LAM++: ENHANCED LOCATEANYTHING VIA HIGH-RESOLUTION POST-TRAINING AND GEOMETRY-AWARE SPECULATIVE DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent vision-language models such as the LocateAnything Model (LAM) have enabled universal localization through generative parallel box decoding, yet they still have gaps with strong dedicated vision models such as SAM3 on dense scenes and small targets. We present **LAM++**, an enhanced LAM with higher accuracy and faster speed via post-training and speculative decoding. We first construct a high-quality corpus of over 30M localization examples. A supervised finetuning recipe that integrates high-resolution visual processing with single-query training is developed to substantially improve localization accuracy on all domains, and a sequence-level GRPO strategy is used to further strengthen performance. We then introduce a geometry-aware speculative decoding (GeoSpec) framework that reuses LAM's native parallel decoder for drafting and employs the autoregressive (AR) policy for verification. Unlike exact token-level verification of speculative decoding, GeoSpec exploits the geometric structure of localization outputs, accepting spatially equivalent coordinate proposals under a relaxed criterion and invoking AR correction only when necessary. This yields a favorable accuracy–efficiency Pareto frontier: GeoSpec achieves a 68% runtime speedup with $\Delta\mathrm{mF1} \leq 0.1$. Across a diverse suite of localization benchmarks that span different task formulations and visual domains, LAM++ achieves state-of-the-art performance, outperforming SAM3 as well as larger or closed-source VLMs. The models, full training corpus, and complete post-training recipes will be released.

1 INTRODUCTION

Object detection (Girshick et al., 2014; Cheng et al., 2024; Redmon et al., 2016; Lin et al., 2017b;a; Liu et al., 2016) has long been a fundamental task in computer vision, aiming to identify object instances and predict their bounding boxes. Conventional dedicated detection systems typically rely on specialized vision architectures (Liu et al., 2024; Carion et al., 2026; Bevli et al., 2026), whereas recent vision-language models (VLMs) have formulated detection as a generative prediction problem, representing bounding-box coordinates as tokens and producing them with a language-model decoder (Chen et al., 2021; Bai et al., 2025; Jiang et al., 2026b). This formulation offers a unified interface for detecting instances that match free-form queries of varying granularity and compositional complexity, while allowing detection results to support subsequent multimodal reasoning (Shao et al., 2024a; Jiang et al., 2026a). The continued rapid progress of frontier multimodal models, including GPT-5.6 (OpenAI, 2026), Gemini-3.7 (Google, 2026), and Qwen-3.8-max (Alibaba, 2026) further highlights the potential of language-conditioned generative object detection.

However, existing VLM-based generative detection and grounding methods still face a pronounced accuracy–efficiency trade-off. Autoregressive coordinate generation incurs substantial decoding latency, especially as the number of targets grows, while compact generative models often suffer from unstable localization in dense scenes (Jiang et al., 2026b). Scaling to substantially larger frontier VLMs can improve localization quality, but introduces considerably higher parameter counts and inference costs. The recently developed LocateAnything Model (LAM) (Wang et al., 2026) takes an important step toward efficient generative detection by introducing *Parallel Box Decoding* (PBD). Instead of autoregressively emitting every coordinate token, LAM treats each bounding box as a structured generation unit and predicts its coordinates in parallel through a multi-token prediction

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.



Figure 1: **Overview of LAM++.** LAM++ improves both localization accuracy and decoding efficiency through four complementary components. **(1) Diverse localization data.** We build a high-quality corpus of over 30M localization samples, covering detection, referring, dense localization, reasoning, GUI grounding, and diverse visual domains. **(2) Failure-driven SFT.** High-resolution visual processing preserves fine-grained evidence for small targets, while single-query training improves instance localization and reduces cross-concept interference. **(3) RL post-training.** We systematically study GRPO for both token-wise AR and box-wise MTP policies, and adopt sequence-level AR-GRPO to improve localization quality. **(4) GeoSpec.** We reuse LAM's MTP policy to draft candidate boxes and the AR policy to verify them with a geometry-aware acceptance criterion. Together, these components enable LAM++ to *see better, optimize better, and decode faster*.

(MTP) policy (Gloeckle et al., 2024b). This substantially alleviates the sequential decoding bottleneck and enables a flexible trade-off between accurate autoregressive (AR) decoding and faster parallel decoding. However, improving decoding efficiency alone cannot resolve the remaining detection errors. Through a controlled analysis of LAM, we find that its largest performance deficits are concentrated on *dense scenes* and *small objects*. This observation motivates us to revisit LAM from the perspective of post-training: *can a generative detector match, or even surpass, strong dedicated vision models such as SAM3 (Carion et al., 2026) while retaining decoding efficiency?*

To achieve the above mentioned goal, we first strengthen LAM through *failure-driven supervised post-training* (SFT). We carefully construct a high-quality corpus containing over 30M examples, which span object detection, referring expression comprehension, dense localization, reasoning, and GUI grounding across diverse visual domains. We increase the effective visual resolution to preserve fine-grained evidence for small objects, and reformulate training into a *single-query* setting that reduces cross-concept interference and encourages the detection of all matching instances. For images associated with multiple queries, the image-prefix tokens can be cached and reused, thereby amortizing the additional cost of issuing separate queries. We further investigate whether reinforcement learning (RL) can improve detection performance beyond SFT. LAM can support two distinct generation policies: token-wise AR decoding and box-wise MTP decoding. By systematically investigating GRPO for both policies, we show that simple sequence-level GRPO provides the most reliable improvement and we adopt AR-GRPO as the final RL post-training strategy.

The AR policy achieves stronger detection performance, while MTP decoding remains substantially faster in practice. LAM's original hybrid decoding partially balances the two, but its reliance on MTP introduces noticeable performance degradation. We therefore present **GeoSpec**, a geometry-aware speculative decoding framework for generative object grounding. For bounding-box prediction, coordinate tokens are discrete, while the geometry they represent is continuous. Different coordinate-token sequences may correspond to nearly identical bounding boxes. GeoSpec exploits

this property by reusing LAM's native MTP policy as the drafter and the stronger AR policy as the verifier. Rather than requiring exact token agreement, GeoSpec accepts coordinate proposals that are geometrically close to the AR predictions and invokes AR correction only when necessary. GeoSpec generates **68% more boxes per second** than AR decoding while maintaining essentially unchanged detection performance. Our main contributions are summarized as follows:

- We identify dense-scene and small-object detection as key bottlenecks of generative detectors and develop a failure-driven post-training recipe built on a high-quality corpus of over 30M examples, with high-resolution visual processing and single-query supervision.
- We introduce a geometry-aware speculative decoding framework GeoSpec that uses LAM's MTP policy for drafting bounding boxes and AR policy for verification, delivering much higher box-generation throughput with essentially unchanged detection performance.
- We systematically study reinforcement learning for both AR and MTP detection policies, showing the effectiveness of sequence-level GRPO and uncovering bidirectional cross-policy transfer between distinct decoding policies that share the same model parameters.

The overview of our proposed method, namely **LAM++**, is illustrated in Fig. 1. LAM++ demonstrates strong detection capabilities and faster decoding than LAM. Across a diverse evaluation suite that covers multiple visual domains, LAM++ achieves state-of-the-art performance, outperforming SAM3 and substantially larger or closed-source VLMs (Qwen, 2026; OpenAI, 2026; Google, 2026).

2 Related Work

Open-Vocabulary and Generative Object Detection. Open-vocabulary object detection extends conventional detection by allowing target concepts to be specified through natural language. GLIP (Li et al., 2022) reformulates detection as phrase grounding, while Grounding DINO (Liu et al., 2024) combines DETR-style (Carion et al., 2020) detection with vision-language fusion to support open-set object detection. Related advances in promptable segmentation, represented by the SAM series (Kirillov et al., 2023; Ravi et al., 2025; Carion et al., 2026), and language-conditioned perception, represented by Falcon Perception (Bevli et al., 2026; Chaybouti et al., 2026), provide additional context for detecting and delineating object instances. In parallel, generative detection formulates object prediction as sequence generation. Pix2Seq (Chen et al., 2021) represents object categories and bounding-box coordinates as discrete tokens, while multimodal language models such as Kosmos-2 (Peng et al., 2023), Shikra (Chen et al., 2023b), and the Qwen-VL family (Bai et al., 2025) incorporate visual grounding into language-model generation. Rex-Omni (Jiang et al., 2026b) uses dedicated quantized coordinate tokens and combines large-scale supervised fine-tuning with GRPO for language-conditioned object detection. The LocateAnything Model (LAM) (Wang et al., 2026) further improves decoding efficiency through *Parallel Box Decoding* (PBD), supporting both autoregressive (AR) and multi-token prediction (MTP) (Gloeckle et al., 2024b) decoding within the same model. In this work, we employ failure-driven supervised post-training and reinforcement learning to improve detection performance, particularly in dense scenes and for small objects.

Efficient Decoding and Speculative Generation. Parallel prediction and speculative decoding provide two major routes for accelerating autoregressive generation. Blockwise Parallel Decoding (Stern et al., 2018) and MTP methods (Gloeckle et al., 2024a) generate several future tokens simultaneously, while Medusa (Cai et al., 2024) uses additional decoding heads to propose multiple candidates. LAM (Wang et al., 2026) adapts this idea to localization by predicting box-coordinate tokens in parallel. Another direction distills autoregressive policies into faster draft or multi-token predictors. On-policy distillation (Agarwal et al., 2024) reduces train–test mismatch by supervising students on their own generated states, and work extends this idea to standalone MTP (Kirchenbauer et al., 2026) and speculative draft models (Lei et al., 2026). However, matching the semantic reliability of AR decoding remains difficult for structured localization outputs. Speculative decoding instead retains the AR model during inference. Classical speculative decoding (Leviathan et al., 2023; Chen et al., 2023a) uses a lightweight drafter followed by target-model verification, while EAGLE (Li et al., 2024b) and DFlash (Chen et al., 2026) improve proposals through block-level drafting. These methods define acceptance in discrete token space. In this work, we exploit a property of localization: different coordinate tokens may still represent nearly identical geometry, enabling speculative acceleration without an additional draft model while retaining efficient decoding.

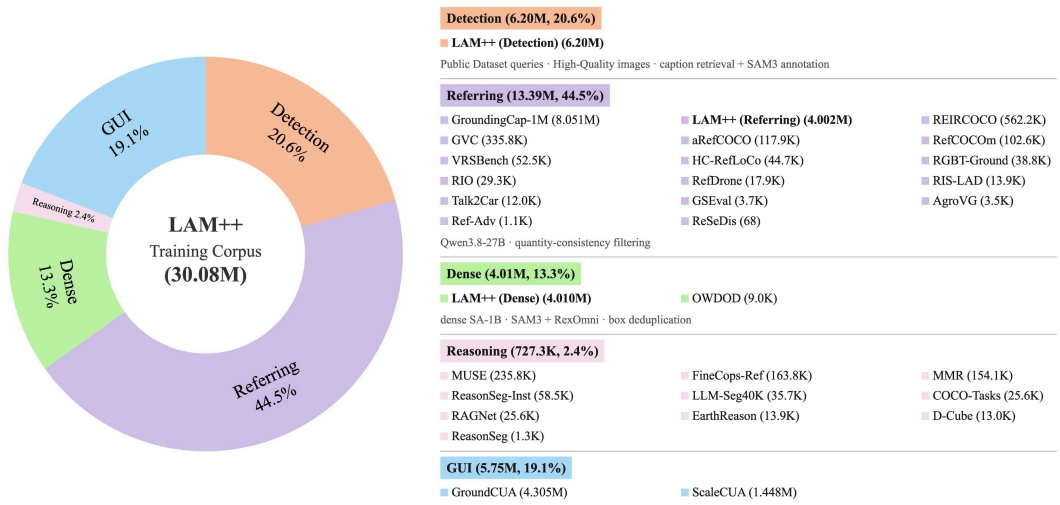

Figure 2: **Composition of LAM++ training corpus.** Constructed data sources are shown in bold.

