PROMPT OPTIMIZATION ENABLES IMPROVING AND SELF-DISTILLING IMAGE SEGMENTATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-Promptable segmentation models (PSMs) like SAM 3 and Falcon Perception can generate masks from natural language. However, we know from other model types like LLMs that prompt optimization can greatly improve outputs, and that these improved outputs can be used to self-train the model for recursive self-improvement. To test this for segmentation models, we study whether PSMs can be improved without ground-truth labels by optimizing the prompts using multimodal LLMs. First, we test and find that training-free iterative prompt refinement, where an MLLM inspects the image and current mask to propose improved prompts, raises segmentation quality by approximately 20 IoU points over category-name prompts. Then, we train MLLMs via GRPO to compress multi-step refinement into a single prompt rewrite, which achieves similar quality at a lower inference cost. Finally, we close the loop by distilling optimized masks back into the segmentation model, so that it produces improved masks directly from the original prompt. Through this work, we enable label-free self-improvement for PSMs for the first time and provide a foundation for autonomous adaptation to new domains.

1 INTRODUCTION

Segmentation models such as SAM (Kirillov et al., 2023, Carion et al., 2026) are strong at detecting objects, and often show generalization to complex and compositional prompts (Bevli et al., 2026). This makes them attractive tools for open-vocabulary segmentation and object grounding. However, segmentation models, like almost all foundation models, are trained on a static dataset and therefore need to adapt to new domains, new object contexts, or tail concepts not well covered in the training data. Unfortunately, this is expensive, as creating ground-truth segmentation masks incurs significant, and often manual, effort. Thus, instead of costly manual annotation, using synthetic data is preferable.

In this work, we are interested in self-improvement techniques for promptable segmentation models. We ask whether we can derive improved segmentation masks using only the initial model and test-time scaling together with a strong scoring model. In this case, we generate multiple prompts, find the best one, and obtain the mask generated from the best prompt. This is an exciting direction, because it could provide high-quality synthetic labels at a small cost. Additionally, if this is possible, we can then fine-tune the segmentation model itself on the created synthetic masks to further improve its capabilities. This is an instance of self-distillation, i.e., we elicit knowledge that the model fundamentally has but that may not easily surface, and then distill it back in a way so the model can improve its performance even without test-time scaling.

Concretely, we study self-improvement in segmentation models from three perspectives. First, we systematically evaluate and confirm that we can obtain significantly better masks through tuning the language prompt for the segmentation model. Best-optimized prompts generally score about 20% IoU higher than simply using the object name, which is a substantial improvement. Building on this insight, we propose an RL-workflow to fine-tune MLLM policies to one-shot generate optimized prompts, which delivers similar gains at reduced deployment cost (less inferences required). Finally, we show that distilling the optimized masks back into the segmentation model also substantially improves their performance, now even with simple prompts, which fundamentally demonstrates that self-improvement in promptable segmentation models is possible.

In summary, our contributions are: (1) We propose and evaluate self-improvement in promptable image segmentation models. (2) We systematically show that MLLMs can iteratively optimize the

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

text prompt for segmentation models like Falcon Perception and SAM3.1. (3) We show that this iterative optimization process can be compressed into one-shot generation by GRPO-tuning of the MLLM policy, and this generalizes to hold-out datasets. (4) We demonstrate that the segmentation model's downstream performance can be improved by using the masks from MLLM optimization as distillation targets, which enables label-free adaptation to downstream datasets. Our work lays the foundation for self-improvement in promptable segmentation models, and we expect future research building on our work to significantly enhance segmentation models and their adaptation abilities.

2 Related Work

Image Segmentation. Deep learning-based segmentation models (Long et al., 2015) have been recently extended to open-vocabulary object-search models. Historically, segmentation has transcended closed-class settings in two ways: (1) We know the object, i.e. the pixel groupings, but need to label them, and (2) we have an object prompt, but want to find it in an image. For (1), most methods map object representations into CLIP space, which is multimodal, so it can be used for zero-shot classification with arbitrary label sets (Li et al., 2022, Ghiasi et al., 2022, Yu et al., 2023). For (2), segment anything (SAM, Kirillov et al. (2023), Ravi et al. (2025)) models accepted point or bounding box prompts and solved the pixel grouping problem without giving class names. Their successors, like SAM3 (Carion et al., 2026) and Falcon Perception (Bevli et al., 2026), accept natural language prompts and return the segmentation mask of the prompted image (if it is present), which also in part replace reasoning segmentation models, i.e., MLLMs fine-tuned to output segmentation masks after reasoning over the user prompt (Rasheed et al., 2024, Lai et al., 2024, Liu et al., 2025, Huang et al., 2025, Zhu et al., 2026a). In this work, we focus on text-promptable segmentation models and how in combination with MLLMs they can be used to obtain high-quality segmentation masks without ground-truth annotations.

Prompt Optimization. For language models, automatic prompt optimization (Zhou et al., 2023, Guo et al., 2024, Yang et al., 2024, Fernando et al., 2024) yields strong performance improvements, and the same is observed for multimodal large language models (Qu et al., 2025, Zhu et al., 2026b, Choi et al., 2026). These methods iteratively improve prompts based on task feedback. This direction has been less explored for promptable segmentation models. For point and bounding-box prompts, i.e., using SAM and SAM2, Zhu et al. (2025) and Du et al. (2026) fine-tune MLLMs on trajectories that refine prompts until the segmentation is correct. Here, the main goal is to have an agent (the MLLM) that can independently perform iterative prompt refinement, but the goal is not to improve the segmentation model itself, and it lacks the language component. Wang & Yu (2026) additionally compose masks over multiple iterations but also don't train the segmentation model. Finally, SAM3 Agent (Carion et al., 2026) iteratively optimizes the prompt with MLLM proposals (also without training the model), and Ye et al. (2025) combine prompt optimization for segmentation with an evolutionary algorithm, but their prompt scoring relies on an expensive tournament selection process that involves pairwise ranking of masks.

Self-Training. Self-training methods attempt to improve a model with its own predictions. For LLMs, this has been demonstrated by Zelikman et al. (2022), who propose to generate reasoning traces, filter out wrong ones, and train on the retained set. Subsequent work extended this to arbitrary reward scores (Gulcehre et al., 2023, Yuan et al., 2024), formalized it as an EM-style algorithm (Singh et al., 2024), and included reasoning over rejected predictions as additional feedback (Dou et al., 2024). Our proposed self-training procedure builds on these approaches by coupling segmentation-model fine-tuning with prompt optimization.

3 Preliminaries

Task and Motivation. Fundamentally, our task is referring expression segmentation (RES): Given an image x from a dataset D and a text prompt p that describes an object in the image, we want to obtain a segmentation mask $m = S(x, p)$ that grounds p in the image. If the described object does not exist in x, then the desired mask is empty. To achieve this, we use a pretrained promptable segmentation model (PSM), such as SAM 3.1 (Carion et al., 2026) or Falcon Perception (Bevli et al., 2026). These models target precisely our application, i.e., they take a text prompt and an image and output a segmentation mask.



Figure 1: Selected training-free trajectories with Gemma 4 31B and Falcon Perception. Green denotes ground truth and fuchsia predictions. Dots mark omitted steps. Prompts are verbatim, and IoU (%) scores the displayed masks.

Prompt Optimization. The property that PSMs accept text prompts enables prompt optimization (Reynolds & McDonell, 2021, Zhou et al., 2023). We formalize the prompt space of a PSM as the set $\mathcal{P}_{\text{text}}$ of all possible sequences of language tokens, i.e., text prompts. Thus, the problem of prompting reduces to searching for a $p^* \in \mathcal{P}_{\text{text}}$ that induces the best output m^* under a given task objective (e.g., mask accuracy conditioned on some downstream reward or supervision). We introduce a policy model π_θ, implemented by a multimodal large language model (MLLM), to explore this space by proposing text prompts. This can be achieved iteratively by including feedback (Pryzant et al., 2023), such as a visualization of the map on the image, which an MLLM can use to optimize the prompt (Sec. 4). Alternatively, we tune the policy π_θ to directly generate optimized prompts (Sec. 5).

Datasets and Evaluation Metrics. Throughout this paper, we use tasks from the following datasets: birds in CUB-200-2011 (Wah et al., 2011), grocery products in D2S (Follmann et al., 2018), clothing in Fashionpedia (Jia et al., 2020), food in FoodInsSeg[1] and FoodSeg103 (Wu et al., 2021), and flowers in Oxford Flowers 102 (Nilsback & Zisserman, 2008). These datasets provide a broad range of segmentable objects. A task in the context of this paper refers to a (image, object) pair, where the object is represented by its class name, and the goal is to find the optimal segmentation mask for this object on the image. For each experiment, we randomly sample subsets of tasks from these datasets, while keeping training, development, and test splits strictly separate. In all cases, generated masks are evaluated by IoU against the ground-truth masks provided by the datasets.

4 Training-Free Iterative Prompt Optimization

In the first experiment, we evaluate how much segmentation performance can improve by optimizing the text prompt. We keep both the segmentation model and the MLLM policy frozen and change only the prompt supplied to the segmentation model. This tests whether better prompts can produce better masks without additional training. By showing that the performance of PSMs can be improved substantially by prompt optimization, this experiment provides the basis for improving the MLLM to generate better prompts directly without iterative optimization, which is more efficient (Sec. 5). It also provides the basis for improving the segmentation model without ground-truth masks in Sec. 6.

