

SAM3-I: Segment Anything with Instructions

Jingjing Li^{1,†}, Yue Feng^{2,†}, Yuchen Guo^{3,†}, Jincan Huang^{4,†}, Wei Ji^{5,2,‡,*},
Qi Bi⁶, Yongri Piao⁷, Miao Zhang⁷, Xiaoqi Zhao⁵,
Qiang Chen², Shihao Zou^{8,*}, Huchuan Lu⁷, Li Cheng¹

¹University of Alberta ²Tencent WeChat ³Northwestern University ⁴SUSTech ⁵Yale University

⁶Utrecht University ⁷Dalian University of Technology ⁸SIAT, Chinese Academy of Sciences

[†]Equal Contribution, [‡]Project Lead, *Corresponding Author

Abstract

Segment Anything Model 3 (SAM3) advances open-vocabulary segmentation through promptable concept segmentation, enabling users to segment all instances associated with a given concept using short noun-phrase (NP) prompts. While effective for concept-level grounding, real-world interactions often involve far richer natural-language instructions that combine attributes, relations, actions, states, or implicit reasoning. Currently, SAM3 relies on external multi-modal agents to convert complex instructions into NPs and conducts iterative mask filtering, leading to coarse representations and limited instance specificity. In this work, we present **SAM3-I**, an instruction-following extension of the SAM family that unifies concept-level grounding and instruction-level reasoning within a single segmentation framework. Built upon SAM3, SAM3-I introduces an instruction-aware cascaded adaptation mechanism with dedicated alignment losses that progressively aligns expressive instruction semantics with SAM3’s vision-language representations, enabling direct interpretation of natural-language instructions while preserving its strong concept recall ability. To enable instruction-following learning, we introduce **HMPL-Instruct**, a large-scale instruction-centric dataset that systematically covers hierarchical instruction semantics and diverse target granularities. Experiments demonstrate that SAM3-I achieves appealing performance across referring and reasoning-based segmentation, showing that SAM3 can be effectively extended to follow complex natural-language instructions without sacrificing its original concept-driven strengths. Code and dataset are available at <https://github.com/debby-0527/SAM3-I>.

1 Introduction

Semantic segmentation (Kirillov et al., 2023; Ravi et al., 2025; Wu et al., 2025) is a fundamental task

Contact Emails: jingjin1@ualberta.ca; wei.ji@yale.edu; {fengyue5717,ethan.chen1988}@gmail.com;sh.zou@siat.ac.cn.

in visual understanding. Traditional methods (Long et al., 2015; Ji et al., 2023) have primarily targeted explicit instance segmentation, where each pixel is assigned to one of a fixed set of predefined categories. However, such closed vocabularies remain considerably narrower than the diverse range of instances present in real-world scenarios. To this end, SAM3 (Carion et al., 2025) represents an important step forward by introducing promptable concept segmentation (**PCS**), a concept-driven paradigm that enables users to segment all related instances using short NP prompts. As shown in Fig. 1 b, a concept prompt such as “soccer player” can be used to segment all associated instances in an image.

However, real-world interactions are typically not conveyed through short NPs alone; users often describe instances using far richer expressions that may incorporate attributes, spatial relations, functions, actions, states, or even implicit reasoning over instances. Such descriptions often specify a concept together with multiple conditions, e.g., “*the soccer player on the far right, wearing a blue jersey, is sliding on the grass to tackle the ball*”, as shown in Fig. 1 c. Handling such inputs requires the model to parse structured semantics and accurately ground them in the scene. This capability is essential for many open-world applications, including home robotics (Quartey et al., 2025), autonomous driving (Feng et al., 2020), and augmented reality (Wahid et al., 2024), where systems must follow complex instructions to guide segmentation and decision-making.

This leads to a natural question: *Can we retain SAM3’s strong concept recall ability gained from large-scale training while enabling it to interpret more complex instructions and ground the corresponding instances?* To approximate this capability, SAM3 currently relies on external multi-modal agents, e.g., Qwen3-VL (Bai et al., 2025), to convert long instructions into short NPs and then perform multiple rounds of mask filtering process.



Figure 1: **Evolution of promptable segmentation in the SAM family.** (a) Promptable Visual Segmentation (PVS) in SAM1/2 relies on interactive visual prompts, with each prompt segmenting a single object. (b) Promptable Concept Segmentation (PCS) in SAM3 supports short NPs as prompts (e.g., “soccer player”) to segment all instances of a given concept. (c) To handle complex natural-language instructions, SAM3 relies on external multi-modal agents to convert instructions into short NPs, which remain overly coarse to capture fine-grained conditions (red icon). (d) Promptable Instruction Segmentation (PIS) in SAM3-I directly interprets complex instructions and grounds the intended target instance (green icon). SAM3-I introduces a unified framework, as the first of its kind in the SAM family, that integrates concept-level understanding and instruction-level reasoning.

Although workable, this agent-driven pipeline separates linguistic reasoning from visual segmentation, adds notable computational overhead, and limits the model’s ability to directly align complex instructions with visual concepts. More importantly, the PCS formulation constrains the agent’s final output to short NPs. These NP-level concepts are overly coarse and often insufficient for precisely grounding specific instances described by detailed instructions, as illustrated in Fig. 1 c and d.

To address these limitations, we propose **SAM3-I**, an instruction-following extension of the SAM family that unifies concept-level grounding and instruction-level reasoning within a single segmentation framework. Built upon SAM3, SAM3-I preserves its strong concept recall ability while enabling the direct interpretation of free-form natural-language instructions, ranging from simple referring expressions to complex reasoning-based queries. At its core, we introduce a dedicated instruction-aware cascaded adaptation mechanism with dedicated alignment losses that progressively aligns expressive instruction semantics with SAM3’s existing vision-language representations. This design injects instruction-following capability in a parameter-efficient manner, without disrupting SAM3’s original concept-driven strengths.

To support instruction-driven learning, we further construct **HMPL-Instruct**, a large-scale instruction-centric dataset derived from existing open-vocabulary segmentation benchmarks. HMPL-Instruct organizes natural-language instructions in a hierarchical semantic space, spanning concept-level prompts, explicit referring expressions, and complex reasoning-based instructions, while supporting diverse target granularities at both

object and part levels. Moreover, the dataset also covers a wide range of instruction-target cardinalities, including *one-to-one*, *one-to-all*, and *one-to-many* grounding scenarios, reflecting the diversity of real-world instructions. HMPL-Instruct is built via a human-in-the-loop pipeline that combines automatic instruction generation, agentic quality inspection, and targeted human correction, ensuring both scalability and precise visual grounding.

Empirical results show that SAM3-I achieves appealing performance on both referring and reasoning segmentation, bringing robust instruction-following capability to the SAM family. We have released SAM3-I together with the HMPL-Instruct dataset and our in-house annotation pipeline to facilitate future research on instruction-based segmentation. Finally, we highlight several promising directions for further exploration.

2 Preliminary: SAM Family

SAM represents a major milestone in segmentation foundation models, demonstrating strong performance in diverse downstream applications, such as medical imaging and industrial inspection (Wu et al., 2025; Peng et al., 2025). Its success is largely driven by the promptable segmentation paradigm introduced in SAM 1 & 2 (Kirillov et al., 2023; Ravi et al., 2025), where users provide visual prompts (points/boxes/masks) to segment a single target per prompt, a setting known as Promptable Visual Segmentation (PVS) (Fig. 1 a). Combined with large-scale, high-quality SA-1B and SA-V datasets, this paradigm forms the foundation of the SAM family’s remarkable segmentation capability.

SAM3 extends this paradigm to PCS, enabling segmentation of all instances of a given concept using short NPs as prompts (e.g., “soccer player”),

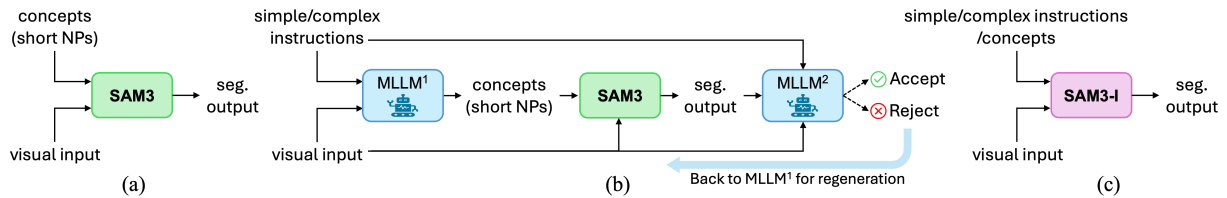


Figure 2: **Comparison of SAM3-series pipelines.** (a) SAM3 takes an image and a short NP as input and directly produces segmentation outputs. (b) SAM3 Agent handles longer instructions by prompting an MLLM¹ to interpret them into short NPs and using an MLLM² to validate and refine SAM3’s predictions, often through multiple rounds of mask filtering. (MLLM¹ and MLLM² denote two separate calls to the same MLLM.) Although this enables SAM3 to process long instructions, it introduces additional system complexity, computational overhead, and may collapse detailed instructions into over-simplified NPs. (c) SAM3-I removes the agent loop entirely and directly supports both concepts and instructions in a single pass. For clarity, visual prompts (mask/point/box) are omitted.

as shown in Fig. 2 a. While this advances open-vocabulary segmentation, SAM3 is fundamentally designed around atomic concept prompts, restricting text inputs to simple NPs rather than multi-condition referring expressions or reasoning instructions. When presented with complex instructions, SAM3 relies on an agentic pipeline (Fig. 2 b), where an external multi-modal large language model (MLLM) converts rich instructions into short NPs followed by iterative mask filtering. This multi-step process depends heavily on external reasoning and often oversimplifies natural-language descriptions, limiting precise instance-level grounding.