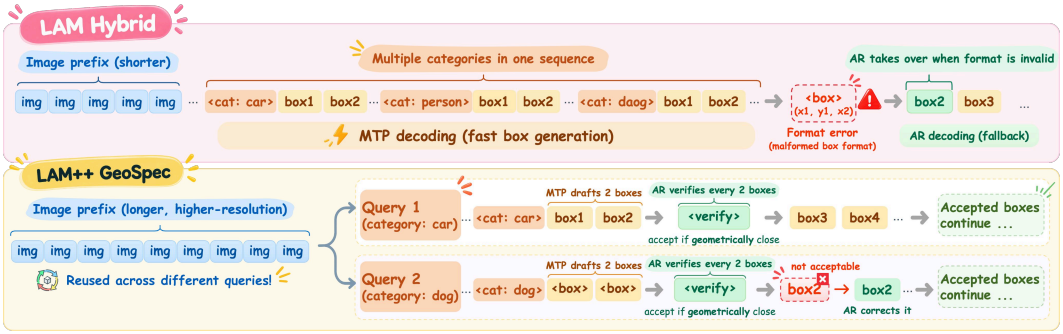

Figure 3: LAM Hybrid uses a shorter visual prefix and packs multiple categories into a single sequence, relying on fast MTP decoding and invoking AR only when malformed outputs occur. LAM++ instead combines high-resolution visual processing with single-query decoding, while computing the image prefix only once and reusing its KV states across queries. GeoSpec further turns the native MTP policy into a speculative drafter: MTP proposes multiple boxes, while AR periodically verifies their coordinates and accepts geometrically consistent drafts or corrects rejected ones.

3 METHOD

LAM++ improves LAM (Wang et al., 2026) in localization accuracy and decoding efficiency through four components. We first build a large-scale localization corpus for challenging regimes (Sec. 3.1), introduce failure-driven SFT with high-resolution processing and single-query training (Sec. 3.2), further improve localization with post-training (Sec. 3.3), and finally introduce GeoSpec for geometry-aware speculative decoding with MTP drafting and AR verification (Sec. 3.4).

3.1 DIVERSE LOCALIZATION DATA

Large-scale localization supervision is central to recent generative localizers like LAM and Rex-Omni (Jiang et al., 2026b), yet existing training mixtures remain dominated by generic detection and grounding data, providing limited coverage of challenging cases such as dense scenes, small targets, complex referring expressions, and reasoning-intensive queries. We therefore construct a failure-oriented localization corpus that explicitly strengthens supervision for these regimes. As summarized in Fig. 2, the resulting corpus contains **30.08M** examples spanning general detection, referring grounding, dense localization, reasoning grounding, and GUI grounding. All data are converted into a unified single-query localization format; details are provided in Appendix Sec. A.

General Detection. Public detection datasets provide accurate supervision but contain a finite set of image–category combinations, making it hard to continuously scale localization data while con-

trolling target density and object size. We therefore construct a scalable **silver-label pool**. We first collect category vocabularies from LVIS (Gupta et al., 2019), SACo-Gold (Carion et al., 2026), COCO (Lin et al., 2015), Objects365 (Shao et al., 2019), and OpenImages (Kuznetsova et al., 2020). Candidate images are obtained from LLaVA-OV-85M (An et al., 2025), Unsplash (Unsplash, 2026), and SA-1B (Kirillov et al., 2023), while visually grounded captions are generated with Qwen3.5-35B-A3B (Team, 2026). We retrieve images by matching the query vocabulary against captions using lexical and regular-expression rules, and apply SAM3 (Carion et al., 2026) to each retrieved image–query pair for silver box annotation. This produces **100M** candidate examples, from which we sample **6.2M** while preserving broad vocabulary coverage and diverse box-count distributions.

Referring Grounding. We curate **9.39M** examples from 16 public referring-grounding datasets spanning natural, embodied, aerial, and remote-sensing domains (Appendix Sec. A), and construct another **4M** examples from LLaVA-OV-85M, Unsplash, and SA-1B. Qwen3.5-35B-A3B generates fine-grained queries involving attributes, spatial relations, and quantities, while Qwen3.8-27B (Qwen, 2026) predicts boxes. For queries with explicit cardinality, we retain only samples whose stated count matches the predicted count. The resulting subset contains **13.39M** examples.

Dense Localization. Dense scenes are difficult for generative localizers, yet conventional detection corpora contain few queries with many matching instances. We therefore construct dense localization supervision by combining 8,969 public OWDOD (Joseph et al., 2021) examples with **4.01M** examples from the top-8% SA-1B (Kirillov et al., 2023) images ranked by ground-truth mask count. Qwen3.5-35B-A3B generates short category queries for repeated, distinguishable objects while excluding ambiguous textures and uncountable regions, and SAM3 and RexOmni (Jiang et al., 2026b) independently localize each image–query pair. We merge their pseudo-labels through cross-model geometric matching, with the filtering and deduplication rules selected by grid search. The resulting subset contains **4.01M** examples. Additional details are provided in Appendix Sec. A.

Reasoning Grounding. We collect **727K** examples from ten reasoning-oriented datasets covering queries about object function, attributes, spatial context, affordances, and implicit user intent (Appendix Sec. A). For segmentation-based sources, we convert each target to its bounding box while preserving the original target granularity and discard examples without valid target regions.

GUI Grounding. For GUI grounding, we retain the large-scale supervision used by LAM (Wang et al., 2026). The resulting subset contains **5.75M** examples from GroundCUA Feizi et al. (2026) (4.3M) and ScaleCUA Lv et al. (2026) (1.45M). We convert multi-turn interaction records and multi-element annotations into single-query localization samples using their native element labels.

3.2 Failure-Driven Supervised Fine-Tuning

The corpus broadens localization supervision, but scale alone does not address the two dominant categories of failure cases of LAM: insufficient visual granularity for small objects and incomplete instance enumeration in dense scenes. Therefore, we adapt and improve the SFT formulation to address these failures, as illustrated in Fig. 3.

High-Resolution Visual Processing. We resize the shorter side of each image to 1024 pixels and use MoonViT's (Kimi, 2025) native-resolution processing scheme, preserving substantially finer spatial evidence for small and crowded targets without modifying the localization output representation.

Single-query Training. LAM supports multiple localization concepts within a single request. We instead decompose each training example into a single query to reduce cross-concept interference. At inference time, multiple queries do not require repeated image encoding: because image tokens precede the query tokens, their **prefix KV states can be cached and reused** across queries. As a result, the additional cost of single-query inference is limited to query-conditioned decoding itself.

3.3 RL Post-Training

We further optimize localization quality with GRPO. LAM exposes two shared-parameter decoding policies: token-wise autoregressive (AR) decoding and box-wise multi-token prediction (MTP) decoding. We systematically study GRPO under both policies. The final LAM++ model adopts AR-GRPO, while MTP-GRPO is used to analyze optimization behavior and cross-policy transfer.

Localization Reward. Directly using thresholded evaluation metrics such as recall or F1 as sequence-level rewards provides little learning signal when imperfect boxes fall below the matching threshold, even if their geometry is already close to the ground truth. Inspired by Perception-R1 (Yu et al., 2025), we therefore introduce a dense localization reward based on matched box IoU:

$$R_{\text{IoU}} = \frac{\sum_{(p,g)\in\mathcal{M}} \text{IoU}(p,g)}{\max(|P|,|G|,1)}, \tag{1}$$

where P and G denote predicted and ground-truth box sets, and $\mathcal{M}$ their one-to-one matching. Unlike threshold metrics, this term provides continuous credit as boxes approach their targets, while the denominator penalizes missed and over-generated instances. However, IoU alone does not fully capture the set-level detection objective. We combine it with F1 at multiple IoU thresholds:

$$R = w_{\text{IoU}} R_{\text{IoU}} + w_{\text{F1}} \frac{1}{|\mathcal{T}|} \sum_{\tau\in\mathcal{T}} F_1@\tau. \tag{2}$$

The IoU term encourages geometrically accurate localization, whereas F1 explicitly accounts for both missed and spurious detections. We use $w_{\text{IoU}} = 0.5$, $w_{\text{F1}} = 1.0$, and $\mathcal{T} = \{0.5, 0.75\}$. For each prompt, GRPO samples multiple localization responses from the current policy, parses them into box sets, and computes the reward for each complete response. AR-GRPO uses standard autoregressive rollouts under the current policy. Further analyses of reward and credit-assignment variants, as well as the MTP-GRPO formulation, are provided in Appendix Secs. B and C.

3.4 GeoSpec: Geometry-Aware Speculative Decoding

After SFT and RL post-training, AR mode provides better localization quality, while MTP remains substantially faster. We introduce **GeoSpec**, which reuses LAM's native MTP policy for drafting and the stronger AR policy for verification, improving decoding speed with negligible accuracy loss.

MTP Drafting. As illustrated in Fig. 3, GeoSpec uses MTP policy to draft up to K consecutive boxes, draft length K controls how many boxes are generated before AR verification: larger K reduces verification frequency, but may waste computation when inaccurate drafts require correction.

AR Verification. The AR policy then verifies the MTP drafts. Conventional speculative decoding requires exact token agreement between draft and target policies, which is unnecessarily strict for localization because discrete coordinate tokens represent continuous geometry. We therefore introduce a coordinate-tolerance hyperparameter ϵ, specifying the maximum allowable deviation between the MTP draft and AR prediction in normalized coordinate space. Given an MTP draft $\tilde{b} = (\tilde{x}_1, \tilde{y}_1, \tilde{x}_2, \tilde{y}_2)$ and its AR prediction $b^{\text{AR}} = (x_1^{\text{AR}}, y_1^{\text{AR}}, x_2^{\text{AR}}, y_2^{\text{AR}})$, GeoSpec accepts $\tilde{b}$ when

$$\max_{j\in\{x_1,y_1,x_2,y_2\}} \left|\tilde{b}_j - b_j^{\text{AR}}\right| \le \epsilon, \tag{3}$$

while structural tokens require exact agreement with the AR verifier. GeoSpec commits the longest accepted draft prefix. At the first mismatch, AR completes the current box and discards the remaining draft suffix. Larger ϵ increases draft acceptance, while smaller ϵ more closely follows AR.

4 Experiments

4.1 Training and Evaluation Details

Within each task, we reserve 10% of the data for GRPO, prioritizing dense scenes and small targets, and use the remainder for SFT. We fine-tune LAM on the resulting 27M-example mixture for ∼45K steps using 128 PPUs (16 nodes × 8 devices), BF16 precision, a global batch size of 128, a peak learning rate of 2×10^{-5}, 3% warm-up, and cosine decay. Starting from this checkpoint, we perform one epoch of AR-GRPO on 3M image–query pairs (∼25K steps) using the reward in Sec. 3.3, with eight rollouts per prompt, a global batch size of 128 prompts, and a learning rate of 2×10^{-6}, using the same high-resolution input. For all open-source datasets in Fig. 2, only the training split is used.

