LangHOPS: Language Grounded Hierarchical Open-Vocabulary Part Segmentation

Yang Miao
INSAIT, Sofia University
"St. Kliment Ohridski"

Jan-Nico Zaech
INSAIT, Sofia University
"St. Kliment Ohridski"

Xi Wang
INSAIT, Sofia University
"St. Kliment Ohridski"
ETH Zurich, TU Munich

Fabien Despinoy
Toyota Motor Europe

Danda Pani Paudel
INSAIT, Sofia University
"St. Kliment Ohridski"

Luc Van Gool
INSAIT, Sofia University
"St. Kliment Ohridski"

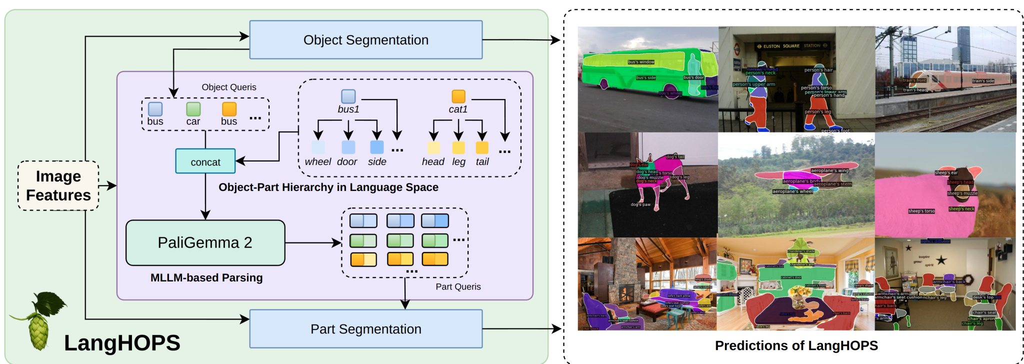

Figure 1: Given a 2D image and user queries of candidate object-part categories, our method LangHOPS grounds the hierarchy between objects and parts in language space and subsequently leverages a Multimodal Large Language Model to break down the segmented objects into parts.

Abstract

We propose LangHOPS, the first Multimodal Large Language Model (MLLM)-based framework for open-vocabulary object–part instance segmentation. Given an image, LangHOPS can jointly detect and segment hierarchical object and part instances from open-vocabulary candidate categories. Unlike prior approaches that rely on heuristic or learnable visual grouping, our approach grounds object–part hierarchies in language space. It integrates the MLLM into the object-part parsing pipeline to leverage its rich knowledge and reasoning capabilities, and link multi-granularity concepts within the hierarchies. We evaluate LangHOPS across multiple challenging scenarios, including in-domain and cross-dataset object-part instance segmentation, and zero-shot semantic segmentation. LangHOPS achieves state-of-the-art results, surpassing previous methods by **5.5%** Average Precision (AP) (in-domain) and **4.8%** (cross-dataset) on the PartImageNet dataset and by **2.5%** $mIOU$ on unseen object parts in ADE20K (zero-shot). Ablation studies further validate the effectiveness of the language-grounded hierarchy and MLLM-driven part query refinement strategy. The code will be released here.

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

1 Introduction

2D instance segmentation is a well-established computer vision research field and has experienced significant progress in object-level instance segmentation in the past decades [6, 17, 54, 64, 69]. While recent efforts have expanded toward higher-level reasoning from visual input [22, 54, 62], the growing demand for finer semantic understanding has led to increased interest in part-level segmentation [37, 50, 55, 58]. Unlike object-level segmentation, part-level understanding introduces new challenges, as it requires richer contextual awareness, reasoning about object-part relationships, and task-dependent interpretation. For example, a car can break down into coarse-grained components such as the body and wheels, or further delineated into finer-grained elements, including windows, doors, headlights, mirrors, or screws, depending on the downstream task and desired granularity.

Open-Vocabulary object-Part Instance Segmentation (OVPIS) emerges as a promising approach to address this challenge and has gained increasing interest in recent years. Unlike open-vocabulary object–part semantic segmentation [8, 9, 55], which assigns part labels to pixels without distinguishing between multiple part instances, object-part instance segmentation requires detecting and segmenting object and part instances separately. This introduces additional complexity, as the model must establish part–whole relationships at the instance level and maintain consistent grouping between objects and their constituent parts. In contrast to closed-vocabulary settings that rely on predefined object-part lists, open-vocabulary models aim to generalize to unseen part categories and novel compositions, which is a key capability for real-world generalization. Among existing works, SAM [20, 43] relies on handcrafted object-part and subpart heuristics for part-level segmentation. However, it does not offer control over the semantic granularity of the parts. Recent works [46, 63] extend SAM with a text prompt module to guide the segmentation process, but lack modeling of object–part hierarchies, which limits their ability to reason about relationships between objects and corresponding parts.

Moving beyond interactive or prompt-tuned variants of SAM, a separate line of work focuses on open-vocabulary part segmentation by leveraging vision–language models. OV-Parts [55], Part-CLIPSeg [9], and PartCATSeg [8] implement object-part hierarchical reasoning implicitly in CLIP embedding space [41] and enable zero-shot transfer to novel part categories. However, the performance of these methods is constrained by the limitations of CLIP in compositional and part-level understanding [1, 51, 55]. PartGLEE [24] explicitly models object-part structures using a Q-Former and performs joint object and part instance segmentation. Nevertheless, it has suboptimal segmentation performance in open-vocabulary scenarios since the Q-Former module lacks mechanisms to handle part granularity variations. Addressing this limitation is essential for improving generalization in real-world applications, where part granularity naturally varies across contexts and user intentions. For example, operating a laptop may require segmenting coarse parts such as the lid, while repairing it demands finer segmentation of detailed components such as screws or hinges.

In contrast, we propose LangHOPS, a novel framework that leverages language-grounded hierarchy and integrates MLLM for the task of OVPIS, as shown in Fig. 1. LangHOPS embeds object–part hierarchies directly in the language space, producing language-grounded part queries with object context. Those queries are further processed by a MLLM to link compositional object-part concepts and to generate adaptive segmentation queries. To verify the performance of LangHOPS, we conduct experiments in multiple settings (in-domain, cross-dataset and zero-shot) and on multiple dataset (PartImageNet, PascalPart-116 and ADE20K). As a result, LangHOPS significantly outperforms baselines by **5.5%** AP (in-domain) and **4.8%** AP (cross-dataset) on the PartImageNet dataset and by **2.5%** $mIOU_{\text{unseen}}$ on ADE20K (zero-shot). Experiment also shows the advanced scalability of LangHOPS with improvement by **10.0%** AP on PartImageNet when trained on more dataset, Ablation study shows that LangHOPS have object-part synergy that part-level instance segmentation can improve object segmentation by **5.4%** AP. In summary, our key contributions are:

- We propose LangHOPS, the first framework integrating an MLLM for the task of OVPIS.
- We propose language-space grounded object-part hierarchy modeling for part query representation and link the multi-granularity concepts with an MLLM to enable context-aware and accurate object-part parsing.
- We conduct experiments and demonstrate superior performance of LangHOPS in in-domain, cross-dataset, and zero-shot settings, as well as its scalability when on larger datasets. Notably, we show for the first time that part-level supervision can significantly enhance object-level segmentation.

2 Related Work

2D Object-Part Segmentation aims to jointly detect and segment both objects and their semantic parts, while preserving the hierarchical structure between them [10, 16, 67]. This task goes beyond traditional object-level understanding [2, 3, 12, 13, 26, 57, 65, 68] by introducing part-level granularity within object instances. This topic has gained attention [8, 11, 24, 25, 35, 39, 53] due to its potential in downstream applications such as image editing [19, 29] and robotics [5, 36]. TAPPS [10] extends Mask2Former [6] to predict jointly objects and parts with a set of shared queries. However, it is limited to a fixed set of predefined categories. PartCLIPSeg [9] applies a two-stage strategy for part-level semantic segmentation by first extracting mask proposals and then applying CLIP [41] to classify the masked image crops. Nevertheless, CLIP-based approaches such as PartCLIPSeg [9] and OV-Part [55] often exhibit suboptimal performance in fine-grained part segmentation, largely due to CLIP's limited capacity for compositional reasoning and explicit modeling of object–part hierarchies [1, 51, 55]. More recently, PartCATSeg [8] introduces a cost aggregation framework with a compositional loss and DINO-based structural guidance to enhance part-level image–text alignment and structural understanding. However, this method still lacks an explicit, language-grounded mechanism for representing hierarchical object–part relationships, which is essential for robust compositional generalization. Separately, PartGLEE [24] adopts a different two-stage pipeline that first segments object instances and then parses object queries into parts using a Q-Former. Since the Q-Former in PartGLEE [24] is not explicitly aware of part granularity during training or inference, it struggles to adapt across datasets with differing levels of annotation detail. For instance, a model trained on fine-grained parts such as "eye," "nose," and "ear" for cats in Pascal-Part-116 performs poorly on PartImageNet, where the same category is annotated only with coarser parts ("head," "body," "foot," and "tail"). Although [8, 24] incorporate object-level context into part segmentation, they do not entirely leverage the hierarchical relationships between objects and parts from candidate category definitions, consequently limiting their overall performance. In contrast, LangHOPS explicitly embeds the object-part hierarchies in language space to guide the MLLM for object-part parsing, as detailed in Sec. 3.4.

