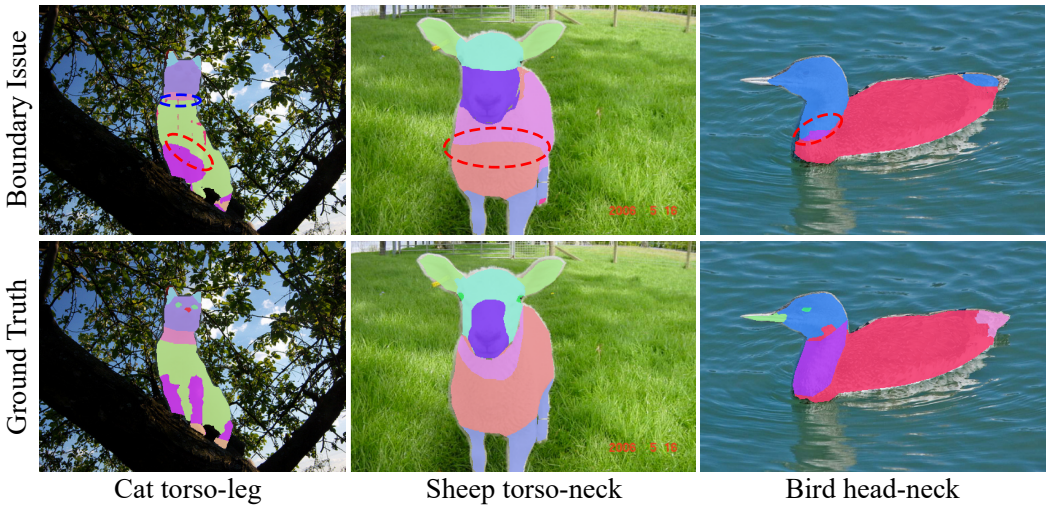


Figure 1: Boundary issues of existing methods on Pascal-Part-116 [21]. Dataset-specific part boundary definition variances limit the validity of pixel-alignment metrics for segmentation evaluation. For example, although the prediction boundary between cat neck and torso (PartCLIPSeg [28], blue dotted) deviates from annotations, it remains anatomically interpretable. Our method prioritizes anatomically/structurally implausible boundaries (red dotted).

The fundamental challenge in segmenting structurally connected part boundaries arises from the incompatibility between continuous, smooth image features and discrete classification mechanisms. Discrete classification enforces mutually exclusive label assignments and fails to account for pixels at structurally connected part boundaries, which exhibit hybrid characteristics of adjacent parts and lack clearly distinguishable features. To achieve accurate segmentation, it is essential to enhance the discrimination of these ambiguous features. Furthermore, since label assignment relies on category feature representations, adaptive optimization of category features can synergistically improve classification precision.

We propose the Boundary-Aware Refinement (BAR) module to refine the structurally connected part boundary. The BAR (1) first locates ambiguous boundary regions by analyzing the matching cost. In this work, we use the cosine similarity between the pixel features and part prototypes as the matching cost [29, 20]. The ambiguous regions are then separated from the original part masks, leaving the remaining deterministic regions as reliable references. (2) BAR then optimizes the pixel features within ambiguous regions by leveraging the context from deterministic regions and adaptively updates the part prototypes to better align with the current image characteristics. (3) Finally, ambiguous boundary regions are reclassified using enhanced pixel features and adapted prototypes.

In this paper, we propose Progressive Boundary-Aware Part Segmentation (PBAPS), a training-free OVPS framework that refines structurally connected part boundaries. Specifically, we first generate visual prototypes for each part class using Stable Diffusion [30], SAM [31], and DINOv2 [32]. Then, based on part structural relationships, a Hierarchical Part Connected Graph (HPCGraph) is constructed. Guided by this HPCGraph, progressive part segmentation is performed, where the BAR module mitigates boundary ambiguities and enhances segmentation precision.

In summary, the contribution of this paper is threefold: (i) We reveal the intrinsic cause of structurally connected part boundary errors: the conflict between continuous image features and discrete classification. (ii) We propose the BAR module, which improves boundary precision through feature optimization and dynamic prototype adaptation. (iii) We propose PBAPS, a novel and effective training-free OVPS method that integrates hierarchical reasoning with iterative boundary refinement, achieving state-of-the-art on Pascal-Part-116 [21], ADE20K-Part-234 [21], and PartImageNet [33].

2 Related Work

Open Vocabulary Semantic Segmentation. OVSS aims to overcome the limitations of predefined classes in traditional segmentation [34, 35, 36, 37, 38] by enabling zero-shot segmentation of unseen classes, requiring integration of the semantic understanding of VLM (e.g., CLIP [13], BLIP [14]) with pixel-level localization. Existing methods can be broadly categorized into two paradigms. One [10, 11, 16, 12, 17] is based on feature clustering and mask classification. OVSegmentor [39] leverages slot-attention [40] for pixel grouping and text alignment. RIM [41] employs image-to-image matching for training-free segmentation, constructing visual references and enhancing robustness via a relation-aware ranking distribution strategy. Our method generates part prototypes based on the process of constructing visual references in RIM. EBSeg [42] balances embeddings between base/novel classes and supplements spatial cues with SAM [31]. The second paradigm focuses on the alignment of the pixel-level features [18, 19, 20], which directly establishes associations between the vision and text features on the pixel scale. ZegCLIP [19] extends CLIP image-level classification capability to the pixel level. CAT-Seg [20] proposes a cost aggregation framework that optimizes CLIP image-text similarity through spatial and category aggregation.

Open Vocabulary Part Segmentation. Compared to OVSS, OVPS imposes higher demands on both model generalization and fine-grained recognition. Existing OVSS methods [43, 20, 21] often suffer significant performance degradation at the part level, mainly due to insufficient object-part structural modeling and limited fine-grained feature extraction. Recent studies have made notable progress. OPS [24] introduces class-agnostic part segmentation via object-aware spatial constraints and self-supervised feature optimization. ViRReq [25] decomposes part segmentation into composable atomic requests, leveraging a knowledge base for multigranular parsing. ViRReq and TAPPS [44] have previously explored object-to-part segmentation. In contrast, hierarchical reasoning in our method PBAPS introduces a cross-hierarchy matching mechanism, fully exploiting hierarchical context to strengthen feature discrimination. VLPart [45] constructs a joint vision-language embedding space through co-training, using DINO [46] dense semantic correspondence to parse novel objects into known components. In terms of benchmark construction, OV-PARTS [21] establishes the first OVPS benchmark (Pascal-Part-116, ADE20K-Part-234) with defined task scenarios. HIPIE [47] decouples hierarchical representations to separate foreground-background features. WPS-SAM [48] proposes a weakly supervised framework based on SAM [31], which reduces annotation dependency through co-training of learnable part prompt tokens and bounding box/point supervision. In feature enhancement, OIParts [27] fuses DINOv2 [32] local features with Stable Diffusion [30] global representations via adaptive channel selection. PartCLIPSeg [28] addresses fine-grained generalization by jointly optimizing separation and enhancement losses for part context modeling. PartCATSeg [49] improves semantic discrimination by constructing part-aware text embeddings combined with contrastive training, but lacks the ability to model structural relationships between parts.

3 Boundary Feature Gradient Analysis

To quantitatively analyze the feature variations across object boundaries, structurally connected part boundaries, and non-structurally connected part boundaries, we compute the spatial feature gradient. This feature gradient refers to the rate of spatial change in image feature vectors. Specifically, we extract pixel-wise features using DINOv2 [32]. For the feature vector $f(i,j) \in \mathbb{R}^{1\times d}$ of each pixel (i,j) in the image, we calculate the gradients in the horizontal (x) and vertical (y) directions for each channel c:

$$G_x(i,j,c) = |f(i+1,j,c) - f(i,j,c)|, \quad G_y(i,j,c) = |f(i,j+1,c) - f(i,j,c)| \tag{1}$$

For each pixel (i,j), we aggregate the gradients across all channels using the Euclidean norm to obtain the overall spatial feature gradient: $\text{Grad}(i,j) = \sqrt{\sum_{c=1}^{d} (G_x(i,j,c)^2 + G_y(i,j,c)^2)}$. A higher value of $\text{Grad}(i,j)$ corresponds to more pronounced local feature variation, whereas a lower value indicates smoother feature transitions. We extract these gradient values for all pixels located within the ground-truth masks of each boundary type and perform a systematic statistical analysis.