We evaluate detection on LVIS (Gupta et al., 2019), COCO (Lin et al., 2015), SACo-Gold (Carion et al., 2026); referring grounding on HumanRef (Jiang et al., 2025), RefCOCOg (Yu et al., 2016),

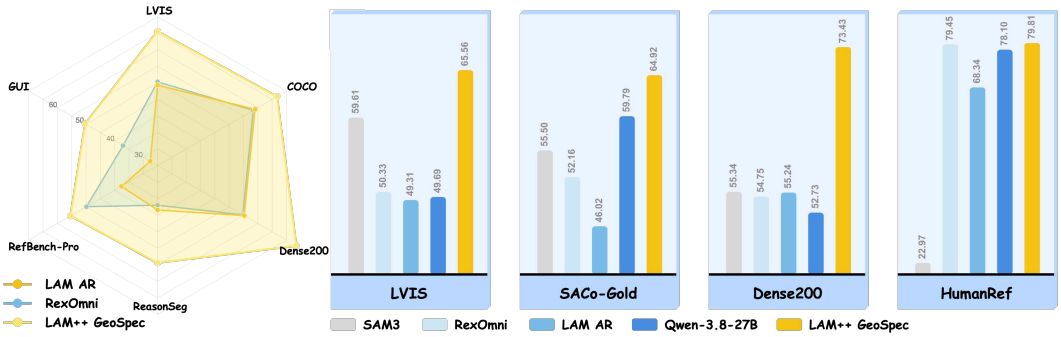

Figure 4: **Comparison with strong localization baselines. Left:** LAM++ consistently outperforms comparable-scale VLMs (Wang et al., 2026; Jiang et al., 2026b) across diverse benchmarks. **Right:** On representative benchmarks, LAM++ further surpasses the dedicated localization model SAM3 (Carion et al., 2026) and the substantially larger model Qwen-3.8-27B (Qwen, 2026).

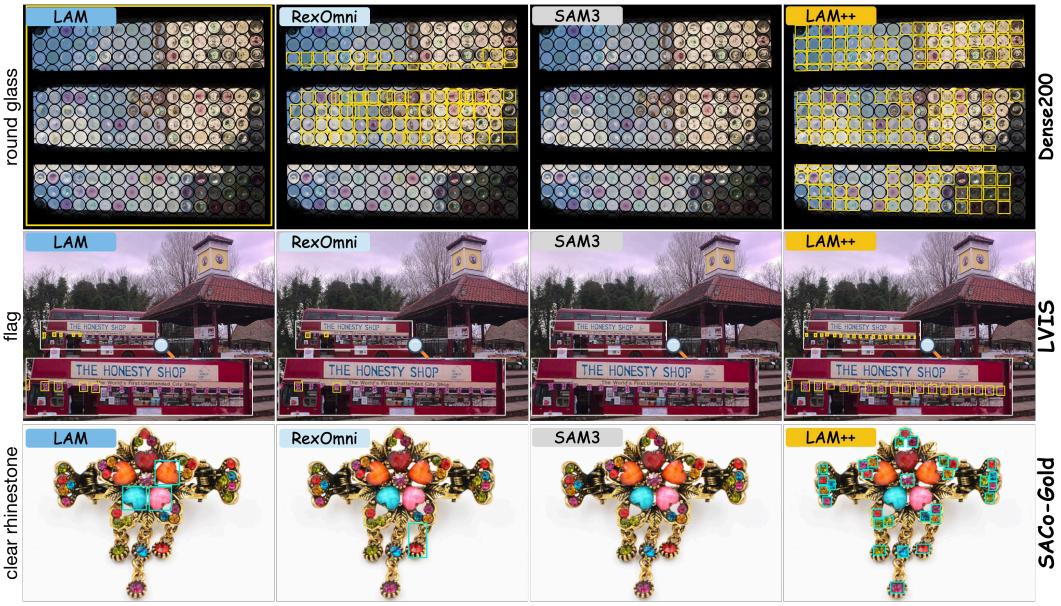

Figure 5: LAM++ localizes small and dense instances while maintaining precise box placement.

RefBench-PRO (Gao et al., 2025); dense localization on Dense200 (Jiang et al., 2026b); reasoning grounding on ReasonSeg (Lai et al., 2024); and GUI grounding on ScreenSpot-Pro (Li et al., 2025a). VisDrone (Zhu et al., 2021) and Talk2Car (Deruyttere et al., 2019) further evaluate aerial and autonomous-driving generalization. We compare against strong open- and closed-source baselines, including general-purpose VLMs Qwen-3.8-27B (Qwen, 2026), Gemini-3.7-Flash (Google, 2026), and GPT-5.6 (OpenAI, 2026), the dedicated localization model SAM3 (Carion et al., 2026), the VLM-based localization model RexOmni (Jiang et al., 2026b), and the original LAM (Wang et al., 2026) under both AR and Hybrid decoding. All benchmarks use positive queries and a unified box-based mF1; for ScreenSpot-Pro, we replace its official point-in-box metric with box-based mF1.

4.2 Main Results

Tab. 1 reports complete results across all evaluated tasks, domains, and baselines, while Fig. 4 provides a compact comparison on representative benchmarks. LAM++ consistently outperforms comparable-scale VLMs, including LAM (Wang et al., 2026) and RexOmni (Jiang et al., 2026b), **on every benchmark**. The advantage is particularly pronounced in dense localization: on Dense200, LAM++ achieves relative improvements of **32.9%** over LAM AR and **32.7%** over SAM3 (Carion et al., 2026). Overall, despite retaining a general-purpose VLM architecture, LAM++ outperforms the dedicated localization model SAM3 on 10 of 11 benchmarks, demonstrating strong general-

Table 1: We compare closed-source and open-source models across different localization tasks and visual domains. All results are evaluated using a box-based mF1 metric, including ScreenSpot-Pro (Li et al., 2025a), whose official metric instead measures whether a predicted point falls inside the ground-truth box. The best result within each group is highlighted in the corresponding color.

Model	LVIS	SACo-Gold	COCO	Dense200	VisDrone	Talk2Car	HumanRef	RefCOCOg	RefBench-PRO	ReasonSeg	ScreenSpot-Pro
Closed-source Models											
Gemini-3.7-Flash	–	–	**59.34**	27.01	15.17	**36.94**	**81.15**	**79.05**	**64.59**	**57.61**	26.46
GPT-5.6-Luna	–	–	41.83	23.66	12.69	25.87	66.19	61.74	52.68	39.22	29.43
GPT-5.6-Sol	**44.34**	–	49.86	40.11	**19.83**	31.52	76.63	71.85	60.13	46.54	**36.22**
Open-source Models											
Qwen-3.8-27B	49.69	59.79	56.60	52.73	28.83	26.60	78.10	76.88	**61.67**	56.73	23.31
RexOmni	50.33	52.16	58.41	54.75	24.70	17.13	79.45	75.48	49.85	36.99	37.07
SAM3	59.61	55.50	63.67	55.34	**35.66**	4.71	22.97	34.56	22.95	6.06	0.38
LAM AR	49.31	46.02	59.04	55.24	27.10	22.96	68.34	64.75	37.61	38.43	27.58
LAM Hybrid	47.21	46.63	57.55	55.59	23.96	22.43	66.91	48.55	34.77	37.33	24.48
LAM++ GeoSpec	**65.56**	**64.92**	**66.69**	**73.43**	32.79	**33.25**	**79.81**	**79.45**	55.42	**54.38**	**50.28**

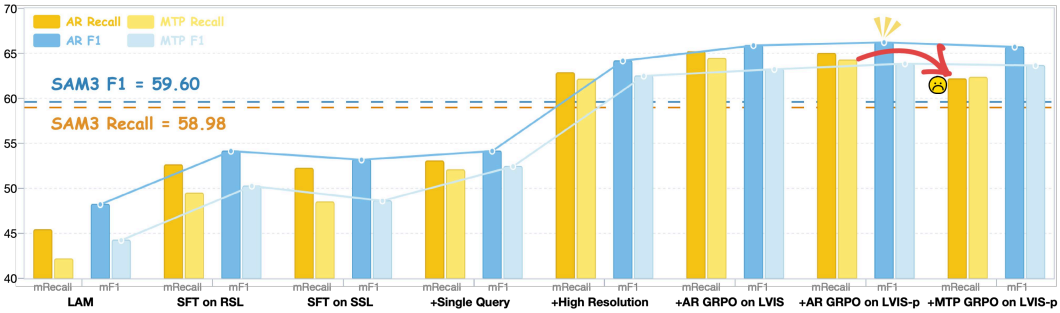

Figure 6: **Training-recipe ablations of LAM++ on LVIS.** Starting from LAM, we progressively study failure-driven silver-label selection, single-query SFT, high-resolution visual processing, and GRPO post-training. **Randomly-sampled Silver Labels (RSL)** are sampled randomly from the silver-label pool, whereas **Selected Silver Labels (SSL)** preferentially retain dense and small-target cases. **LVIS-p** is a failure-focused RL subset containing about 40% of the full LVIS training data. AR and MTP report the two decoding policies of the same checkpoint at each training stage.

ization across substantially different localization settings. Notably, we use *no images or instance annotations from the COCO or LVIS training sets*, making both evaluations **zero-shot**, as in SAM3.

The qualitative comparison in Fig. 5 further reveals where these gains arise. LAM and RexOmni frequently under-enumerate repeated objects in crowded scenes, either missing many instances or merging nearby targets into coarse boxes, while SAM3 can also fail to recover a large fraction of instances in these challenging examples. In contrast, LAM++ produces substantially more complete instance-wise predictions and separates densely packed small objects into individual boxes, directly addressing the persistent low-recall behavior that dominates these challenging dense cases.

LAM++ also performs favorably against substantially larger and closed-source frontier VLMs. It outperforms Qwen-3.8-27B (Qwen, 2026) on most benchmarks while remaining competitive on the rest, and shows similarly strong results against GPT-5.6 (OpenAI, 2026) and Gemini-3.7-Flash (Google, 2026). Overall, LAM++ delivers strong localization performance across detection, dense scenes, referring and reasoning grounding, cross-domain settings, and GUI interaction. Detailed quantitative results and additional qualitative examples are provided in Appendix Sec. D.

4.3 Training Recipe Ablations

Since full-scale controlled retraining on our 30M-example corpus is prohibitively expensive, we study the main training choices on LVIS (Gupta et al., 2019) for its sufficient scale, dense scenes, and small targets. As summarized in Fig. 6, SSL yields larger gains than RSL on the targeted dense and small-object failure modes. Single-query SFT further improves both AR and MTP and narrows their performance gap. High-resolution visual processing yields the largest SFT gain and

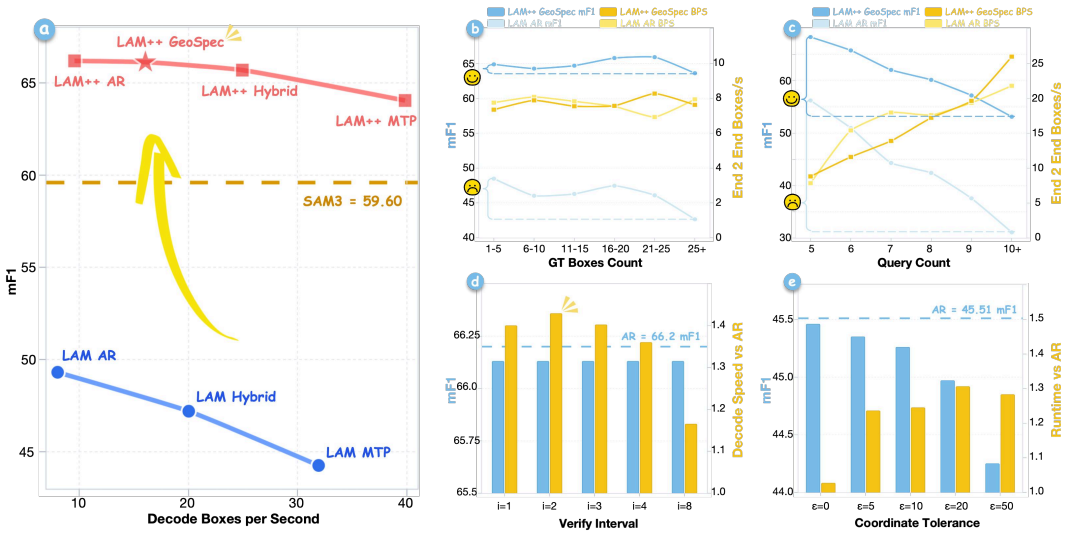

Figure 7: **Accuracy–efficiency analysis of GeoSpec on LVIS.** All panels use the final AR-GRPO checkpoint selected in Fig. 6; GeoSpec changes only the inference procedure without additional training. **(a)** GeoSpec retains near-AR accuracy while substantially increasing decoding throughput. **(b,c)** Accuracy and end-to-end throughput as scene density and query count increase. **(d)** Ablation on the draft length K, with $K = 2$ providing the best efficiency. **(e)** Ablation on the coordinate tolerance ϵ, showing the accuracy–efficiency trade-off of geometry-aware verification.

pushes the model beyond SAM3. AR-GRPO further improves both decoding policies through cross-policy transfer; notably, a failure-focused subset using only 40% of the LVIS training data achieves gains comparable to full-data GRPO. In contrast, direct MTP-GRPO shifts the model toward higher precision but lower recall. We therefore adopt high-resolution single-query SFT followed by AR-GRPO as the final training recipe. Detailed settings and analyses are provided in Appendix Sec. F.