Method. We assume that the PSM S understands the images and prompts in D, but may not localize the target reliably. We combine S with an MLLM policy π_θ that proposes prompts in $\mathcal{P}_{\text{text}}$. Given the original image x, a prompt p, and its mask overlay $x \odot m$, the policy proposes a revised prompt p'. The segmentation model then produces a new mask $m' = S(x, p')$. At refinement step i, the policy

[1] https://github.com/jamesjg/FoodInsSeg

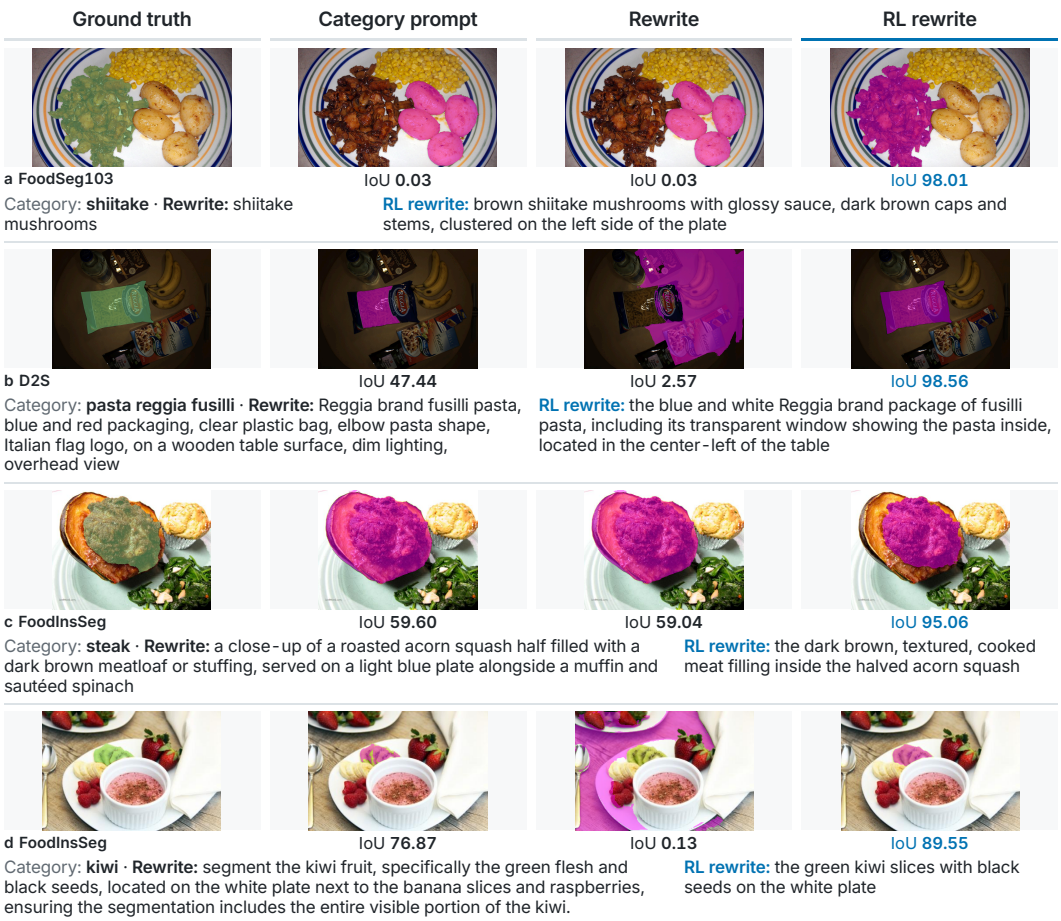

Figure 2: Single-rewrite examples with Qwen 3.5 4B and frozen Falcon Perception. Rewrite and RL rewrite use the policy before and after 100 RL steps. Green denotes ground truth and fuchsia predictions. Prompts are verbatim, and IoU scores the displayed masks.

observes the category prompt at step zero and the history $\{(x \odot m_j, p_j)\}_{j=0}^{i-1}$ as multi-image input. The policy receives neither ground-truth masks nor numerical scores. It must judge the masks from the image and overlays alone. See Fig. 1 for an example.

Models and Datasets. We evaluate ten pretrained MLLMs with Falcon Perception and SAM 3.1. For each MLLM, we use the same 300 test tasks from each of six datasets in Sec. 3 for every MLLM–PSM pair, giving 1,800 attempted trajectories per pair. To construct our evaluation set, we randomly sample 300 masks from each dataset without replacement, but limit the samples to non-empty masks.

Evaluation and Baselines. We evaluate prompt optimization in two settings. In *Fixed Iterations*, each trajectory starts with the category prompt at step zero and makes exactly ten MLLM refinements. Early termination is not allowed. In *Submission*, the MLLM may submit its current mask when it judges that no further refinement is needed, including at step zero. Otherwise, it can make up to ten refinements. This lets the MLLM decide when to stop from the observed masks.

We compare with two baselines. *Category prompting* uses the object's category name with underscores and dashes replaced by spaces, for example, "Black_Tern" becomes "Black Tern". *Base* uses the first MLLM rewrite without RL training. This rewrite receives the original image, target category, and category-mask overlay, so it already attempts to correct the initial segmentation. The comparison with later steps measures the effect of further refinement. The instructions are given in Sec. B.

We score masks with IoU against the target annotation. In Tab. 1, the four fixed-length scores use accepted traces containing the category mask and all ten refinements, without filling missing steps. Category uses step zero, Base step one, and Final step ten. Max averages each trace's highest IoU,

including the category mask. Each row averages accepted tasks directly, so task sets and dataset proportions differ between models. Ground truth is used only to report scores, not to select a mask.

With submission enabled, Accepted includes both valid submissions and traces that reach ten refinements without an invalid action. *Submitted only* averages IoU over voluntary submissions. *Submitted + Last* includes all 1,800 attempts, using the last observed mask whenever the model did not submit, including traces ending with an error. *Submitted + First* instead uses the first refinement for these traces, or the category mask if no refinement was produced. Both fallback columns retain genuine zero scores. The submission *Max* averages each submitted trace's best observed IoU, including the category mask. It uses the same traces as Submitted only.

Table 1: Training-free IoU from two runs on the 1,800 task test set per model pair. Accepted counts traces completed without invalid actions. Submitted (%) is the percentage of all 1,800 attempts that submit. Mean step averages the refinement step among submitted traces only. Submitted only and the submission Max use voluntary submissions, while +Last and +First columns use all 1,800 attempts. Each Max averages the highest IoU within each trace.

		Fixed 10 refinements No submission					Maximum 10 refinements Submission enabled						
MLLM	Params. (B)	Accepted	Category	Base	Max	Final	Accepted	Submitted (%)	Mean step	Submitted only	Submitted + Last	Submitted + First	Max
Falcon Perception													
Qwen 3.5 0.8B	0.8	886	54.90	59.73	71.48	56.06	971	44.8	0.59	67.58	60.45	60.69	70.68
Qwen 3.5 2B	2	1,627	56.36	62.65	73.01	60.93	833	26.8	0.59	69.40	64.02	64.25	73.33
LFM 2.5 VL 3B	3	661	54.66	50.40	70.83	39.51	1,547	85.9	0.86	38.03	39.91	40.22	54.13
Qwen 3.5 4B	4	1,800	55.62	68.21	77.05	64.78	1,364	68.2	7.45	70.15	66.58	67.67	78.09
Gemma 4 E2B	5.1	1,793	55.77	64.07	78.46	56.38	1,800	100.0	1.30	65.77	65.77	65.77	70.71
Gemma 4 E4B	8	1,800	55.62	67.66	78.70	62.19	1,800	94.4	2.04	67.20	67.30	67.61	72.80
Qwen 3.5 9B	9	1,800	55.62	65.58	75.86	62.79	1,295	51.4	1.17	74.41	69.41	69.39	76.59
Qwen 3.8 27B	27	1,800	55.62	71.62	79.51	69.93	1,797	94.8	1.58	75.26	74.45	73.98	77.94
Muse Glimmer 30B	30	1,800	55.62	69.55	80.71	68.63	1,799	81.2	2.17	76.96	71.55	72.40	81.17
Gemma 4 31B	30.7	1,800	55.62	71.38	80.70	69.14	1,778	64.6	2.86	78.31	71.24	71.97	82.11
SAM 3.1													
Qwen 3.5 0.8B	0.8	901	41.59	37.33	57.22	38.86	847	32.6	1.19	70.46	45.61	45.02	73.75
Qwen 3.5 2B	2	1,594	42.35	47.30	62.53	44.69	825	19.9	0.87	71.16	49.81	49.16	74.21
LFM 2.5 VL 3B	3	683	40.48	33.35	52.85	12.73	1,634	90.8	0.61	20.04	21.49	21.64	39.62
Qwen 3.5 4B	4	1,800	42.08	49.19	70.86	47.60	1,224	51.1	7.34	64.23	47.79	51.98	76.13
Gemma 4 E2B	5.1	1,794	42.13	45.13	73.30	37.92	1,799	99.9	1.80	57.87	57.86	57.91	64.95
Gemma 4 E4B	8	1,800	42.08	49.50	74.31	43.19	1,800	93.9	2.73	62.14	61.22	61.70	70.56
Qwen 3.5 9B	9	1,800	42.08	52.14	68.31	44.71	1,369	41.9	1.57	73.62	57.83	55.07	75.90
Qwen 3.8 27B	27	1,800	42.08	50.81	77.41	54.31	1,798	90.3	2.10	76.14	72.20	70.73	77.49
Muse Glimmer 30B	30	1,800	42.08	50.24	75.42	46.50	1,800	64.8	2.23	75.74	60.71	59.35	79.75
Gemma 4 31B	30.7	1,800	42.08	52.62	77.66	49.07	1,780	53.4	2.95	76.72	60.02	60.59	81.44

Results. Fig. 3 summarizes the results across all ten MLLMs. The first rewrite raises mean IoU from 55.54 to 65.09 for Falcon and from 41.90 to 46.76 for SAM 3.1. Further refinement is not monotonic. The final mask scores 61.03 and 41.96, respectively, below the first rewrite for both PSMs. Yet the mean per-trace maximum reaches 76.63 for Falcon and 68.99 for SAM 3.1. The gap between Max and Final shows that the last prompt often loses improvements made earlier in the trajectory. The maximum column is computed by averaging the maximum IoU for each rollout across all 10 refinement steps, or across the ≤ 10 steps in the setting where submission is enabled.

Allowing the MLLM to submit a mask gives 68.31 IoU for Falcon and 64.81 for SAM 3.1 on submitted traces. These means describe only the tasks on which the MLLM chose to submit. Including every attempt with the last-mask fallback in case of no submission gives 65.07 and 53.45 IoU, while the first-mask fallback gives 65.39 and 53.32. Both fallbacks score close to the first rewrite for Falcon and above it for SAM 3.1. The per-dataset comparison is given in Fig. 8 in the appendix.