As shown in Table 1, the mean gradients and standard deviations (SD) exhibit consistent patterns across all three datasets: the object boundary shows the highest mean and SD values, while the

Table 1: Quantitative analysis of boundary feature gradients on Pascal-Part-116 [21], ADE20K-Part-234 [21], and PartImageNet [33]. The mean reflects the overall intensity of feature changes at the boundary. The standard deviation indicates the spatial consistency of boundary feature changes.

Boundary Type	Pascal-Part-116		ADE20K-Part-234		PartImageNet	
	Mean	SD	Mean	SD	Mean	SD
Object	0.5291	0.1300	0.5102	0.1253	0.5487	0.1352
Non-structurally connected part	0.5043	0.1164	0.4875	0.1108	0.5036	0.1204
Structurally connected part	0.4356	0.0762	0.4219	0.0721	0.4015	0.0658

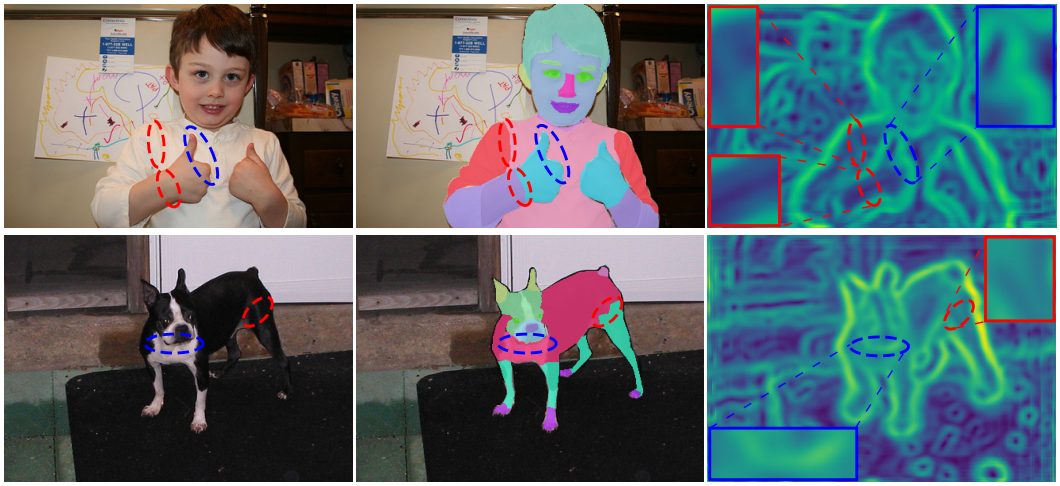


Figure 2: Heatmap of feature gradients. (1) Object boundary: feature gradients at the boundaries between human/dog and other objects are the highest. (2) Non-structurally connected part boundary (blue dotted): these have notably higher gradients than interior regions, indicating clear local variations. (3) Structurally connected part boundary (red dotted): feature gradients are relatively low, reflecting smooth feature transitions between adjacent parts.

structurally connected part boundary presents the lowest values. This indicates that, regardless of the scale and diversity of the dataset, the object boundary exhibits the most significant feature differences and the most drastic changes. The structurally connected part boundary exhibits smooth feature transitions, while the non-structurally connected part boundary shows moderate feature changes, reflecting local differences. These results are consistent with Figure 2, provide theoretical support for optimizing structurally connected part boundaries in the BAR module.

The low feature gradients at structurally connected part boundaries reflect the conflict between the continuous feature space and the discrete semantic space. When pixel features exhibit smooth transitions between structurally connected parts, their feature vectors blend characteristics from both adjacent parts, making it difficult for discrete classification methods based on thresholds or similarity metrics to assign accurate labels. Moreover, the low feature gradients at these boundaries violate the common assumption that "boundaries have high gradients", rendering conventional boundary localization ineffective for such boundary.

To address these challenges, we design the Boundary-Aware Refinement (BAR) module, which explicitly locates ambiguous and deterministic regions at structurally connected part boundaries through cost-divergence maps. By leveraging context from deterministic regions, BAR enhances the discriminative features of ambiguous regions and adaptively optimizes part prototypes, thereby improving the segmentation accuracy of structurally connected part boundaries.

4 Method

Given an input image, the OVPS method is required to assign the correct part class label to each pixel. Unlike closed-set semantic segmentation, OVPS allows the test set D_{test} to contain unseen part classes C_{unseen} that are not included in the training set. In our training-free framework, all classes in the test set belong to C_{unseen}.

As illustrated in Figure 3, we present PBAPS, a training-free OVPS framework that operates in three stages: (1) generating visual prototypes via foundation models, (2) constructing HPCGraph based on structural prior knowledge, and (3) performing hierarchical segmentation guided by HPCGraph, during which the BAR module refines the structurally connected part boundaries.

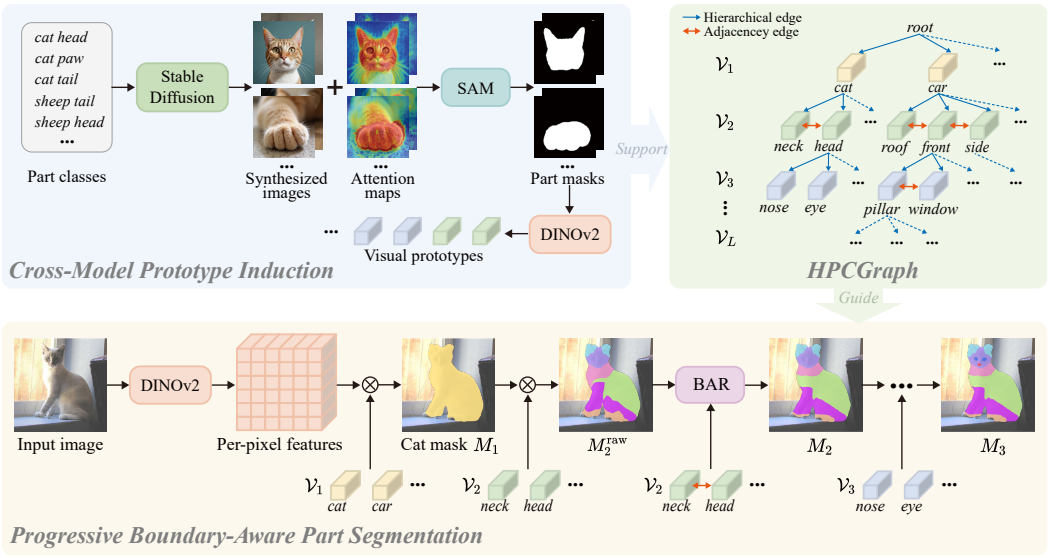


Figure 3: The architecture of PBAPS. Object-level segmentation is first performed through feature-prototype matching in $\mathcal{V}_1$, generating the object mask M_1.Taking the cat mask M_1 as an example, PBAPS then calculates part-level matching costs between the pixels and the corresponding cat part prototypes in $\mathcal{V}_2$, yielding the initial part mask M_2^{raw}. The BAR module subsequently refines structurally connected part boundaries in M_2^{raw} to obtain the optimized part mask M_2.

4.1 Cross-Model Prototype Induction

It is important to note that the paradigm of semantic segmentation via visual prototype construction has been adopted and validated as effective in existing studies, such as OVDiff [50] and RIM [41]. The novelty of this work does not lie in the prototype construction itself, but rather in addressing the low segmentation accuracy of the structurally connected part boundaries in OVPS. Building upon the prototype-based paradigm, our method performs hierarchical segmentation guided by the HPCGraph, while the Boundary-Aware Refinement (BAR) module further improves boundary precision through feature optimization and dynamic prototype adaptation.