Real-world usage, however, involves rich, compositional instructions beyond simple NPs. This motivates us to extend SAM3 with the ability to follow complex instructions while preserving its strong concept-level grounding capability.

3 Promptable Instruction Segmentation

We extend the PCS paradigm to a more expressive instruction-following setting, which we term Promptable Instruction Segmentation (**PIS**). We formulate instructions into three progressively finer hierarchical levels:

- *i) Concept instruction* represents the most atomic form, expressed as short NPs compatible with SAM3.
- *ii) Simple instruction* retains explicit NP mentions while incorporating additional referring conditions such as attributes, spatial relations, or local context.
- *iii) Complex instruction* removes explicit NP references and instead requires the model to identify targets by interpreting functions, actions, affordance, or broader contextual semantics.

These three levels establish a hierarchy that enables SAM-I to unify concept-level understanding and instruction-level reasoning within the SAM family.

4 SAM3-I Model

Overview. As shown in Fig. 3, SAM3-I builds upon the SAM3 architecture and extends it to the PIS task, while remaining fully compatible with PCS and PVS. Briefly, SAM3 comprises a dual vision-language encoder and a detector-based segmentation head, along with a tracker and a memory module for video segmentation.

To equip SAM3 with the ability to interpret instructions from simple referring expressions to complex reasoning, we introduce an instruction-aware cascaded adaptation mechanism with distribution alignment losses, while keeping the SAM3 backbone frozen. This design preserves SAM3’s original capabilities, prevents catastrophic forgetting, and injects new instruction-following ability in a parameter-efficient manner.

Instruction-Aware Cascaded Adapter. Instruction understanding spans a wide range of linguistic complexity. Simple referring expressions rely on explicit NPs combined with additional conditions, which requires the model to detect explicit linguistic anchors and align them with visual evidence. Complex instructions demand richer linguistic competence, requiring the model to interpret functional descriptions, implicit references, or perform contextual reasoning. Mastering this spectrum is essential for robust instruction-following segmentation.

To address this challenge, we introduce an instruction-aware cascaded adapter, a hierarchical module inserted into each layer of SAM3’s text encoder. The design contains two adapters:

- *i) S-Adapter* learns attribute, position, and relation semantics, enabling effective grounding for simple referring expression where the target instance’s NP remains explicitly available.
- *ii) C-Adapter* builds upon S-Adapter and is responsible for handling complex instructions that omit explicit target NP mentions and require contextual reasoning.

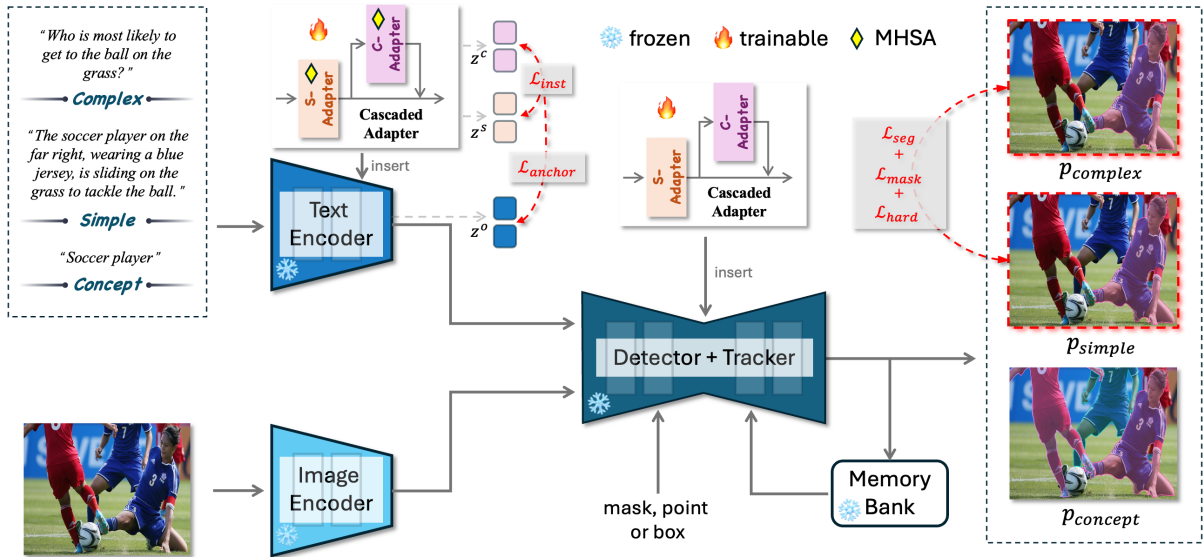


Figure 3: Overview of the proposed SAM3-I framework. (Sec. 4)

Both adapters employ a bottleneck structure (down-projection, GELU activation function, and up-projection) to enhance alignment with visual features, and a multi-head self-attention (MHSA) layer is incorporated to capture long-range textual dependencies. During training, S-Adapter is used exclusively for simple instructions, whereas complex instructions activate the full cascade so that C-Adapter can refine and enrich the representation produced by S-Adapter. This progressive design mirrors the natural hierarchy of linguistic difficulty, yielding structured linguistic embeddings that generalize across instruction forms. The cascaded adapter (without MHSA) is also integrated into SAM3’s detector to propagate instruction semantics into the segmentation stage.

Distribution Alignment Losses. Allowing the two branches to interpret instructions independently may lead to *semantic drift* between them. To maintain coherent grounding across different instruction levels, we introduce distribution alignment losses that regularize the two branches at both text embedding-level and output-level, beyond the segmentation loss \mathcal{L}_{seg} used in SAM3. The alignment losses comprise four components as detailed below.

Instruction Contrastive Loss. In SAM3-I, simple and complex instructions referring to the same target instance should yield consistent semantic representations, whereas instructions referring to different instances—even within the same concept category—should remain distinguishable. To achieve this, we introduce an instruction contrastive loss to regularize the adapter-enhanced text embeddings.

Specifically, for each training sample i , we treat the simple and complex text embeddings (z_i^s, z_i^c)

extracted by the text encoder as a positive pair, and treat all instructions from other samples within a mini-batch as negatives. The instruction contrastive loss is defined as:

$$\mathcal{L}_{\text{inst}} = \sum_i (\ell_i^{s \rightarrow c} + \ell_i^{c \rightarrow s}), \quad (1)$$

$$\ell_i^{s \rightarrow c} = -\log \frac{\exp(\langle z_i^s, z_i^c \rangle / \tau)}{\sum_k w_{ik} \exp(\langle z_i^s, z_k^c \rangle / \tau)}, \quad (2)$$

where the weighting factor w_{ik} is defined as

$$w_{ik} = \begin{cases} 1, & y_k \neq y_i, \\ \rho, & y_k = y_i, \end{cases} \quad (3)$$

with $\rho < 1$ down-weighting negative samples from the same concept category to stabilize training, and y denotes the concept category. $\langle \cdot, \cdot \rangle$ denotes cosine similarity. For clarity, we present only the formulation of $\ell_i^{s \rightarrow c}$ in Eq. (2), while $\ell_i^{c \rightarrow s}$ is defined symmetrically and omitted for brevity. This design encourages invariance across different instruction expressions of the same instance while retaining fine-grained discriminability among different instances.

Parent-rank Concept Anchoring Loss. While the instance-level contrastive loss effectively structures the instruction embedding space, it does not explicitly constrain instructions to remain within their semantic concept boundaries. Directly aligning instruction embeddings with concept embeddings as in $\mathcal{L}_{\text{inst}}$ is undesirable, as concept prompts often describe all instances of a category, whereas instructions may refer to subsets.

To this end, we introduce a parent-rank concept anchoring loss that enforces a relative semantic constraint: an instruction embedding should be more similar to its own concept than to concepts

from other categories. Specifically, for each instruction embedding z_i^{sc} (averaged from the simple and complex text embeddings), we impose a margin-based ranking loss against concept embeddings from other categories:

$$\mathcal{L}_{\text{anchor}} = \sum_i \max \left(0, m + \max_{j: y_j \neq y_i} \langle z_i^{sc}, z_j^o \rangle - \langle z_i^{sc}, z_i^o \rangle \right), \quad (4)$$

where z_i^o is the corresponding concept embedding and m is a margin hyper-parameter. Gradients are not propagated through the concept embeddings. This loss acts as a semantic guardrail, anchoring instruction embeddings to SAM3’s original concept-level representation while preserving the flexibility required for subset-level instruction understanding.