Larger MLLMs generally produce better refinements. Across all ten models, the Pearson correlation between parameter count and fixed-length Max is 0.78 for Falcon and 0.71 for SAM 3.1. This comparison spans model families and does not isolate the effect of parameter count. Final IoU is highest for the 27B+ group, at 69.24 for Falcon and 49.96 for SAM 3.1 (Fig. 4). These models recover during later refinement steps, but end 1.61 and 1.26 points below their first refinement. Smaller models lose more IoU between steps 1 and 10, with drops of 4.85 and 5.63 points for 4B–9B models and 5.43 and 7.23 points below 4B.

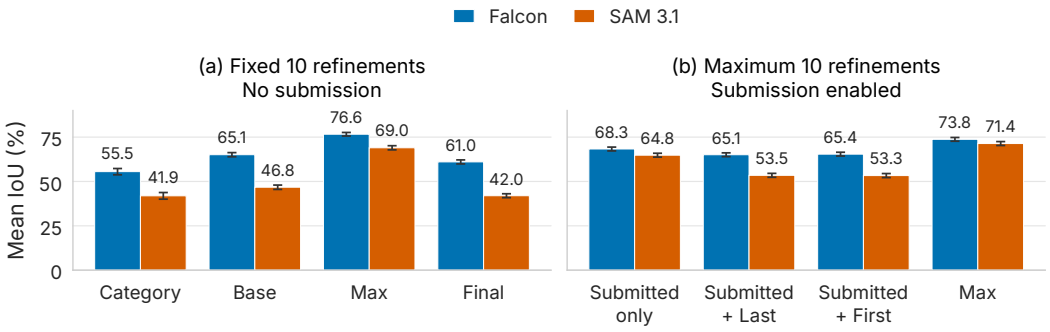

Figure 3: Training-free IoU averaged across all ten MLLMs, with 95% CIs as error bars. Left: category prompting, the first rewrite, the per-trace maximum, and the final mask on accepted fixed-length traces. Right: submission strategies and the per-trace maximum on voluntary submissions.

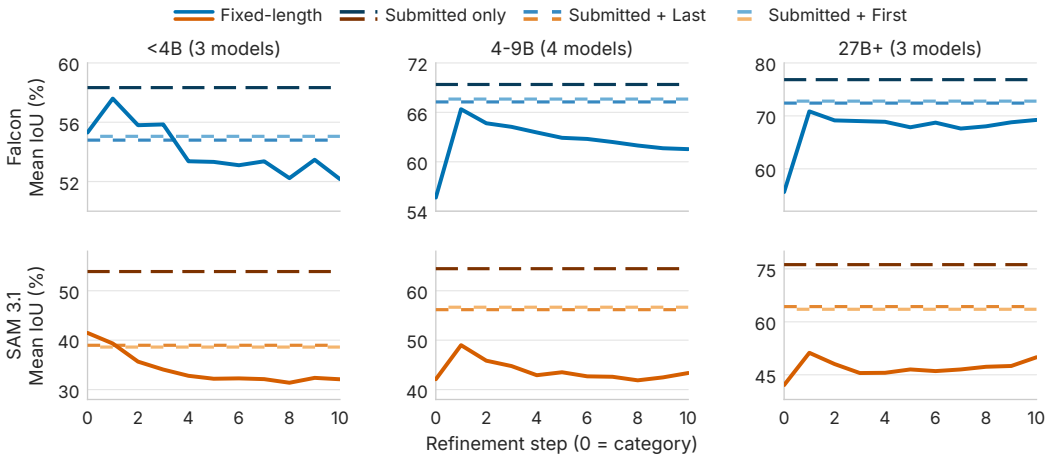

Figure 4: Training-free refinement curves (solid) and final submission-strategy scores (horizontal dashed), weighting MLLMs equally within each size group. Panels separate PSMs and MLLM sizes and use separate y-axis limits. Blue denotes Falcon and orange SAM 3.1, with shades distinguishing strategies. Step zero uses category prompting. Fixed-length scores use accepted complete traces. Submitted only uses voluntary submissions, while both fallback strategies include all attempts.

While there is no monotonous improvement, we observe that within each rollout, there are refinement steps where the MLLM achieves a high IoU, as evidenced by the high average of per-rollout maximum scores. If we were able to reliably select the maximum, we could therefore substantially improve the segmentation quality. Next, we explore whether we can get closer to the maximum IoU score by learning to directly output the optimal prompt $p^* \in \mathcal{P}_{\text{text}}$ using RL training of π_θ.

5 Training Prompt Optimizers with Reinforcement Learning

We have shown that performance of PSMs can be substantially improved through optimizing the text prompt. However, this has two disadvantages: (1) As we have seen, improvement is not monotonous, so we need an additional mechanism to select the best mask from each trajectory. (2) Iterative optimization is expensive, as it requires multiple MLLM inferences for each image-object pair. To address both problems simultaneously, we train MLLM policies to produce a single prompt refinement, i.e., a fine-tuned MLLM receives the image and the object name, and directly outputs the optimized prompt. This is faster, because we only need one inference, and we do not have to select from multiple options. To train the MLLM policies, we use Falcon Perception and SAM 3.1 as separate environments with distinct prompt spaces $\mathcal{P}_{\text{text}}$ to be explored, because which kind of prompt is optimal depends on the particular PSM.

Method. We train the MLLM policy π_θ with Group-Relative Policy Optimization (GRPO; Shao et al. (2024)) using IoU as the reward. During training, the segmentation model S remains fixed, and the policy receives the image x, the category prompt p, and the overlay $x \odot m$ of the mask $m = S(x, p)$. It proposes one rewritten prompt p', which the segmentation model maps to a mask $m' = S(x, p')$. The IoU of m' against the ground-truth annotation then provides the training reward. Ground-truth masks and numerical rewards are never included in the policy's input.

Training and Evaluation. Each run starts from the pretrained MLLM and uses 3,000 training tasks, with 500 from each dataset as described in Sec. 3. We train each model for 100 RL steps, where a single RL step has a batch size of 256 rollouts, with eight rollouts for each of 32 tasks. After training, all policies are evaluated on the same 100 held-out tasks per dataset, and we report the final checkpoint at RL step 100 (we provide the training curves in Fig. 9). As baselines, we compare to category prompting and the base MLLM policy before RL tuning, with the same inputs and instruction for the base and trained policies (Sec. B). For MLLMs, we train Qwen3.5 0.8B, Qwen3.5 2B, and Qwen3.5 4B.

For evaluation, our primary metric assigns zero score to malformed outputs, empty prompts, and repetitions of the initial prompt. To assess the effect of penalizing repetitions, we also report scores that assign repeated prompts the IoU of the category mask. Confidence intervals for RL comparisons describe variation across test images for one training run per condition. Additional technical details on training and evaluation are in Sec. A.

Table 2: Single-turn IoU (%) on 600 held-out tasks. Base is a single rewrite from the pretrained policy, and RL is the policy after 100 RL steps. Δ cat. and Δ base report RL minus category prompting and the pretrained rewrite, respectively, in IoU points.

	Falcon Perception				SAM 3.1			
Policy	Base	RL	Δ cat.	Δ base	Base	RL	Δ cat.	Δ base
Qwen 3.5 0.8B	51.10	63.35	+5.90	+12.25	30.89	33.35	-8.42	+2.46
Qwen 3.5 2B	61.53	67.08	+9.63	+5.55	48.08	53.18	+11.41	+5.10
Qwen 3.5 4B	69.02	70.17	+12.72	+1.15	51.39	54.40	+12.63	+3.01
Category only	—	57.45	—	—	—	41.77	—	—

Results. When 3.5 4B achieves the highest final IoU with both PSMs, reaching 70.17 with Falcon and 54.40 with SAM 3.1 (Tab. 2). These scores improve on direct category prompting by 12.72 and 12.63 points, and on the pretrained rewrite by 1.15 and 3.01 points, respectively. All three policies improve over their own base scores. All three also exceed category prompting with Falcon, and two do so with SAM 3.1. Fig. 2 shows four selected Falcon examples with valid rewrites from both policies. The shiitake and pasta masks move to the intended objects, while the steak and kiwi masks exclude surrounding regions. These results show that RL-tuning can teach MLLMs to efficiently propose prompts that effectively locate objects and are adjusted to the preferences of the respective segmentation model.

5.1 Ablation: Leave-One-Out Generalization under RL

To test whether learned refinements transfer to datasets excluded from RL training, we train six Qwen3.5 0.8B MLLM policies with Falcon as the PSM, each leaving out one dataset from RL training, which we then evaluate the MLLM policy on. This tests whether the MLLM policy can learn prompting strategies that generalize across datasets. Each policy starts from the base MLLM and receives 40 training tasks from each of the other five datasets, resulting in 200 total tasks for RL training. We train for 50 RL steps using GRPO and evaluate mask quality on 100 tasks from the omitted dataset. We compare category prompting, a native rewrite from the pretrained MLLM before RL, and the rewritten prompt from the trained policy.

RL improves over the standard rewrite on five of the six omitted datasets (Fig. 5). Gains range from 4.25 points on FoodSeg103 to 18.12 on CUB, while Fashionpedia decreases from 56.06 to 43.29. The trained policies exceed category prompting on CUB, D2S, and Flowers, but remain below it on Fashionpedia, FoodInsSeg, and FoodSeg103 (Tab. 6). These datasets are excluded from RL training, although we cannot rule out exposure during MLLM or PSM pretraining, as the corresponding

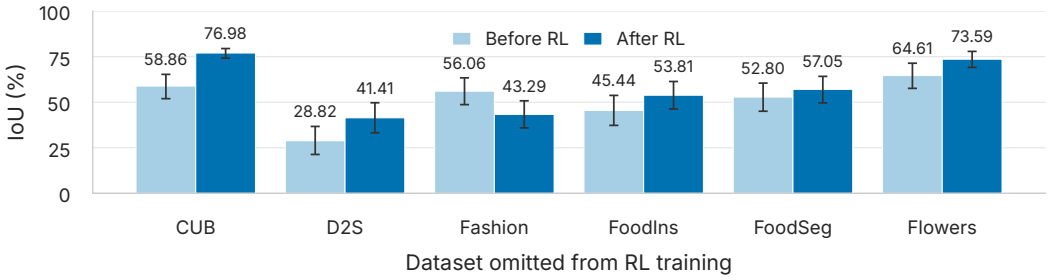

Figure 5: Generalization to datasets excluded from RL training. Bars show Falcon IoU before and after RL. We train Qwen3.5 0.8B for 50 RL steps on the other five datasets and evaluate them on 100 tasks from the omitted dataset. Error bars show 95% CIs. Full results are in Tab. 6.

datasets are not public. FoodInsSeg and FoodSeg103 also share source photographs and ingredient categories, making these two task domains a test of transfer between closely related annotation tasks.