To generate visual prototypes for part classes, we follow the proven pipeline [41] that integrates Stable Diffusion [30], SAM [31] and DINOv2 [32]. (1) First, Stable Diffusion generates synthetic images for each part c in D_{test} using text prompts (e.g., "a photo of c"), while extracting cross-attention maps to localize the target part c across multiple layers and timesteps [51, 50, 41]. These attention maps are processed via normalization and thresholding to generate a high-confidence region mask M. (2) Within regions where $M = 1$, we randomly sample k foreground prompt points for SAM, which produces a binary mask corresponding to part c. (3) Finally, DINOv2 extracts features from the masked regions of the synthetic images, and the visual prototype $\mathbf{p}_c \in \mathbb{R}^{1\times d}$ for c is computed as the average feature vector in these regions, serving as a global semantic representation for subsequent matching and boundary refinement.

4.2 Hierarchical Part Connected Graph

We construct the Hierarchical Part Connected Graph (HPCGraph) based on part structural relationships. The HPCGraph is defined as $G = (\mathcal{V}, \mathcal{E})$, where the node set $\mathcal{V} = \{\mathcal{V}_0, \mathcal{V}_1, \ldots, \mathcal{V}_L\}$ consists of $L+1$ hierarchical layers. Specifically, $\mathcal{V}_0 = \{\text{root}\}$ serves as the starting symbol, $\mathcal{V}_1$ denotes the object nodes, and $\mathcal{V}_l$ ($l > 1$) represents the part nodes of different granularity. The edge set $\mathcal{E}$ comprises two types of relationships: (1) Hierarchical edge $e_{p\to c}$ from the parent node $v_p \in \mathcal{V}_l$ to the child node $v_c \in \mathcal{V}_{l+1}$, representing the "composition" relationship (e.g., "cat head" $\to$ "cat eye"). (2) Adjacency edge e_{i-j} between nodes $v_i, v_j \in \mathcal{V}_l$ if they have structural connection (e.g., "cat head" $\leftrightarrow$ "cat neck").

Cross-hierarchy Matching Cost. To effectively exploit object-part and part-part relationships, we define the matching cost between a pixel feature $f \in \mathbb{R}^{1\times d}$ and a node $v \in \mathcal{V}$ as the maximum cosine similarity within its dominance set $\mathcal{D}(v) = \{v\} \cup \{\textit{descendant nodes of } v\}$:

$$S(f, v) = \max_{u\in\mathcal{D}(v)} \left[\frac{f^\top \mathbf{p}_u}{|f| \cdot |\mathbf{p}_u|}\right] \tag{2}$$

4.3 Progressive Boundary-Aware Part Segmentation

Given an input image $I \in \mathbb{R}^{h\times w\times 3}$, we extract pixel features $F \in \mathbb{R}^{h\times w\times d}$ using DINOv2, adopting a sliding-window strategy [52] to preserve fine spatial details. The progressive part segmentation is defined as $Segment(F, \mathcal{V}_l) \to M_l$, where $\mathcal{V}_l$ denotes the node set of the layer l and M_l represents the corresponding segmentation mask produced for the layer l.

Object-level Segmentation ($l = 1$). For each pixel feature $f(i, j)$ in the feature map F, we calculate cross-hierarchy matching costs with all nodes in $\mathcal{V}_1$. The resulting object-level segmentation mask M_1 is obtained by assigning to each pixel (i, j) the node with the highest matching score: $M_1(i, j) = \arg\max_{v\in\mathcal{V}_1} S\left(f(i, j), v\right)$.

Part-level Segmentation ($l \geq 2$). For each parent node $p \in \mathcal{V}_{l-1}$, perform localized segmentation within its corresponding mask $M_{l-1}(p)$. Specifically, we extract the parent-region feature $F_p = F \odot M_{l-1}(p)$ and calculate the cross-hierarchy matching costs between F_p and its child nodes $C \subseteq \mathcal{V}_l$ to generate raw mask $M^{\text{raw}}_{l,p} = \arg\max_{c\in C} S(F_p, c)$. The overall segmentation mask at layer l is $M^{\text{raw}}_l = \bigcup_{p\in\mathcal{V}_{l-1}} M^{\text{raw}}_{l,p}$. Subsequently, the BAR module optimizes structurally connected part boundaries through feature optimization and prototype adaptation, generating the final refined masks M_l. The progressive part segmentation ends when the current layer l contains only atomic parts (i.e., $\forall v \in \mathcal{V}_l, C(v) = \emptyset$).

4.4 Boundary-Aware Refinement

The BAR module refines structurally connected part boundaries via four steps, as shown in Figure 4.

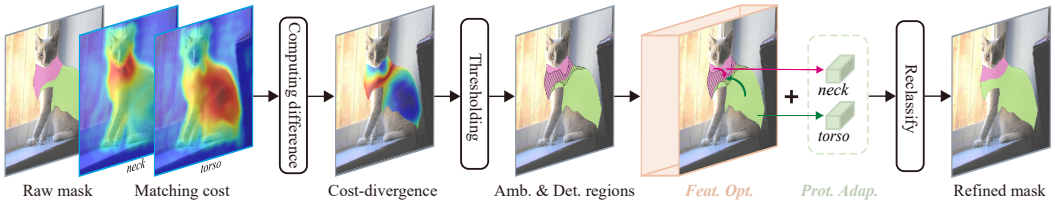


Figure 4: Example of Boundary-Aware Refinement. When segmenting the cat neck and torso, the corresponding matching-cost maps are obtained. Taking their absolute difference quantifies the classification ambiguity, where salient regions indicate similar matching costs. Thresholding this difference map yields ambiguous regions (shaded areas), and the rest as deterministic regions. Ambiguous-region feature optimization and prototype adaptation are then applied to refine the boundary between the cat neck and torso.

Ambiguous Boundary Region Localization. To address the uncertainty at structurally connected part boundaries, we first quantify the classification ambiguity using a cost-divergence map defined as $D_{A,B} = |S_A - S_B|$, where $S_A, S_B \in \mathbb{R}^{h \times w}$ denote the matching cost maps of the structurally connected parts A and B. The low divergence values correspond to regions where both parts exhibit similar matching costs, indicating semantic ambiguity. We normalize $D_{A,B}$ and apply a threshold to identify the ambiguous region $U = \mathbb{I}(\mathrm{Norm}(D_{A,B}) \leq \lambda)$, where $\lambda \in [0, 1]$ is the ambiguity threshold. The high-confidence deterministic regions are then obtained by excluding U from the initial part masks A_{raw}, B_{raw}:

$$A_{\mathrm{det}} = A_{\mathrm{raw}} \odot (1 - U), \quad B_{\mathrm{det}} = B_{\mathrm{raw}} \odot (1 - U) \tag{3}$$

Ambiguous Pixel Feature Optimization. The deterministic regions serve as reliable context for refining ambiguous pixels, as their features are distinct and free from classification uncertainty. For each ambiguous pixel $m \in U$, we enhance its discriminative characteristics by incorporating the context from deterministic regions A_{det} and B_{det}. Specifically, we first compute attention weights $w_A(m, a)$ by normalizing cosine similarities between the ambiguous pixel feature f_m and all pixels $a \in A_{\mathrm{det}}$. These weights quantify the relevance of each pixel in A_{det} to the ambiguous pixel m. Using $w_A(m, a)$ to aggregate contextual features from A_{det}:

$$w_A(m, a) = \frac{\exp(\cos(f_m, f_a))}{\sum_{e \in A_{\mathrm{det}}} \exp(\cos(f_m, f_e))}, \quad \mathbf{c}_A(m) = \sum_{a \in A_{\mathrm{det}}} w_A(m, a) f_a \tag{4}$$

Similarly, we compute $w_B(m, b)$ and aggregate the context of B_{det} into $\mathbf{c}_B(m)$. The optimized feature $\tilde{f}_m$ is then derived by fusing the original feature f_m with the contextual feature of both parts, this fusion mitigates the ambiguity of hybrid features by "pulling" f_m toward the discriminative characteristics of deterministic regions, thereby improving feature separability and boundary precision:

$$\tilde{f}_m = \gamma \cdot f_m + (1 - \gamma) \cdot \frac{\mathbf{c}_A(p) + \mathbf{c}_B(p)}{2} \tag{5}$$

Visual Prototype Adaptive Refinement. To adapt global part knowledge to the specific image context, we refine visual prototypes by integrating global priors with local image features. For the deterministic region of part A, we cluster its features $F^A_{\mathrm{det}} = \{f_a \mid a \in A_{\mathrm{det}}\}$ using K-means to obtain a local prototype $\mathbf{q}_A$, which captures the dominant appearance patterns of part A in the current image. The adaptive prototype is then computed by fusing the global prototype $\mathbf{p}_A$ with its corresponding local prototype $\mathbf{q}_A$:

$$\tilde{\mathbf{p}}_A = \alpha \cdot \mathbf{p}_A + (1 - \alpha) \cdot \mathbf{q}_A \tag{6}$$

where $\alpha \in [0, 1]$. This adaptation allows $\tilde{\mathbf{p}}_A$ to preserve the universal characteristics of part A while incorporating image-specific variations (e.g., pose, texture), making it more relevant for matching ambiguous pixels in the current scene. Similarly, obtain $\tilde{\mathbf{p}}_B$ for part B.