Mask Distribution Alignment Loss. To ensure that the simple- and complex-instruction branches produce compatible predictions for the same target, we align their mask distributions using a Kullback-Leibler (KL) divergence loss:

$$\mathcal{L}_{\text{mask}} = (\text{KL}(p_s \parallel p_c) + \text{KL}(p_c \parallel p_s))/2 \quad (5)$$

where p_s and p_c represent instruction-conditioned mask predictions for both simple and complex branches. This further encourages the two branches to share a consistent semantic space.

Uncertainty-Aware Hard-Region Supervision Loss. Discrepancies between the two branches often arise in semantically difficult regions, e.g., occlusions, relational cues, or context-dependent areas. To emphasize learning on these challenging pixels, we compute an uncertainty map $w_{\text{uncertainty}}$ based on the disagreement between the simple- and complex-instruction predictions, measured using the Jensen-Shannon Divergence (Lin, 1991). This uncertainty map is then used as an adaptive weight in an auxiliary cross-entropy (CE) loss between the mask predictions and ground-truth (GT):

$$\mathcal{L}_{\text{hard}} = w_{\text{uncertainty}} \cdot (\text{CE}(p_s, GT) + \text{CE}(p_c, GT)). \quad (6)$$

By guiding the model to focus on ambiguous or reasoning-intensive regions, this loss enhances robustness to complex linguistic instructions.

The overall training objective is $\mathcal{L}_{\text{train}} = \mathcal{L}_{\text{seg}} + \mathcal{L}_{\text{inst}} + \mathcal{L}_{\text{anchor}} + \mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{hard}}$.

Multi-Stage Training. SAM3-I is trained through a three-stage curriculum designed to ensure stable convergence and progressively acquire increasingly complex instruction-following capabilities.

In stage 1, the SAM3 backbone is frozen and the S-Adapter is trained on simple instructional data to establish category-, attribute-, and relation-

level grounding ability. In stage 2, the S-Adapter is frozen, and the C-Adapter is initialized from it and trained with complex instructions to enable the model with functional, and reasoning-level grounding ability. Finally, in stage 3, all adapters are jointly activated and fine-tuned using the proposed alignment objectives, harmonizing the two branches and ensuring consistent instruction-conditioned predictions.

5 HMPL-Instruct Dataset

In this section, we elaborate on the proposed *Hierarchical Multi-grained PACO-LVIS-Instruct (HMPL-Instruct)* dataset.

Motivation. As summarized in Tab. 1, instruction-guided segmentation benchmarks have progressively expanded from simple referring expressions to more implicit, reasoning-oriented instructions. Along this evolution, three key attributes have emerged as increasingly important for practical instruction-following segmentation. First, real-world instructions often refer to multiple targets within a scene, requiring models to identify a specific subset or all instances of visually similar objects, rather than a single isolated instance. Second, instruction semantics naturally exhibit a hierarchical structure, spanning concept-level NPs, explicit referring expressions, and more complex instructions that rely on relational or functional reasoning without explicitly naming the target. Third, effective instruction following frequently requires perception at different semantic granularities, ranging from object-level understanding to fine-grained part-level recognition. These observations motivate the design of *HMPL-Instruct* as a unified benchmark that jointly captures instruction hierarchy (concept, simple, and complex), multi-grained target cardinality (one-to-one, one-to-many, and one-to-all), and semantic granularity (object-level and part-level) within a single dataset.

Dataset Construction. To enable a unified formulation of hierarchical instructions and multi-grained targets, we develop a *human-in-the-loop data engine* that enriches the PACO-LVIS dataset (Ramanathan et al., 2023) with compositional natural-language instructions. The pipeline has four stages (detailed diagram is provided in Appendix B):

Stage 1: Automatic Instruction Annotation. Given an image with instance masks and class labels, we construct a visual prompt comprising the original image, mask-overlaid image, and cropped

Table 1: **Statistics of representative instruction-guided segmentation datasets.** The symbol [†] indicates datasets that explicitly support one-to-many scenarios, as highlighted in the corresponding papers.

Dataset	Instruction Level			Instruction Granularity			Obj.	Part	#Samples	#Masks	#Instruct.
	Concept	Simple	Complex	1-to-1	1-to-many [†]	1-to-all					
RefCOCO (Kazemzadeh et al., 2014)	✓	✓	✗	✓	✗	✗	✓	✗	19,994	50,000	142,209
RefCOCO+ (Kazemzadeh et al., 2014)	✓	✓	✗	✓	✗	✗	✓	✗	19,992	49,856	141,564
RefCOCOG (Yu et al., 2016)	✓	✓	✗	✓	✗	✗	✓	✗	25,799	49,822	95,010
gRefCOCO (Liu et al., 2023)	✓	✓	✗	✓	✓	✓	✓	✗	19,994	60,287	278,232
Ref-ZOM (Hu et al., 2023)	✓	✓	✗	✓	✓	✗	✓	✗	55,078	74,942	90,199
ReasonSeg (Lai et al., 2024)	✗	✗	✓	✓	✗	✓	✓	✓	1,218	2,058	5,078
MMR (Jang et al., 2025)	✗	✗	✓	✓	✗	✓	✓	✓	57,431	341,182	194,398
InstructPart (Wan et al., 2025)	✓	✓	✓	✓	✗	✗	✗	✓	2,400	2,400	14,400
HMPL-Instruct	✓	✓	✓	✓	✓	✓	✓	✓	44,109	347,120	849,792

masked regions. Conditioned on a system prompt and few-shot examples, an annotator MLLM generates diverse instruction candidates for each target instance, including simple instructions that explicitly mention the target NP and describe perceptual attributes, and complex instructions that omit the NP and rely on higher-level reasoning cues such as actions or functions. Both instruction types are produced in declarative and question formats. The annotator additionally synthesizes negative instructions by contradicting visual semantics, as well as a concise concept-level NP compatible with SAM3’s PCS-style prompting. In this way, each target instance is associated with four positive instructions, four negative instructions, and one concept NP.

Stage 2: Agentic Quality Inspection. All generated instructions are verified by a second MLLM via a multiple-choice consistency check with the visual content and target mask. An instruction set is accepted only if the inspector achieves 100% accuracy across all candidates; otherwise, it is rejected and returned to Stage 1 for regeneration, iteratively filtering semantic inconsistencies.

Stage 3: Human-in-the-Loop Correction. Samples that repeatedly fail automatic verification due to ambiguous visuals or subtle linguistic cues are reviewed by three human annotators, who collaboratively revise or discard problematic instructions to ensure accurate and precise grounding. Through the first three stages, we obtain high-quality instruction-mask pairs that primarily support *one-to-one* and *one-to-all* grounding.

Stage 4: Human Annotation for One-to-Many Setting. The *one-to-many* setting, which requires selecting a subset of visually similar instances, remains challenging for automatic MLLM-based annotation, as models often fail to consistently group subsets or generate unambiguous instructions. We therefore adopt a fully human-driven process using a dedicated web-based interface, allowing annotators to select instance subsets and associate them with corresponding simple and complex in-

structions. Ten annotators create initial annotations, and four inspectors refine them to ensure semantic correctness and compliance with predefined rules, such as excluding target NPs in complex instructions. Ambiguous cases are removed. Finally, the instructions are lightly polished using an MLLM-based rewriting step to improve fluency while preserving the original semantics. Using this process, we collect 2,941 one-to-many samples with 11,764 paired instructions and 7,449 masks, averaging 2.53 masks per image.

In summary, HMPL-Instruct comprises 133,960 object-level masks and 213,160 part-level masks across 44,109 images, paired with 849,792 positive instructions. The average instruction length is 15 words. We follow the original PACO-LVIS benchmark for the train/validation/test splits. Representative examples are provided in Appendix Fig. 7.

6 Experiments

6.1 Evaluation Metrics

We evaluate our approach using $gIoU$ and $P@50$. $gIoU$ (Lai et al., 2024) is computed as the average Intersection-over-Union (IoU) over all test images. $P@50$ (Hu et al., 2016) reports the proportion of test samples whose predicted mask achieves an IoU of at least 0.5 with the ground-truth mask.

6.2 Quantitative Results

Comparison with SAM3 and SAM3 Agent. We first assess the instruction-following ability of SAM3-I compared with the original SAM3 and SAM3 with MLLM agents on HMPL-Instruct. As shown in Tab. 2, SAM3 is limited to short NP concept prompts and only supports concept-level segmentation, while the agent-based variant enables instruction processing at the cost of multi-pass inference and substantially increased model size. In contrast, SAM3-I directly supports concept, simple, and complex instructions in a single-pass pipeline. Importantly, SAM3-I preserves SAM3’s concept-level performance (29.5 $gIoU$ / 31.9 $P@50$) since we explicitly keep the SAM3 backbone frozen, while

Table 2: **Comparison of SAM3-I with SAM3 and SAM3 Agent baselines.** “**” indicates that we retain naive SAM3’s concept-level scores, and “-” denotes settings not applicable to the corresponding method.