4.4 Accuracy–Efficiency Analysis

All experiments in Fig. 7 use the final AR-GRPO checkpoint in Sec. 4.3; GeoSpec requires no additional training and only changes inference. As shown in Fig. 7(a), GeoSpec retains near-AR localization accuracy while increasing decoding throughput by approximately 68%, recovering much of MTP's parallel-decoding efficiency. Figs. 7(b,c) show that, as scene density or query count increases, LAM++ remains substantially more robust than LAM. Figs. 7(d,e) ablate the two GeoSpec hyperparameters and yield $K = 2$ and $\epsilon = 7$. In multi-query settings, reusable image-prefix KV states amortize the cost of separate single-query decoding, keeping end-to-end throughput competitive even at high query counts. Detailed settings and analyses are provided in Appendix Sec. H.

5 Conclusion

We presented **LAM++**, a stronger and more efficient generative model for universal localization. Starting from LAM, we addressed dense-scene and small-object failures through failure-driven data construction, high-resolution single-query SFT, and AR-GRPO, and observed bidirectional transfer between the shared AR and MTP policies. To recover decoding efficiency without sacrificing AR-level accuracy, we introduced **GeoSpec**, a geometry-aware speculative decoding framework that uses native MTP predictions as drafts and relaxes verification according to spatial equivalence. Across diverse tasks and domains, LAM++ substantially and consistently improved over the original LAM, surpasses strong dedicated localization models as well as VLMs with substantially larger parameter counts, and achieves a favorable accuracy–efficiency trade-off. We hope LAM++ provides a practical foundation for accurate, efficient, and general-purpose visual localization.

Limitations. Due to the substantial computational cost of full-scale post-training, controlled training ablations were conducted on LVIS rather than repeatedly retraining all variants on the complete 30M-example corpus. GeoSpec was evaluated with a fixed geometry-aware acceptance rule, while more adaptive verification strategies and hardware-specific optimizations are left for future work.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript drafting, language polishing, and structural editing, as well as with literature search and code development.

Generative AI models were also used as components of the data-construction pipeline, as described in Sec. 3.1.

AI-generated suggestions were used only as assistance and were reviewed, revised, and verified by the authors. All technical claims, experimental results, interpretations, and final written content were validated by the authors, who take full responsibility for the work.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Alibaba. Qwen3.8-max. https://www.alibabacloud.com/help/en/model-studio/qwen3-8-max, 2026. Accessed: 2026-09-16.

Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Didi Zhu, Chunsheng Wu, Huajie Tan, Chunyuan Li, Jing Yang, Jie Yu, Xiyao Wang, Bin Qin, Yumeng Wang, Zizhen Yan, Ziyong Feng, Ziwei Liu, Bo Li, and Jiankang Deng. Llava-onevision-1.5: Fully open framework for democratized multimodal training, 2025. URL https://arxiv.org/abs/2509.23661.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

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.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. Medusa: Simple llm inference acceleration framework with multiple decoding heads. *arXiv preprint arXiv:2401.10774*, 2024.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

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 *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

Sofian Chaybouti, Yasser Dahou, Ngoc Dung Huynh, Reda Alami, and Hilde Kuehne. Falcon perception-hd: High density perception via reinforcement learning. *arXiv preprint arXiv:2608.18881*, 2026.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023a.

Jian Chen, Yesheng Liang, and Zhijian Liu. Dflash: Block diffusion for flash speculative decoding. *arXiv preprint arXiv:2602.06036*, 2026.

Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023b.

Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852*, 2021.

Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. YOLO-World: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16901–16911, June 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Cheng_YOLO-World_Real-Time_Open-Vocabulary_Object_Detection_CVPR_2024_paper.html.

Thierry Deruyttere, Simon Vandenhende, Dusan Grujicic, Luc Van Gool, and Marie Francine Moens. Talk2car: Taking control of your self-driving car. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 2088–2098, 2019.

Qihua Dong, Kuo Yang, Lin Ju, Handong Zhao, Yitian Zhang, Yizhou Wang, Huimin Zeng, Jianglin Lu, and Yun Fu. Ref-adv: Exploring mllm visual reasoning in referring expression tasks. *arXiv preprint arXiv:2602.23898*, 2026.

Aarash Feizi, Shravan Nayak, Xiangru Jian, Kevin Qinghong Lin, Kaixin Li, Rabiul Awal, Xing Han Lù, Johan S Obando Ceron, Juan A Rodriguez, Nicolas Chapados, et al. Grounding computer use agents on human demonstrations. In *International Conference on Learning Representations*, volume 2026, pp. 5181–5215, 2026.

Shenghao Fu, Qize Yang, Qijie Mo, Junkai Yan, Xihan Wei, Jingke Meng, Xiaohua Xie, and Wei-Shi Zheng. Llmdet: Learning strong open-vocabulary object detectors under the supervision of large language models, 2025. URL https://arxiv.org/abs/2501.18954.

Tianyi Gao, Hao Li, Han Fang, Xin Wei, Xiaodong Dong, Hongbo Sun, Ye Yuan, Zhongjiang He, Jinglin Xu, Jingmin Xin, et al. Refbench-pro: Perceptual and reasoning oriented benchmark for referring expression comprehension. *arXiv preprint arXiv:2512.06276*, 2025.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 580–587, 2014.

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. *arXiv preprint arXiv:2404.19737*, 2024a.

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. *arXiv preprint arXiv:2404.19737*, 2024b.

Google. Gemini 3.7 flash. https://ai.google.dev/gemini-api/docs/models/gemini-3.7-flash, 2026. Accessed: 2026-09-16.

Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5351–5359. IEEE, 2019.

Xiangzhao Hao, Kuan Zhu, Hongyu Guo, Haiyun Guo, Ning Jiang, Quan Lu, Ming Tang, and Jinqiao Wang. Referring expression instance retrieval and a strong end-to-end baseline. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 4464–4473, 2025.

Rui Hu, Lianghui Zhu, Yuxuan Zhang, Tianheng Cheng, Lei Liu, Heng Liu, Longjin Ran, Xiaoxin Chen, Wenyu Liu, and Xinggang Wang. Groundingsuite: Measuring complex multi-granular pixel grounding, 2025. URL https://arxiv.org/abs/2503.10596.

Ziling Huang, Yidan Zhang, and Shin'ichi Satoh. Resedis: A dataset for referring-based object search across large-scale image collections. *arXiv preprint arXiv:2506.15180*, 2025.

Donggon Jang, Yucheol Cho, Suin Lee, Taehyeon Kim, and Dae Shik Kim. Mmr: A large-scale benchmark dataset for multi-target and multi-granularity reasoning segmentation. In *International Conference on Learning Representations*, volume 2025, pp. 67406–67431, 2025.

Chaoya Jiang, Yongrui Heng, Wei Ye, Haiyang Xu, Ming Yan, Ji Zhang, Fei Huang, and Shikun Zhang. Vlm-r^3: Region recognition, reasoning, and refinement for enhanced multimodal chain-of-thought. *Advances in Neural Information Processing Systems*, 38:63841–63869, 2026a.

Qing Jiang, Lin Wu, Zhaoyang Zeng, Tianhe Ren, Yuda Xiong, Yihao Chen, Liu Qin, and Lei Zhang. Referring to any person. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21667–21678. IEEE, 2025.

Qing Jiang, Junan Huo, Xingyu Chen, Yuda Xiong, Zhaoyang Zeng, Yihao Chen, Tianhe Ren, Junzhi Yu, and Lei Zhang. Detect anything via next point prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25472–25483, 2026b.

KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5826–5836. IEEE, 2021.

Kimi. Kimi-vl technical report, 2025. URL https://arxiv.org/abs/2504.07491.

John Kirchenbauer, Abhimanyu Hans, Brian Bartoldson, Micah Goldblum, Ashwinee Panda, and Tom Goldstein. Multi-token prediction via self-distillation. *arXiv preprint arXiv:2602.06019*, 2026.

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 *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 3992–4003. IEEE, 2023.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, March 2020. ISSN 1573-1405. doi: 10.1007/s11263-020-01316-z. URL http://dx.doi.org/10.1007/s11263-020-01316-z.

Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9579–9589. IEEE, 2024.

Haodi Lei, Yafu Li, Haoran Zhang, Shunkai Zhang, Qianjia Cheng, Xiaoye Qu, Ganqu Cui, Bowen Zhou, Ning Ding, Yun Luo, et al. Draft-opd: On-policy distillation for speculative draft models. *arXiv preprint arXiv:2605.29343*, 2026.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

Haocheng Li, Juepeng Zheng, Zenghao Yang, Kaiqi Du, Guilong Xiao, Gengmeng Pu, Haohuan Fu, and Jianxi Huang. Agrovg: A large-scale multi-source benchmark for agricultural visual grounding. *arXiv preprint arXiv:2605.22034*, 2026.

Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. Screenspot-pro: Gui grounding for professional high-resolution computer use. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 8778–8786, 2025a.

Kaiyu Li, Zepeng Xin, Li Pang, Chao Pang, Yupeng Deng, Jing Yao, Guisong Xia, Deyu Meng, Zhi Wang, and Xiangyong Cao. Segearth-r1: Geospatial pixel reasoning via large language model. *arXiv preprint arXiv:2504.09644*, 2025b.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10955–10965. IEEE, 2022.

Xiang Li, Jian Ding, and Mohamed Elhoseiny. Vrsbench: A versatile vision-language benchmark dataset for remote sensing image understanding. *Advances in Neural Information Processing Systems*, 37:3229–3242, 2024a.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*, 2024b.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. URL https://arxiv.org/abs/1405.0312.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2117–2125, July 2017a. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Lin_Feature_Pyramid_Networks_CVPR_2017_paper.html.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, October 2017b. URL https://openaccess.thecvf.com/content_iccv_2017/html/Lin_Focal_Loss_for_ICCV_2017_paper.html.

Junzhuo Liu, Xuzheng Yang, Weiwei Li, and Peng Wang. Finecops-ref: A new dataset and task for fine-grained compositional referring expression comprehension, 2025. URL https://arxiv.org/abs/2409.14750.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot MultiBox detector. In *Computer Vision – ECCV 2016*, pp. 21–37. Springer International Publishing, 2016. ISBN 9783319464480. doi: 10.1007/978-3-319-46448-0_2. URL https://doi.org/10.1007/978-3-319-46448-0_2.

Bowen Lv, Xiao Liu, Yanyu Ren, Hanyu Lai, Bohao Jing, Hanchen Zhang, Yanxiao Zhao, Shuntian Yao, Jie Tang, and Yuxiao Dong. Scalecua: Scaling computer use agents with verifiable task synthesis and efficient online rl. *arXiv preprint arXiv:2607.11185*, 2026.

Zhenjie Mao, Yuhuan Yang, Chaofan Ma, Dongsheng Jiang, Jiangchao Yao, Ya Zhang, and Yanfeng Wang. Safire: Saccade-fixation reiteration with mamba for referring image segmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Yingqian Min, Kun Zhou, Yifan Li, Yuhuan Wu, Han Peng, Yifan Du, Wayne Xin Zhao, Min Yang, and Ji-Rong Wen. Improving vision-language models with perception-centric process reward models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 33099–33109, June 2026.