Notably, the similarity between synthetic prototypes and real test images can affect segmentation performance. Our approach achieves a balance between effectiveness and generalization by employing synthetic prototypes with moderate similarity and combining them with the dynamic prototype adaptation mechanism. This design maintains high segmentation accuracy without overfitting to specific datasets, enabling robust performance across diverse domains. Moreover, even when the attention-based masks derived from Stable Diffusion contain minor inaccuracies (e.g., small non-target regions), the prototype adaptation mechanism effectively mitigates such issues. By combining global prototypes with contextual features from deterministic regions, the resulting prototypes dynamically align with the true object parts present in the image, thus enhancing segmentation precision and robustness.

Ambiguous Region Reclassification. For each ambiguous pixel $m \in U$, we perform a reclassification using the optimized pixel feature $\tilde{f}_m$ and the adaptive prototypes $\tilde{\mathbf{p}}_A, \tilde{\mathbf{p}}_B$:

$$y(m) = \arg \max_{c \in \{A, B\}} \cos(\tilde{f}_m, \tilde{\mathbf{p}}_c) \tag{7}$$

5 Experiments

5.1 Datasets and Evaluation

We evaluate PBAPS on three benchmarks: (1) Pascal-Part-116 [21], which refines the Pascal-Part [53] by merging over-segmented parts and removing redundant descriptors. The validation set includes 17 object classes, 116 part classes, and 850 images. (2) ADE20K-Part-234 [21], derived from ADE20K [54] via low-frequency class filtering and synonym merging, containing 44 object classes, 234 part classes, and 1016 validation images. (3) PartImageNet [33], which groups 158 ImageNet [55] classes into 11 superclasses with uniform part structures, follows prior work [28] to evaluate 40 common object categories on 2957 validation images. In line with previous works [45, 28], we use the mean Intersection over Union (mIoU) to assess overall segmentation quality. Additionally, the boundary Intersection over Union (bIoU) [56] is introduced to specifically evaluate the accuracy of part boundary.

5.2 Implementation Details

Visual Prototype Generation. For each part class, we generate hundreds of 512×512 synthetic images using Stable Diffusion v1.4 [30], along with their corresponding cross-modal attention maps. After binarizing attention maps (threshold=0.7), 5 prompt points are randomly sampled for ViT-B SAM [31] to obtain the corresponding part masks. The features within the masked regions are then extracted using ViT-B DINOv2 [32]. Finally, K-means clustering ($K = 4$) is applied to these features to construct subcategory prototypes, thus capturing intra-class morphological diversity and improving the robustness of the part prototype.

Model Inference. We extract pixel-wise features from input images using ViT-B DINOv2 [32] with a sliding-window strategy [52], using the "key" values from the final attention layer as feature representations [57]. The hyperparameters of the BAR module are fixed as follows: ambiguity threshold $\lambda_{\text{amb}} = 0.3$, feature fusion weight $\gamma = 0.8$, and prototype adaptation coefficient $\alpha = 0.7$.

5.3 Comparison with Existing Methods

Table 2: Comparison with existing methods. * denotes our re-implementation. Bold and underline indicate the best and second-best results, respectively.

Method	Backbone	Supervision	Zero-shot transfer	Pascal-Part-116		ADE20K-Part-234		PartImageNet	
				mIoU	bIoU	mIoU	bIoU	mIoU	bIoU
ZSSeg+ [16, 21]	ResNet-50	class label	✗	24.91	18.18	19.84	12.89	-	-
VLPart [45]	ResNet-50	class label	✗	25.98	16.79	-	-	-	-
CLIPSeg [43]	ViT-B/16	class label	✗	24.23	15.98	5.88	4.87	26.98	18.54
CAT-Seg [20]	ViT-B/16	class label	✗	30.53	21.25	8.88	7.71	28.56	20.21
PartCATSeg [49]	ViT-B/16	class label	✗	29.54	20.96	14.68	11.99	30.18	21.29
PartCLIPSeg [28]	ViT-B/16	class label	✗	35.96	26.72	12.64	9.69	30.38	21.04
OVDiff [50]	UNet	Training-free		41.55	31.57	<u>22.13</u>	<u>15.83</u>	38.22	<u>26.71</u>
RIM* [41]	UNet+ViT-B/16	Training-free		<u>43.19</u>	<u>32.13</u>	20.91	13.30	<u>39.32</u>	25.06
PBAPS (ours)	UNet+ViT-B/16	Training-free		**46.35**	**34.46**	**24.70**	**16.41**	**42.61**	**29.31**

We compare PBAPS with ZSSeg+, VLPart, CLIPSeg, CAT-Seg, PartCATSeg and PartCLIPSeg, which are fine-tuned on target benchmarks. As shown in Table 2, PBAPS outperforms the state-of-the-art full-supervised method PartCLIPSeg [28] by more 10% in mIoU across the three datasets, with bIoU improvements of 7.74%, 6.72% and 8.27% on Pascal-Part-116, ADE20K-Part-234, and PartImageNet, respectively. We also compare PBAPS with training-free OVSS methods OVDiff [50] and RIM [41], using identical visual prototype set for fairness. All three visual prototype matching methods outperform finetuning-based methods, validating the effectiveness of this paradigm in OVPS. Our method achieves state-of-the-art mIoU and bIoU scores on all three datasets, surpassing OVDiff and RIM. Figure 5 visually demonstrates its superiority. Unlike RIM, which under-segments when target parts lack distinct boundaries, PBAPS captures subtle feature transitions between connected parts, enhancing segmentation accuracy.

5.4 Analysis and Ablation Study

Table 3: Ablation study for PBAPS on Pascal-Part-116. The baseline model (1st row) excludes all components, directly matching pixel-wise features extracted by DINOv2 [32] with visual prototypes.

HPCGraph		BAR			Pascal-Part-116	
w/ Hier. Seg.	w/ X-Hier. Match	w/ Boun. Loc.	w/ Feat. Opt.	w/ Proto. Adapt.	mIoU	bIoU
					40.74	31.09
✓					43.37	32.66
✓	✓				44.08	32.99
✓	✓		✓		43.86	32.89
✓	✓			✓	44.83	33.42
✓	✓	✓	✓		45.58	34.04
✓	✓	✓		✓	<u>45.79</u>	<u>34.16</u>
✓	✓	✓	✓	✓	**46.35**	**34.46**

Ablation study for HPCGraph. The HPCGraph enhances part-level semantic modeling via hierarchical segmentation and cross-hierarchy matching. As shown in Table 3, incorporating hierarchical segmentation (2nd row) increases mIoU and bIoU by 2.63% and 1.57%, respectively, indicating that modeling top-down part relationships effectively improves segmentation consistency. Further integration of cross-hierarchy matching (3rd row) raises mIoU and bIoU to 44.08% and 32.99%, demonstrating that cross-level prototype comparison strengthens fine-grained part discrimination and improves boundary precision.