Open-Vocabulary Segmentation requires models to detect and segment object parts from novel categories guided by free-form text descriptions, without relying on category-specific training data. Early works, such as MaskCLIP [12] and GroupViT [57], initiated this paradigm by using Vision Language Models (VLMs) to transfer knowledge from text supervision to pixel-level tasks. Follow-up methods [26, 66] further enhance this capability by introducing text embeddings into mask prediction, contrastive learning, or region-level alignment. These approaches demonstrate the potential of using language as a flexible and scalable supervision signal for segmentation tasks. However, most of the existing works [12, 26, 30, 38, 44, 45, 57, 66] focus on object-level semantics, consequently lacking fine-grained part-level reasoning. OV-Part [55] and VLPart [50] establish benchmarks for open-vocabulary part segmentation by augmenting existing datasets with part-level annotations [4, 16, 42, 67]. Although recent methods [8, 24, 50, 55] make progress towards an open-vocabulary setting, they still exhibit limited generalization, particularly in zero-shot and cross-dataset scenarios where both the label space and data distribution differ from those seen during training. LangHOPS leverages MLLMs and object-part hierarchies to improve generalization and accuracy in the OVPIS task, setting a new benchmark in the open-vocabulary zero-shot and cross-dataset settings.

MLLM-based Image Segmentation integrates multimodal language models into image segmentation tasks, unlocking strong performance in various domains such as open-vocabulary panoptic segmentation, referring segmentation, interactive segmentation, and reasoning-based segmentation [22, 54, 62, 64]. LISA [22] introduces "reasoning segmentation", allowing MLLMs to generate the mask token in response to complex and implicit textual queries. PSALM [64] extends LLMs with a vision encoder and a mask decoder with a flexible input prompt to handle diverse segmentation tasks. OMG-LLaVA [62] proposes an end-to-end MLLM-based framework capable of image-, object- and pixel-level understanding including pixel-level segmentation. While effective, these methods focus on object-level understanding and lack the ability to decompose objects into fine-grained semantic parts. Osprey [58] achieves part-level visual understanding but relies on off-the-shelf class-agnostic part masks (e.g. from SAM [20]) and cannot control over part granularity. More recently, CALICO [37] leverages MLLM for multi-image part-focused object comparison by identifying unique and common parts of certain object across images. In contrast, LangHOPS is the first framework to leverage MLLMs for open-vocabulary object-part instance segmentation, enabling fine-grained parsing at the instance level, beyond the semantic and multi-image settings explored in prior work.

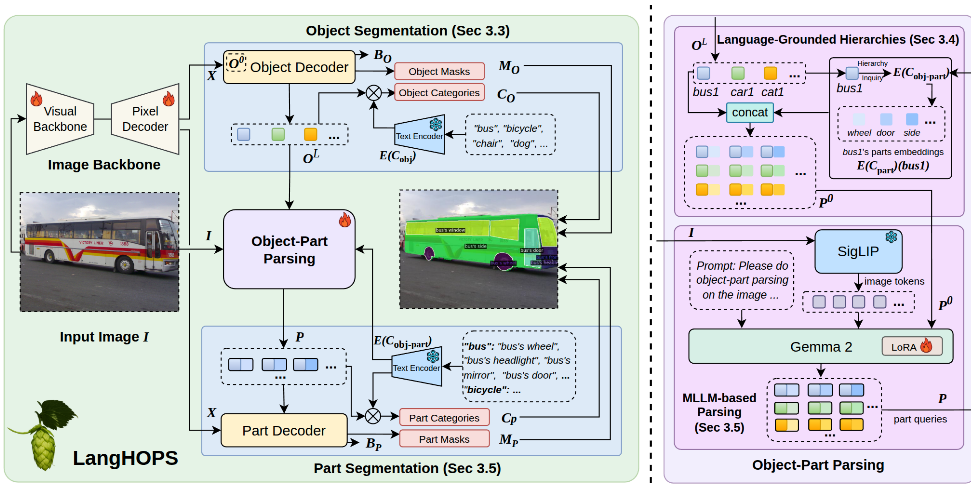

Figure 2: **LangHOPS framework.** The left block illustrates the overall architecture, with an image backbone, an object segmentation module, object-part parser and a part segmentation module. The right block illustrates the ideas on the object-part parser, consisting of a "Language-Grounded Hierarchies" module embedding the object-part hierarchy in language space, and a "MLLM-based Parsing" module producing the part queries for segmentation using a MLLM.

3 Method

3.1 Problem Definition

OVPIS aims to segment an image into distinct object-level instances and object-specific part-level instances, with the capability to generalize in novel object-part categories. Given an image and user-defined open-vocabulary object-part categories, the model outputs masks and categories of objects with their corresponding parts (e.g., “bus 1”, “bus 1’s headlight 1”, “bus 1’s headlight 2”, etc.). Note that, in contrast to the semantic part segmentation task proposed in [55], OVPIS also distinguishes between different instances of the same object category. For open-vocabulary segmentation, we adopt the commonly used settings in prior work [50] where the model takes images and ground-truth mask pairs, for one set of object and part categories $\mathbf{C}^{\text{train}}$ during training, and segment objects and parts of novel categories $\mathbf{C}^{\text{novel}}$ during inference.

3.2 Method Overview

Our model is illustrated in Fig. 2. Given an input RGB image $\mathbf{I}$ with a set of user-defined open-vocabulary candidate objects and part categories $\mathbf{C}$, the model outputs the masks and categories of the segmented instances of objects and parts, as well as the object-part hierarchies between the instances. Specifically, our framework in Fig. 2 is composed as follow:

Object Segmentation. We derive the initial object queries $\mathbf{O^0}$ with prior information from image features $\mathbf{X}$, and apply the object decoder utilized in [24] together with CLIP text encoder to obtain predictions of object-level categories $\mathbf{C_O}$, bounding boxes $\mathbf{B_O}$ and segmentation masks $\mathbf{M_O}$.
Language-grounded Hierarchies. We first extract hierarchies between objects and parts from the input candidate categories $\mathbf{C}$ and encode them in CLIP’s language space $\mathbf{E}(\mathbf{C})$. Subsequently, given predictions of objects’ categories, we construct initial part-level queries $\mathbf{P^0}$ by retrieving object-conditioned part embeddings from $\mathbf{E}(\mathbf{C})$, and concatenating them with object queries, enabling context-aware and granularity-adaptive object-part parsing.
MLLM-based Parsing. We leverage a MLLM to refine initial object-part-concatenated queries $\mathbf{P^0}$ by linking visual concepts with object-part-concatenated queries through structured prompt guidance, producing enriched part queries $\mathbf{P}$ that capture hierarchical part relationships across both language and visual domains for subsequent decoding.
Part segmentation. We use $\mathbf{P}$ and $\mathbf{X}$ as inputs to the part decoder with the same structure as the object decoder and predict categories $\mathbf{C}_P$, bounding boxes $\mathbf{B_P}$ and masks $\mathbf{M}_P$ of part instances.

By integrating the language-grounded hierarchies and MLLM into the object-part parsing framework, our object and part segmentation modules are tightly coupled. Joint information between parts and objects is utilized through the following information flow: For each segmented object, a set of part queries is constructed by a concatenation of the object embedding generated in the object decoder and the language-encoded part description. These queries are processed by the MLLM, generating queries capable of open-vocabulary part segmentation. The processed queries are used in the part decoder to segment each requested part. The details are provided in the following subsections.