OpenAI. Gpt-5.6: Frontier intelligence that scales with your ambition. https://openai.com/index/gpt-5-6/, 2026. Accessed: 2026-09-16.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023.

Mengxue Qu, Yu Wu, Wu Liu, Xiaodan Liang, Jingkuan Song, Yao Zhao, and Yunchao Wei. Rio: A benchmark for reasoning intention-oriented objects in open environments. *Advances in Neural Information Processing Systems*, 36:43041–43056, 2023.

Qwen. Qwen3.8-27B. https://huggingface.co/Qwen/Qwen3.8-27B, 2026. Hugging Face model card.

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 *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, June 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/Redmon_You_Only_Look_CVPR_2016_paper.html.

Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. Pixellm: Pixel reasoning with large multimodal model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26364–26373. IEEE, 2024.

Johann Sawatzky, Yaser Souri, Christian Grund, and Jurgen Gall. What object should i use?-task driven object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7605–7614, 2019.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642, 2024a.

Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8429–8438, 2019. doi: 10.1109/ICCV.2019.00852.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024b.

Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. Blockwise parallel decoding for deep autoregressive models. *Advances in Neural Information Processing Systems*, 31, 2018.

Zhichao Sun, Yepeng Liu, Zhiling Su, Huachao Zhu, Yuliang Gu, Yuda Zou, Zelong Liu, Gui-Song Xia, Bo Du, and Yongchao Xu. Refdrone: A challenging benchmark for referring expression comprehension in drone scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 1752–1762, June 2026.

Qwen Team. Qwen3.5-omni technical report, 2026. URL https://arxiv.org/abs/2604.15804.

Unsplash. Unsplash dataset. https://unsplash.com/data, 2026. Accessed: 2026-09-19.

Junchi Wang and Lei Ke. Llm-seg: Bridging image segmentation and large language model reasoning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1765–1774. IEEE, 2024.

Shihao Wang, Shilong Liu, Yuanguo Kuang, Xinyu Wei, Yangzhou Liu, Zhiqi Li, Yunze Man, Guo Chen, Andrew Tao, Guilin Liu, et al. Locateanything: Fast and high-quality vision-language grounding with parallel box decoding. In *European Conference on Computer Vision*, pp. 336–357. Springer, 2026.

Wenxuan Wang, Tongtian Yue, Yisi Zhang, Longteng Guo, Xingjian He, Xinlong Wang, and Jing Liu. Unveiling parts beyond objects: Towards finer-granularity referring expression segmentation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12998–13008. IEEE, 2024.

Fangyun Wei, Jinjing Zhao, Kun Yan, Hongyang Zhang, and Chang Xu. A large-scale human-centric benchmark for referring expression comprehension in the lmm era. *Advances in Neural Information Processing Systems*, 37:69566–69587, 2024.

Dongming Wu, Yanping Fu, Saike Huang, Yingfei Liu, Fan Jia, Nian Liu, Feng Dai, Tiancai Wang, Rao Muhammad Anwer, Fahad Shahbaz Khan, et al. Ragnet: Large-scale reasoning-based affordance segmentation benchmark towards general grasping. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11980–11990. IEEE, 2025.

Chi Xie, Zhao Zhang, Yixuan Wu, Feng Zhu, Rui Zhao, and Shuang Liang. Described object detection: Liberating object detection with flexible expressions. *Advances in Neural Information Processing Systems*, 36:79095–79107, 2023.

Kai Ye, YingShi Luan, Zhudi Chen, Guangyue Meng, Pingyang Dai, and Liujuan Cao. Ris-lad: A benchmark and model for referring low-altitude drone image segmentation. *arXiv preprint arXiv:2507.20920*, 2025.

En Yu, Kangheng Lin, Liang Zhao, Jisheng Yin, Yana Wei, Yuang Peng, Haoran Wei, Jianjian Sun, Chunrui Han, Zheng Ge, Xiangyu Zhang, Daxin Jiang, Jingyu Wang, and Wenbing Tao. Perception-r1: Pioneering perception policy with reinforcement learning, 2025. URL https://arxiv.org/abs/2504.07954.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European conference on computer vision*, pp. 69–85. Springer, 2016.

Hao Zhang, Hongyang Li, Feng Li, Tianhe Ren, Xueyan Zou, Shilong Liu, Shijia Huang, Jianfeng Gao, Lei Zhang, Chunyuan Li, and Jianwei Yang. Llava-grounding: Grounded visual chat with large multimodal models, 2023. URL https://arxiv.org/abs/2312.02949.

Tianyi Zhao, Jiawen Xi, Linhui Xiao, Junnan Li, Xue Yang, Maoxun Yuan, and Xingxing Wei. Rgbt-groundbench: Visual grounding beyond rgb in complex real-world scenarios. *arXiv preprint arXiv:2512.24561*, 2025.

Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Heng Fan, Qinghua Hu, and Haibin Ling. Detection and tracking meet drones challenge. *IEEE transactions on pattern analysis and machine intelligence*, 44(11):7380–7399, 2021.

Appendix

In this Appendix, we provide the following materials:

1. Sec. A: detailed description of the localization data construction pipeline;
2. Sec. B: additional analysis of AR-GRPO, including structured action masking and fine-grained credit assignment;
3. Sec. C: formulation and analysis of GRPO for the MTP policy;
4. Sec. D: additional quantitative and qualitative evaluation results across detection, referring grounding, dense localization, reasoning, and cross-domain benchmarks;
5. Sec. E: detailed LVIS performance breakdown across dense, small-object, long-tail, and normal subsets;
6. Sec. F: detailed settings and analyses of the training-recipe ablations;
7. Sec. G: additional GeoSpec analysis, including coordinate tolerance, target-size sensitivity, verification behavior, box ordering, and draft rejection;
8. Sec. H: detailed accuracy–efficiency analysis of GeoSpec and multi-query inference.

A Detail of data construction

Dense localization. We combine 8,969 public OWDOD (Joseph et al., 2021) examples with **4.01M** constructed examples from the top-8% SA-1B (Kirillov et al., 2023) images ranked by ground-truth mask count. For each image, Qwen3.5-35B-A3B generates short category queries targeting repeated, individually distinguishable objects while excluding ambiguous textures and uncountable regions. SAM3 (Carion et al., 2026) and RexOmni (Jiang et al., 2026b) then independently localize each image–query pair.

As illustrated in Fig. A1, the two models provide complementary predictions, motivating cross-model pseudo-label merging. We first discard RexOmni outputs containing more than 100 boxes for a query, which are typically unreliable. The remaining SAM3 and RexOmni boxes are aligned by one-to-one Hungarian matching that maximizes total IoU. A matched pair is considered duplicate if its IoU exceeds 0.5, or if the intersection covers at least 90% of the smaller box and 60% of the larger box. Requiring substantial coverage of the larger box prevents near-containment alone from collapsing distinct instances: for example, a small background object may lie almost entirely inside the bounding box of a larger foreground object of the same category, yet should remain a separate instance. For duplicate pairs, we retain the tighter box. The thresholds and filtering rules are selected through a small grid search. The resulting dense-localization subset contains **4.01M** examples and over **100M** annotated boxes.

Referring grounding. Our referring-grounding corpus contains **9.39M** curated examples from 16 public datasets: aRefCOCO (Mao et al., 2025), GroundingCap-1M (Fu et al., 2025), GSEval (Hu et al., 2025), GVC (Zhang et al., 2023), HC-RefLoCo (Wei et al., 2024), Ref-Adv (Dong et al., 2026), RefCOCOm (Wang et al., 2024), REIRCOCO (Hao et al., 2025), ReSeDis (Huang et al., 2025), RGBT-GroundBench (Zhao et al., 2025), RIO (Qu et al., 2023), Talk2Car (Deruyttere et al., 2019), AgroVG (Li et al., 2026), RefDrone (Sun et al., 2026), RIS-LAD (Ye et al., 2025), and VRSBench (Li et al., 2024a). Together, these datasets cover natural, embodied, aerial, and remote-sensing imagery and provide substantial variation in object attributes, spatial relationships, target granularity, and scene composition.

We additionally construct approximately **4M** referring examples from LLaVA-OV-85M, Unsplash, and SA-1B. Given an image, Qwen3.5-35B-A3B (Team, 2026) generates fine-grained referring expressions emphasizing discriminative attributes, spatial relations, and object quantities, and Qwen3.8-27B (Qwen, 2026) predicts the corresponding bounding boxes. For expressions containing explicit cardinality constraints, we retain a sample only when the stated object count agrees with the number of predicted boxes, which removes a substantial class of inconsistent pseudo-labels. After combining public and constructed data, the referring-grounding subset contains **13.39M** examples.

Reasoning grounding. Our reasoning-grounding subset contains **727K** unique examples from ten datasets: ReasonSeg, ReasonSeg-Inst (Lai et al., 2024), MUSE (Ren et al., 2024), MMR (Jang et al.,

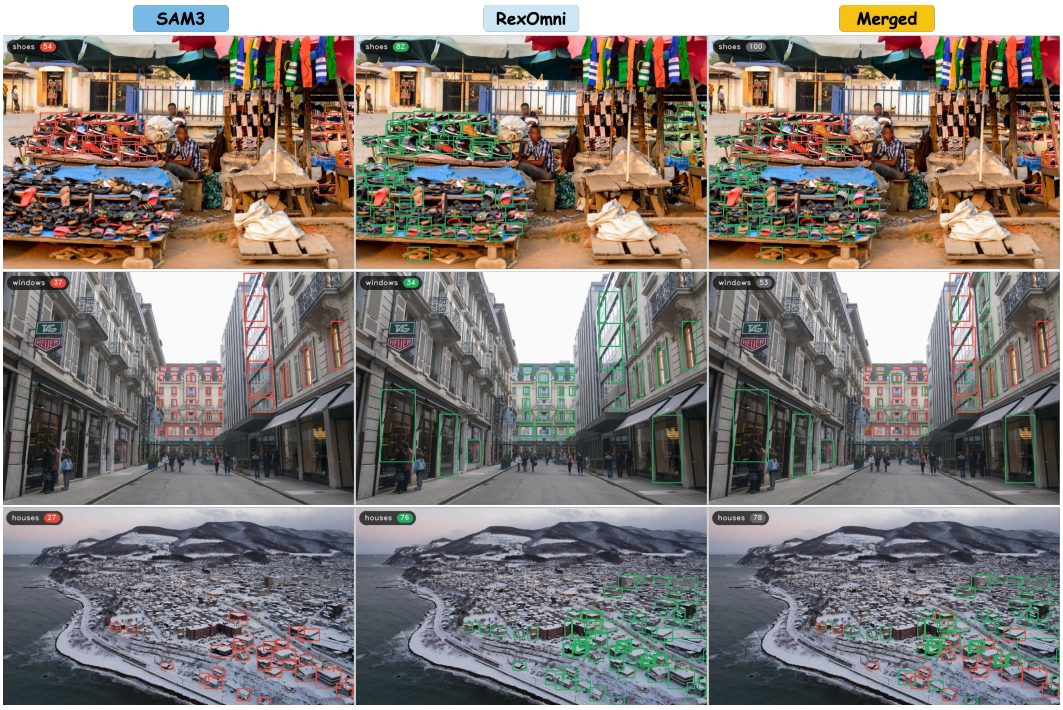

Figure A1: **Cross-model pseudo-label merging for dense localization.** SAM3 (Carion et al., 2026) and RexOmni (Jiang et al., 2026b) provide complementary predictions in densely populated scenes. We geometrically align their predictions and remove redundant boxes while preserving complementary instances, producing substantially more complete dense annotations. Red and green boxes in the merged results denote predictions originating from SAM3 and RexOmni, respectively.

2025), LLM-Seg40K (Wang & Ke, 2024), COCO-Tasks (Sawatzky et al., 2019), RAGNet (Wu et al., 2025), D-Cube (Xie et al., 2023), EarthReason (Li et al., 2025b), and FineCops-Ref (Liu et al., 2025). These datasets extend localization beyond explicit object names to queries involving object function, attributes, relations, spatial context, affordances, and implicit user intent. For segmentation-based datasets, each annotated target is converted into its tight bounding box without splitting disconnected components into separate targets. Examples without valid target regions are removed.