Ablation study for BAR. As shown in Table 3, using feature optimization alone without ambiguous boundary localization (4th row) leads to slight performance degradation, mIoU and bIoU drop by 0.22% and 0.1% compared to the 3rd row, indicating that indiscriminate context fusion weakens local discriminative features. Introducing boundary region localization (6th row) improves mIoU and bIoU by 1.5% and 1.05%, validating the importance of distinguishing deterministic from ambiguous regions. Moreover, the 5th and 7th rows demonstrate the synergistic effect between prototype adaptation and boundary localization, which jointly improve boundary precision. When all components are integrated (8th row), mIoU and bIoU increase by 5.61% and 3.37% over the baseline, confirming the overall effectiveness of the BAR module in refining part boundaries.

Effectiveness of boundary region localization. As shown in Table 4, we assess the boundary localization capability of the BAR module by computing boundary recall, which measures the proportion of true boundary pixels correctly covered by the detected ambiguous regions. When the ambiguity threshold is set to $\lambda_{\text{amb}} = 0.3$, the model achieves the optimal balance between the boundary recall (78.93%) and the segmentation precision (46.35% mIoU). A strict threshold ($\lambda_{\text{amb}} = 0.1$) filters out 64.78% of true boundary pixels, leading to insufficient refinement, while a lenient threshold ($\lambda_{\text{amb}} = 0.5$) improves recall to 88.97% but introduces non-boundary noise, reducing mIoU and bIoU.

Table 4: Pascal-Part-116 results with different ambiguity threshold.

λ_{amb}	Boundary Recall	mIoU	bIoU
0.1	35.22	44.48	33.42
0.2	58.70	45.54	33.81
0.3	78.93	**46.35**	**34.46**
0.4	<u>82.15</u>	<u>45.71</u>	<u>34.01</u>
0.5	**88.97**	44.29	33.10

Table 5: Impact of part boundary optimization types on Pascal-Part-116.

Optimization Type	mIoU	bIoU	Inference Time (s)
Baseline	44.08	32.99	**1.47**
Non-struct. boundary	43.19	32.48	2.06
All part boundary	<u>45.43</u>	<u>33.95</u>	2.19
Struct. boundary	**46.35**	**34.46**	<u>1.81</u>

Impact of BAR for different boundary. As shown in Table 5, we analyze the impact of the BAR module on different types of part boundaries. The baseline (1st row) employs the HPCGraph without BAR. For structurally connected part boundaries, BAR achieves the most significant improvements (2.27% mIoU, 1.47% bIoU). In contrast, applying BAR to non-structurally connected part boundaries (2nd row) slightly decreases both mIoU and bIoU, while increasing inference time by 0.25s compared

to the 4th row. This degradation occurs because such boundaries are already well separated, and additional context fusion introduces noise that blurs the boundaries. Moreover, the greater number of boundaries increases the overall inference time. Although applying BAR to all part boundaries improves mIoU by 1.35%, it increases the inference time by 48%, confirming that selective optimization of structurally connected boundaries offers the best balance between accuracy and efficiency.

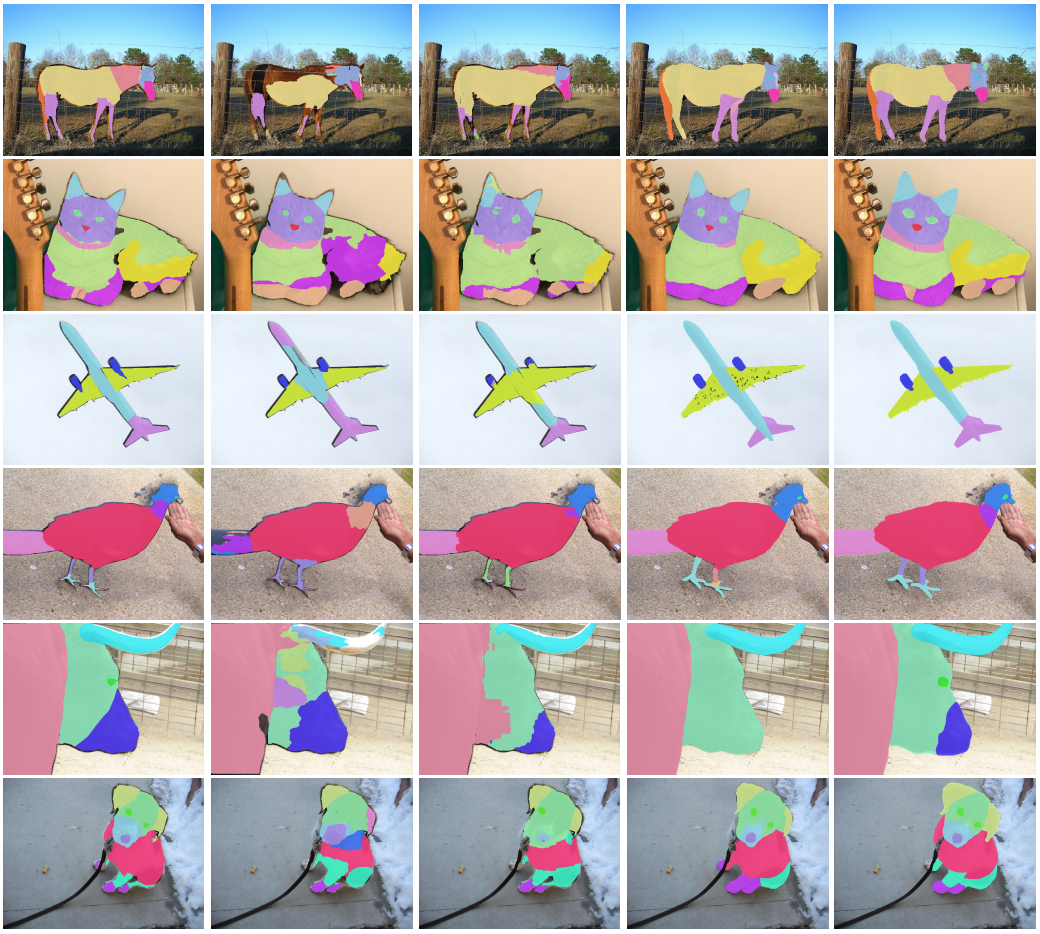


Figure 5: Qualitative results on Pascal-Part-116.

6 Conclusion

This study addresses the inaccurate segmentation of structurally connected part boundaries by introducing PBAPS, a training-free OVPS framework. PBAPS integrates a progressive segmentation strategy guided by HPCGraph with a BAR module to enhance boundary precision. Extensive experiments demonstrate that PBAPS consistently outperforms state-of-the-art methods across multiple benchmarks, confirming its effectiveness and generalizability. This study offers a novel and interpretable solution for fine-grained part segmentation in open-world scenarios.

7 Acknowledgement

This research was supported by the National Natural Science Foundation of China (No.62476192) and the Natural Science Foundation of Tianjin (No.23JCQNJC02010). The support of these foundations has been instrumental in advancing this study, for which we express our sincere gratitude.

References

[1] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[4] Wei Jiang, Yan Wu, Linting Guan, and Junqiao Zhao. Dfnet: Semantic segmentation on panoramic images with dynamic loss weights and residual fusion block. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5887–5892. IEEE, 2019.

[5] Nabil Ibtehaz and M Sohel Rahman. Multiresunet: Rethinking the u-net architecture for multimodal biomedical image segmentation. *Neural networks*, 121:74–87, 2020.

[6] Di Lin, Dingguo Shen, Yuanfeng Ji, Siting Shen, Mingrui Xie, Wei Feng, and Hui Huang. Tagnet: Learning configurable context pathways for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2475–2491, 2022.

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[8] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018.

[9] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111:98–136, 2015.

[10] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *European conference on computer vision*, pages 540–557. Springer, 2022.

[11] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11583–11592, 2022.

[12] Xiaoyi Dong, Jianmin Bao, Yinglin Zheng, Ting Zhang, Dongdong Chen, Hao Yang, Ming Zeng, Weiming Zhang, Lu Yuan, Dong Chen, et al. Maskclip: Masked self-distillation advances contrastive language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10995–11005, 2023.