3.3 Object Segmentation

Following PartGLEE [24], we apply the transformer decoder implemented in [23] as object decoder. Object queries $\mathbf{O^0} \in R^{N \times D_q}$ are initialized given priors from the multi-scale image features $\mathbf{X_s} \in R^{D_s \times \frac{H}{2^s} \times \frac{W}{2^s}}, s = \{2, 3, 4, 5\}$, with N as hyper-parameter denoting the number of object queries. Next, L layers of a deformable transformer decoder module [23] are applied for cross-attention computation between $\mathbf{X_s}$ and $\mathbf{O^i}$, as well as self-attention of $\mathbf{O^i}$, $i \in [1, L]$. The output queries $\mathbf{O}^L$ are utilized to perform object-level detection, classification and segmentation with 3 separate prediction head. For detection, a 3-layer MLP is utilized to map $\mathbf{O}^L$ to the coordinates of the bounding boxes $\mathbf{B^O} \in R^{N \times 4}$:

$$\mathbf{B^O} = MLP(\mathbf{O}^L). \tag{1}$$

For open-vocabulary object-level classification, we apply the CLIP text encoder to process the user-defined candidate object categories and obtain the object-level class embeddings:

$$\mathbf{E}(\mathbf{C}_{\text{obj}}) = CLIP_{\text{text}}(\mathbf{C}_{\text{obj}}) \tag{2}$$

Then the classification logits are calculated by:

$$\mathbf{S^O} = f_{CO}(\mathbf{O}^L) \cdot \mathbf{E}(\mathbf{C}_{\text{obj}}), \tag{3}$$

where f_{CO} is the linear layer mapping $\mathbf{O}^L$ to CLIP embedding space. By taking the maximum logits over the candidate categories, semantic categories predictions, $\mathbf{C^O}$, are obtained. For object segmentation, masks are generated by calculating the inner-product between $\mathbf{O}^L$ and the dense mask features $f_M(\mathbf{X_2})$ obtained with a 2D convolutional network f_M on the dense features $\mathbf{X_2}$:

$$\mathbf{M^O} = f_{MO}(\mathbf{O}^L) \cdot f_M(\mathbf{X_2}), \tag{4}$$

where f_{MO} is 3-layer MLP mapping queries into mask features's embedding space.

3.4 Language-grounded Hierarchies

Following object segmentation, the next objective is to decompose each segmented object into its corresponding part-level instances. This requires first modeling the relationship between objects and their constituent parts. PartGLEE [24] addresses this by introducing a set of learnable, universal parsing queries that, together with object queries, are processed by a Q-Former to generate a fixed number of part queries for each object. However, such Q-Former-based object-part parsing method has inherent limitations. First, it lacks of context awareness as it does not incorporate user-defined open-vocabulary categories $\mathbf{C}$ during object-part parsing. Consequently, the model may fail to generalize across domains where the definition of parts differs (e.g., coarse vs. fine-grained part sets). Second, the Q-Former-based method suffers from limited generalization from data priors. In fact, it has to be entirely trained and lacks external knowledge, which makes it highly dependent on the distribution and coverage of the training data.

To effectively address these issues, we explicitly model the hierarchical object-part structure from $\mathbf{C}$ in the well-generalizable language space. Specifically, given one object $\mathbf{O}^L \in R^{1 \times D_q}$ and its predicted object category C_o from Sec. 3.3, we query its potential part categories using $\mathbf{C}$: $\mathbf{C}_{\text{part}}$. For example, if an object with the query $\mathbf{O}^L$ is classified as a "bus", then we retrieve all the parts belonging to "bus" from $\mathbf{C}$ ("bus's wheel", "bus's window", "bus's door", etc). Subsequently, we encode the retrieved part categories into the CLIP text embedding space:

$$\mathbf{E}(\mathbf{C}_{\text{part}}(C_o)) = \{CLIP_{\text{text}}(C_o^p)\}, C_o^p \in \mathbf{C}_{\text{part}}(C_o) \tag{5}$$

where $\mathbf{C}_{\text{part}}(C_o)$ represents part categories belonging to a corresponding object category C_o. Ultimately, the embedding of each candidate part is concatenated with the corresponding object query separately as the initial part queries, with both object-level context and part-level language priors:

$$\mathbf{p_i^0} = (\ \mathbf{O}^L \parallel f_{CO}(\mathbf{e_o^i})\ ), \mathbf{e_o^i} \in \mathbf{E}(\mathbf{C}_{\text{part}}(C_o)), \tag{6}$$

where $(\cdot\|\cdot)$ represents the concatenation of two tensors. The query $\mathbf{p_i^0}$ is beyond pure text embeddings. Instead, it incorporates both visual information and language semantics for open-vocabulary classification and segmentation tasks. Each initialized part query $\mathbf{p_i^0}$ is repeated N_p times to accommodate multiple instances of the same part category within a single object (e.g., a bus having four wheels). Consequently, the part queries of the same part category and the same object are identical, e.g., $\mathbf{p^0}$ of bus1's wheel1 and bus1's wheel2. On the other hand, the queries of the same category but different objects are different due to the distinct visual information from the objects, e.g., $\mathbf{p^0}$ of bus1's wheel1 and bus2's wheel1. The initialized query is further refined by a MLLM to link the visual and text information between the object and its corresponding part, as detailed in the following subsection.

3.5 MLLM-based Parsing

To parse the multi-granularity concepts embedded in $\mathbf{P^0} = \{\mathbf{p_i^0}\}$, we utilize PaliGemma 2 [49], a lightweight and state-of-the-art MLLM that takes the image I, the concatenated object-part queries $\mathbf{P^0}$, and prompt guidance as input and implements object-part parsing in our framework. From the prompt, the MLLM receives the object query $\mathbf{O}^L$ followed by part queries $\mathbf{P^0}$ and outputs refined part queries that integrate both object- and part-level information. This design enables the MLLM to leverage object-level context to infer part semantics, and also allows bidirectional information flow - from parts back to the object - during training (see Sec. 4.3 Object-Part Synergy). Subsequently, the image tokens from SigLIP [49] and the prompt with queries $\mathbf{P^0}$ are provided to Gemma 2 model in a structured text prompt as follows:

Please do object−part parsing on the image <img><img_tokens></img>.

For each object, you will be given a list of object−part queries:
<obj_part>part_query1, part_query 2, ..., part_query n</obj_part>,
please implement object−part parsing by refine the queries so that it can be used
for later part category and mask prediction.

These are all the candidate object−part queries:
object 1 with parts <obj_part>part_query1, part_query 2, ...,
part_query n1</obj_part> ;
object 2 with parts <obj_part>part_query1, part_query 2, ...,
part_query n2</obj_part>;
...

This stage processes object-part queries jointly and outputs part queries $\mathbf{P}$ integrated with visual information and object context. Note we utilize Gemma 2 as a feed-forward model, instead of utilizing auto-regressive generation to ensure a controlled output structure. $\mathbf{P}$ is obtained from the last hidden states of the corresponding input part queries. $\mathbf{P}$ will be used as input for a separate part decoder with same structure as the object decoder introduced in Sec. 3.3.

3.6 Implementation Details

We employ a two-stage training strategy. In the first stage, we train the model with an object instance segmentation loss only:

$$L^1 = \lambda_{\text{cls}} \cdot L_{\text{cls}}^{\text{obj}} + \lambda_{\text{bbox}} \cdot L_{\text{bbox}}^{\text{obj}} + \lambda_{\text{mask}} \cdot L_{\text{mask}}^{\text{obj}}. \tag{7}$$

$L_{\text{cls}}^{\text{obj}}$ is the focal loss [27] on the prediction logits $\mathbf{S^O}$. $L_{\text{bbox}}^{\text{obj}}$ is the L1 loss on predicted object bounding boxes $\mathbf{B^O}$. $L_{\text{mask}}^{\text{obj}}$ is the combination of focal loss and dice loss [34] on the predicted object masks $\mathbf{M^O}$. In the second stage, joint object and part segmentation training is implemented with losses on both object and part predictions:

$$L^2 = \lambda_{\text{cls}} \cdot (L_{\text{cls}}^{\text{obj}} + L_{\text{cls}}^{\text{part}}) + \lambda_{\text{bbox}} \cdot (L_{\text{bbox}}^{\text{obj}} + L_{\text{bbox}}^{\text{part}}) + \lambda_{\text{mask}} \cdot (L_{\text{mask}}^{\text{obj}} + L_{\text{mask}}^{\text{part}}). \tag{8}$$

The loss functions on part segmentation are the same with the ones on object. The parameters of the Swin-L backbone and MaskDINO decoder are initialized with the pre-trained checkpoints from GLEE [18]. Following MaskDINO, the hyperparameters are set to $\lambda_{\text{cls}} = 4, \lambda_{\text{bbox}} = 2, \lambda_{\text{mask}} = 5, L = 9$. The number of repeated part queries $N_p = 3$. The training is conducted on 4 x H200 GPUs with a batch size of 16.