Model	Parameter	Inference	Concept-level		Simple Instruct.		Complex Instruct.	
			gIoU	P@50	gIoU	P@50	gIoU	P@50
SAM3 (Carion et al., 2025)	0.8B	single	29.5	31.9	-	-	-	-
SAM3 Agent (Qwen3-VL-8B)	8.8B	multi	-	-	28.4	28.9	26.4	27.4
SAM3-I	1.1B	single	29.5*	31.9*	59.7	65.3	49.0	51.0
					+31.3	+36.4	+22.6	+23.6

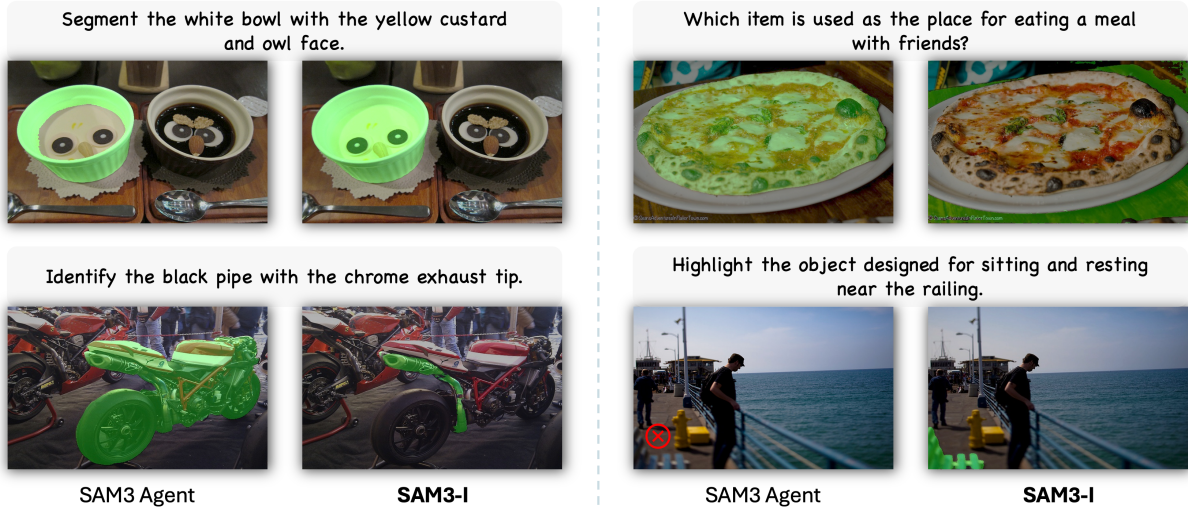


Figure 4: **Qualitative comparison between SAM3-I and SAM3 Agent.** Here Qwen3-VL-8B is used for the agent.

achieving 59.7 / 65.3 on simple instructions and 49.0 / 51.0 on complex instructions. This demonstrates that instruction-level learning does not degrade standard concept prompting. Moreover, compared with the agent-based variant, SAM3-I attains appealing accuracy using fewer parameters and without relying on external MLLM reasoning. These results demonstrate that SAM3-I unifies concept grounding and instruction-level reasoning in an efficient extension of the SAM family.

Analysis of Different Target Granularities.

Tab. 3 evaluates instruction-following segmentation on HMPL-Instruct across three target granularities: *one-to-one*, *one-to-many*, and *one-to-all*, under both simple and complex instructions. SAM3-I achieves consistently strong performance across settings. Notably, SAM3-I significantly outperforms the SAM3 agent-based baseline in the challenging *one-to-many* setting, for both simple and complex instructions, demonstrating its ability to perform effective subset-level reasoning and instance discrimination among visually similar objects, rather than overfitting to category-level segmentation. Additional results at both object and part levels are provided in Appendix C.

Evaluation on Representative Benchmarks. We evaluate SAM3-I on two representative segmentation benchmarks, RefCOCO (Kazemzadeh et al., 2014) and Ref-ZOM (Hu et al., 2023), in Tab. 4. SAM3-I achieves competitive or superior perfor-

mance, despite being trained with substantially less data (Tab. 1) than recent large-scale non-agent method UniPixel (Liu et al., 2025). This demonstrates that SAM3-I performs well to diverse instruction-driven benchmarks and highlights the benefits of building upon SAM3, a concept-driven segmentation foundation pretrained on large-scale pixel-level corpora, which provides robust and transferable concept representations for instruction-level extension.

6.3 Qualitative Results

Fig. 4 provides a qualitative comparison between SAM3-I and SAM3 Agent (Qwen3-VL-8B) on instruction-following segmentation. The left column shows simple referring instructions, while the right column presents more challenging complex instructions. Across both settings, SAM3-I more accurately localizes the intended target than the agent-based baseline, while operating in a single-pass pipeline without external MLLM-driven multi-step mask filtering. For simple referring instructions, SAM3-I demonstrates stronger instance-level discrimination. For example, when asked to segment the “black pipe with the chrome exhaust tip”, SAM3 Agent incorrectly segments the entire motorcycle, whereas SAM3-I precisely isolates the specified component. For complex reasoning instructions, the advantage of SAM3-I is more evident. In the final example, SAM3 Agent fails to

Table 3: **Quantitative results on the HMPL-Instruct dataset under different target granularities**, including *one-to-one*, *one-to-many*, and *one-to-all*, evaluated with simple and complex instructions.

Model	Simple Instruct.						Complex Instruct.					
	1-to-1		1-to-many		1-to-all		1-to-1		1-to-many		1-to-all	
	gIoU	P@50	gIoU	P@50	gIoU	P@50	gIoU	P@50	gIoU	P@50	gIoU	P@50
LISA (Lai et al., 2024)	24.1	20.2	21.8	14.4	19.4	11.8	18.2	13.8	18.2	10.8	17.8	10.2
UniPixel (Liu et al., 2025)	36.7	35.1	38.6	36.0	30.0	23.4	33.3	31.8	36.1	34.3	28.8	23.0
SAM3 Agent	28.3	28.6	14.9	14.9	33.6	34.9	27.1	28.3	20.3	20.6	26.9	27.7
SAM3-I	60.8	66.7	48.9	50.0	60.9	67.8	48.1	49.6	39.9	39.5	54.7	58.7

Table 4: **Results on the RefCOCO and Ref-ZOM.**

Model	RefCOCO		Ref-ZOM	
	gIoU	P@50	gIoU	P@50
LISA (Lai et al., 2024)	57.3	58.8	51.8	52.4
ReLA (Liu et al., 2023)	73.8	83.9	66.7	74.1
UniPixel (Liu et al., 2025)	78.1	88.4	67.0	72.4
SAM3 Agent	71.1	81.9	65.0	73.8
SAM3-I	76.3	85.5	73.6	80.0

Table 5: **Ablation studies of the proposed SAM3-I.** * denotes using a single adapter. \mathcal{L}_{align} represents using all alignment losses.

Stages	Enabled Losses	Simple Instruct.		Complex Instruct.	
		gIoU	P@50	gIoU	P@50
Original SAM3					
1	\mathcal{L}_{seg}	57.6	62.0	32.9	31.0
1,2	\mathcal{L}_{seg}	57.6	62.0	45.9	46.5
2	\mathcal{L}_{seg}	40.8	40.5	43.5	44.7
1,2,3	$\mathcal{L}_{seg} + \mathcal{L}_{inst}$	58.6	64.1	47.4	49.1
1,2,3	$\mathcal{L}_{seg} + \mathcal{L}_{anchor}$	58.6	63.8	47.1	48.0
1,2,3	$\mathcal{L}_{seg} + \mathcal{L}_{mask}$	58.0	63.1	46.4	47.3
1,2,3	$\mathcal{L}_{seg} + \mathcal{L}_{hard}$	58.1	63.4	46.7	48.3
1,2,3*	$\mathcal{L}_{seg} + \mathcal{L}_{align}$	57.7	62.3	46.7	48.4
3	$\mathcal{L}_{seg} + \mathcal{L}_{align}$	57.3	62.0	45.4	46.4
1,2,3	SAM3-I (Ours)	59.7	65.3	49.0	51.0

produce a valid segmentation, while SAM3-I correctly identifies the instructed object by directly grounding the implicit functional semantics.

Additional qualitative results are provided in Appendix C, covering a wide range of instruction-following scenarios, including one-to-one, one-to-many, and one-to-all settings, as well as object-level and part-level targets under both referring and reasoning-based instructions.

6.4 Ablation Studies

We ablate key components of SAM3-I on HMPL-Instruct, with results summarized in Tab. 5.