[13] 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 *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[14] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[15] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021.

[16] 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 *European Conference on Computer Vision*, pages 736–753. Springer, 2022.

[17] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2955–2966, 2023.

[18] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. San: side adapter network for open-vocabulary semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):15546–15561, 2023.

[19] Ziqin Zhou, Yinjie Lei, Bowen Zhang, Lingqiao Liu, and Yifan Liu. Zegclip: Towards adapting clip for zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11175–11185, 2023.

[20] Seokju Cho, Heeseong Shin, Sunghwan Hong, Anurag Arnab, Paul Hongsuck Seo, and Seungryong Kim. Cat-seg: Cost aggregation for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4113–4123, 2024.

[21] Meng Wei, Xiaoyu Yue, Wenwei Zhang, Shu Kong, Xihui Liu, and Jiangmiao Pang. Ov-parts: Towards open-vocabulary part segmentation. *Advances in Neural Information Processing Systems*, 36:70094–70114, 2023.

[22] Zheng Ding, Jieke Wang, and Zhuowen Tu. Open-vocabulary panoptic segmentation maskclip. 2022.

[23] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. *Advances in Neural Information Processing Systems*, 36:32215–32234, 2023.

[24] Tai-Yu Pan, Qing Liu, Wei-Lun Chao, and Brian Price. Towards open-world segmentation of parts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15392–15401, 2023.

[25] Chufeng Tang, Lingxi Xie, Xiaopeng Zhang, Xiaolin Hu, and Qi Tian. Visual recognition by request. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15265–15274, 2023.

[26] Bin Xie, Jiale Cao, Jin Xie, Fahad Shahbaz Khan, and Yanwei Pang. Sed: A simple encoder-decoder for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3426–3436, 2024.

[27] Zhenqi Dai, Ting Liu, Xingxing Zhang, Yunchao Wei, and Yanning Zhang. One-shot in-context part segmentation. In *ACM Multimedia*, volume 2024, 2024.

[28] Jiho Choi, Seonho Lee, Seungho Lee, Minhyun Lee, and Hyunjung Shim. Understanding multi-granularity for open-vocabulary part segmentation. *arXiv preprint arXiv:2406.11384*, 2024.

[29] Biyang Liu, Huimin Yu, and Guodong Qi. Graftnet: Towards domain generalized stereo matching with a broad-spectrum and task-oriented feature. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13012–13021, 2022.

[30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[31] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[32] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[33] Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jie-Neng Chen, Shuai Liu, Cheng Yang, Qihang Yu, and Alan Yuille. Partimagenet: A large, high-quality dataset of parts. In *European Conference on Computer Vision*, pages 128–145. Springer, 2022.

[34] Di Lin, Dingguo Shen, Siting Shen, Yuanfeng Ji, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Zigzagnet: Fusing top-down and bottom-up context for object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7490–7499, 2019.

[35] Di Lin and Hui Huang. Zig-zag network for semantic segmentation of rgb-d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2642–2655, 2019.

[36] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7262–7272, 2021.

[37] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

[38] Naisong Luo, Yuwen Pan, Rui Sun, Tianzhu Zhang, Zhiwei Xiong, and Feng Wu. Camouflaged instance segmentation via explicit de-camouflaging. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17918–17927, 2023.

[39] Jilan Xu, Junlin Hou, Yuejie Zhang, Rui Feng, Yi Wang, Yu Qiao, and Weidi Xie. Learning open-vocabulary semantic segmentation models from natural language supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2935–2944, 2023.

[40] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in neural information processing systems*, 33:11525–11538, 2020.

[41] Yuan Wang, Rui Sun, Naisong Luo, Yuwen Pan, and Tianzhu Zhang. Image-to-image matching via foundation models: A new perspective for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3952–3963, 2024.

[42] Xiangheng Shan, Dongyue Wu, Guilin Zhu, Yuanjie Shao, Nong Sang, and Changxin Gao. Open-vocabulary semantic segmentation with image embedding balancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28412–28421, 2024.

[43] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7086–7096, 2022.

[44] Daan De Geus and Gijs Dubbelman. Task-aligned part-aware panoptic segmentation through joint object-part representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3174–3183, 2024.

[45] Peize Sun, Shoufa Chen, Chenchen Zhu, Fanyi Xiao, Ping Luo, Saining Xie, and Zhicheng Yan. Going denser with open-vocabulary part segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15453–15465, 2023.

[46] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022.

[47] Xudong Wang, Shufan Li, Konstantinos Kallidromitis, Yusuke Kato, Kazuki Kozuka, and Trevor Darrell. Hierarchical open-vocabulary universal image segmentation. *Advances in Neural Information Processing Systems*, 36:21429–21453, 2023.

[48] Xin-Jian Wu, Ruisong Zhang, Jie Qin, Shijie Ma, and Cheng-Lin Liu. Wps-sam: Towards weakly-supervised part segmentation with foundation models. In *European Conference on Computer Vision*, pages 314–333. Springer, 2024.

[49] Jiho Choi, Seonho Lee, Minhyun Lee, Seungho Lee, and Hyunjung Shim. Fine-grained image-text correspondence with cost aggregation for open-vocabulary part segmentation. *arXiv preprint arXiv:2501.09688*, 2025.

[50] Laurynas Karazija, Iro Laina, Andrea Vedaldi, and Christian Rupprecht. Diffusion models for zero-shot open-vocabulary segmentation. *arXiv e-prints*, pages arXiv–2306, 2023.

[51] Weijia Wu, Yuzhong Zhao, Mike Zheng Shou, Hong Zhou, and Chunhua Shen. Diffumask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1206–1217, 2023.

[52] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Hrda: Context-aware high-resolution domain-adaptive semantic segmentation. In *European conference on computer vision*, pages 372–391. Springer, 2022.

[53] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1971–1978, 2014.

[54] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

[55] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[56] Bowen Cheng, Ross Girshick, Piotr Dollár, Alexander C Berg, and Alexander Kirillov. Boundary iou: Improving object-centric image segmentation evaluation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15334–15342, 2021.

[57] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2(3):4, 2021.

[58] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

[59] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[60] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[61] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in neural information processing systems*, 34:17864–17875, 2021.

[62] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014.

[63] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9404–9413, 2019.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The Introduction (Section 1) explicitly states our contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our method in Appendix A.1.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The mathematical formulations in Sections 3 and 4 are methodological components rather than theoretical propositions. The work focuses on algorithmic design and empirical validation.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The method PBAPS is detailed in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer:[Yes]

Justification: We open-source the code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5.2 comprehensively specifies the datasets, the foundation models, the hyperparameters.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Statistical significance is not a focus of our experiments, as the emphasis lies on comparing method effectiveness rather than statistical hypothesis testing.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: These can be found in the Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research does not involve sensitive data and the datasets used are standard benchmarks in the field (Section 5).

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss societal impacts in the Appendix A.2.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper focuses on a segmentation framework using publicly available part segmentation datasets and does not involve the release of high-risk models

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper appropriately cites all assets used (Section 1), such as datasets and models.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper introduces a novel framework (PBAPS) and methodology but does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The research does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The study does not involve human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The PBAPS relies on computer vision techniques without employing LLMs as components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Discussion

A.1 Limitations and Future Work

Although our method shows excellent performance in the OVPS task, it still has several limitations. First, PBAPS focuses on part-level semantic segmentation and cannot distinguish different instances of the same category. For example, when multiple cats appear in an image, PBAPS segments all "cat head" without assigning each head to its respective individual. Second, our method relies on the HPCGraph, which defines hierarchical and adjacency relationships between objects and parts via manual design or external knowledge bases. However, when processing objects with complex, irregular structures (e.g., modular furniture), the statically predefined graph may fail to capture their intrinsic part topology and semantic associations.

To address these limitations, future work can explore two directions. On the one hand, introducing instance-aware mechanisms could enable simultaneous semantic segmentation and instance separation. On the other hand, developing a dynamic hierarchical reasoning framework that can adaptively adjust part hierarchical structures based on specific objects.