5.2 Ablation: No Hint vs. Hint on Prompting Style

For some models, certain information about the preferred prompting style is provided by the model developer. We evaluate whether guidance on prompt wording improves RL for all three policies with SAM 3.1. We add a request to use short, specific noun phrases, which is the official recommendation for prompting, with "red car" and "person in blue shirt" as examples, and append this hint to the prompt as a single sentence. The hint contains no dataset annotations and is included during RL training and in all evaluations, including that of the base policy. Both conditions use the same training pool, test tasks, and budget of 100 RL steps, and we train Qwen3.5 0.8B, 2B, and 4B.

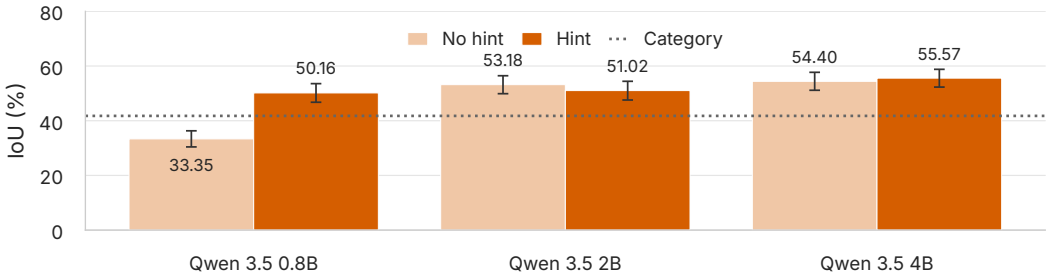

Figure 6: Final SAM 3.1 test IoU with and without a noun-phrase hint for each policy, with 95% confidence intervals from resampling test photographs. The hint is present throughout training and evaluation. The grey reference gives category-only IoU. Paired differences and checkpoint curves are in Tab. 7 and Fig. 10.

Providing the hint has the largest effect on Qwen 3.5 0.8B (Fig. 6). Its base IoU increases by 5.82 points with the hint. After RL, it reaches 50.16 IoU with the hint, compared with 33.35 without it, a difference of 16.81 points (95% confidence interval: 13.23–20.42). The final differences for the other two policies range from -2.17 to $+1.17$ points, and both intervals include zero. These results indicate that hints might help smaller models to explore the prompt space more efficiently, but do not show a consistent benefit across policies.

6 Self-Training the Segmentation Model

In Sec. 4 and Sec. 5, we have shown that we can generate high-quality masks for image-object pairs only using an MLLM. Critically, the masks we obtain from our methods are generally better than category prompting. This naturally motivates an application where we distill the better masks to the segmentation model, so that even for category prompting, the model outputs a better mask. This eliminates the prompt optimization step and instead improves the segmentation model directly.

Method. For an original prompt p and a refined prompt p', the segmentation model S produces masks $m = S(x, p)$ and $m' = S(x, p')$, respectively. We then fine-tune S to predict the refined mask m' when prompted with the original prompt p, i.e., the unoptimized prompt or class name.

We distinguish three sources of training masks: *category masks* from the class name alone, *base rewrite masks* from one rewrite by the pretrained MLLM before RL training, and *RL masks* from one rewrite by the same MLLM after RL training. All three use the same frozen Falcon Perception model to predict masks from the resulting prompts. For the rewrites, the MLLM receives the image, category name, and the mask predicted from that category. Each mask source defines the targets for fine-tuning a separate Falcon student. The students receive the original category prompts during training and evaluation, so the refined wording is not needed at test time.

Training and Evaluation. As MLLM policy, we use Qwen3.5 0.8B before RL training and at RL step 100 to generate the two sets of rewritten prompts (in addition to the raw class names). For each dataset, we distill a separate student segmentation model with the same initialization, optimizer settings, category prompts, and 200-step training budget. This yields three distilled segmentation models per dataset, one for each supervision type (category, non-RL, RL prompts). All student models are then evaluated by IoU on the same 100 held-out tasks per dataset.

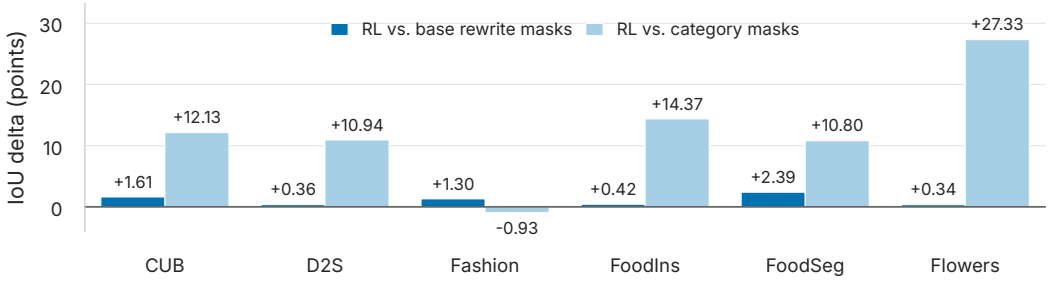

Figure 7: Test IoU gains for Falcon. Bars compare the RL-mask student with students trained on base rewrite masks or category masks, evaluated at training step 200 on the same 100 held-out tasks per dataset. All evaluations use category prompts. Failed base rewrites are excluded from training.

Results. Across the six datasets, RL-mask students improve mean IoU by 1.07 points over base-rewrite students (95% confidence interval: 0.50–1.74) and by 12.44 points over category-mask students (10.11–14.87). Their mean IoU is higher than that of base-rewrite students on all six datasets (Fig. 7). They exceed category-mask students on five datasets, with a decrease of 0.93 points on Fashionpedia. The mean IoUs are 66.20 for RL masks, 65.13 for base rewrite masks, and 53.76 for category masks. Full results and technical details are in Sec. I, and results for fine-tuning with ground-truth masks in Sec. H.

7 Conclusion

We have demonstrated self-improvement in promptable segmentation models. MLLMs can optimize text prompts to produce substantially better masks, and these improved masks can be distilled back into the segmentation model without ground-truth annotations. Training-free iterative refinement confirms that better prompts unlock higher segmentation quality, with per-trace maxima exceeding category prompting by over 20 IoU points. GRPO-tuned MLLM policies compress this iterative process into one-shot generation, and achieve direct category prompting for both Falcon and SAM 3.1, though gains depend on the dataset, policy, and required output format. These learned refinements generalize to five of six datasets omitted from RL training. In the self-distillation experiments, Falcon students trained on refined masks improve over students trained on category-prompt masks in five of six datasets, which closes the self-improvement loop. Together, these results establish a label-free perspective to adapt promptable segmentation models to new domains. Future work can build upon our work by extending the loop to additional modalities and domains, scaling the RL policies, and investigating whether distilled improvements can themselves seed further rounds of refinement, moving toward fully autonomous self-improvement.

AI USE STATEMENT

In this work, we used generative AI tools for generating synthetic data sets, designing or providing feedback on research methodology or experiments, implementing methods, supporting qualitative and thematic data analysis, and interpreting results. We have not used generative AI tools for formulating mathematical claims, assisting with translation, helping develop theoretical models or conceptual frameworks, proposing or refining hypotheses, and cleaning and reformatting datasets. Assisting in the writing of proofs is not applicable to this work. Beyond this, we used generative AI tools for suggesting experimental parameters, creating or editing software code, drafting parts of the paper, brainstorming, searching for information, and identifying relevant literature. That being said, we checked and revised all AI-generated output, including code, and we read all cited literature ourselves. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

This work uses publicly available data for all evaluations (CUB-200-2011, D2S, Fashionpedia, FoodInsSeg, FoodSeg103, Oxford Flowers 102), and all models (SAM3.1, Falcon Perception) as well as MLLMs are open-weight. Our paper contains all instructions and prompts necessary to understand and reproduce our results. Additional artifacts and all code will be released upon acceptance.

REFERENCES

Aviraj Bevli, Sofian Chaybouti, Yasser Dahou, Hakim Hacid, Ngoc Dung Huynh, Phuc H Le Khac, Sanath Narayan, Wamiq Reyaz Para, and Ankit Singh. Falcon perception. *arXiv preprint arXiv:2603.27365*, 2026.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *ICLR*, 2026.

Yumin Choi, Dongki Kim, Jinheon Baek, and Sung Ju Hwang. Multimodal prompt optimization: Why not leverage multiple modalities for mllms. In *ICLR*, 2026.

Zi-Yi Dou, Cheng-Fu Yang, Xueqing Wu, Kai-Wei Chang, and Nanyun Peng. Re-rest: Reflection-reinforced self-training for language agents. In *EMNLP*, 2024.

Tianyuan Du, Haopeng Li, Zhen Fan, Jiarui Zhang, Panwang Pan, and Yang Zhang. Sam-veteran: An mllm-based human-like sam agent for reasoning segmentation. In *ICLR*, 2026.

Chrisantha Fernando, Dylan Sunil Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: Self-referential self-improvement via prompt evolution. In *ICML*, 2024.

Patrick Follmann, Tobias Böttger, Philipp Härtinger, Rebecca König, and Markus Ulrich. MVTec D2S: Densely segmented supermarket dataset. In *ECCV*, 2018.

Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *ECCV*, 2022.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. Reinforced self-training (rest) for language modeling. In *arXiv*, 2023.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *ICLR*, 2024.

Jiaqi Huang, Zunnan Xu, Jun Zhou, Ting Liu, Yicheng Xiao, Mingwen Ou, Bowen Ji, Xiu Li, and Kehong Yuan. Sam-r1: Leveraging sam for reward feedback in multimodal segmentation via reinforcement learning. In *NeurIPS*, 2025.

Menglin Jia, Mengyun Shi, Mikhail Sirotenko, Yin Cui, Claire Cardie, Bharath Hariharan, Hartwig Adam, and Serge Belongie. Fashionpedia: Ontology, segmentation, and an attribute localization dataset. In *ECCV*, 2020.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023.

Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *CVPR*, 2024.

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

Yuqi Liu, Bohao Peng, Zhisheng Zhong, Zihao Yue, Fanbin Lu, Bei Yu, and Jiaya Jia. Seg-zero: Reasoning-chain guided segmentation via cognitive reinforcement. In *arXiv*, 2025.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. In *EMNLP*, 2023.

Xiangyan Qu, Gaopeng Gou, Jiamin Zhuang, Jing Yu, Kun Song, Qihao Wang, Yili Li, and Gang Xiong. Proapo: Progressively automatic prompt optimization for visual classification. In *CVPR*, 2025.

Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S Khan. Glamm: Pixel grounding large multimodal model. In *CVPR*, 2024.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *ICLR*, 2025.

Laria Reynolds and Kyle McDonell. Prompt programming for large language models: Beyond the few-shot paradigm. In *CHI Extended Abstracts*, 2021.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. In *arxiv*, 2024.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, et al. Beyond human data: Scaling self-training for problem-solving with language models. In *TMLR*, 2024.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Zilin Wang and Stella X Yu. Vision harnessing agent for open ad-hoc segmentation. In *arXiv*, 2026.

Xiongwei Wu, Xin Fu, Ying Liu, Ee-Peng Lim, Steven C. H. Hoi, and Qianru Sun. A large-scale benchmark for food image segmentation. In *ACM International Conference on Multimedia*, 2021.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *ICLR*, 2024.

Kai Ye, Xiaotong You, Jianghang Lin, Jiayi Ji, Pingyang Dai, and Liujuan Cao. Evolving, not training: Zero-shot reasoning segmentation via evolutionary prompting. In *arXiv*, 2025.

Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. In *NeurIPS*, 2023.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. Self-rewarding language models. In *ICML*, 2024.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. In *NeurIPS*, 2022.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *ICLR*, 2023.

Lianghui Zhu, Bin Ouyang, Yuxuan Zhang, Tianheng Cheng, Rui Hu, Haocheng Shen, Longjin Ran, Xiaoxin Chen, Li Yu, Wenyu Liu, et al. Lens: Learning to segment anything with unified reinforced reasoning. In *AAAI*, 2026a.

Muzhi Zhu, Yuzhuo Tian, Hao Chen, Chunluan Zhou, Qingpei Guo, Yang Liu, Ming Yang, and Chunhua Shen. Segagent: Exploring pixel understanding capabilities in mllms by imitating human annotator trajectories. In *CVPR*, 2025.

Qipeng Zhu, Yanzhe Chen, Huasong Zhong, Jie Chen, Yan Li, Zhixin Zhang, Junping Zhang, and Zhenheng Yang. Uniapo: Unified multimodal automated prompt optimization. In *AAAI*, 2026b.

Supplementary Material

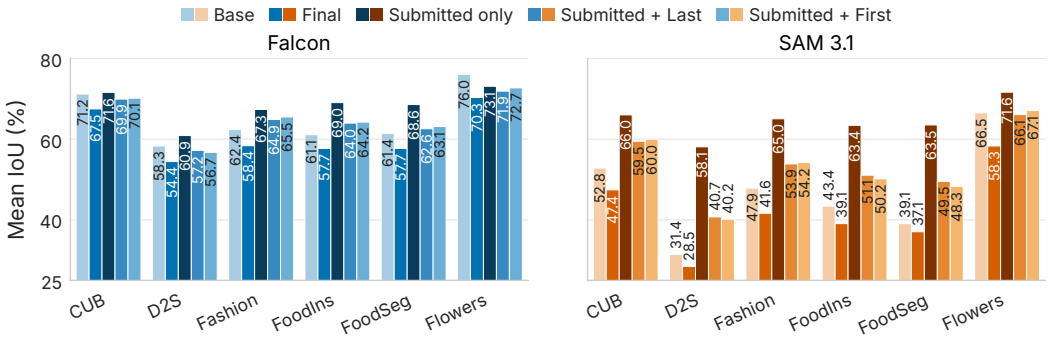

Figure 8: Training-free IoU by dataset, averaged equally across all ten MLLMs. Base uses the first rewrite (step 1) and Final uses step 10. Fixed-length scores use accepted complete traces. Submitted only uses voluntary submissions, while both fallback strategies include all attempts. Blue denotes Falcon and orange SAM 3.1, with shades distinguishing strategies. The y-axis spans 25–80% IoU.

A Additional Evaluation Details

Single-turn training and evaluation. Each of the six main runs starts from the corresponding pretrained VLM and uses LoRA with rank 64 and scaling factor 64, AdamW with learning rate 5×10^{-6}, and 100 RL steps with GRPO. Each RL step uses a batch of 256 rollouts, with eight rollouts for each of 32 sampled tasks. There are 500 distinct training tasks from each of the six datasets. Training, test-task selection, and resampling use fixed random seeds. Greedy checkpoint evaluations use fixed subsets of 100 training and 100 held-out tasks per dataset at RL steps 0, 10, . . ., 100. The training curves therefore describe a training subset, not held-out performance.

We compute 95% confidence intervals by drawing 10,000 new samples from the evaluated photographs, with replacement. Tasks from the same photograph stay together, including shared FoodInsSeg/FoodSeg103 images, and each sample uses the same tasks for both conditions. We take the 2.5th and 97.5th percentiles of the resulting mean differences. Test photographs are excluded from training. Each condition has one training run, so these intervals measure how the estimate changes with the sampled test images. They do not measure variation between training runs.

Error bars on absolute RL and hint scores apply the same procedure to mean IoU. For the training-free averages, we draw 300 tasks with replacement within each dataset, using the same draw for every MLLM and PSM. In each draw we apply each column's trace selection, recompute each model's mean, and average the ten models equally. We report the 2.5th and 97.5th percentiles over 10,000 draws. The evaluated models remain fixed.

Output formats. Qwen 3.5 2B and 4B use a strict single-key JSON object containing the prompt. Qwen 3.5 0.8B also accepts that object inside one complete markdown JSON code block. The same format is used before RL, during training, and after RL for each policy. The repeated-prompt sensitivity in Tab. 4 keeps malformed and empty outputs at zero and assigns repeated category prompts the score of their cached mask.

B Prompt Instructions

The instruction blocks below reproduce the prompts in the code. Braced fields in the task messages are filled separately for each image. Category prompts replace each run of underscores or dashes with a space and collapse repeated whitespace.

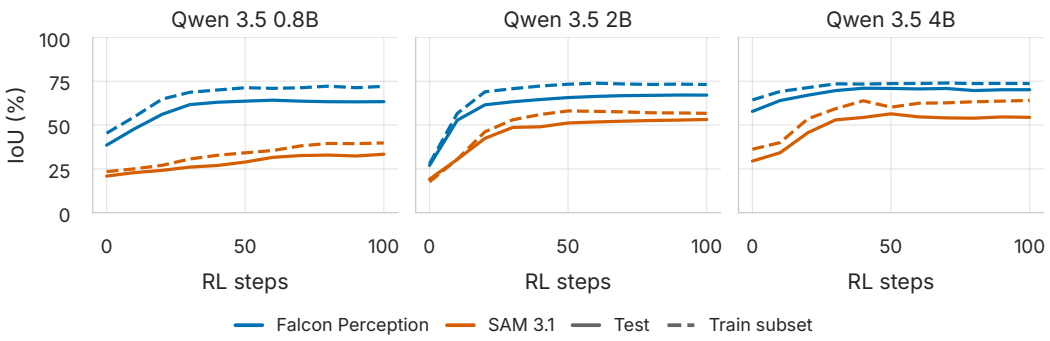

Figure 9: Single-turn checkpoint IoU. Blue denotes Falcon and orange SAM 3.1 throughout the paper. Solid lines use held-out tasks and dashed lines the fixed training subset. Each curve averages 600 tasks. The final checkpoint is fixed at RL step 100.

Training-free refinement. All ten MLLMs receive the same system instruction below with both PSMs in Sec. 4. The first refinement receives the original image, target category, category prompt, and its predicted mask overlay. Later refinements retain the full conversation and all previous overlays.

> You are a visual segmentation prompt-refinement agent. The immutable target class and the original image are supplied below. Inspect every segmentation overlay in the complete conversation history. In an overlay, translucent colored regions with numbered contours are exactly the instances selected by the segmenter; uncolored content is not selected, and a red NO MASK DETECTED notice means nothing was selected. Compare the latest selection with the original image and preserve the immutable target's semantic intent while correcting omissions, excess regions, or prompt ambiguity. Call exactly one available tool per turn. Use segment with a new, nonempty textual expression when the latest mask can be improved; use finish only when the latest selection matches the intended target as closely as you can judge. The latest mask is always selected. You cannot select an earlier mask except by calling segment again with that earlier prompt. Think as thoroughly as useful before your tool call.
>
> This trajectory is a fixed-horizon systems evaluation. Until 11 segmentation results exist, finish is deliberately unavailable even if the latest mask already looks excellent. Use a distinct, semantically faithful diagnostic reformulation of the immutable target (for example, vary specificity, visible attributes, extent, or scene relation) and inspect how the new overlay changes. Never repeat the immediately preceding prompt. After result 11, call finish with your assessment.