Multi-Stage Training of Cascaded Adapters. We begin with the original SAM3 as the baseline, which is fundamentally designed to support concept-level prompting but not compatible to handle either simple or complex natural-language instructions. Introducing Stage 1 training enables the S-Adapter to learn attribute-, position-, and relation-level semantics from simple referring instructions. As shown in Tab. 5, this stage yields

appealing performance on simple instructions (57.6 gIoU / 62.0 P@50), while complex instruction performance remains undesired, as expected. Extending training to Stage 2 further activates the C-Adapter and exposes the model to complex, NP-free instructions. This addition enables effective reasoning-based grounding, leading to clear improvements on complex instructions (45.9 gIoU / 46.5 P@50), while maintaining stable performance on simple ones. This progression confirms that complex instruction understanding cannot be acquired implicitly from simple supervision alone, but instead requires explicit reasoning-oriented training. When only C-Adapter is introduced and trained directly on complex instructions, the model can learn a certain degree of reasoning capability, but consistently under-performs the full cascaded design in which the C-Adapter is initialized from the S-Adapter for complex instruction processing. This result suggests that attribute- and relation-level grounding learned by the S-Adapter provides an essential semantic foundation for subsequent high-level reasoning.

Effect of Alignment Losses. Building upon stage 1 and 2 early training, we further examine the effect of introducing alignment losses in Stage 3. Adding the instruction contrastive loss (\mathcal{L}_{inst}) improves both simple and complex instruction performance (e.g., complex: 45.9 \rightarrow 47.4 gIoU), demonstrating its role in enforcing semantic consistency across different instruction formulations while preserving instance-level discrimination. Incorporating the parent-rank concept anchoring loss (\mathcal{L}_{anchor}) also brings noticeable performance improvement (e.g., complex: 45.9 \rightarrow 47.1 gIoU), highlighting the importance of constraining instruction embeddings within appropriate concept boundaries. At the prediction level, introducing the mask distribution alignment loss (\mathcal{L}_{mask}) yields consistent gains across instruction types (e.g., simple: 62.0 \rightarrow 63.1 P@50), indicating that aligning outputs from different instruction branches helps maintain coherent grounding. Finally, adding the uncertainty-aware hard-region supervision loss (\mathcal{L}_{hard}) leads to per-

formance gains (e.g., complex: 46.5 \rightarrow 48.3 P@50), demonstrating the importance of emphasizing semantically difficult regions to improve robustness. With all losses enabled, SAM3-I achieves the best overall performance (59.7 gIoU / 65.3 P@50) on simple instructions and (49.0 / 51.0) on complex instructions, demonstrating that the alignment objectives play complementary roles in regulating instruction semantics and enforcing consistent and robust visual grounding.

Cascaded Adapters vs. Single Adapter. We further validate the necessity of the cascaded design by comparing it with a single-adapter variant (marked as * in Tab. 5). It is observed that a single-adapter variant trained with the same curriculum and alignment losses under-performs the cascaded S-/C-Adapter design (e.g., complex: 46.7 vs. 49.0 gIoU). This suggests that explicitly separating linguistic roles enables progressive refinement of structured, layered instruction representations—an ability that a single adapter struggles to acquire within a shared parameter space.

Multi-stage vs. Single-stage. We further remove the proposed staged curriculum and instead train the cascaded adapters with all losses in a single stage. This results in clear performance degradation on both simple and complex instructions. Without progressive training, the model is required to learn attribute grounding, relation modeling, and high-level reasoning simultaneously, which makes optimization substantially more difficult and prevents the cascaded adapters from forming well-structured, hierarchical linguistic representations.

Ablation results for hyper-parameters ρ and m in Eq. (3) and Eq. (4) are provided in Appendix C.

7 Conclusion

In this work, we presented SAM3-I, an instruction-following extension of the SAM family that unifies concept-level grounding and instruction-level reasoning within a single segmentation framework. Supported by a scalable instruction-centric data construction pipeline, SAM3-I demonstrates that SAM3 can be effectively extended to follow natural-language instructions while preserving its strong concept-driven segmentation capability. We further introduce HMPL-Instruct, a dataset that systematically covers hierarchical instruction semantics and diverse target granularities, thereby providing a unified benchmark for instruction-following segmentation. We hope that SAM3-I serves as an

initial yet meaningful step toward more capable and general instruction-driven segmentation foundation models, and encourages future exploration in this emerging direction.

Acknowledgment

Thanks to the anonymous reviewers for their helpful feedback. This work is partly supported by NSERC Discovery, CFI-JELF, NSERC Alliance, Alberta Innovates, PrairiesCan grants, Shenzhen Science and Technology Program under Grant No. JCYJ20250604182948064 and Tencent Project UP. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agency.

Limitations

While SAM3-I demonstrates the feasibility of extending the SAM family toward instruction-following segmentation, several limitations and opportunities for further improvement remain.

i) Scaling training data for richer instruction knowledge. Our data engine offers a prototype for generating rich instructional corpus, yet scaling to larger and more diverse instruction-mask datasets would further strengthen the model’s ability to understand and reason over natural-language instructions. Similar to how large-scale datasets such as SA-1B, SA-V, and SA-Co (Kirillov et al., 2023; Ravi et al., 2025; Carion et al., 2025) fueled the evolution of the SAM family, scaling PIS data across broader object categories, part-level annotations, and reasoning patterns represents an important step toward a more comprehensive instruction-grounded segmentation foundation model.

ii) Potential bias of instruction-instance pairs. Existing instruction-based segmentation benchmarks are still dominated by *one-to-one* and *one-to-all* instruction-instance mappings, where a single instruction refers to a single instance or all instances of a given concept. In contrast, more challenging *one-to-many* settings, where an instruction simultaneously refers to multiple specific instances, have received comparatively less attention. This imbalance introduces a data bias that may limit the model’s generalization ability to group-level or collective reasoning scenarios. Addressing this limitation calls for curating data with more diverse instruction-instance cardinalities and explicitly modeling multi-instance grounding in training.

iii) Beyond instruction-driven segmentation to-

ward higher-level understanding. Following the design philosophy of the SAM family, SAM3-I primarily focuses on enhancing segmentation foundations through instruction-driven prompting, without explicitly integrating high-level dense grounded understanding or conversational capabilities. While the current design choice allows SAM3-I to remain lightweight and modular, it also leaves room for future extensions. An interesting direction is to couple instruction-following segmentation with deeper semantic reasoning, or conversational feedback, potentially by integrating large language models with stronger contextual understanding (e.g., dense grounded understanding frameworks such as Sa2VA (Yuan et al., 2025)). Such extensions could enable more comprehensive human-model interaction while preserving SAM3-I’s powerful segmentation foundation.

Ethical Considerations

The proposed SAM3-I framework and the accompanying dataset are designed to support research on instruction-based segmentation and do not involve any content that violates ethical norms or societal standards. The proposed HMPL-Instruct dataset does not contain personally identifiable information and is not intended for surveillance, identification, or privacy-invasive applications. We envisage that the most proximate impacts of this work will be positive, providing an open and valuable resource for research community.

References

- Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.
- Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, and 19 others. 2025. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607.
- Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. 2020. Deep multi-modal object detection and semantic segmentation for autonomous driving: datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22:1341–1360.
- Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. 2016. Segmentation from natural language expressions. In *European Conference on Computer Vision*, pages 108–124.
- Yutao Hu, Qixiong Wang, Wenqi Shao, Enze Xie, Zhenguo Li, Jungong Han, and Ping Luo. 2023. Beyond one-to-one: Rethinking the referring image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4067–4077.
- Donggon Jang, Yucheol Cho, Suin Lee, Taehyeon Kim, and Dae-Shik Kim. 2025. Mmr: A large-scale benchmark dataset for multi-target and multi-granularity reasoning segmentation. *International Conference on Learning Representations*.
- Wei Ji, Jingjing Li, Cheng Bian, Zhicheng Zhang, and Li Cheng. 2023. Semantict: A large-scale dataset and method for robust semantic segmentation in multispectral images. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 3307–3316.
- Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 787–798.
- 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. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026.
- Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. 2024. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9579–9589.
- Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. 2023. Open-vocabulary semantic segmentation with mask-adapted clip. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7061–7070.
- Jianhua Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.
- Chang Liu, Henghui Ding, and Xudong Jiang. 2023. Gres: Generalized referring expression segmentation. In *Proceedings of the IEEE/CVF Conference*