A.2 Social Impact

Our method offers significant societal value in multiple domains. In healthcare, its precise segmentation of structurally connected parts can improve the efficiency of pathological diagnosis and provide a low-cost, annotation-free diagnostic aid for remote regions, thereby promoting equitable access to medical resources. In industrial manufacturing, its hierarchical part reasoning capability supports automated quality inspection and disassembly of mechanical equipment.

Despite these advantages, PBAPS also has potential risks. In safety-critical applications such as medical imaging and autonomous driving, reliance on predefined structural priors may lead to segmentation errors in complex scenarios. For example, misidentifying the boundaries of rare pathological organs, causing diagnostic inaccuracies.

B Additional Experiments

B.1 Further Details

Stable Diffusion. A text-conditioned latent diffusion model [30] that generates images via iterative denoising and has three main components: (1) a pre-trained VAE [58] for image encoding/decoding, (2) a text encoder τ converting the prompt into embedding vector, and (3) a time-conditional U-Net ϕ that denoises an initial Gaussian noise to produce the image. During image synthesis, the corresponding cross-attention maps A [51] are extracted to locate the target part. Specifically, the prompt p is encoded in text embedding $\tau(p) \in \mathbb{R}^{N\times d}$, where N denotes the length of the token sequence. At each U-Net ϕ timestep t, visual features $\phi(z_t) \in \mathbb{R}^{H\times W\times C}$ from a noisy image z_t are flattened and projected to query Q, while text embedding $\tau(p)$ produces the key K and the value V via learnable layers ℓ_K and ℓ_V:

$$Q = \ell_Q(\phi(z_t)), \quad K = \ell_K(\tau(p)), \quad V = \ell_V(\tau(p)) \tag{8}$$

The cross-attention weights $A = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \in \mathbb{R}^{H\times W\times N}$. To obtain robust class attention maps, attention maps are aggregated over multiple layers and time steps:

$$\tilde{A}^j = \frac{1}{|B||T|}\sum_{b\in B}\sum_{t\in T}\frac{A^j_{b,t}}{\max(A^j_{b,t})} \tag{9}$$

where j is the index of the text token, B and T denote the layers and timestep sets, respectively.

Segment Anything Model. SAM [31] is a training-free image segmentation framework that enables prompt-based rapid segmentation of arbitrary objects using point, box, or mask input. Its architecture consists of: (1) an image encoder that extracts the global visual feature F_i from the input image, (2) a prompt encoder that encodes prompts into unified features F_p, and (3) a mask decoder that generates candidate masks by integrating F_i with F_p.

DINOv2. DINOv2 [32] is a discriminative self-supervised ViT that distills general visual features from large-scale unlabeled data. Through joint global-image and local-patch level learning, its robust features excel in tasks including image classification, semantic segmentation, and patch matching.

For each part class c, we mainly adopt the generic prompt template "a photo of c" to guide Stable Diffusion [30] in synthesizing images of the part c. To further enrich visual diversity, we incorporate synonyms and subclasses of the original class names [51]. We also standardize class names (e.g., "tvmonitor" $\rightarrow$ "tv monitor") and refine them to resolve ambiguities (e.g., "cat hand" $\rightarrow$ "cat paw").

During image generation, we set an independent random seed for each class to ensure reproducibility. The image generation time scales linearly with the number of images: 124s for 32 images, 227s for 64 images, and 443s for 128 images. Each part prototype, extracted via DINOv2 [32], occupies 0.45 MB. During inference, PBAPS employs a sliding window of size 224×224 with a stride of 64 for feature extraction using DINOv2 (identical to all baseline methods), achieving an average processing time of 1.81s per image. All results are measured with a single NVIDIA A6000 GPU.

B.2 Ablation on Feature Extractors

Table 6: Comprison of different feature extractors on Pascal-Part-116 [21].

Feature Extractor	mIoU	bIoU
MAE [59]	37.94	27.47
DINO [60]	41.20	30.79
CLIP [13]	42.39	30.41
Stable Diffusion [30]	<u>44.12</u>	<u>33.31</u>
DINOv2 [32]	**46.35**	**34.46**

Table 7: Comprison of mask generators on Pascal-Part-116 [21].

Mask Generator	mIoU	bIoU
MaskFormer [61]	41.28	30.97
SAM [31]	<u>43.82</u>	<u>32.43</u>
None	**46.35**	**34.46**

Our method can integrate with any pretrained visual feature extractor to construct visual prototypes and extract image features. As shown in Table 6, to demonstrate the superiority of using DINOv2 [32] for image feature extraction in our framework, we compare it with several self-supervised ViT feature extractors. DINOv2 significantly outperforms other methods, benefiting from its pretraining based on image-level and patch-level discriminative learning, which empowers it with fine-grained feature representation capabilities. CLIP exhibits limitations in fine-grained feature alignment, likely due to its contrastive learning that focuses on global feature alignment (Figure 6). MAE yields the weakest performance due to its lack of explicit semantic discriminative learning. Stable Diffusion performs secondarily by leveraging structural information implicitly learned through generative tasks.

Ground Truth w/ CLIP w/ DINOv2

Figure 6: Qualitative ablation on feature extractors in Pascal-Part-116 [21].

B.3 Ablation on Mask Generators

Our PBAPS framework supports two segmentation paradigms: (1) direct per-pixel feature classification and (2) region-level classification following mask generation by a pre-trained segmenter. To validate the superiority of our pixel-wise classification strategy, we compare it against MaskFormer-based, SAM-based variants. Note that when using MaskFormer [61] or SAM [31] for mask generation, the segmentation process operates only under the structural constraints of HPCGraph without boundary refinement via the BAR module, as these generators only produce binary masks, lacking class-specific matching cost maps. As shown in Table 7, introducing MaskFormer or SAM significantly degrades performance. This is because MaskFormer, trained on object-level COCO [62, 8, 63], tends to propose coarse object regions and fails to capture subtle part-level distinctions. Although SAM exhibits zero-shot generalization, its segmentation relies on prominent visual changes (e.g., texture, color). The part-level segmentation often involves only local, subtle variations, leading to frequent under-segmentation by SAM (Figure 7).

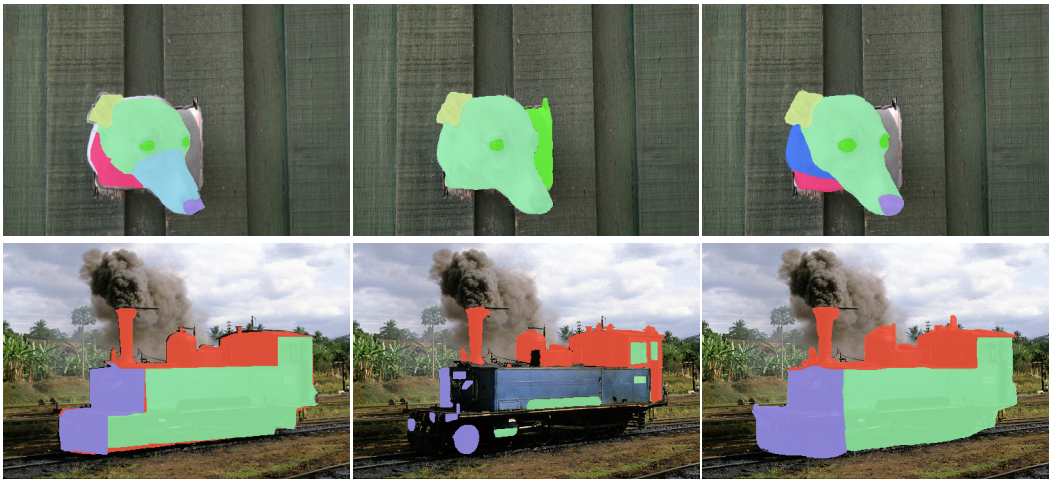
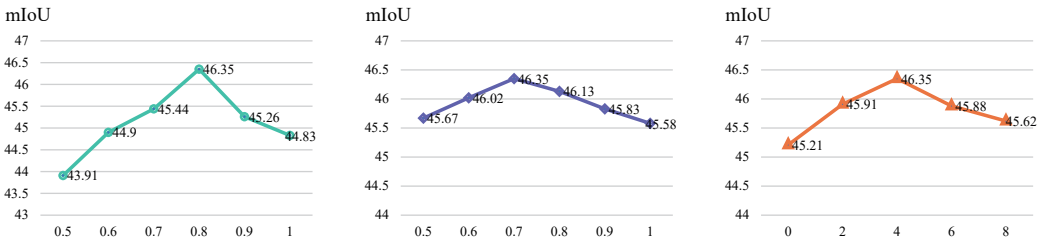


Figure 7: Qualitative ablation on mask generators in Pascal-Part-116 [21].