B More AR GRPO Analysis

We further investigate several variants of Group Relative Policy Optimization (GRPO) (Shao et al., 2024b) for AR policy. All variants use the localization reward R defined in Sec. 3.3, and each is evaluated against a matched sequence-level GRPO control under the same data and optimization settings.

Sequence-level GRPO. For each input prompt x, consisting of an image and a localization query, the rollout policy π_{old} samples a group of G responses $\{y_i\}_{i=1}^{G}$. Here, i indexes a response within the group, and $y_i = (y_{i,1}, \ldots, y_{i,T_i})$ contains T_i generated tokens. Each response receives a scalar localization reward $R_i = R(y_i)$. Following standard GRPO (Shao et al., 2024b), we define

$$\bar{R} = \frac{1}{G}\sum_{i=1}^{G} R_i, \qquad A_i = \frac{R_i - \bar{R}}{\sigma_R + \eta}, \tag{4}$$

where $\bar{R}$ and σ_R are respectively the mean and standard deviation of the G rewards for the same prompt, A_i is the group-relative advantage of response i, and η is a small constant for numerical stability. Thus, $A_i > 0$ indicates that response i performs better than the group average, while $A_i < 0$ indicates the opposite.

The same sequence-level advantage A_i is assigned to all generated tokens in response y_i. For token position t, we define the policy ratio

$$r_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\pi_{\text{old}}(y_{i,t} \mid x, y_{i,<t})}, \tag{5}$$

where $y_{i,t}$ is the sampled token at position t, $y_{i,<t}$ denotes its preceding tokens, π_θ is the policy being optimized with parameters θ, and π_{old} is the rollout policy that generated the response. Following the clipped GRPO objective, the token contribution is

$$\ell_{i,t} = \min(r_{i,t}(\theta)A_i,\ \text{clip}(r_{i,t}(\theta), 1-\delta, 1+\delta)\, A_i)\,, \tag{6}$$

where δ is the clipping range and $\text{clip}(z, l, u)$ restricts z to $[l, u]$. Our default sequence-level objective is

$$\mathcal{L}_{\text{seq}} = -\frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i}\ell_{i,t}, \tag{7}$$

where $\mathcal{L}_{\text{seq}}$ is the training loss. We use no additional KL penalty.

Structured action masking. Inspired by the structured treatment of autoregressive perception actions in Falcon Perception-HD (Chaybouti et al., 2026), we explicitly compress LAM's sampled action space according to its box-generation format. At the first action of each box, the policy chooses only between $\langle\text{box}\rangle$ and $\langle\text{EOS}\rangle$. Each of the following four coordinate actions is restricted to $\mathcal{C} = \{\langle 0\rangle, \ldots, \langle 1000\rangle\}$. The closing $\langle/\text{box}\rangle$ token is inserted deterministically and is excluded from the sampled action space. For sampled state s_t, the valid action set is therefore

$$\mathcal{V}(s_t) = \begin{cases} \{\langle\text{box}\rangle, \langle\text{EOS}\rangle\}, & t \text{ is a box-start position,} \\ \mathcal{C}, & t \text{ is a coordinate position.} \end{cases} \tag{8}$$

We renormalize the original full-vocabulary policy over this set:

$$\pi_\theta^{\text{mask}}(a \mid s_t) = \frac{\pi_\theta(a \mid s_t)\,\mathbf{1}[a \in \mathcal{V}(s_t)]}{\sum_{v\in\mathcal{V}(s_t)}\pi_\theta(v \mid s_t)}. \tag{9}$$

Here, a is a candidate token, $\pi_\theta(a \mid s_t)$ is its original full-vocabulary probability, $\pi_\theta^{\text{mask}}(a \mid s_t)$ is the renormalized probability, $\mathbf{1}[\cdot]$ is the indicator function, and v indexes the valid actions in $\mathcal{V}(s_t)$. Although this compressed action space eliminates structurally invalid sampled actions, it does not improve performance relative to its matched sequence-level GRPO control under standard decoding.

Fine-grained credit assignment. Standard GRPO assigns the same sequence-level advantage A_i to every token, even when only part of a predicted box is responsible for a localization error. Perceval (Min et al., 2026) addresses a related credit-assignment problem using token-level process supervision. Inspired by this idea, we construct a GT-privileged counterfactual verifier for localization.

We first match predicted and ground-truth boxes. For each matched predicted box, we replace one coordinate at a time with its corresponding ground-truth coordinate while keeping the matching fixed. A coordinate is marked as harmful if this replacement allows the matched pair to satisfy additional predefined IoU thresholds. For an unmatched predicted box, its box-start action is marked as harmful. Let $m_{i,t} \in \{0, 1\}$ denote this error indicator for token t in response i. We define

$$A_{i,t}^{\text{perc}} = A_i - \lambda m_{i,t}|A_i|, \qquad \lambda = 0.1, \tag{10}$$

where $A_{i,t}^{\text{perc}}$ is the token-specific advantage, A_i is the original sequence-level advantage, $m_{i,t}$ indicates whether the action is identified as harmful, and λ controls the strength of the local correction.

We evaluate two credit granularities. The *token-level* variant modifies only the identified coordinate or box-start action. The *box-level* variant broadcasts the correction to the five semantic actions of the corresponding box, namely its box-start token and four coordinate tokens. Both variants use the same counterfactual verifier. Neither provides a consistent improvement over sequence-level GRPO across our fixed validation sets, and box-level credit does not outperform token-level credit.

Overall, neither structured action masking nor finer token- or box-level credit assignment provides a stable advantage over clean sequence-level GRPO. We therefore adopt the simpler sequence-level AR-GRPO formulation in LAM++.

C MTP GRPO ANALYSIS

We further investigate whether GRPO (Shao et al., 2024b) can directly optimize LAM's MTP policy. Multi-token prediction (Gloeckle et al., 2024b) predicts multiple tokens in parallel, and LAM (Wang et al., 2026) adapts this mechanism to box generation with six parallel prediction positions. Native LAM MTP inference, however, applies format handling and post-processing to raw model predictions before they are used by subsequent decoding steps. Directly optimizing this processed trajectory would make the sampled raw actions differ from the actions represented in the executed trajectory. We therefore define an explicit *raw MTP policy* that preserves action–trajectory consistency during GRPO.

Raw MTP policy. The category prefix $\langle \mathrm{ref} \rangle query \langle / \mathrm{ref} \rangle$ is provided deterministically and excluded from the policy loss. At MTP step b, the model produces six full-vocabulary categorical distributions in parallel. For response i, we denote the sampled block by

$$\mathbf{a}_{i,b} = (a_{i,b,0}, a_{i,b,1}, \ldots, a_{i,b,5}), \tag{11}$$

where i indexes a sampled response, b indexes the MTP decoding step, $j \in \{0, \ldots, 5\}$ indexes one of the six parallel positions, and $a_{i,b,j}$ is the token sampled at position j. We use $h_{i,b}$ to denote the committed raw history before block b.

Termination is controlled only by position 0. If $a_{i,b,0} = \langle \mathrm{EOS} \rangle$, only this EOS action is committed and generation terminates. Otherwise, all six sampled positions are committed. We denote the committed action positions by

$$\mathcal{I}_{i,b} = \begin{cases} \{0\}, & a_{i,b,0} = \langle \mathrm{EOS} \rangle, \\ \{0, 1, \ldots, 5\}, & \text{otherwise}, \end{cases} \tag{12}$$

where $\mathcal{I}_{i,b}$ is therefore either the single termination action or all six positions of a non-terminal block. EOS tokens sampled at positions 1–5 do not terminate generation.

No grammar constraint or corrective post-processing is applied to the raw training trajectory. Invalid markup, non-coordinate tokens, malformed coordinates, and other raw predictions are committed exactly as sampled and are scored as part of the resulting sequence. This ensures that policy optimization is performed on the actions actually sampled from the raw MTP distribution.

Exact policy replay. For each sampled action, we preserve its block boundary, committed history, parallel position, and rollout log-probability. During replay, the same raw history and sampled token are used without applying the native MTP post-processing. For position j of block b, the policy ratio is

$$r_{i,b,j}(\theta) = \frac{\pi_\theta(a_{i,b,j} \mid h_{i,b}, j)}{\pi_{\mathrm{old}}(a_{i,b,j} \mid h_{i,b}, j)}, \tag{13}$$

where π_{old} is the rollout policy, π_θ is the policy being optimized, $h_{i,b}$ is the committed raw history before block b, and j identifies the parallel prediction position. The numerator and denominator therefore evaluate the same sampled token under the same raw MTP state.

Using the sequence-level group-relative advantage A_i defined in Sec. B, each sampled MTP action contributes

$$\ell_{i,b,j} = \min(r_{i,b,j}(\theta) A_i,\ \mathrm{clip}(r_{i,b,j}(\theta), 1-\delta, 1+\delta)\, A_i)\,, \tag{14}$$

where δ is the same clipping range used in AR-GRPO and $\ell_{i,b,j}$ is the clipped objective for that sampled MTP action.

Let

$$N_i = \sum_{b=1}^{B_i} |\mathcal{I}_{i,b}| \tag{15}$$

be the total number of sampled MTP actions in response i, where B_i is its number of generated MTP blocks and $|\mathcal{I}_{i,b}|$ is the number of committed actions in block b. The MTP objective is

$$\mathcal{L}_{\mathrm{MTP}} = -\frac{1}{G} \sum_{i=1}^{G} \frac{1}{N_i} \sum_{b=1}^{B_i} \sum_{j \in \mathcal{I}_{i,b}} \ell_{i,b,j}, \tag{16}$$

Table A1: **Detailed results on SACo-Gold.** We report positive-query F1 and recall (%) at representative IoU thresholds. The overall columns report mF1 and mRecall averaged over IoU thresholds.

Model / Decode	IoU 0.50		IoU 0.60		IoU 0.70		IoU 0.80		IoU 0.90		Overall	
	F1	R	F1	R	F1	R	F1	R	F1	R	mF1	mR
SAM3	62.74	64.88	61.28	63.37	59.16	61.21	55.44	57.42	47.12	48.82	55.50	57.44
RexOmni	65.05	59.76	61.49	56.59	57.12	52.73	50.67	47.02	38.77	36.45	52.16	48.29
Qwen-3.8-27B	73.76	71.99	69.93	68.31	64.77	63.35	57.56	56.43	45.42	44.67	59.79	58.52
LAM AR	56.20	52.15	52.72	49.09	49.04	45.83	44.09	41.34	36.54	34.47	46.02	43.03
LAM Hybrid	57.45	53.47	53.84	50.31	49.88	46.79	44.86	42.28	36.43	34.58	46.63	43.80
LAM++ Hybrid	75.59	78.27	72.49	75.03	68.68	71.05	63.18	65.35	53.42	55.24	64.67	66.71
LAM++ GeoSpec	75.56	77.58	72.56	74.48	68.74	70.56	63.43	65.11	53.98	55.42	**64.92**	**66.92**

where G is the number of sampled responses per prompt and $\mathcal{L}_{\text{MTP}}$ is the resulting raw-MTP GRPO loss. Thus, AR-GRPO and MTP-GRPO use the same sequence-level reward and group-relative objective, while differing in how their policy actions and trajectories are constructed.

Optimization behavior. MTP-GRPO is considerably more expensive to train in our implementation because on-policy optimization requires block-wise rollout and replay together with the corresponding KV-history management. For matched-scale, one-epoch runs, MTP-GRPO takes approximately $2.7\times$ the wall-clock training time of AR-GRPO.