- on *Computer Vision and Pattern Recognition*, pages 23592–23601.
- Ye Liu, Zongyang Ma, Junfu Pu, Zhongang Qi, Yang Wu, Shan Ying, and Chang Wen Chen. 2025. Unipixel: Unified object referring and segmentation for pixel-level visual reasoning. In *Advances in Neural Information Processing Systems*.
- Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3431–3440.
- Shuyi Ouyang, Hongyi Wang, Shiao Xie, Ziwei Niu, Ruofeng Tong, Yen-Wei Chen, and Lanfen Lin. 2023. Slvit: Scale-wise language-guided vision transformer for referring image segmentation. In *International Joint Conference on Artificial Intelligence*, pages 1294–1302.
- Yun Peng, Xiao Lin, Nachuan Ma, Jiayuan Du, Chuangwei Liu, Chengju Liu, and Qijun Chen. 2025. Samlad: Segment anything model meets zero-shot logic anomaly detection. *Knowledge-Based Systems*, 314:113176.
- Benedict Quartey, Eric Rosen, Stefanie Tellex, and George Konidaris. 2025. Verifiably following complex robot instructions with foundation models. In *IEEE International Conference on Robotics and Automation*, pages 1–8. IEEE.
- 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. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PmLR.
- 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. 2023. Paco: Parts and attributes of common objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7141–7151.
- 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. 2025. Sam 2: Segment anything in images and videos. *International Conference on Learning Representations*.
- Heeseong Shin, Chaehyun Kim, Sunghwan Hong, Seokju Cho, Anurag Arnab, Paul Hongsuck Seo, and Seungryoung Kim. 2024. Towards open-vocabulary semantic segmentation without semantic labels. *Advances in Neural Information Processing Systems*, 37:9153–9177.
- Peize Sun, Shoufa Chen, Chenchen Zhu, Fanyi Xiao, Ping Luo, Saining Xie, and Zhicheng Yan. 2023. Going denser with open-vocabulary part segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15453–15465.
- Azhar Wahid, Miftachul Huda, Moh Abdul Rohim, Abdul Halim Ali, Khairul Ghufuran Kaspin, Maskanatul Fiqiyah, and Muhammad Talhah Ajmain Jima'ain. 2024. Augmented reality model in supporting instruction process: a critical review. In *International Congress on Information and Communication Technology*, pages 69–83. Springer.
- Zifu Wan, Yaqi Xie, Ce Zhang, Zhiqiu Lin, Zihan Wang, Simon Stepputtis, Deva Ramanan, and Katia P Sycara. 2025. Instructpart: Task-oriented part segmentation with instruction reasoning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 24202–24227.
- Junde Wu, Ziyue Wang, Mingxuan Hong, Wei Ji, Huazhu Fu, Yanwu Xu, Min Xu, and Yueming Jin. 2025. Medical sam adapter: Adapting segment anything model for medical image segmentation. *Medical Image Analysis*, 102:103547.
- Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. 2023. Side adapter network for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2945–2954.
- Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr. 2022. Lavt: Language-aware vision transformer for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18155–18165.
- Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. 2016. Modeling context in referring expressions. In *European Conference on Computer Vision*, pages 69–85.
- Haobo Yuan, Xiangtai Li, Tao Zhang, Yueyi Sun, Zilong Huang, Shilin Xu, Shunping Ji, Yunhai Tong, Lu Qi, Jiashi Feng, and Ming-Hsuan Yang. 2025. Sa2va: Marrying sam2 with llava for dense grounded understanding of images and videos. *arXiv preprint arXiv:2501.04001*.
- Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, Nanyun Peng, Lijuan Wang, Yong Jae Lee, and Jianfeng Gao. 2023a. Generalized decoding for pixel, image, and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15116–15127.
- Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. 2023b. Segment everything everywhere all at once. *Advances in Neural Information Processing Systems*, 36:19769–19782.

Appendix

This appendix provides supplementary material supporting the main paper. We first review recent efforts on language-guided segmentation and instruction-centric segmentation benchmarks related to SAM3-I in Appendix A. We then present additional details on dataset construction in Appendix B, followed by extended experimental results in Appendix C, including hyperparameter analysis, additional visualizations, and further qualitative and quantitative evaluations. Implementation details are provided in Appendix D. Finally, Appendix E presents statements on artifacts and ethics.

A Related Work

A.1 Evolving Language-Guided Segmentation

Segmenting objects described by language has attracted increasing attention in recent years, driven by the demand for flexible, open-ended, and human-centric visual understanding systems. Along this line, prior studies can be broadly categorized by the level of linguistic abstraction they support.

Early efforts primarily focus on *open-vocabulary semantic segmentation*, which aims to generalize segmentation models beyond a fixed closed set of categories (Liang et al., 2023; Sun et al., 2023; Xu et al., 2023; Shin et al., 2024) by leveraging large-scale vision-language pretraining (Radford et al., 2021). By aligning visual representations with textual embeddings learned from image-text corpora, these approaches enable open-set category recognition and segmentation, laying an important foundation for language-driven segmentation. However, most open-vocabulary methods are limited to short NPs or category names, and are not designed to handle richer instructional language or instance-specific reasoning.

To support more precise instance-level grounding, *referring expression segmentation* (Hu et al., 2016) is introduced to segment a target object specified by an explicit natural language description. A large body of work (Ouyang et al., 2023; Yang et al., 2022; Liu et al., 2023; Hu et al., 2023) explores cross-modal feature fusion, attention mechanisms, and relational modeling to resolve attributes, spatial relations, and inter-object dependencies. More recent approaches (Zou et al., 2023a,b; Yuan et al., 2025) further benefit from vision and vision-language foundation models, such as SAM and

CLIP, significantly improving robustness and generalization. Despite these advances, referring expression segmentation typically assumes that the target object is explicitly mentioned and that each instruction corresponds to a single, well-defined instance, limiting its applicability to more implicit or compositional instructions.

More recently, research (Lai et al., 2024) has begun to explore *reasoning-based segmentation*, where the target is specified implicitly through functional, relational, or commonsense descriptions, rather than explicit object names, marking an important step toward higher-level semantic understanding. Following efforts (Wan et al., 2025; Jang et al., 2025) investigate part-level segmentation with instruction reasoning, enhancing the ability of segmentation models to recognize each part of the target and its corresponding affordances before grounding to task-related regions.

Overall, the evolution from category-level open-vocabulary segmentation, to instance-level referring expression segmentation, and further toward reasoning-centric segmentation reveals a clear trend toward richer language understanding and more flexible visual grounding. In parallel, the SAM family has evolved from relying on low-level visual prompts to support concept-level prompts, demonstrating strong generalization and robust concept understanding across diverse domains. Nevertheless, existing SAM-based models are not explicitly designed to follow free-form natural language instructions or perform instruction-level reasoning. These observations motivate us to extend the SAM paradigm toward *instruction-following segmentation*, leading to SAM3-I, which aims to unify concept-level understanding and instruction-level reasoning within the SAM family.

A.2 Instruction-Centric Segmentation Benchmarks

A variety of datasets have been proposed to support referring expression and reasoning-based segmentation. Early benchmarks such as RefCOCO, RefCOCO+, and RefCOCOg (Kazemzadeh et al., 2014; Yu et al., 2016) focus on explicit object references with increasing linguistic complexity, encouraging models to reason over attributes and relations rather than relying solely on location cues. Subsequent extensions (Liu et al., 2023; Hu et al., 2023) explore multi-target referring scenarios, but still assume that target categories are explicitly specified in the text.

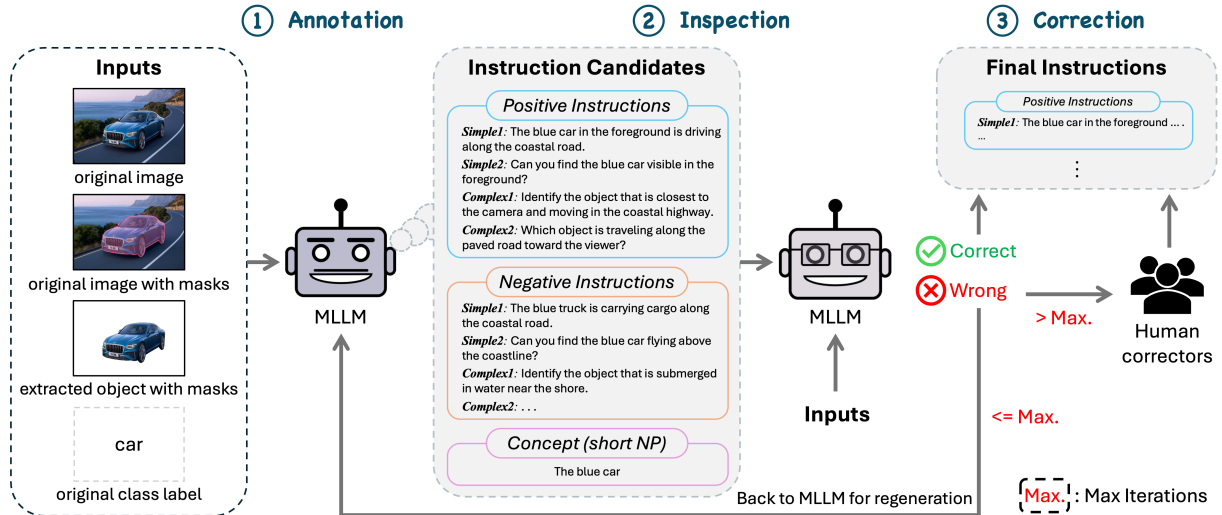


Figure 5: Overview of the scalable instructional data construction pipeline. (Sec. 5)

More recent datasets shift toward implicit and reasoning-driven instructions, where target names are intentionally omitted and segmentation relies on functional or semantic descriptions (Lai et al., 2024). This dataset significantly advances the field by introducing higher-level reasoning requirements, and have been further extended to part-level and multi-granularity settings (Wan et al., 2025; Jang et al., 2025). While these efforts substantially enrich instruction diversity, they typically address specific referring or reasoning paradigms and target scopes in isolation.