Method	PPS-116			+*INS*			+*INS*+*PART*			PartImageNet		
	obj	part	AP	obj	part	AP	obj	part	AP	obj	part	AP
VLPart	–	4.5	–	–	–	–	–	–	–	–	29.7	–
PSALM†	31.6	8.27	13.4	48.0	10.7 (+2.4)	18.9 (+5.5)	58.6	<u>11.6</u> (+3.3)	<u>21.9</u> (+8.5)	79.2	40.1	48.7
PartGLEE	<u>38.4</u>	**9.20**	<u>15.6</u>	<u>58.7</u>	<u>11.0</u> (+1.8)	<u>21.5</u> (+5.9)	<u>61.0</u>	9.57 (+0.4)	21.0 (+5.4)	<u>81.4</u>	<u>41.5</u>	<u>50.4</u>
LangHOPS	**44.5**	<u>8.86</u>	**16.7**	**60.5**	**11.4** (+2.5)	**22.3** (+5.6)	**62.8**	**16.4** (+7.5)	**26.7** (+10.)	**83.9**	**49.2**	**56.9**

Table 1: Cross-dataset experiment: **PascalPart-116** (training) → PartImageNet (evaluation) and in-domain experiment: PartImageNet (training) → PartImageNet (evaluation). We report object-level (obj), part-level (part), and overall (AP) mAP. The best result is in **bold** and the second best one is in <u>underline</u>. The notations "+*INS*" and "+*INS*+*PART*" indicate additional training dataset for scalability. Green values reflect relative AP gains over the PPS-116 baseline; Cyan values reflect relative mAP gains over the PPS-116 baseline. Gray columns shows in-domain performance.

Method	PartImgNet			+*INS*			+*INS*+*PART*			PPS-116		
	obj	part	AP	obj	part	AP	obj	part	AP	obj	part	AP
PSALM	<u>8.58</u>	1.87	2.89	20.0	<u>3.33</u> (+1.5)	5.87 (+3.0)	20.1	<u>3.58</u> (+1.7)	6.09 (+3.2)	48.7	13.2	18.6
PartGLEE	8.00	<u>2.18</u>	<u>3.06</u>	<u>23.1</u>	3.06 (+0.9)	<u>6.17</u> (+3.1)	<u>22.2</u>	3.37 (+1.2)	<u>6.33</u> (+3.3)	<u>53.2</u>	<u>14.5</u>	<u>20.4</u>
LangHOPS	**9.57**	**2.20**	**3.32**	**23.3**	**3.64** (+1.4)	**6.63** (+3.3)	**22.6**	**4.67** (+2.5)	**7.39** (+4.1)	**54.6**	**15.0**	**21.0**

Table 2: Cross-dataset experiment: **PartImageNet** (training) → PPS-116 (evaluation) and in-domain experiment: PPS-116 (training) → PPS-116 (evaluation).

4 Experiments

4.1 Cross-dataset Object-Part Instance Segmentation

We conduct experiments to evaluate the cross-dataset generalization performance of LangHOPS, as well as baseline methods for the object-part instance segmentation task.

Experiment Setup. We follow the setup proposed in VLPart [50] where each method is trained on one base dataset and evaluated on another unseen dataset, without finetuning. Two settings are implemented: Pascal-Part-116 [55] → PartImageNet [16] and PartImageNet → Pascal-Part-116 (i.e., the model is trained on Pascal-Part-116 and evaluated on PartImageNet, and vice versa). We further evaluate the scalability of LangHOPS by integrating two additional sets of datasets into training, including object-level datasets *INS* (consisting of COCO [28], VisualGenome [21] and LVIS [14], with object annotations) and part-level datasets (*PART* consisting of ADE20K [67], SA1B [20] and PACO [42], with object and part annotations). Note the granularity of the part-level annotations across the datasets within *PART* are different. The metric is mAP_{mask} on the evaluation set of PartImageNet and Pascal-Part-116 dataset.

Baseline methods. The existing methods for the OVPIS task include VLPart [50] and PartGLEE [24]. To extend the set of baselines for comparison, we further adapt PSALM [64], a state-of-the-art LLM-based 2D object-level segmentation method by extending the LLM mask tokens with learnable part queries for object-part parsing and part segmentation. The adapted PSALM is denoted as PSALM†.

Cross-dataset and in-domain evaluation on PartImageNet. As shown in Tab. 1, LangHOPS achieves the best performance of object-part instance segmentation in both cross-dataset and in-domain settings (i.e.,trained with Pascal-Part-116 and evaluated on PartImageNet). LangHOPS surpasses PartGLEE by 1.1% and PSALM† by 3.3% in mAP on object-part instance segmentation. Our experiments further show that LangHOPS has better scalability with additional training datasets containing part-level annotations. Trained on Pascal-Part-116+*INS*, all methods achieve similar performance gains in both part-level mAP and overall AP. However, when the training set is extended with additional part-level datasets (Pascal-Part-116 + *INS* + *PART*) our approach achieves a significant performance boost in both part-level mAP ($+7.5$) and overall AP ($+10.0$). In contrast, the performance gain of PartGLEE in part-level segmentation drops ($+5.9 \rightarrow +5.4$) compared to the Pascal-Part-116+*INS* setting, mainly due to lacking the object-part hierarchy context during part parsing phase, as illustrated in Sec. 3.4.

Cross-dataset and in-domain evaluation on Pascal-Part-116. As shown in Tab. 2, training the model on PartImageNet and implementing evaluation on Pascal-Part-116 is more challenging than the previous condition for all the evaluated methods. Indeed, the latter dateset contains multiple novel object categories and finer-granularity parts than the former. LangHOPS achieves the best

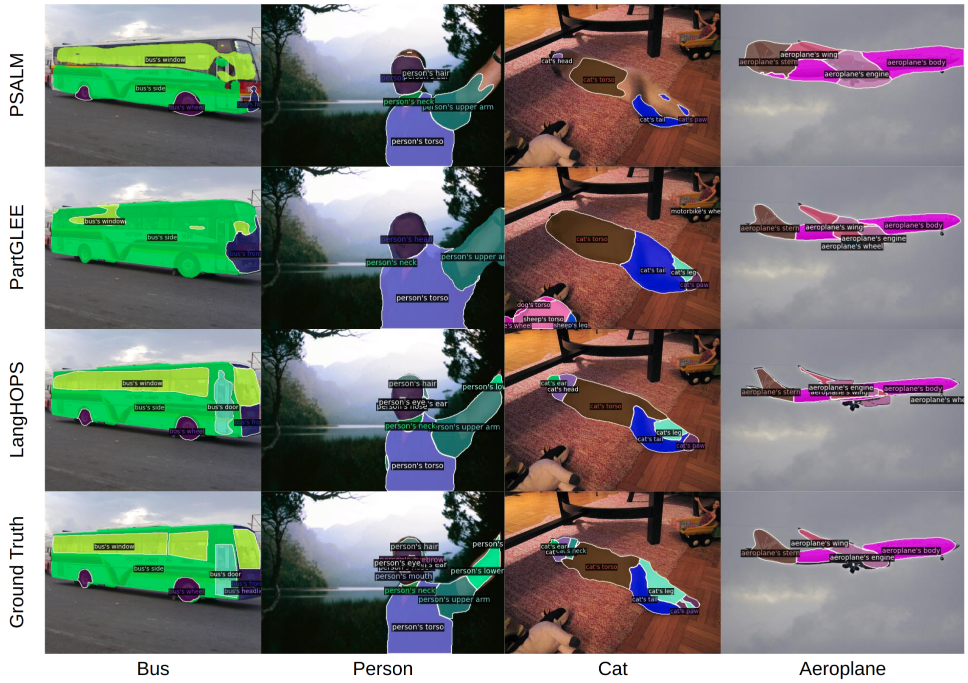

Figure 3: Qualitative results of part-level segmentation if LangHOPS and baselines.

performance in both cross-dataset and in-domain object-part segmentation on Pascal-Part-116. The experiments also shows the advantage of LangHOPS in scalability, especially in the setting with PartImageNet+ *INS* + *PART* as training dataset (+4.1 over +3.2 and +3.3) when trained only on PartImageNet dataset. Here, LangHOPS performs better than both baselines PartGLEE and PSALM†.

Qualitative Results are shown in Fig. 3 in the setting of PartImageNet + *INS* + *PART* $\rightarrow$ Pascal-Part-116. As the figure shows, LangHOPS achieves more accurate part segmentation than other baselines. Importantly, segmentation results on "person" and cat demonstrates LangHOPS's superior generalization performance to finer part granularity in the cross-dataset condition.

4.2 Zero-shot Part Segmentation

We further carry out experiments on the OV-Part benchmark [55] and PartImageNet dataset [16] for the zero-shot segmentation. One must note that this benchmark is evaluating open-vocabulary semantic segmentation of parts, which is not the core application of LangHOPS. The metric used is the harmonic mean of intersection-over-union (hIoU), for both seen and unseen categories [56]:

$$hIOU = \frac{2 \cdot mIoU_{\text{seen}} \cdot mIoU_{\text{unseen}}}{mIoU_{\text{seen}} + mIoU_{\text{unseen}}}. \tag{9}$$

As shown in Tab. 3, LangHOPS achieves the best performance on Pascal-Part-116 and PartImageNet datasets, and reaches second-best performance on ADE20K-234 dataset, achieving competitive performance with PartCATSeg [8]. LangHOPS obtains the highest $mIoU_{\text{seen}}$ on all three datasets, demonstrating the superior generalization ability to unseen object and part categories. Noticeably, our method is designed for open-vocabulary object-part instance segmentation while most others, including PartCATSeg [8], are designed specifically for the OV-Part benchmark (semantic part segmentation). Directly evaluating LangHOPS in the semantic segmentation still leads to superior performance (hIOU) in PPS-116 and PartImageNet datasets, showing its great potential.