Despite optimizing only the raw MTP policy, MTP-GRPO also changes the AR policy through their shared model parameters, providing the reverse direction of the cross-policy transfer observed with AR-GRPO. In our controlled LVIS study, MTP-GRPO increases overall precision but decreases recall for both AR and native MTP decoding relative to the SFT initialization. This more conservative operating point is undesirable for dense localization, where incomplete instance coverage is a primary failure mode. We therefore use AR-GRPO in the final training recipe and retain MTP as the efficient drafting policy for GeoSpec.

D More Evaluation Results

D.1 Detection

We provide detailed detection results on SACo-Gold (Carion et al., 2026), COCO (Lin et al., 2015), and LVIS (Gupta et al., 2019) in Tabs. A1, A2, and A3, respectively. Across all three benchmarks, LAM++ substantially improves over the original LAM and remains consistently strong across a wide range of IoU thresholds. LAM++ GeoSpec achieves the strongest overall mF1 among the evaluated LAM-family decoding strategies and remains highly competitive with substantially larger VLMs and dedicated localization models. Importantly, the gains persist under stricter IoU thresholds, indicating that the improvement comes not only from better instance coverage but also from more accurate box geometry. The qualitative comparisons in Figs. A2 and A3 are consistent with this trend, showing fewer missed instances and more precise localization on representative examples. Together, these results demonstrate that LAM++ improves both detection completeness and fine-grained geometric accuracy across diverse detection settings.

D.2 Referring Grounding and Cross-Domain Generalization

We report detailed referring-grounding results on HumanRef (Jiang et al., 2025), RefCOCOg (Yu et al., 2016), and RefBench-PRO (Gao et al., 2025), together with cross-domain evaluation on VisDrone (Zhu et al., 2021) and Talk2Car (Deruyttere et al., 2019), in Tabs. A4, A5, A6, A7, and A8. Across all five benchmarks, LAM++ substantially improves over the original LAM under both AR and Hybrid decoding, showing that the gains transfer beyond generic detection to fine-grained referring expressions and distinct visual domains. GeoSpec is also the strongest or near-strongest LAM++ decoding strategy throughout these evaluations.

On HumanRef and RefCOCOg, LAM++ remains competitive with substantially larger general-purpose VLMs while maintaining particularly strong localization at strict IoU thresholds. For exam-

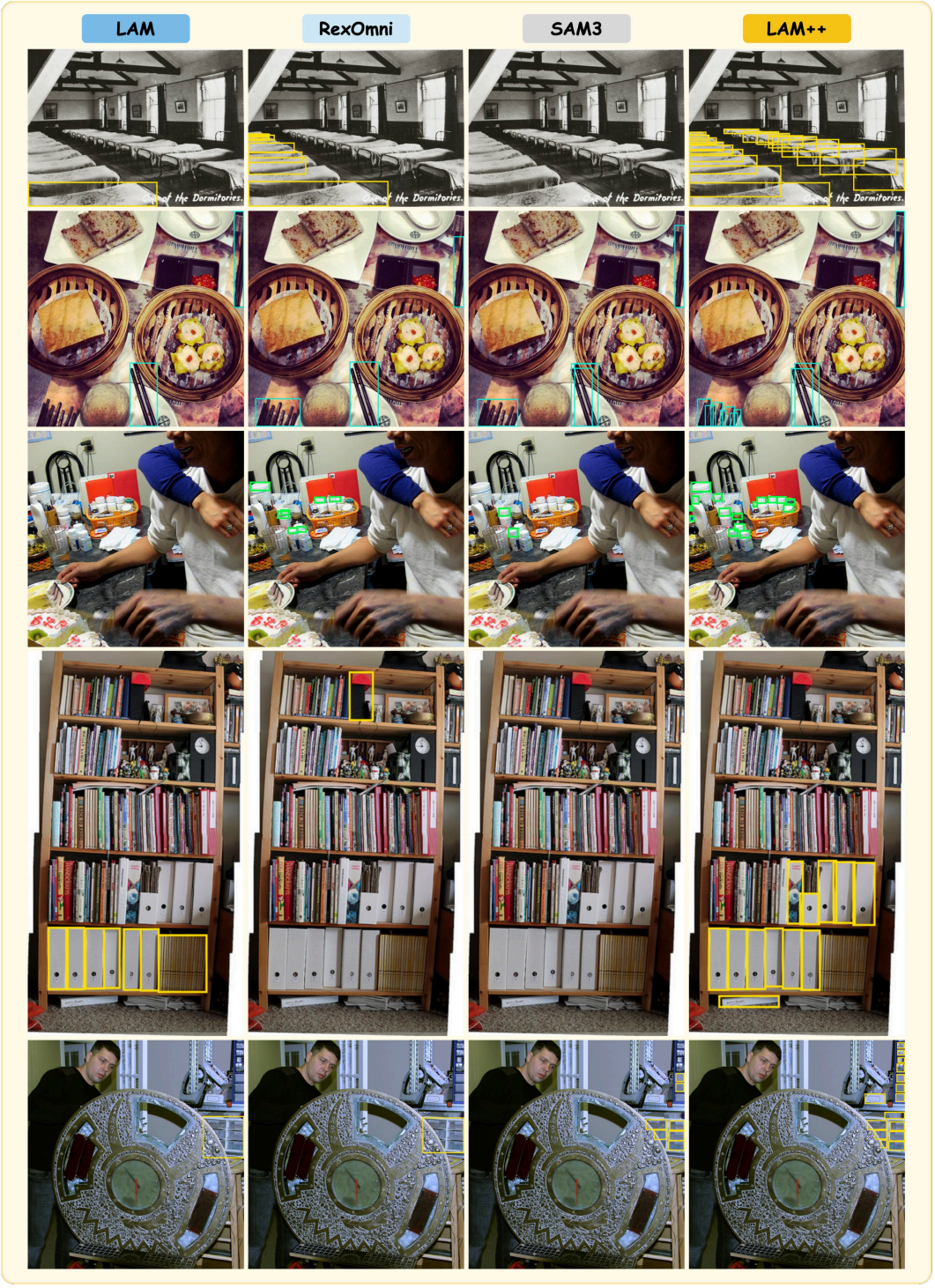

Figure A2: **Qualitative results on LVIS.** LAM++ provides more complete instance coverage and more precise box localization than the original LAM on representative detection examples.

ple, at IoU 0.9, GeoSpec reaches 68.75 F1 on RefCOCOg, exceeding Qwen-3.8-27B (Qwen, 2026) and RexOmni (Jiang et al., 2026b) despite their competitive overall scores. RefBench-PRO remains more challenging, yet GeoSpec substantially improves over both the original LAM and RexOmni. The cross-domain results show a similar pattern: on VisDrone, GeoSpec approaches the dedicated



Figure A3: **Qualitative results on SACo-Gold.** LAM++ more reliably localizes individual targets with accurate boxes, particularly in challenging scenes with multiple instances.

SAM3 (Carion et al., 2026) while outperforming the evaluated VLM baselines, whereas on Talk2Car it surpasses all evaluated open-source baselines and remains competitive with closed-source frontier VLMs. Together, these results indicate that LAM++ improves not only instance coverage but also precise geometric grounding across diverse referring formulations and visual domains.

D.3 Dense Scene

We further evaluate dense localization on Dense200 (Jiang et al., 2026b), which contains challenging scenes with many small and repeated instances. Detailed results are reported in Tab. A9. LAM++ GeoSpec achieves the strongest overall performance, reaching **73.43 mF1** and **71.51 mRecall**, substantially outperforming the original LAM, RexOmni, SAM3, and substantially larger VLMs. Compared with LAM Hybrid, LAM++ improves mF1 from 55.59 to 73.43 and mRecall from 53.09 to 71.51, demonstrating a large reduction in missed instances under crowded conditions.

Table A2: **Detailed results on COCO val2017.** We report positive-query F1 and recall (%) at representative IoU thresholds. The overall columns report mF1 and mRecall averaged over IoU.

Model / Decode	IoU 0.50		IoU 0.60		IoU 0.70		IoU 0.80		IoU 0.90		Overall	
	F1	R	F1	R	F1	R	F1	R	F1	R	mF1	mR
Gemini-3.7-Flash	74.09	72.77	71.23	70.02	66.96	65.93	59.42	58.64	41.65	41.32	59.34	58.47
GPT-5.6-Luna	62.07	61.52	55.29	54.83	47.06	46.68	37.19	36.93	22.43	22.31	41.83	41.51
GPT-5.6-Sol	68.94	69.70	63.32	63.85	56.28	56.65	46.38	46.63	30.22	30.37	49.86	50.24
Qwen-3.8-27B	75.79	75.36	70.65	70.32	63.82	63.57	53.63	53.46	35.91	35.79	56.60	56.36
SAM3	77.08	79.72	75.05	77.62	71.62	74.08	64.84	67.09	46.57	48.14	63.67	65.84
RexOmni	77.08	75.37	72.32	70.89	65.97	64.85	56.09	55.30	37.96	37.54	58.41	57.40
LAM AR	75.19	73.52	71.22	69.76	65.99	64.73	57.61	56.57	41.39	40.75	59.04	57.91
LAM Hybrid	74.47	72.99	70.08	68.83	64.37	63.32	55.99	55.18	39.24	38.77	57.55	56.60
LAM++ Hybrid	81.90	82.43	78.44	78.98	73.90	74.40	65.92	66.36	48.37	48.60	66.26	66.68
LAM++ GeoSpec	81.97	82.11	78.67	78.85	74.33	74.51	66.81	67.00	48.94	49.03	**66.69**	**66.83**

Table A3: **Detailed results on LVIS.** We report positive-query F1 and recall (%) at representative IoU thresholds. The overall columns report mF1 and mRecall averaged over IoU thresholds.

Model / Decode	IoU 0.50		IoU 0.60		IoU 0.70		IoU 0.80		IoU 0.90		Overall	
	F1	R	F1	R	F1	R	F1	R	F1	R	mF1	mR
GPT-5.6-Sol	64.94	63.24	58.34	56.89	49.99	48.81	38.89	38.07	24.11	23.70	44.34	43.31
Qwen-3.8-27B	69.48	66.19	63.57	60.73	55.74	53.47	45.26	43.60	29.61	28.71	49.69	47.66
SAM3	70.29	69.22	68.33	67.40	65.41	64.55	59.88	59.23	46.92	46.66	59.61	58.86
RexOmni	68.90	64.81	63.70	60.18	56.73	53.87	46.72	44.66	30.69	29.60	50.33	47.80
LAM++ Hybrid	78.91	76.20	75.33	72.88	70.74	68.56	63.57	61.74	49.81	48.57	64.89	62.93
LAM++ GeoSpec	79.24	75.97	75.82	72.80	71.35	68.63	64.46	62.13	50.70	49.07	**65.56**	**63.11**

The advantage remains pronounced under stricter localization criteria. At IoU 0.9, LAM++ GeoSpec achieves 54.00 F1, compared with 39.50 for SAM3, 28.95 for RexOmni, and 42.92 for Qwen-3.8-Max. This indicates that the gain does not come merely from predicting more boxes: LAM++ also preserves accurate box geometry for densely packed objects. Together with the qualitative examples in Fig. 5, these results show that LAM++ substantially alleviates the under-localization behavior of previous generative localizers, which often miss repeated instances or merge multiple nearby objects into coarse predictions.

D.4 Reasoning

We further evaluate reasoning-intensive grounding on ReasonSeg (Lai et al., 2024), with detailed results reported in Tab. A10. LAM++ substantially improves over the original LAM and RexOmni, demonstrating that the proposed post-training recipe also transfers to queries requiring semantic reasoning beyond explicit object references. The advantage remains pronounced under strict localization criteria: at IoU 0.9, GeoSpec achieves 46.82 F1, outperforming Qwen-3.8-27B and substantially exceeding LAM AR and RexOmni. This suggests that LAM++ improves reasoning-based target identification while retaining accurate box geometry.