In contrast, our **HMPL-Instruct** dataset aims to provide a unified benchmark that systematically covers the full spectrum of instruction types. It integrates concept-level prompts, explicit referring expressions, and implicit complex instructions within a single dataset, while supporting object-level and part-level annotations as well as diverse target cardinalities, ranging from one-to-one and one-to-many to one-to-all scenarios. By unifying instruction hierarchy and target granularity in a coherent benchmark, HMPL-Instruct is designed to facilitate the development and evaluation of general instruction-following segmentation models.

B Data Construction Details

B.1 Human-in-the-Loop Data Construction

As described in Sec. 5, we develop a *human-in-the-loop data engine*. Fig. 5 illustrates its overall workflow in detail. This engine yields rich instruction-mask pairs that primarily support *one-to-one* and *one-to-all* grounding scenarios. For the

more challenging *one-to-many* setting, where an instruction refers to a *subset* of visually similar instances, we develop a web-based dual-panel instruction annotation tool to facilitate high-quality instruction-grounding annotations, especially for subset-level (*one-to-many*) scenarios. The tool supports both *Annotation* and *Inspection* workflows.

Design and Rationale. As shown in Fig. 6, the interface displays the same sample in two synchronized panels. The left panel (*Annotate*) is interactive and is used to place point prompts on an image with mask overlays. Importantly, in our setting, point prompts are used as a subset selection signal: when multiple instances of the same concept exist in a scene (e.g., ten persons), annotators may write an instruction that refers to only a subset (e.g., five of them). The clicks explicitly indicate which instances are intended by the instruction. Since the source dataset provides instance masks, we subsequently assign each click to its corresponding mask region (via a point-in-mask check), thereby determining the selected target subset in a reliable and scalable manner. The right panel (*Reference*) is non-interactive and shows the original image without overlays, enabling annotators and inspectors to verify fine visual cues (e.g., color, material, and subtle boundaries) that might be partially occluded by mask visualizations.

Annotation Fields and Their Roles. For each image, the tool records three complementary supervision signals: i) *Point prompts*. Annotators click on the left panel to add or delete points. All coordinates are stored in the original image resolution to remain invariant to display scaling. These

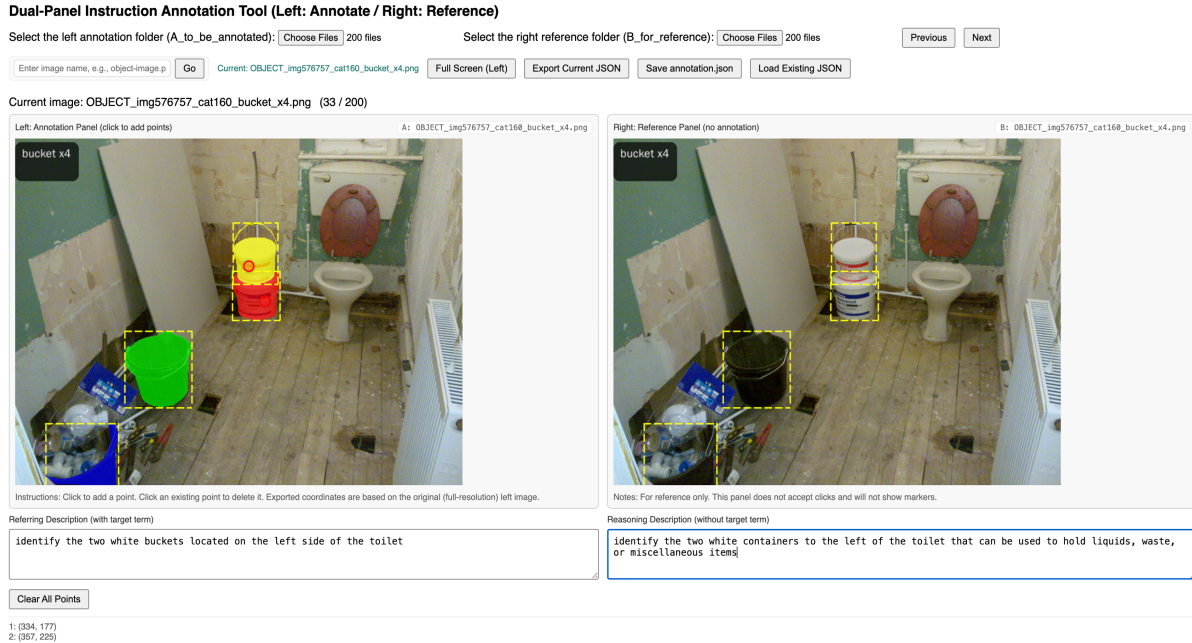


Figure 6: **Overview of our dual-panel instruction annotation tool.** The interface presents the same image in two synchronized panels. The left *Annotate* panel supports interactive point placement with mask overlays to specify target instances or subsets (e.g., *one-to-many* selection), while the right *Reference* panel displays the original image without overlays for visual verification. For each sample, annotators provide point prompts, a referring description with the target term, and a reasoning description without the target term, enabling unified collection of simple and complex instruction-grounding annotations.

points serve as the instance selection interface for subset-level grounding and enable unambiguous mapping from an instruction to the corresponding mask subset. ii) *Referring description (with target term)*. Annotators write a concise referring instruction that explicitly contains the target term (e.g., “bucket”), serving as a direct grounding cue for simple instructions. iii) *Reasoning description (without target term)*. Annotators provide a semantically equivalent instruction that intentionally excludes the target term, requiring implicit identification through functional, relational, or contextual reasoning. This supports complex instruction evaluation and training, where the model must infer targets without explicit naming. Together, these components form a structured annotation unit that supports both explicit referring and implicit reasoning under a unified interface.

Usability features for annotation and inspection.

To support efficient annotation and iterative review, the tool provides: i) a full-screen mode for the left panel with $1 \times 2 \times 3 \times$ zoom and Shift+drag panning for precise point placement; ii) auto-save to local storage after any update to prevent annotation loss; iii) export/import utilities, including per-sample JSON export, saving all annotations into

a single `annotation.json`, and loading existing JSON files for resuming and secondary inspection; and iv) efficient navigation via previous/next browsing and jump-by-filename for targeted verification and error correction.

Annotation and Inspection Protocol. We adopt a two-stage human workflow. During *Annotation*, annotators i) place points to indicate the intended target instances (or subsets), ii) write a referring description with the target term, and iii) write a reasoning description without the target term while following predefined constraints (e.g., complex instructions must not reveal the target NP). During *Inspection*, reviewers use the right reference panel to verify the faithfulness of the textual descriptions to the original visual evidence and check the consistency between selected points and intended targets. Samples with unavoidable referential ambiguity (particularly in subset selection) are flagged and removed.

Overall, the dual-panel tool enables collection of point-assisted subset grounding together with paired referring and reasoning descriptions, while supporting robust human verification via a clear separation between interactive annotation and non-interactive reference views.