4.3 Ablation Study

Ablation studies further demonstrate the effectiveness of LangHOPS.

Setting	Detached	Obj-Part Seg
obj	0.76	0.82
part	0.58	0.67

Table 5: Attention score.

Object-Part Synergy. To showcase the object-part synergy enabled by LangHOPS (i.e., a performance improvement from joint training of object and part instance segmentation) we reported the our

Method	PPS-116 [55]			PartImageNet [16]			ADE20K [55]		
	seen	unseen	$hIoU$	seen	unseen	$hIoU$	seen	unseen	$hIoU$
VLPart [50]	42.6	18.7	26.0	*	*	*	*	*	*
ZSSeg+ [31]	54.4	19.0	28.2	*	*	*	43.2	27.8	33.9
CLIPSeg [52, 55]	48.9	27.5	35.2	53.9	37.2	44.0	38.2	30.9	34.2
CAT-Seg [7, 55]	43.8	27.7	33.9	47.3	35.1	40.3	33.8	25.9	29.3
PartCLIPSeg [9]	50.0	31.7	38.8	56.3	51.7	53.9	38.4	38.8	38.6
PartGLEE [24]	57.4	27.4	37.1	*	*	*	<u>51.3</u>	35.3	41.8
PartCATSeg [8]	<u>57.5</u>	<u>44.9</u>	<u>50.4</u>	**73.8**	<u>71.5</u>	<u>72.7</u>	**53.1**	<u>47.2</u>	**50.0**
LangHOPS	**59.2**	**46.5**	**52.1**	<u>71.9</u>	**73.7**	**72.8**	49.3	**49.7**	<u>49.5</u>

Table 3: h-IoU. Zero-shot evaluation on PPS-116, PartImageNet and ADE20K.

Training setup	Obj Seg		Detached Obj-Part Seg		Obj-Part Seg	
Eval Dataset	Obj	Part	Obj	Part	Obj	Part
PPS-116	25.8	0.00	25.2	9.66	26.2	10.3
PartImageNet	67.9	2.08	62.9	13.2	68.3	14.9

Table 4: mAP of object and part instance segmentation. Ablations on Object-Part Synergy.

performances in following training setups: a) "Obj Seg": LangHOPS is trained only with the loss of object instance segmentation; b) "Detached Obj-Part Seg": LangHOPS is trained using losses of both object and part instance segmentation. However, the gradient flow coming from "MLLM-based parsing" module is interrupted, meaning that the gradients of $\mathbf{P}$ from the part segmentation loss will not directly propagate to object queries $\mathbf{O}^L$. One can note that object and part segmentation will still affect each other indirectly since both tasks use the same dense image features $\mathbf{X}$. c) "Obj-Part Seg": This setup allows a joint training of object and part instance segmentation without gradient flow cut. As show in Tab. 4, in "Obj Seg" setting, the mAP of part segmentation performance is near 0, as the loss of part segmentation is not used. Compared to "Obj Seg", the performance of object segmentation of "Detached Obj-Part Seg" drops 0.6% on PascalPart116 dataset and more significantly on PartImageNet dataset by 5.0%, due to the absence of the gradient flow by the MLLM-based parsing. In contrast, LangHOPS shows improved object segmentation performance in "Obj-Part Seg" than "Obj Seg", and gains significant boost in both object (by $\mathbf{5.4\%}$) and part segmentation (by $\mathbf{1.7\%}$) over "Detached Obj-Part Seg". This demonstrates that the proposed MLLM-based object-part parsing enables beneficial synergy effect in both cross-dataset and in-dataset conditions. We further investigate the object-part synergy mechanism by reporting the average attention score. The average attention score is calculated by summing attention scores of true positive predictions inside the ground truth masks M, divided by the area of the masks. The attention is the normalized cos similarity between object queries and the dense features of the final layer of the object/part decoder.

$$S_a = \sum_{u \in M} \frac{1 + cos(f_u, p_M)}{2 \cdot |M|}, \tag{10}$$

where u is the pixel within the ground truth mask M, $(f_u$ is the mapped feature for segmentation and p_M is the refined object/part query of the predicted instance matched to the ground truth instance. The score shows the amount of attention correctly assigned by the model to the ground truth area, and is in the range of [0, 1]. In the setting of PPS116+INS+PART -> PartImageNet, as shown in Tab. 5, compared to the "detached object-part seg.", the synergized object-part segmentation leads to higher attention scores for both object and part segmentation, proving strong evidence of the synergy between both segmentation tasks.

Effect of MLLM-based Parsing. We implement an ablation study to demonstrate the effectiveness of the MLLM-based Object-Part Parsing module by replacing it with a Q-Former. The Q-Former takes the object queries $\mathbf{O}$ as key and value, and hierarchical part queries $\mathbf{P^0}$ as query. In the end, the Q-former-based module outputs part queries $\mathbf{P^Q}$ for part segmentation purposes. As shown in Tab. 7, the ablated version, denoted as "w/o MLLM" shows inferior performances with both PartImageNet and PPS116 datasets, demonstrating the effectiveness of the MLLM module in object-part parsing.

Ablation on two-stage. We also provide an ablation study on the training strategy of the model. Two-stage refers to firstly training the model on object segmentation and secondly training it on object-part segmentation. One-stage means we directly train the model on object-part segmentation

Method	PPS-116			+*INS*			+*INS+PART*			PartImageNet		
	obj	part	AP	obj	part	AP	obj	part	AP	obj	part	AP
One-Stage	40.6	8.50	15.6	57.8	10.6	21.1	60.2	15.5	25.4	84.6	51.2	58.6
Two-Stage	44.5	8.86	16.7	60.5	11.4	22.3	62.8	16.4	26.7	83.9	49.2	56.9

Table 6: Ablations on training strategy in the cross-dataset setting of **PascalPart-116** (training) → PartImageNet (evaluation).

from scratch. Tab. 6 shows the model trained with the two-stage strategy achieves better cross-dataset performance, though its in-domain performance is inferior compared to one-stage.

Effect of Language-grounded Hierarchies. To investigate the effectiveness of the language-space-aligned object-part hierarchies, we conduct an ablation study by replacing the representation proposed in Sec. 3.5 with N learnable queries, denoted as "w/o hierarchy" in Tab. 7. Specifically, we initialize N learnable queries and concatenate them to each object query $\mathbf{o}^L$ from $\mathbf{O}^L$ to form the initial part queries $\mathbf{P^N}$. Subsequently, the MLLM uses the $\mathbf{O}^L$, $\mathbf{P^N}$ and as input, and outputs parsed part queries that are forwarded to the part decoder for final part segmentation. As shown in Tab. 7, by leveraging hierarchies between object and parts, and formulating part queries within language space, LangHOPS achieves better performance than the ablated version with learnable initial queries in both datasets.

Module	PartImageNet	PPS116
w/o MLLM	23.2	18.4
w/o hierarchy	22.5	19.1
LangHOPS	26.7	19.8

Table 7: mAP on object-part instance segmentation in the cross-dataset setting. Ablations on architecture design.

4.4 Limitation and Future Work

As shown in the supplementary material (Section A.4), the computational cost of LangHOPS is nontrivial compared to the baselines, primarily due to the integration of the MLLM for object–part parsing. Improving efficiency is essential for deploying LangHOPS in real-time or on-board computer vision and robotics applications. In addition, the training datasets [4, 14, 16, 21, 28] used in this work mainly contain common object and part categories, which may not fully cover all potential application scenarios. Therefore, additional datasets with task-specific annotations may still be required for fine-tuning in specialized cases (e.g., interactable articulated objects for robotic manipulation), even though LangHOPS demonstrates strong generalization capabilities compared to existing baselines. Additionally, as 2D-to-3D lifting [32, 40, 48, 59] is increasingly popular, leveraging LangHOPS for 3D computer vision tasks [15, 33, 47, 60, 61] is also a promising future direction.

5 Conclusion

We propose a new method LangHOPS that performs Open-vocabulary Part Instance Segmentation through hierarchical modeling in language space. Using language-grounded hierarchies improves both the context awareness and the accuracy of object-part parsing. In experiments, we show that LangHOPS performs notably better than existing state-of-the-art methods across multiple benchmark settings. Notably, our method achieves significant improvements in in-domain and cross-dataset object-part instance segmentation, where we outperform existing state-of-the-art approaches by **5.5%** AP. LangHOPS further achieves the best $mIOU$ on unseen object-parts in OVPIS tasks, on all PartImageNet, PascalPart-116 and ADE20K datasets, consequently demonstrating strong generalization ability in unseen object and part categories. In conclusion, LangHOPS establishes a novel foundation for Open-vocabulary Part Segmentation and highlights the potential of MLLM-based methods for fine-grained visual understanding, with the aim of encouraging further research into scalable language-driven approaches for structured scene parsing.