The initial user message supplies the target fields and original image.

```
Immutable target class: {class_label}
Immutable initial prompt: {initial_prompt}
Original RGB image:
```

The `segment` tool takes one string field, `prompt`, and returns the resulting segmentation overlay. It is the only available tool during the ten refinements. After the eleventh mask, including the category mask, only `finish` is available. It takes a string field, `reason`, and selects the latest mask. Generation uses a 1,024-token response budget and temperature zero. Incomplete traces are excluded from the reported means.

Single-turn rewrite and RL. The base policy and the RL-trained policy in Sec. 5 receive the same instruction below. It is also used during RL training, for both Falcon and SAM 3.1. It forms the start of a single user message.

> You are a visual segmentation prompt-refinement policy. The immutable target class, original image, label-derived initial prompt, and its baseline segmentation overlay are supplied. Ground truth and metrics are unavailable. Inspect what the baseline selected and produce one improved, nonempty segmentation prompt faithful to the target. Do not repeat the initial prompt. Return exactly one JSON object with one key, prompt: {"prompt": "your

> refined segmentation prompt"}. Do not include reasoning, markdown, or any other text.
> This JSON invokes segment once on the original image and ends the episode.

The instruction is followed by the original image, the text "Original RGB image.", the category-mask overlay, and the following task-specific text.

```
Baseline segmentation overlay.
Immutable target class: {class_label}
Immutable initial prompt: {initial_prompt}
Return the JSON object containing the best refined prompt.
```

The JSON response supplies the prompt for one segmentation call. Qwen 3.5 2B and 4B require the JSON object directly. Qwen 3.5 0.8B also accepts one complete JSON code fence. These rules are fixed before and after RL.

Prompting-style hint. For Sec. 5.2, the following sentence is appended to the single-turn instruction during both RL training and evaluation, including evaluation of the base policy.

> Use short, specific noun phrases that clearly name the visual object or region you want segmented, e.g. "red car" or "person in blue shirt."

C Single-Turn Results by Dataset

Tab. 3 reports final-checkpoint primary scores for each dataset, alongside the direct category baseline. The task-level IoU average is distinct from a class-macro semantic-segmentation benchmark score. Tab. 4 reports acceptance and paired base-to-RL gains, including the scores with repeated prompts included. A policy can improve over its base score and still perform worse than category prompting.

Tab. 5 compares the checkpoint evaluations at RL steps 0 and 100 on the same tasks where both produce an accepted, non-repeated action. Valid outputs with empty masks or zero IoU remain included. On the 541 retained Falcon tasks for Qwen 3.5 4B, the mean gain is 7.10 points, compared with 12.39 over all tasks in these checkpoint evaluations. This comparison measures changes among valid refinements, but selecting tasks by the policies' actions can change task difficulty. It does not isolate how much of the overall gain is caused by better formatting. The paired intervals use the photograph resampling procedure in Sec. A, restricted to the retained tasks.

Table 3: Final single-turn IoU by dataset, with 100 held-out tasks in each column. Top: Falcon Perception. Bottom: SAM 3.1.

Falcon Perception

Policy	CUB	D2S	Fashion	FoodIns	FoodSeg	Flowers
Qwen 3.5 0.8B	76.26	49.51	54.99	61.93	62.21	75.22
Qwen 3.5 2B	77.68	55.35	61.63	68.22	64.96	74.62
Qwen 3.5 4B	77.71	72.42	61.82	67.12	68.30	73.65
Category only	61.23	34.42	70.90	61.87	65.80	49.33

SAM 3.1

Policy	CUB	D2S	Fashion	FoodIns	FoodSeg	Flowers
Qwen 3.5 0.8B	53.88	19.31	8.89	23.06	24.38	70.58
Qwen 3.5 2B	61.57	30.68	45.68	55.79	51.92	73.47
Qwen 3.5 4B	53.48	38.26	52.35	56.00	52.61	73.69
Category only	54.54	13.66	48.20	43.30	46.10	44.81

D Leave-One-Out Results

Each omitted-dataset policy starts from the base Qwen 3.5 0.8B model. Its 200 training tasks come from a shared pool of 40 tasks per included dataset, with evaluation images excluded. Training runs

Table 4: Action acceptance (%) and paired IoU changes in the checkpoint evaluations on all 600 test tasks. Base and RL denote RL steps 0 and 100. Δ repeats uses the scores with repeated prompts included. The intervals reflect variation across test photographs for the recorded run.

PSM	Policy	Base acc.	RL acc.	Δ IoU	95% CI	Δ repeats
Falcon	Qwen 3.5 0.8B	96.7	100.0	+24.85	[+21.69, +28.07]	+24.85
Falcon	Qwen 3.5 2B	37.5	99.8	+40.14	[+36.62, +43.66]	+20.86
Falcon	Qwen 3.5 4B	90.2	100.0	+12.39	[+9.78, +15.01]	+9.01
SAM 3.1	Qwen 3.5 0.8B	96.3	99.8	+12.50	[+9.77, +15.17]	+12.56
SAM 3.1	Qwen 3.5 2B	38.8	99.7	+34.08	[+30.52, +37.71]	+21.38
SAM 3.1	Qwen 3.5 4B	89.8	99.0	+24.99	[+21.25, +28.72]	+23.26

Table 5: Single-turn IoU on tasks with accepted actions in both checkpoint evaluations. Paired n is the number retained from the same 600 held-out tasks for each condition. Base and RL denote RL steps 0 and 100. Repeated category prompts and malformed actions are excluded from both sides of each pair. Valid zero-IoU masks are retained. The intervals describe variation across the retained test photographs.

Policy	Paired n	Base	RL	Δ IoU	95% CI
Falcon Perception					
Qwen 3.5 0.8B	580	39.83	63.94	+24.11	[+21.03, +27.31]
Qwen 3.5 2B	225	71.84	73.80	+1.96	[-1.16, +5.16]
Qwen 3.5 4B	541	64.08	71.18	+7.10	[+4.86, +9.41]
SAM 3.1					
Qwen 3.5 0.8B	577	21.68	33.88	+12.19	[+9.47, +14.90]
Qwen 3.5 2B	233	49.19	61.96	+12.77	[+7.76, +17.83]
Qwen 3.5 4B	533	32.64	57.41	+24.77	[+20.86, +28.75]
SAM 3.1 with hint					
Qwen 3.5 0.8B	565	28.33	51.07	+22.74	[+19.05, +26.50]
Qwen 3.5 2B	357	51.51	55.95	+4.44	[+0.63, +8.32]
Qwen 3.5 4B	584	43.50	55.39	+11.89	[+8.38, +15.42]

for 50 RL steps with 256 rollouts per RL step. The corresponding 100 test tasks for the omitted dataset are held fixed. Tab. 6 compares direct category prompting, the native rewrite before RL, and the trained policy at RL step 50. Each difference subtracts the corresponding baseline from the after-RL score.

Table 6: Falcon leave-one-out test IoU (%) on 100 tasks per omitted dataset. Before RL uses a native rewrite from pretrained Qwen 3.5 0.8B. After RL uses the policy trained on the other five datasets for 50 RL steps. Δ cat. and Δ rewrite give the gains over category prompting and the native rewrite in IoU points.

Omitted dataset	Category	Before RL (native rewrite)	After RL	Δ cat.	Δ rewrite
CUB-200	62.08	58.86	76.98	+14.90	+18.12
D2S	35.26	28.82	41.41	+6.15	+12.59
Fashionpedia	70.56	56.06	43.29	-27.27	-12.77
FoodInsSeg	62.32	45.44	53.81	-8.51	+8.37
FoodSeg103	65.14	52.80	57.05	-8.09	+4.25
Flowers 102	49.35	64.61	73.59	+24.24	+8.98

E PROMPTING-STYLE HINT RESULTS

Tab. 7 compares the final policies trained with and without the noun-phrase hint. The hint also changes the base-policy scores, before training, so a smaller base-to-RL gain does not by itself imply a worse final policy. We report the final difference and the base difference separately. Each

comparison uses the same 600 test tasks. Fig. 10 gives checkpoint curves for all three policies, and Tab. 8 gives final differences by dataset.

Table 7: SAM 3.1 IoU with and without the prompting-style hint. The first two columns give final-checkpoint scores. Differences are hint minus no hint, with a paired interval for the final difference.

Policy	No hint	Hint	Δ final	95% CI	Δ base
Qwen 3.5 0.8B	33.35	50.16	+16.81	[+13.23, +20.42]	+5.82
Qwen 3.5 2B	53.18	51.02	-2.17	[-4.96, +0.63]	+11.69
Qwen 3.5 4B	54.40	55.57	+1.17	[-2.05, +4.36]	+12.93

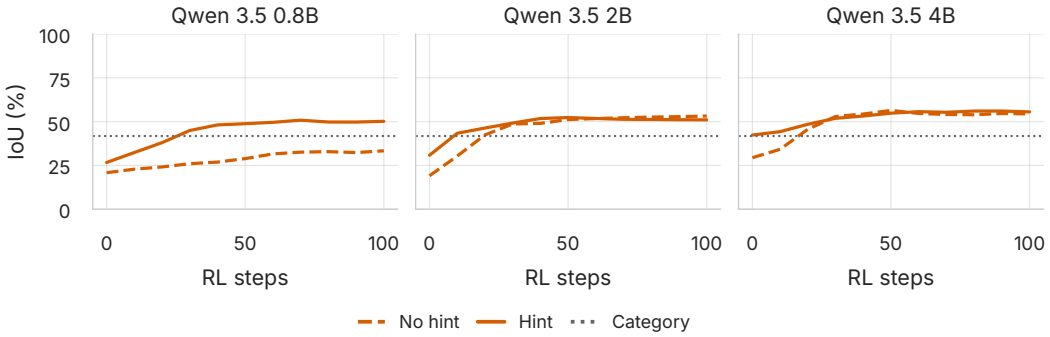

Figure 10: SAM 3.1 test IoU during RL with and without the noun-phrase hint. Grey lines give category-only IoU. Both conditions use fixed checkpoint evaluations on the same 600 test tasks.

Table 8: Final SAM 3.1 IoU differences by dataset, with hint minus without hint. Each cell uses 100 matched test tasks.