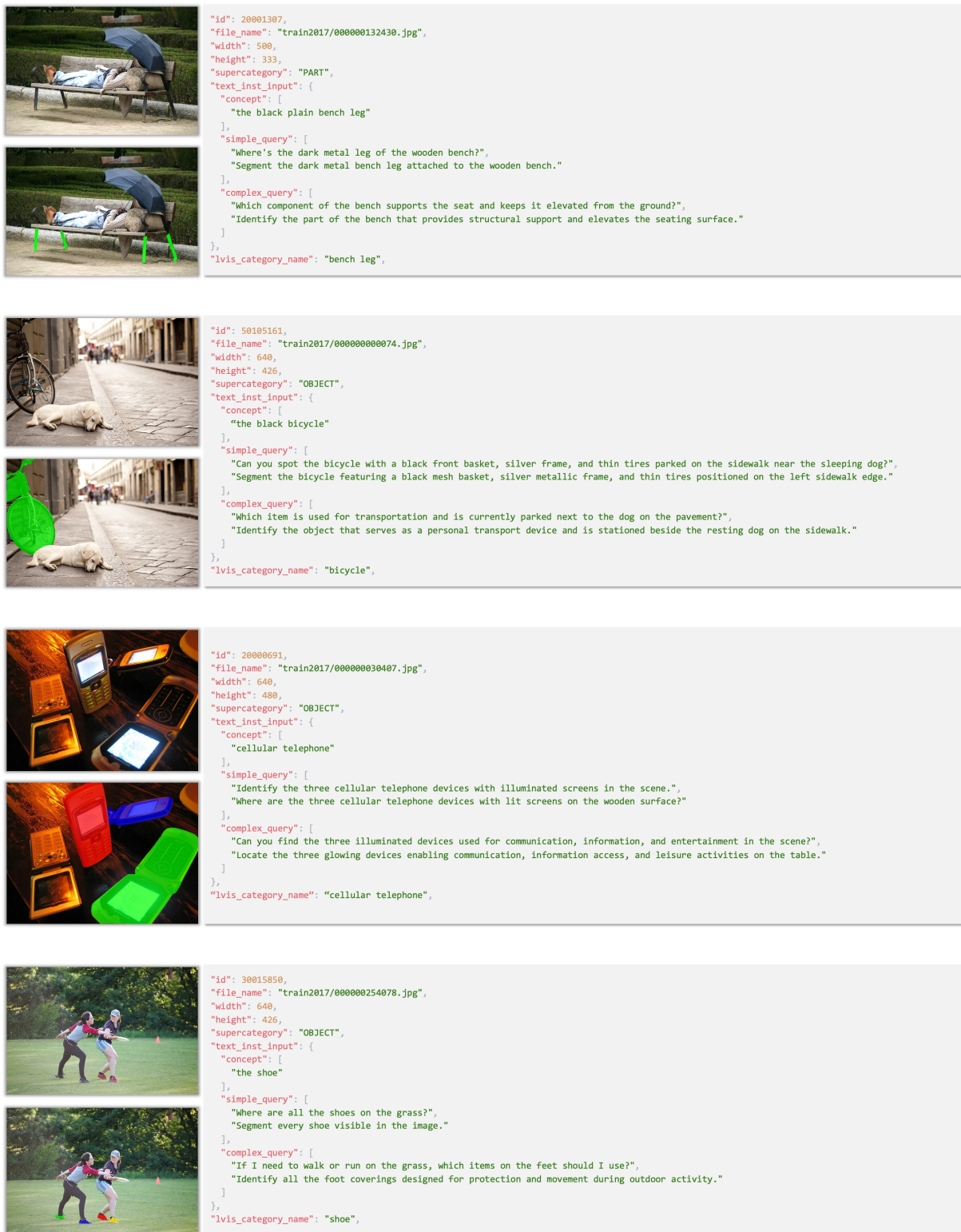


Figure 7: Examples from the proposed HMPL-Instruct dataset. Each sample is represented as a structured JSON entry, incorporating a hierarchical instruction taxonomy with *concept*-, *simple*-, and *complex*-level queries for both object- and part-level segmentation. Target instances are visualized with colored masks. The first example illustrates a part-level segmentation case, while the remaining examples demonstrate object-level scenarios with *one-to-one*, *one-to-many*, and *one-to-all* instruction-instance mappings.

Segment the object that serves as a visual representation of a drink in the skateboard's artwork.



Can you show me the two dark cups with lettering behind the pizzas?



Locate the two gray containers under the blue pole used for collecting and sorting garbage.



Locate the two white items on the sofa that provide back support and aid in body relaxation.



Would you kindly find the two white objects with handles on either side of the white plate, used for holding drinks?



If I need to carry personal items like coins, keys, or a phone, which objects should I use?



Identify all the soft objects designed for head and neck support during rest.



Input Image

Ground Truth

UniPixel

SAM3-I

Figure 8: Visualization results on representative instruction-guided segmentation approaches.

C More Experimental Results

C.1 Hyperparameter Analysis

We further analyze the sensitivity of SAM3-I to the key hyper-parameters ρ and m , which respectively control the weighting factor in the instruction contrastive loss and the margin in the parent-rank concept anchoring loss.

In Tab. 6, we observe that setting $\rho = 0.5$ yields the best overall performance. When ρ is too small, negative samples from the same concept category are overly down-weighted, weakening intra-class discrimination and causing instruction embeddings to collapse toward coarse concept-level representations. Conversely, a large ρ enforces excessive repulsion among same-concept instances, which harms semantic consistency across different instruction expressions and destabilizes training. Setting $\rho = 0.5$ provides a balanced trade-off between semantic invariance and instance-level discriminability, leading to optimal performance.

In Tab. 7, we also examine the effect of the margin parameter m in $\mathcal{L}_{\text{anchor}}$. We find that larger margins result in noticeable performance degradation, indicating overly restrictive semantic constraints. A moderate margin of $m = 0.2$ achieves the best performance. These results suggest that the proposed parent-rank concept anchoring loss effectively regularizes instruction embeddings to remain aligned with their parent concept semantics, while preserving flexibility for subset-level and instance-specific instruction understanding.

Table 6: **Ablation analysis on hyper-parameter ρ .** The performance is the average of simple and complex scenarios.

	$\rho = 0.3$	$\rho = 0.5$	$\rho = 0.7$
gIoU	54.1	54.4	52.8

Table 7: **Ablation analysis on hyper-parameter m .** The performance is the average of simple and complex scenarios.

	$m = 0.1$	$m = 0.2$	$m = 0.3$
gIoU	54.2	54.4	53.7

C.2 More Quantitative and Qualitative Results

In Tables 8, we report additional quantitative results on the HMPL-Instruct dataset at both the object and part levels. These supplementary results pro-

Table 8: **Quantitative results on the HMPL-Instruct dataset under both object and part levels.**

Model	Object Level		Part Level	
	gIoU	P@50	gIoU	P@50
LISA (Lai et al., 2024)	31.5	26.7	13.9	8.2
UniPixel (Liu et al., 2025)	42.5	45.4	23.4	17.5
SAM3 Agent	38.7	42.4	20.7	19.8
SAM3-I	66.6	72.2	47.2	50.0

vide a more comprehensive view of model behavior across different target granularities.

We further present additional visualizations to illustrate the characteristics of the HMPL-Instruct dataset, as shown in Fig. 7. The examples span object- and part-level scenarios, covering multiple instruction hierarchies (*concept*, *simple*, and *complex*) as well as diverse instruction–instance mappings (*one-to-one*, *one-to-many*, and *one-to-all*). In addition, Fig. 8 visualizes annotated targets alongside model predictions, offering qualitative insights into how the most recent UniPixel (Liu et al., 2025) and our SAM3-I respond to varying *one-to-many* instructions reasoning. Importantly, in these cases, the target cannot be determined from a concept category alone and must be inferred from higher-level semantics. For example, in the fifth row of Fig. 8, the instruction refers to “the two white objects with handles on either side of the white plate, used for holding drinks”. Target identification relies on a combination of attributes, quantity constraints, spatial relations, and functional cues, rather than a concept label such as cup. These results demonstrate that SAM3-I goes beyond latent concept bias and genuinely learns to ground instructions through relational and functional reasoning, particularly in complex, NP-free scenarios.

D Implementation Details

For data engine construction, we adopt two separate Qwen3-VL-8B (Instruct) models (Bai et al., 2025) to perform the annotation–inspection loop described in Sec. 5. The entire process is executed on 32 NVIDIA H100 GPUs to support large-scale instruction generation and verification. Subsequent model training is conducted with a batch size of 32 on 16 NVIDIA H100 GPUs. The hyperparameters ρ and m are set to 0.5 and 0.2, respectively. The temperature parameter τ in Eq. (2) is set to 0.1 following (Chen et al., 2020). Unless otherwise specified, all other hyper-parameters follow the original SAM3 settings without modification. For the SAM3 agent, we use Qwen3-VL-8B as

the external MLLM agent and set the maximum number of iterative checking rounds to 3. *Source code and HMPL-Instruct dataset are available at <https://github.com/debby-0527/SAM3-I> to support future research on instruction-based segmentation.*

For the cascaded adapters, they are inserted as residual-parallel branches to the residual architectures in SAM3’s text encoder and detector. Specifically, in the SAM3 text encoder, adapters are inserted into each Transformer block as parallel residual modules alongside the feed-forward network, enabling instruction-aware modulation of textual features without modifying the original representations. In the SAM3 detector, adapters are inserted in two locations. First, they are added to the transformer blocks in the multimodal decoder using the same residual-parallel design as in the text encoder. Second, adapters are inserted into the pixel decoder as parallel residual modules along the feature pyramid network connections, allowing instruction semantics to effectively propagate into mask prediction. In detail, each adapter adopts a bottleneck structure with a down-projection to dimension 64, followed by a GELU activation and an up-projection back to the original dimension. The internal MHSA operates at feature dimension 1024 with 4 attention heads. The total number of trainable parameters introduced by the adapters is approximately 311.9M (0.3B). All backbone parameters remain frozen during training.

E Artifact & Ethics Statement

E.1 External Artifacts and Attribution

This work builds upon several publicly available datasets and models. We use PACO-LVIS (Ramanathan et al., 2023) as the base dataset for constructing HMPL-Instruct, and SAM3 (Carion et al., 2025) as the segmentation foundation. For agent-based baselines and instruction verification, we use Qwen-VL models (Bai et al., 2025). All external artifacts are properly cited in the main paper.

E.2 License and Intended Use

All external datasets and models used in this work are released for research purposes, and our usage complies with their original licenses and intended scope. HMPL-Instruct dataset is derived solely from publicly available annotations and images and is intended for academic research on instruction-following segmentation. The dataset is not de-

signed for commercial deployment.

E.3 Privacy

HMPL-Instruct does not introduce new personally identifiable information. All images originate from existing public datasets and retain their original privacy and ethical characteristics. No additional personal data is collected, inferred, or annotated in the construction process.

E.4 Dataset Statistics and Documentation

We provide comprehensive dataset statistics in Sec. 5 and Tab. 1, including the number of samples, instruction categories (concept, simple, complex), target granularities (one-to-one, one-to-many, one-to-all), and object- and part-level annotations. Detailed descriptions of the data construction pipeline and annotation process are also provided.