6 Acknowledgement

This research was funded by Toyota Motor Europe and the Ministry of Education and Science of Bulgaria (support for INSAIT, part of the Bulgarian National Roadmap for Research Infrastructure).

References

[1] Ethan Baron, Idan Tankel, Peter Tu, and Guy Ben-Yosef. Real classification by description: Extending clip's limits of part attributes recognition, 2024.

[2] Yihong Cao, Jiaming Zhang, Xu Zheng, Hao Shi, Kunyu Peng, Hang Liu, Kailun Yang, and Hui Zhang. Unlocking constraints: Source-free occlusion-aware seamless segmentation, 2025.

[3] Jialei Chen, Xu Zheng, Dongyue Li, Chong Yi, Seigo Ito, Danda Pani Paudel, Luc Van Gool, Hiroshi Murase, and Daisuke Deguchi. Split matching for inductive zero-shot semantic segmentation. *British Machine Vision Conference (BMVC)*, 2025.

[4] 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 *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[5] Zoey Chen, Aaron Walsman, Marius Memmel, Kaichun Mo, Alex Fang, Karthikeya Vemuri, Alan Wu, Dieter Fox, and Abhishek Gupta. Urdformer: A pipeline for constructing articulated simulation environments from real-world images, 2024.

[6] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[7] Seokju Cho, Heeseong Shin, Sunghwan Hong, Anurag Arnab, Paul Hongsuck Seo, and Seungryong Kim. Cat-seg: Cost aggregation for open-vocabulary semantic segmentation, 2024.

[8] Jiho Choi, Seonho Lee, Minhyun Lee, Seungho Lee, and Hyunjung Shim. Fine-grained image-text correspondence with cost aggregation for open-vocabulary part segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[9] Jiho Choi, Seonho Lee, Seungho Lee, Minhyun Lee, and Hyunjung Shim. Understanding multi-granularity for open-vocabulary part segmentation, 2024.

[10] Daan de Geus, Panagiotis Meletis, Chenyang Lu, Xiaoxiao Wen, and Gijs Dubbelman. Part-aware panoptic segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[11] Daan de Geus, Panagiotis Meletis, Chenyang Lu, Xiaoxiao Wen, and Gijs Dubbelman. Task-aligned part-aware panoptic segmentation through joint object-part representations. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[12] Zheng Ding, Jieke Wang, and Zhuowen Tu. Open-vocabulary universal image segmentation with maskclip. *arXiv preprint arXiv:2208.08984*, 2022.

[13] Yuqian Fu, Runze Wang, Yanwei Fu, Danda Pani Paudel, Xuanjing Huang, and Luc Van Gool. Objectrelator: Enabling cross-view object relation understanding in ego-centric and exo-centric videos. *International Conference on Computer Vision (ICCV)*, 2025.

[14] Agrim Gupta, Piotr Dollár, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[15] Anna-Maria Halacheva, Yang Miao, Jan-Nico Zaech, Xi Wang, Luc Van Gool, and Danda Pani Paudel. Articulate3d: Holistic understanding of 3d scenes as universal scene description, 2025.

[16] Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jie-Neng Chen, Shuai Liu, Cheng Yang, and Alan Yuille. Partimagenet: A large, high-quality dataset of parts. In *European Conference on Computer Vision (ECCV)*, 2022.

[17] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn for object detection and instance segmentation on keras and tensorflow, 2017.

[18] Wu Junfeng, Jiang Yi, Liu Qihao, Yuan Zehuan, Bai Xiang, and Bai Song. General object foundation model for images and videos at scale, 2024.

[19] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[20] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *International Conference on Computer Vision (ICCV)*, 2023.

[21] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations, 2017.

[22] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[23] Feng Li, Hao Zhang, Huaizhe xu, Shilong Liu, Lei Zhang, Lionel M. Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation, 2023.

[24] Junyi Li, Junfeng Wu, Weizhi Zhao, Song Bai, and Xiang Bai. Partglee: A foundation model for recognizing and parsing any objects. In *European Conference on Computer Vision (ECCV)*, 2024.

[25] Xiangtai Li, Shilin Xu, Yibo Yang, Haobo Yuan, Guangliang Cheng, Yunhai Tong, Zhouchen Lin, Ming-Hsuan Yang, and Dacheng Tao. Panoptic-partformer++: A unified and decoupled view for panoptic part segmentation. In *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2024.

[26] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2017.

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014.

[29] Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. Editgan: High-precision semantic image editing. *International Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[30] Qing Ma, Jiancheng Pan, and Cong Bai. Direction-oriented visual–semantic embedding model for remote sensing image–text retrieval. *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[31] Xu Mengde, Zhang Zheng, Wei Fangyun, Lin Yutong, Cao Yue, Hu Han, and Bai Xiang. A simple baseline for openvocabulary semantic segmentation with pre-trained visionlanguage model, 2022.

[32] Yang Miao, Iro Armeni, Marc Pollefeys, and Daniel Barath. Volumetric semantically consistent 3d panoptic mapping, 2024.

[33] Yang Miao, Francis Engelmann, Olga Vysotska, Federico Tombari, Marc Pollefeys, and Dániel Béla Baráth. Scenegraphloc: Cross-modal coarse visual localization on 3d scene graphs, 2024.

[34] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation, 2016.

[35] Shishir Muralidhara, Sravan Kumar Jagadeesh, René Schuster, and Didier Stricker. Jppf: Multi-task fusion for consistent panoptic-part segmentation. In *SN Computer Science*, 2024.

[36] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[37] Kiet A. Nguyen, Adheesh Juvekar, Tianjiao Yu, Muntasir Wahed, and Ismini Lourentzou. Calico: Part-focused semantic co-segmentation with large vision-language models. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[38] Jiancheng Pan, Qing Ma, and Cong Bai. A prior instruction representation framework for remote sensing image-text retrieval. In *Proceedings of the 31st ACM International Conference on Multimedia*, 2023.

[39] Tai-Yu Pan, Qing Liu, Wei-Lun Chao, and Brian Price. Towards open-world segmentation of parts. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[40] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[42] Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, Amir Mousavi, Yiwen Song, Abhimanyu Dubey, and Dhruv Mahajan. PACO: Parts and attributes of common objects. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[43] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[44] Bin Ren, Yawei Li, Jingyun Liang, Rakesh Ranjan, Mengyuan Liu, Rita Cucchiara, Luc V Gool, Ming-Hsuan Yang, and Nicu Sebe. Sharing key semantics in transformer makes efficient image restoration. *International Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[45] Bin Ren, Yahui Liu, Yue Song, Wei Bi, Rita Cucchiara, Nicu Sebe, and Wei Wang. Masked jigsaw puzzle: A versatile position embedding for vision transformers. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[46] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

[47] Erik Sandström, Ganlin Zhang, Keisuke Tateno, Michael Oechsle, Michael Niemeyer, Youmin Zhang, Manthan Patel, Luc Van Gool, Martin Oswald, and Federico Tombari. Splat-slam: Globally optimized rgb-only slam with 3d gaussians. In *International Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025.

[48] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder. Panoptic lifting for 3d scene understanding with neural fields. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[49] Andreas Steiner, André Susano Pinto, Michael Tschannen, Daniel Keysers, Xiao Wang, Yonatan Bitton, Alexey Gritsenko, Matthias Minderer, Anthony Sherbondy, Shangbang Long, Siyang Qin, Reeve Ingle, Emanuele Bugliarello, Sahar Kazemzadeh, Thomas Mesnard, Ibrahim Alabdulmohsin, Lucas Beyer, and Xiaohua Zhai. Paligemma 2: A family of versatile vlms for transfer, 2024.