B.4 Sensitivity Analysis of Hyperparameters



(a) The feature fusion weight γ (b) The prototype adaptation coefficient α (c) The number of subprototypes K

Figure 8: The sensitivity analysis of γ, α and K on Pascal-Part-116 [21].

To evaluate the impact of key hyperparameters in PBAPS, we conduct a sensitivity analysis on Pascal-Part-116 for three core parameters: feature fusion weight γ, prototype adaptation coefficient α, and the number of subprototypes K in visual prototype generation. Figure 8 shows that when $\gamma = 0.8$, PBAPS achieves the highest mIoU of 46.35%. A moderate γ balances the retention of critical original feature information and the integration of discriminative context from deterministic regions. When γ is low (0.5), excessive reliance on contextual features suppresses original discriminative information, degrading performance. When γ is high (1.0), the neglect of contextual guidance leaves

feature ambiguity unresolved. Similarly, a moderate α (0.7) enables adaptive prototypes to retain both universal part knowledge and image-specific variations (e.g., pose, texture). When α is small, over-adaptation to local features may introduce noise or cause overfitting to image details. When α approaches 1.0, the prototypes lack adaptability to variations in the current image. Regarding the number of subprototypes, $K = 4$ allows the model to fully capture intra-class diversity while avoiding noise interference. When $K = 0$, the global prototype is overly generic; when $K > 4$, excessive subprototypes may introduce noisy clusters, leading to performance degradation.

B.5 Additional Qualitative Results

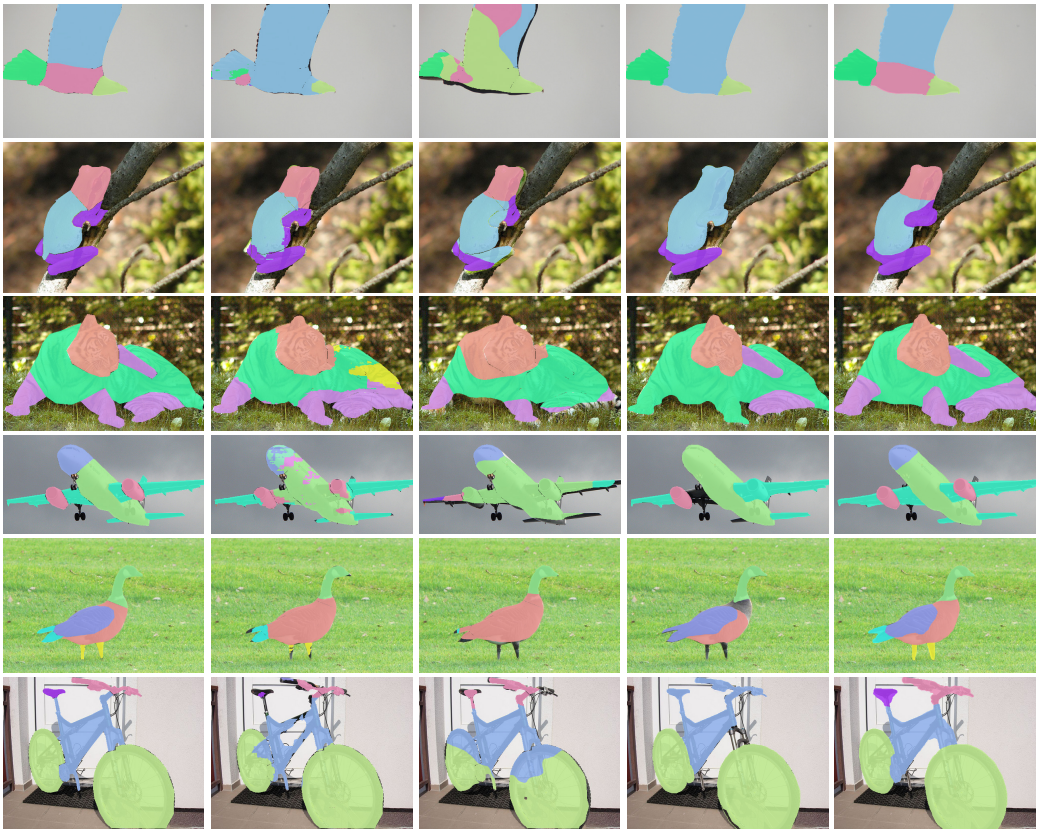


Figure 9: Qualitative results on PartImageNet [33].

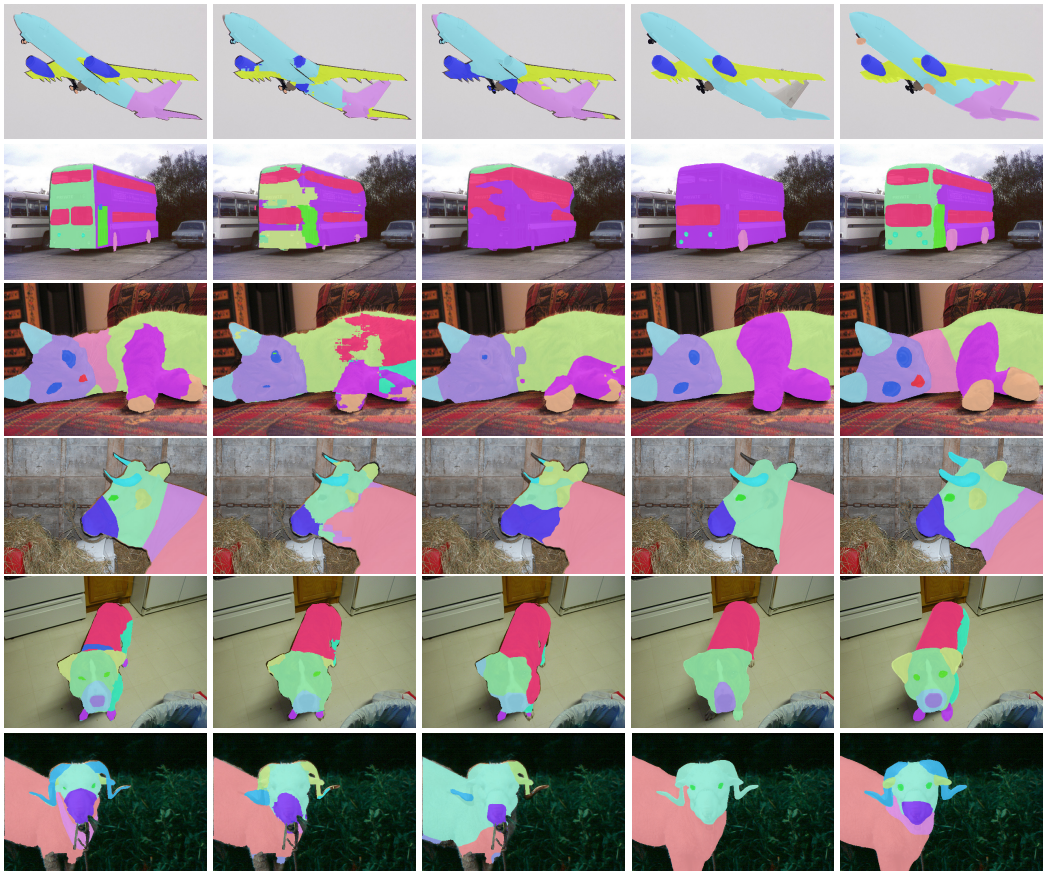


Figure 10: Qualitative results on Pascal-Part-116 [21].