Policy	CUB	D2S	Fashion	FoodIns	FoodSeg	Flowers
Qwen 3.5 0.8B	+5.60	+8.67	+27.56	+27.10	+26.21	+5.71
Qwen 3.5 2B	-5.66	+1.76	-1.42	+1.10	-9.24	+0.47
Qwen 3.5 4B	+6.66	+0.29	-2.71	+2.07	+0.67	+0.03

F Prompt Changes after RL

Sample and review. We select 25 of the 100 held-out task IDs in each dataset by sorting the SHA-256 hashes of a fixed seed, dataset name, and task ID. The same 150 tasks are used for both PSMs, giving 300 prompt pairs from 149 photographs. The base and RL prompts come from the fixed evaluations at RL steps 0 and 100 of Qwen 3.5 4B.

We categorize the changes using GPT-6 Astra on `"max"` reasoning effort. We use the original category prompt and both extracted policy prompts. Labels describe wording only. The review does not inspect the images or judge whether an attribute is visually correct.

Edit definitions. *Object naming* changes the object noun or class name, including broader or narrower terms. Spelling changes alone do not count. *Appearance* adds, removes, or changes a target's color, shape, material, texture, size, or visible state. Synonyms and photographic requests such as "sharp focus" do not count. *Position / relations* changes the target's location, orientation, or relation to another object. It includes removing such a cue. *Target extent* changes explicit coverage of the whole object, parts, packaging or contents, or an object group.

Shortening requires at most 60% of the base prompt's word count and at least eight fewer words. Words contain letters or numbers, with optional internal apostrophes or hyphens. *Instruction removal* requires a segmentation, focus, inclusion, exclusion, or boundary request in the base prompt and none in the RL prompt. The code specifies the matching expressions. Descriptive phrases such as

"including its wings" are not counted as requests. *Other wording* contains accepted pairs with none of the six edit labels. A pair with a rejected endpoint receives no edit labels and stays in the separate rejected group. An accepted action that predicts an empty mask remains eligible for edit labels.

Table 9: Results by prompt-change group in the fixed 300-pair sample. Cat., Base, and RL are mean IoUs for the same tasks within each row. Improved counts pairs with higher RL than base IoU. The six edit groups overlap. Other wording and Rejected endpoint are separate groups. All selected and Both accepted are reference totals.

Change	Pairs	Cat.	Base	RL	Δ	Improved
Falcon Perception						
All selected	150	65.95	60.64	74.12	+13.48	77/150
Both accepted	137	67.49	66.39	75.56	+9.17	66/137
Object naming	22	67.25	61.43	73.21	+11.78	8/22
Appearance	106	68.00	71.10	78.46	+7.36	52/106
Position / relations	57	63.40	57.86	72.24	+14.38	33/57
Target extent	64	63.20	60.46	73.74	+13.28	34/64
Shortening	43	69.36	64.71	80.50	+15.79	24/43
Instruction removal	29	60.99	69.74	76.64	+6.90	14/29
Other wording	5	85.46	67.20	67.18	-0.02	2/5
Rejected endpoint	13	49.69	0.00	58.95	+58.95	11/13
SAM 3.1						
All selected	150	45.47	32.77	58.41	+25.64	86/150
Both accepted	136	46.64	36.14	61.08	+24.94	79/136
Object naming	33	38.51	28.01	63.30	+35.29	22/33
Appearance	102	47.94	40.78	63.00	+22.22	63/102
Position / relations	71	48.49	26.29	57.46	+31.17	42/71
Target extent	66	48.96	35.95	62.11	+26.17	39/66
Shortening	62	43.91	30.73	62.58	+31.85	35/62
Instruction removal	34	36.71	28.59	61.41	+32.81	20/34
Other wording	2	0.00	35.21	64.99	+29.78	1/2
Rejected endpoint	14	34.02	0.00	32.39	+32.39	7/14

Interpretation. The sample contains 137 accepted Falcon pairs and 136 accepted SAM pairs. Falcon improves on 66 of these pairs, ties on three, and worsens on 68. SAM improves on 79, ties on 26, and worsens on 31. A mean gain can therefore coexist with frequent regressions. For example, the Falcon object-naming group gains 11.78 points on average, but only eight of its 22 pairs improve.

G Qualitative Mask Examples

Fig. 11 and Tab. 11 use the first test task ID in CUB, D2S, and Fashionpedia, shared across the two Gemma runs.

H Supervised Falcon Adaptation

The mask-head controls fine-tune 2,361,344 Falcon parameters using annotated masks, with the remaining model frozen. D2S uses 450 training, 50 validation, and 100 test images, with 200 AdamW training steps, batch size four, and learning rate 10^{-5}. CUB uses 5,000 training, 500 validation, and 500 test images, with 1,000 training steps and effective batch size 16 at the same learning rate. Each dataset uses a fixed random seed. For both datasets, the validation-selected checkpoint is the final checkpoint. Their budgets and data splits differ, so the gains do not isolate a dataset effect.

Training uses ground-truth boxes and masks with a mask loss comprising 200 times focal loss plus ten times soft Dice loss. Evaluation uses native predictions from literal category prompts without ground-truth boxes or an RL policy. CUB improves by 6.15 IoU points on the held-out test set. The D2S estimate is smaller, at 1.23 points, and its confidence interval includes zero (Tab. 12). These supervised controls use assigned masks. The generated-mask experiment is reported in Sec. I.

Table 10: Additional pairs from the same sample, with verbatim prompts and IoU below each. The first two improve after RL. The last two regress. All actions are accepted, including those with zero IoU.

Change / PSM	Category	Base policy	RL policy
Added visual details Falcon / FoodIns	popcorn **57.64**	popcorn kernels, white and puffed, mixed with other snacks on a blue plate **58.43**	white fluffy popcorn kernels clustered together, irregularly shaped, some coated with caramel or sugar, distinct from fruit and cereal pieces **85.53**
Entire bird SAM 3.1 / CUB	Pacific Loon **0.00**	Pacific Loon, a water bird with a long pointed beak, floating on calm green water, displaying a grey head and neck with a white throat, and a dark brown body with white speckled patterns on its back and wings. **0.00**	Pacific Loon, the entire bird including its head, neck, speckled brown and white body, and long pointed beak, floating on water **74.18**
Changed target Falcon / Fashion	tassel **83.80**	tassel on the hem of the denim jacket sleeves and the hem of the ripped jeans **49.16**	pink lace-up platform boots with thick soles and decorative laces, including the tassel details on the boot tops **0.00**
Removed detail SAM 3.1 / FoodIns	lettuce **0.00**	a single green leaf of lettuce with visible veins and a slightly crinkled texture, positioned on the left side of the image, adjacent to a cluster of dark purple grapes and slices of dense brown bread with visible seeds and dried fruit. **90.57**	green lettuce leaves, fresh salad greens, leafy vegetable **0.00**

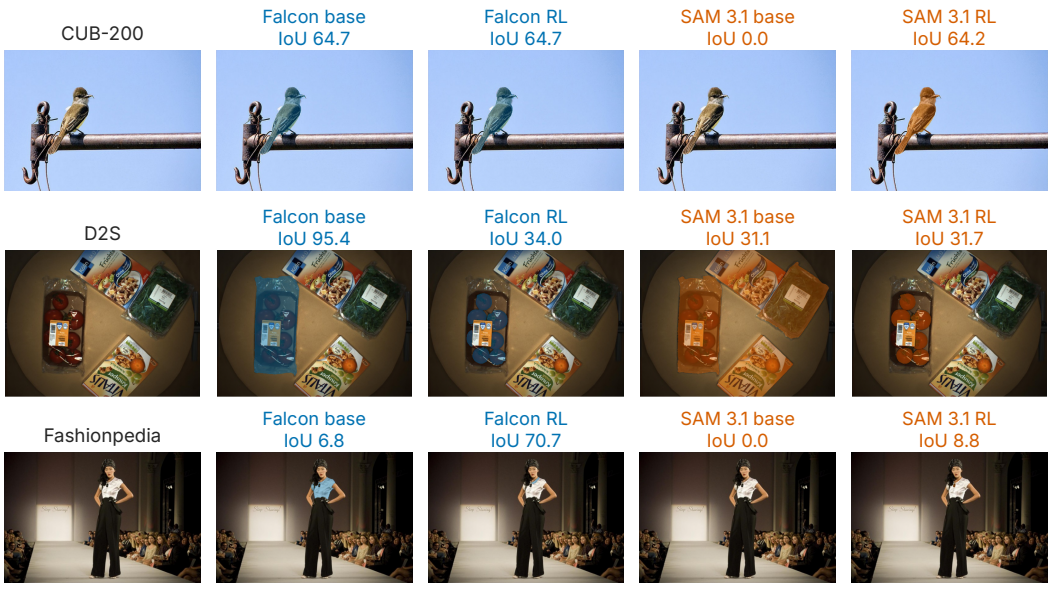

Figure 11: Retained Gemma 3 4B predictions before and after single-turn RL on the same test images. Blue overlays denote Falcon predictions and orange overlays SAM 3.1 predictions. IoU values use the assigned target masks. Those masks are not policy inputs. The examples include little change, improvement, and regression.

Table 11: Verbatim category, base-policy, and RL-policy prompts for Fig. 11, with IoU in parentheses.

Task / PSM	Category (IoU)	Base prompt (IoU)	RL prompt (IoU)
CUB-200 / Falcon	Great Crested Flycatcher (65.24)	a small greenish-gray bird with a crested head perched on a rusty metal pole (64.68)	a small greenish-gray bird with a prominent crest, perched on a metal pole (64.69)
CUB-200 / SAM 3.1	Great Crested Flycatcher (63.58)	a small, olive-green bird with a gray head and a yellow beak, perched on a rusty metal pole (0.00)	bird (64.17)
D2S / Falcon	vine tomatoes (33.68)	fresh vine tomatoes, salad greens, fruit packaging, fruit (95.43)	fresh vine tomatoes in plastic packaging (34.00)
D2S / SAM 3.1	vine tomatoes (31.70)	fresh vine tomatoes, salad greens, fruit packaging, fruit (31.10)	tomatoes in plastic packaging (31.72)
Fashionpedia / Falcon	collar (74.46)	woman's white shirt collar (6.85)	white shirt collar (70.68)
Fashionpedia / SAM 3.1	collar (0.00)	fashion model wearing a black and white outfit, runway, audience (0.00)	woman's neckpiece (8.75)

Table 12: Supervised Falcon mask-head controls using annotated masks. IoU and paired differences are on the 0–100 scale. Train and test counts refer to images.