[50] Peize Sun, Shoufa Chen, Chenchen Zhu, Fanyi Xiao, Ping Luo, Saining Xie, and Zhicheng Yan. Going denser with open-vocabulary part segmentation. In *International Conference on Computer Vision (ICCV)*,

2023.
[51] Yingtian Tang, Yutaro Yamada, Yoyo Zhang, and Ilker Yildirim. When are lemons purple? the concept association bias of vision-language models. In *Conference on Empirical Methods in Natural Language Processing*, 2023.
[52] Alexander Ecker Timo Luddecke. Image segmentation using text and image prompts, 2022.
[53] Xudong Wang, Shufan Li, Konstantinos Kallidromitis, Yusuke Kato, Kazuki Kozuka, and Trevor Darrell. Hierarchical open-vocabulary universal image segmentation, 2023.
[54] Cong Wei, Yujie Zhong, Haoxian Tan, Yong Liu, Zheng Zhao, Jie Hu, and Yujiu Yang. Hyperseg: Towards universal visual segmentation with large language model. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.
[55] Meng Wei, Xiaoyu Yue, Wenwei Zhang, Shu Kong, Xihui Liu, and Jiangmiao Pang. Ov-parts: Towards open-vocabulary part segmentation. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2023.
[56] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero- and few-label semantic segmentation, 2019.
[57] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
[58] Yuqian Yuan, Wentong Li, Jian Liu, Dongqi Tang, Xinjie Luo, Chi Qin, Lei Zhang, and Jianke Zhu. Osprey: Pixel understanding with visual instruction tuning, 2024.
[59] Deheng Zhang, Clara Fernandez-Labrador, and Christopher Schroers. Coarf: Controllable 3d artistic style transfer for radiance fields. In *International Conference on 3D Vision (3DV)*, 2024.
[60] Deheng Zhang, Jingyu Wang, Shaofei Wang, Marko Mihajlovic, Sergey Prokudin, Hendrik P.A. Lensch, and Siyu Tang. Rise-sdf: A relightable information-shared signed distance field for glossy object inverse rendering. In *International Conference on 3D Vision (3DV)*, 2025.
[61] Ganlin Zhang, Erik Sandström, Youmin Zhang, Manthan Patel, Luc Van Gool, and Martin R. Oswald. Glorie-slam: Globally optimized rgb-only implicit encoding point cloud slam, 2024.
[62] Tao Zhang, Xiangtai Li, Hao Fei, Haobo Yuan, Shengqiong Wu, Shunping Ji, Change Loy Chen, and Shuicheng Yan. Omg-llava: Bridging image-level, object-level, pixel-level reasoning and understanding. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2024.
[63] Yuxuan Zhang, Tianheng Cheng, Rui Hu, Lei Liu, Heng Liu, Longjin Ran, Xiaoxin Chen, Wenyu Liu, and Xinggang Wang. Evf-sam: Early vision-language fusion for text-prompted segment anything model. *arXiv preprint arXiv:2406.20076*, 2024.
[64] Zheng Zhang, Yeyao Ma, Enming Zhang, and Xiang Bai. Psalm: Pixelwise segmentation with large multi-modal model. In *European Conference on Computer Vision (ECCV)*, 2024.
[65] Ding Zhong, Xu Zheng, Chenfei Liao, Yuanhuiyi Lyu, Jialei Chen, Shengyang Wu, Linfeng Zhang, and Xuming Hu. Omnisam: Omnidirectional segment anything model for uda in panoramic semantic segmentation, 2025.
[66] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
[67] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
[68] Yuli Zhou, Guolei Sun, Yawei Li, Yuqian Fu, Luca Benini, and Ender Konukoglu. Camsam2: Segment anything accurately in camouflaged videos. *NeurIPS*, 2025.
[69] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once, 2023.

A Technical Appendices and Supplementary Material

This section provides additional visualization and ablation studies.

A.1 Visualization

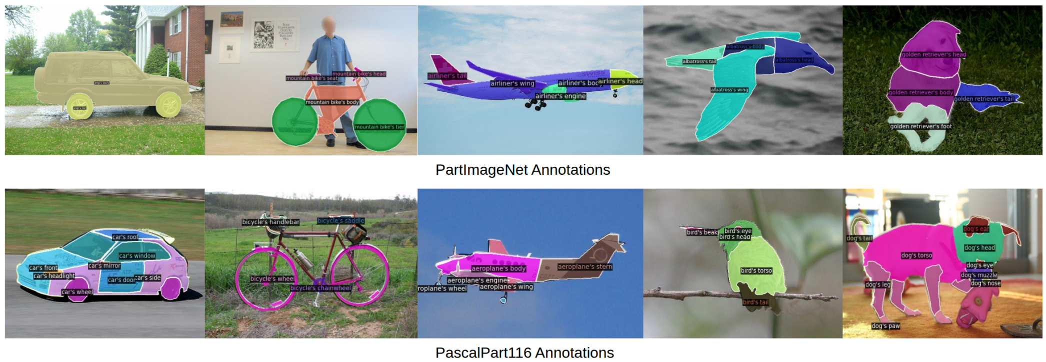

Figure 4: Visualization on Annotations of PartImageNet and PascalPart116 datasets.

Granularity Difference Across Dataset. In Fig. 4, we provide additional visualizations of the annotations of PartImageNet and PascalPart116 datasets. The figure shows that the two datasets provide annotations of parts in different granularity. Generally, PascalPart116 has finer part definition and thus it is more challenging to implement part segmentation on PascalPart116 than on PartImageNet, which explains that in both cross-dataset and in-domain settings, LangHOPS and baselines achieve less mAP on PascalPart116 than on PartImageNet.

Failure Cases. We further provide failure cases of LangHOPS in the cross-dataset setting. As shown in Fig. 5, LangHOPS can fail in several cases:

- when the object is distant to the camera and has small area in the image, LangHOPS may not be able to detect all the parts (one motorbike's wheel missing);
- in the cross-dataset setting, LangHOPS have difficulties in generalizing to some novel parts which it has not see during training (bird's eye, cat's eye). As shown in Fig. 4, the training dataset (PartImagenet) only contain annotations of animal's head and no annotation of eyes.
- when the training and evaluation dataset have different annotation styles, the trained model tends to predict the part segmentation in the style of training dataset (bicycle's wheel, all the pixels within the wheel circle).

A.2 Robustness Analysis

Statistical Robustness of Evaluation. We conduct repetitive experiments the same in Sec. 4.1 with 3 different random seeds. The average and standard deviation are calculated and reported in Tab. 8. The table shows the statistical stability of the cross-dataset evaluation and verifies the superiority of the proposed LangHOPS over the baseline.

Robustness to Prompt Formulation. We conducted two ablation studies on the ordering and wording of the structured input prompts to assess the robustness of our method to prompt formulation. **(a) robustness to prompt ordering**: We randomly shuffled (i) the order of object queries, and (ii) the order of part queries within each object, multiple times during inference. For instance, object 3 may appear before object 1, or part queries within an object may be permuted (e.g., "part 9, part 4, part 6"). As shown in Tab. 9, our method remains highly stable across these permutations, with minimal performance degradation, demonstrating robustness to input ordering. **(b) robustness to wording**: We further test the model's robustness to unseen part names by replacing the subset (from 0 to 100%) of the original part category names with GPT-4o-generated synonyms (e.g., "foot" $\rightarrow$ "leg").

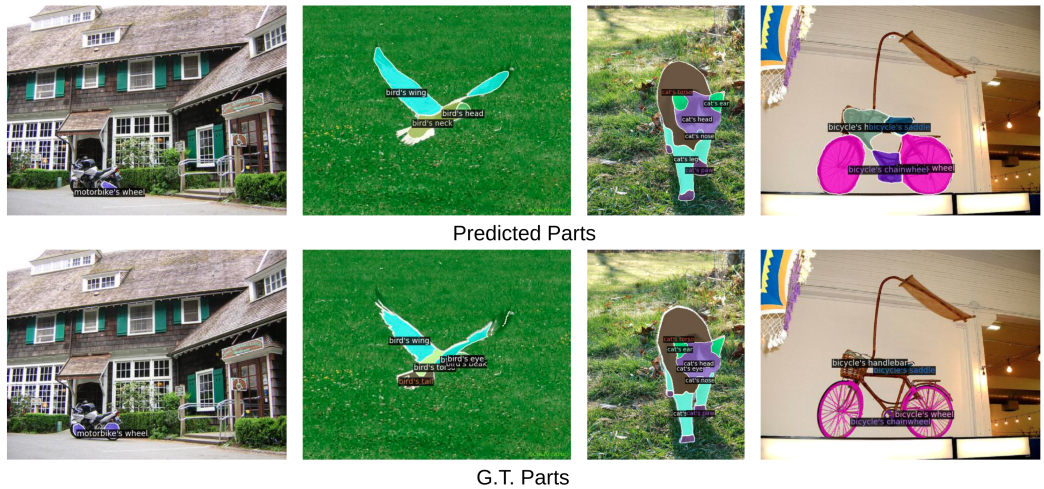

Figure 5: Failure cases of LangHOPS in the cross-dataset setting of **PartImageNet**+*INS*+*PART* (training) $\rightarrow$ PPS-116(evaluation).

Method	PPS-116			+*INS*			+*INS*+*PART*		
	obj	part	AP	obj	part	AP	obj	part	AP
PartGLEE	38.4±0.5	8.61±0.46	15.2±0.5	57.6±1.8	11.3±0.2	21.5±0.6	60.1±0.9	10.5±0.7	21.5±0.7
LangHOPS	48.7±3.3	8.89±0.24	17.7±0.9	60.9±0.7	12.1±1.0	22.9±1.0	63.7±1.4	16.6±0.3	27.0±0.5

(a) PPS-116 $\rightarrow$ PartImageNet

Method	PPS-116			+*INS*			+*INS*+*PART*		
	obj	part	AP	obj	part	AP	obj	part	AP
PartGLEE	8.53±0.52	2.05±0.09	3.04±0.16	23.3±0.1	3.10±0.16	6.17±0.16	22.5±0.4	3.60±0.24	6.46±0.26
LangHOPS	11.0±1.0	2.17±0.03	3.50±0.18	22.9±0.7	3.68±0.11	6.59±0.19	23.2±0.5	4.51±0.25	7.34±0.28

(b) PartImageNet $\rightarrow$ PPS-116

Table 8: Evaluation of PartGLEE and LangHOPS with mean ± standard deviation over 3 runs with different random seeds in the cross-dataset settings .

As shown in Tab. 10, LangHOPS significantly outperforms PartGLEE under increasing synonym replacement ratios, indicating strong generalization to semantically similar but unseen phrasing. Note that synonym substitutions may introduce granularity mismatches with the dataset's ground-truth annotations (e.g., "leg" may exclude "paw" in the ground truth for “foot”), which partially explains the observed performance drop.

Robustness to Noisy Hierarchy. We test on the common OVS setting using clean object-part hierarchies, but believe in the value of closing the gap towards a noisy real-world deployment. To evaluate the robustness of LangHOPS to noisy or automatically mined hierarchies, we replace a portion of the clean object-part taxonomy with GPT-4o-generated object-part hierarchies. These auto-mined hierarchies are constructed solely from the object category names and may introduce ambiguity, inconsistency, or irrelevant parts. In Tab. 11, we report performance under the varying noise hierarchies as the input prompt while the remaining the clean dataset annotations for evaluation. We observe that: LangHOPS consistently outperforms PartGLEE across all noise levels; LangHOPS degrades more gracefully as noise increases, maintaining reasonable AP even when the hierarchies are noisy; The performance gap widens especially at high noise levels, demonstrating LangHOPS's stronger resilience to imperfect or automatically mined hierarchies. Please note that the auto-generated hierarchies are often inconsistent with the ground truth annotations in the dataset, leading to lower evaluation metrics. Overall, developing evaluation protocols for adaptive, task-specific hierarchies remains an open problem and a promising direction for future benchmark design.

Method	PPS-116			+INS			+INS+PART		
	obj	part	AP	obj	part	AP	obj	part	AP
Shuffling of Object	47.8±2.7	8.57±0.36	17.1±0.9	61.1±0.8	11.7±0.8	22.6±0.8	65.1±0.9	15.8±0.3	26.7±0.4
Shuffling of Part	46.9±2.4	9.08±0.33	17.5±0.8	58.8±1.1	13.6±0.9	23.6±0.9	64.2±1.6	16.9±0.3	27.4±0.6
No shuffling	48.7±3.3	8.89±0.24	17.7±0.9	60.9±0.7	12.1±1.0	22.9±0.97	63.7±1.4	16.6±0.3	27.0±0.5

Table 9: Ablations on the ordering of the object and part queries – PPS116→PartImageNet.

Method	0%		25%		50%		75%		100%	
	part	AP	part	AP	part	AP	part	AP	part	AP
PartGLEE	11.2	21.8	9.3	20.3	8.6	19.7	6.6	18.2	5.1	17.0
LangHOPS	17.0	27.1	16.2	26.5	16.5	26.7	14.6	25.3	12.7	23.8

Table 10: Ablation on the robustness to input part category names. PPS116 + INS + PART → PartImageNet. Different percentages of part category names replaced with GPT-4o generated synonyms.

A.3 Additional Ablation study

Ablation on N_p. We further provide ablation study on the number of repeated part queries for each object N_p in the cross-dataset setting of **PPS-116**+*INS*+*PART* (training) → PartImageNet (evaluation). As shown in Tab. 12, the object-part segmentation performance drops when the N_p is too small (1, 2) or too large (4, 5, 6), .

Ablation on backbone finetuning. We further conduct an ablation study to show the necessity of finetuning the visual backbone and pixel decoder during training. As we can see in the Tab. 13, finetuning the visual backbone and pixel decoder leads to improved performance especially in the part segmentation task. This effect is mainly due to the fact that the used visual backbone and pixel [6, 23] are pretrained only on object-level tasks, and the extracted dense features lack part-level understanding. Thus, finetuning them on the object-part-level tasks is beneficial.

A.4 Computation Cost

We report the footprint of GPU hours, carbon cost, inference cost and model size of PartGLEE, PSALM and LangHOPS. The gpu hours and inference time are reported with Nvidia H200 GPU(s). The spec. power (700W) of H200 and world average carbon intensity of electricity (0.475 $kgCO_2/kWh$) are used for calculating the footprint. The Tab. 14 shows that LangHOPS has the largest model size, mainly due to the usage of MLLM (Paligemma2-3B). PSALM† has the longest training time and carbon footprint since it trains the LLM instead of using LoRA, and needs to process all candidate category names, which leads to long input prompts to the LLM. LangHOPS achieves the best performance with reasonable training and inference cost compared to the baselines.

Method	**0%**	-	**25%**	-	**50%**	-	**75%**	-	**100%**	-
	part	AP	part	AP	part	AP	part	AP	part	AP
PartGLEE	11.2	21.8	10.3	21.1	9.8	20.7	8.2	19.4	3.6	15.8
LangHOPS	17.0	27.1	13.1	24.1	12.4	23.5	8.8	20.7	6.7	19.1

Table 11: **Ablations on the noisy hierarchy construction.** Different percentages of obj–part hierarchies from the dataset are replaced with GPT-4o generated ones.

N_p	1	2	3	4	5	6
Obj AP	61.7	62.1	62.8	62.4	61.4	61.8
Part AP	15.4	15.8	16.4	16.0	16.1	15.9

Table 12: Ablation Study on N_p.

Method	PPS-116			+INS			+INS+PART			PartImageNet		
	obj	part	AP	obj	part	AP	obj	part	AP	obj	part	AP
Frozen Bk+Pd	48.2	6.99	16.1	64.1	8.85	21.1	66.1	9.34	22.0	80.6	30.1	41.3
Frozen Bk	47.6	7.36	16.3	63.0	6.98	19.4	63.4	12.4	23.8	83.2	34.7	45.4
LangHOPS	49.1	8.62	17.6	61.8	13.6	24.3	62.7	17.0	27.1	85.5	47.9	55.8

(a) (PPS-116 → PartImageNet.

Method	PartImageNet			+INS			+INS+PART			PPS-116		
	obj	part	AP	obj	part	AP	obj	part	AP	obj	part	AP
Frozen Bk+Pd	10.5	1.71	3.05	23.4	2.57	5.73	23.2	2.76	5.86	53.3	7.48	17.7
Frozen Bk	11.8	1.95	3.45	23.3	2.92	6.01	23.0	3.34	6.32	44.2	7.65	15.8
LangHOPS	11.3	2.17	3.47	21.9	3.82	6.55	23.8	4.16	7.13	56.4	15.3	21.4

(b) PartImageNet → PPS-116.

Table 13: Ablations on frozen image backbones and pixel decoder in the cross-dataset settings. "BK" refers to the visual encoder and "Pd" refers to the pixel decoder in the Fig. 2

Method	**Model Size**	**Training GPU Hours**	**Training Footprint (kg CO_2e)**	**Inference Time (ms)**
PSALM†	1.5B	92	30.6	628
PartGLEE	1B	40	13.3	240
LangHOPS	4B	72	23.9	396

Table 14: Computation Cost of LangHOPS and the baselines. PPS116 + INS + PART → PartImageNet.

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 abstract claims that the paper proposes a method which has key novelties in the model design and achieves state-of-the-part the performance. The introduction, method and experiment parts in the main paper clearly illustrate the model design, model's novelty and experiment implementation and demonstrate the contribution of the paper.

   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: It's in experiment section.

   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 paper does not include theoretical results

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 implementation details demonstrate the hyperparameters and training strategy. Supplementary material will provide further details due to the page limit.

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: The code will be released in Open Access, under some license.

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: It's in the method and experiment sections.

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 the 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: [Yes]

Justification:

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: We mention this in the implementation details.

   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 paper conform, in every respect, with the NeurIPS Code of Ethics

   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: [NA]

    Justification:

    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: no such risks

    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 models and data used in the paper are properly credited and the license and terms of use are properly respected.

    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: [Yes]

    Justification: They will be provided upon acceptance of the paper.

    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 paper does not involve crowdsourcing nor research with 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 paper does not involve crowdsourcing nor research with 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: [Yes]

Justification: The proposed model leverages a open-source MLLM as the component of the framework.

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.