Dataset	Train	Test	Steps	Base	Tuned	Δ	95% CI
D2S	450	100	200	35.88	37.12	+1.23	[-0.13, +2.61]
CUB-200	5000	500	1000	68.66	74.80	+6.15	[+5.78, +6.53]

I Paired Self-Distillation

Dataset class assignments provide the oracle category names described in Sec. 6. Mask generation uses these names and the images without annotated locations. We train three Falcon students per dataset on category masks, base rewrite masks, and RL masks. Frozen Falcon generates all targets from either the category name or one rewrite by Qwen 3.5 0.8B before RL or at RL step 100. We initially select 500 tasks per dataset. Three truncated RL teacher actions are excluded from both the category-mask and RL-mask conditions, leaving 499 tasks each in D2S, Fashionpedia, and FoodInsSeg and 500 in the other datasets. Each pair holds out the same 50 tasks for monitoring loss on generated targets. The remaining 449 or 450 tasks are used for training.

The base-rewrite condition attempts one rewrite on each retained training and validation task with the pretrained policy and no RL adapter. Failed, truncated, or invalid rewrites are excluded without replacement or a fallback mask. This leaves 382–443 training tasks and 38–50 validation tasks per dataset, so the three conditions do not use identical training sets. Category-mask overlays for this condition are reconstructed from the original images and retained masks. Its unchanged-Falcon evaluation reproduces the existing baseline predictions on all 600 test tasks.

All three students start from the same pretrained Falcon checkpoint and tune the mask projection and two output-vocabulary rows for presence and absence, totaling 2,363,392 parameters. The backbone and geometry decoder remain frozen. Training combines response-token cross-entropy with 200 times focal loss and ten times Dice loss on positive generated instances at their original resolution. Empty targets contribute response loss but no positive-mask loss. Each student uses 200 AdamW training steps, batch size four, learning rate 10^{-5}, and a fixed random seed.

Tab. 13 evaluates each dataset's students on the same 100 native test tasks using the original category prompts. Unchanged denotes the original pretrained Falcon checkpoint before any self-distillation fine-tuning. It receives the same category prompts and uses the same inference settings as the students, without an MLLM rewrite. Ground-truth test annotations provide the reference for computing IoU for every condition. Both rewrite conditions exceed category-mask students on five datasets, with lower scores on Fashionpedia. RL-mask students exceed base-rewrite students by 1.07 IoU points

overall (95% confidence interval: 0.50–1.74). The six-dataset average gives equal weight to each dataset. Intervals draw paired samples from the 586 photograph families underlying the 600 tasks, keeping shared food photographs together. They do not measure variation between training seeds. The oracle assumption applies to mask generation for student training. The RL prompt policy was trained beforehand with rewards computed from annotated masks.

Table 13: Self-distillation with Falcon. Category masks, Base rewrite masks, and RL masks identify the generated training targets for three separate students, evaluated at training step 200. Unchanged is pretrained Falcon without self-distillation fine-tuning. All conditions use category prompts on the same 100 test tasks per dataset. The change and interval compare RL-mask students with category-mask students. IoU is on the 0–100 scale.

Dataset	Unchanged	Category masks	Base rewrite masks	RL masks	Δ	95% CI
CUB-200	62.08	62.57	73.09	74.70	+12.13	[+7.03, +17.68]
D2S	34.29	29.90	40.47	40.83	+10.94	[+5.57, +16.92]
Fashionpedia	71.00	70.89	68.66	69.96	-0.93	[-1.43, -0.48]
FoodInsSeg	62.30	54.92	68.87	69.29	+14.37	[+8.35, +21.09]
FoodSeg103	65.08	61.62	70.03	72.42	+10.80	[+5.80, +16.52]
Flowers 102	49.25	42.69	69.68	70.02	+27.33	[+19.74, +35.03]
All six	57.33	53.76	65.13	66.20	+12.44	[+10.11, +14.87]

J Dataset Reconstruction

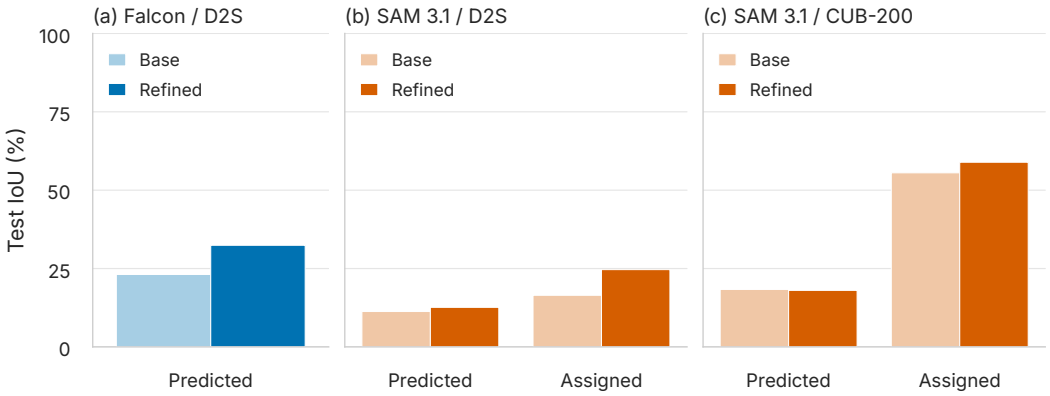

Figure 12: Full dataset reconstruction before and after one RL prompt refinement. Categories are predicted by Qwen 3.5 4B or supplied from assignments. All native test images are included, with missing categories scored as zero.

Category prediction. Qwen 3.5 4B receives each image and the global training taxonomy, without per-image assignments or masks. Greedy decoding allows at most 512 output tokens. We process all 88,974 images and retain invalid and truncated replies. Evaluation excludes 220 Flowers images without reference assignments. Precision, recall, and F1 pool image-category decisions within each dataset and split (Tab. 14). The Flowers inventory contains two exact photographs shared between training and test.

For assigned categories G and predicted categories L, perfect masks for correctly predicted categories give upper bounds of $|G \cap L|/|G|$ native-category IoU and $|G \cap L|/|G \cup L|$ label-union IoU. We average these bounds over scored images. On CUB-200, the correct category is absent from 4,120 of 5,794 test predictions, limiting native-category IoU to 28.89.

Mask reconstruction. Each segmenter uses its own Qwen 3.5 0.8B refiner at RL step 100, with the no-hint instruction and greedy decoding. Category lists are fixed across base and refined runs, with assigned lists supplied only in the SAM controls. Refiners receive neither assigned masks nor quality

Table 14: Category prediction on full native test splits. Scored counts exclude unavailable assignments and retain failed predictions. The IoU bounds assume perfect masks for correctly predicted categories. Scores are on the 0–100 scale.

			Category prediction			IoU upper bound	
Dataset	Images	Scored	Precision	Recall	F1	Native	Union
CUB-200	5794	5794	28.89	28.89	28.89	28.89	28.88
D2S	3600	3600	62.68	39.57	48.51	45.70	41.53
Fashionpedia	1158	1158	75.72	61.77	68.04	64.65	54.07
FoodInsSeg	2135	2135	67.46	61.64	64.42	62.96	52.30
FoodSeg103	2135	2135	68.82	61.09	64.73	62.68	52.60
Flowers 102	6149	5993	68.68	70.08	69.38	70.08	69.14

scores. Missing categories and invalid refinements give empty predictions. All native images remain included. No test image exactly matches a decoded training image from the teacher's pool.

Native-category IoU averages over assigned categories within each image and then over images. Label-union IoU also scores spurious categories as zero. With predicted labels, both segmenters improve on D2S, while SAM 3.1 has no clear gain on CUB-200 (Tab. 15). With assigned categories, SAM gains 8.18 and 3.38 points, respectively. Changing label sets also changes batching and can affect masks and subsequent rewrites, so the controls do not isolate category availability.

Intervals sample paired exact-RGB photograph groups, without measuring training-seed or inference variation. Training-split scores describe reconstruction of training images. These experiments use a supplied taxonomy and policies trained with annotated rewards. They evaluate prompt refinement without a distilled student.

Table 15: Full reconstruction on D2S (3,600 test and 4,380 training images) and CUB-200 (5,794 test and 5,994 training images). Each row uses the same images and category lists before and after refinement. IoU and changes are on the 0–100 scale. Intervals sample paired photograph groups.

		Native-category IoU				Label-union IoU	
PSM / Dataset	Labels	Base	Refined	Δ	95% CI	Base	Refined
Test images							
Falcon / D2S	Predicted	23.11	32.42	+9.31	[+8.59, +10.06]	21.34	29.79
SAM 3.1 / D2S	Predicted	11.29	12.62	+1.33	[+0.57, +2.12]	10.19	11.36
SAM 3.1 / D2S	Assigned	16.46	24.65	+8.18	[+7.20, +9.23]	16.46	24.65
SAM 3.1 / CUB-200	Predicted	18.33	18.03	-0.30	[-0.77, +0.17]	18.32	18.02
SAM 3.1 / CUB-200	Assigned	55.53	58.91	+3.38	[+2.43, +4.33]	55.53	58.91
Training images							
Falcon / D2S	Predicted	31.29	42.64	+11.35	[+10.48, +12.21]	31.10	42.33
SAM 3.1 / D2S	Predicted	12.43	25.63	+13.20	[+12.11, +14.30]	12.23	25.38
SAM 3.1 / D2S	Assigned	17.72	48.14	+30.42	[+29.01, +31.85]	17.72	48.14
SAM 3.1 / CUB-200	Predicted	18.34	18.20	-0.14	[-0.63, +0.35]	18.34	18.20
SAM 3.1 / CUB-200	Assigned	54.21	58.62	+4.41	[+3.46, +5.35]	54.21	58.62