E Performance Details on LVIS

To better understand where each post-training component improves localization, we provide the complete LVIS breakdown in Tab. A11. We partition LVIS queries into four diagnostic subsets: *dense*, containing at least eight ground-truth boxes; *small*, whose mean target-box area is below 0.5% of the image area; *long-tail*, corresponding to LVIS categories marked as rare; and *normal*, containing none of the above properties. The statistics of these subsets are summarized in Tab. A12. The failure subsets are not mutually exclusive, since a query may simultaneously contain dense, small, and long-tail targets. Notably, small-target cases account for 32.6% of the validation queries but only 13.04% of the training queries, making the validation set substantially more challenging.

Table A4: **Detailed results on HumanRef.** We report positive-query F1 and recall (%) at representative IoU thresholds. mF1 and mR are averaged over all IoU thresholds from 0.50 to 0.95.

Model / Decode	IoU 0.5		IoU 0.6		IoU 0.7		IoU 0.8		IoU 0.9		Overall	
	F1	R	F1	R	F1	R	F1	R	F1	R	mF1	mR
Closed-source VLMs												
Gemini-3.7-Flash	90.12	88.39	88.43	86.87	85.84	84.41	82.40	81.10	72.11	71.09	**81.15**	**79.82**
GPT-5.6-Luna	83.55	80.44	79.71	76.79	73.88	71.22	64.12	61.97	46.10	44.68	66.19	63.89
GPT-5.6-Sol	87.36	83.90	85.46	82.07	82.18	78.91	76.41	73.38	63.66	61.20	76.63	73.61
Open-source and Dedicated Models												
Qwen-3.8-27B	84.63	79.74	83.12	78.31	80.89	76.22	77.60	73.11	71.02	67.01	78.10	73.62
SAM3	24.62	27.96	24.20	27.55	23.72	27.05	23.12	26.37	21.68	24.78	22.97	26.17
RexOmni	85.03	85.17	83.84	83.99	82.00	82.15	79.21	79.32	74.13	74.23	79.45	79.58
LAM Family												
LAM AR	72.45	71.79	71.55	70.91	70.01	69.38	67.88	67.29	64.01	63.42	68.34	67.72
LAM Hybrid	71.29	71.15	70.39	70.27	68.72	68.60	66.47	66.40	62.44	62.28	66.91	66.79
LAM++ Hybrid	82.20	84.07	80.92	82.75	79.29	81.08	77.00	78.70	72.58	74.13	77.31	79.02
LAM++ GeoSpec	83.45	84.99	82.64	83.25	80.38	82.01	77.91	79.12	73.14	74.75	**79.81**	**79.14**

Table A5: **Detailed results on RefCOCOg test.** We report positive-query F1 and recall (%) at representative IoU thresholds. mF1 and mR are averaged over all IoU thresholds from 0.50 to 0.95.

Model / Decode	IoU 0.5		IoU 0.6		IoU 0.7		IoU 0.8		IoU 0.9		Overall	
	F1	R	F1	R	F1	R	F1	R	F1	R	mF1	mR
Closed-source VLMs												
Gemini-3.7-Flash	90.83	92.00	88.91	90.01	85.67	86.71	81.14	82.07	67.31	68.09	**79.05**	**80.00**
GPT-5.6-Luna	83.34	84.08	78.26	78.89	70.69	71.21	58.81	59.20	38.05	38.27	61.74	62.21
GPT-5.6-Sol	88.48	89.08	84.98	85.49	80.14	80.57	72.15	72.48	53.30	53.51	71.85	72.24
Open-source and Dedicated Models												
Qwen-3.8-27B	90.45	90.54	87.90	87.98	84.25	84.32	78.06	78.12	62.21	62.25	76.88	76.95
SAM3	39.14	46.44	38.36	45.52	37.20	44.12	35.48	42.10	30.03	35.55	34.56	40.98
RexOmni	88.06	88.09	85.74	85.77	82.17	82.20	76.66	76.68	62.75	62.78	75.48	75.50
LAM Family												
LAM AR	74.26	74.93	72.30	72.94	69.82	70.42	65.68	66.21	55.21	55.62	64.75	65.30
LAM Hybrid	55.84	56.73	54.48	55.33	52.34	53.13	49.43	50.14	41.20	41.77	48.55	49.27
LAM++ Hybrid	87.96	88.78	85.59	86.38	82.39	83.15	77.37	78.08	66.37	66.98	76.82	77.52
LAM++ GeoSpec	90.25	89.81	88.08	87.25	84.94	84.04	79.90	79.04	68.75	67.61	**79.45**	**77.94**

Failure analysis of LAM. The detailed results confirm that LAM's performance gap is highly concentrated in dense scenes and small targets. Under AR decoding, LAM trails SAM3 by only 0.59 mF1 on long-tail cases, but the gap increases to 7.87 on normal cases, 16.77 on small-target cases, and 23.51 on dense cases. The degradation is even larger under MTP decoding, particularly for dense scenes, where the original LAM achieves only 12.61 mF1. These results motivate our failure-driven post-training strategy rather than uniformly scaling supervision across all LVIS cases.

Effect of supervised post-training. Both randomly sampled silver labels (RSL) and failure-focused silver labels (SSL) substantially improve the original LAM. Compared with RSL, SSL further improves AR performance on dense cases from 27.14 to 28.82 mF1 and on small targets from 29.97 to 30.57 mF1, confirming that targeted data selection improves the intended failure modes. However, this specialization slightly reduces performance on normal cases and therefore lowers the overall

Table A6: **Detailed results on RefBench-PRO.** We report positive-query F1 and recall (%) at representative IoU thresholds. mF1 and mR are averaged over all IoU thresholds from 0.50 to 0.95.

Model / Decode	IoU 0.5		IoU 0.6		IoU 0.7		IoU 0.8		IoU 0.9		Overall	
	F1	R	F1	R	F1	R	F1	R	F1	R	mF1	mR
Closed-source VLMs												
Gemini-3.7-Flash	82.44	83.58	78.72	79.80	72.94	73.96	63.17	64.06	43.65	44.26	**64.59**	**65.49**
GPT-5.6-Luna	74.00	74.42	68.30	68.66	60.61	60.92	48.93	49.18	29.05	29.16	52.68	52.95
GPT-5.6-Sol	79.98	80.40	75.28	75.66	68.21	68.52	57.24	57.52	38.21	38.36	60.13	60.42
Open-source and Dedicated Models												
Qwen-3.8-27B	78.47	78.82	74.68	75.00	69.08	69.38	60.36	60.62	42.43	42.58	61.67	61.93
SAM3	29.08	38.66	27.66	36.82	25.72	34.26	22.98	30.52	15.71	20.78	22.95	30.44
RexOmni	62.19	66.02	59.62	63.28	55.11	58.54	48.41	51.42	36.11	38.18	49.85	52.88
LAM Family												
LAM AR	46.64	48.26	44.58	46.14	41.27	42.70	37.00	38.30	27.57	28.46	37.61	38.89
LAM Hybrid	43.72	45.62	41.59	43.36	38.34	39.98	34.03	35.46	24.96	25.90	34.77	36.21
LAM++ Hybrid	69.93	73.54	66.60	70.04	61.49	64.62	53.56	56.10	37.86	39.50	54.95	57.65
LAM++ GeoSpec	70.15	73.28	66.81	69.82	61.96	64.76	54.39	56.74	38.66	40.22	**55.42**	**57.84**

Table A7: **Detailed results on VisDrone test.** We report positive-query F1 and recall (%) at representative IoU thresholds. mF1 and mR are averaged over all IoU thresholds from 0.50 to 0.95.

Model / Decode	IoU 0.5		IoU 0.6		IoU 0.7		IoU 0.8		IoU 0.9		Overall	
	F1	R	F1	R	F1	R	F1	R	F1	R	mF1	mR
Closed-source VLMs												
Gemini-3.7-Flash	23.06	20.04	21.42	18.73	18.53	16.35	13.59	12.11	5.27	4.79	15.17	13.36
GPT-5.6-Luna	25.36	23.92	20.12	18.97	14.28	13.57	8.09	7.65	2.24	2.12	12.69	12.00
GPT-5.6-Sol	35.45	35.24	30.16	29.86	23.38	22.98	14.80	14.53	4.47	4.41	19.83	19.60
Open-source and Dedicated Models												
Qwen-3.8-27B	44.27	47.95	40.18	42.93	34.26	36.10	25.79	26.69	11.37	11.57	28.83	30.51
SAM3	53.45	54.15	49.36	50.23	42.89	43.99	32.37	33.42	14.14	14.71	**35.66**	**36.48**
RexOmni	41.09	37.46	36.17	33.08	29.51	27.07	20.34	18.75	7.20	6.67	24.70	22.65
LAM Family												
LAM AR	41.94	38.32	37.41	34.19	31.93	29.21	24.13	22.08	11.08	10.14	27.10	24.79
LAM Hybrid	37.45	34.47	33.39	30.75	28.58	26.31	21.04	19.43	9.08	8.33	23.96	22.07
LAM++ Hybrid	47.85	52.96	43.56	48.37	37.53	41.94	28.37	31.95	12.82	14.48	31.53	35.19
LAM++ GeoSpec	49.69	53.82	45.21	49.11	38.86	42.49	29.49	32.53	13.47	14.95	32.79	35.83

mF1 from 54.13 to 53.17. This behavior is consistent with the composition of LVIS: normal queries still dominate both training and validation, while aggressive dense-scene supervision encourages longer predictions and can increase over-generation on ordinary scenes.

Reformulating SSL training into the single-query setting consistently improves both decoding policies, increasing overall mF1 to 54.14 for AR and 52.43 for MTP. The largest SFT improvement then comes from high-resolution visual processing. It raises overall AR mF1 from 54.14 to 64.16, with especially large gains on dense cases (29.80 $\rightarrow$ 43.16) and small targets (31.19 $\rightarrow$ 47.13). At this stage, the AR policy already surpasses the SAM3 reference on every LVIS subset, suggesting that insufficient visual granularity is a major source of the original localization gap.

Table A8: **Detailed results on Talk2Car test.** We report positive-query F1 and recall (%) at representative IoU thresholds. mF1 and mR are averaged over all IoU thresholds from 0.50 to 0.95.

Model / Decode	IoU 0.5		IoU 0.6		IoU 0.7		IoU 0.8		IoU 0.9		Overall	
	F1	R	F1	R	F1	R	F1	R	F1	R	mF1	mR
Closed-source VLMs												
Gemini-3.7-Flash	75.40	75.40	65.14	65.14	44.99	44.99	16.63	16.63	1.68	1.68	**36.94**	**36.94**
GPT-5.6-Luna	59.17	59.17	46.79	46.79	29.30	29.30	8.50	8.50	1.02	1.02	25.87	25.87
GPT-5.6-Sol	66.82	66.82	55.41	55.41	37.43	37.43	13.32	13.32	1.14	1.14	31.52	31.52
Open-source and Dedicated Models												
Qwen-3.8-27B	60.03	60.03	48.63	48.63	30.69	30.69	9.11	9.11	0.53	0.53	26.60	26.60
SAM3	9.76	10.67	8.79	9.60	6.10	6.66	1.49	1.68	0.20	0.20	4.71	5.16
RexOmni	34.88	37.39	29.63	31.75	21.13	22.64	7.98	8.66	0.77	0.86	17.13	18.39
LAM Family												
LAM AR	47.55	47.90	40.33	40.58	27.96	28.16	10.02	10.13	1.23	1.23	22.96	23.13
LAM Hybrid	46.88	47.32	39.57	39.93	27.44	27.71	9.36	9.48	1.08	1.10	22.43	22.64
LAM++ Hybrid	66.24	66.57	55.39	55.62	40.17	40.34	14.35	14.38	1.96	1.96	32.36	32.49
LAM++ GeoSpec	67.99	68.12	57.15	57.25	40.98	41.07	14.67	14.71	2.04	2.04	**33.25**	**33.31**

Table A9: **Detailed results on Dense200.** We report positive-query F1 and recall (%) at representative IoU thresholds. mF1 and mR are averaged over all IoU thresholds from 0.50 to 0.95.

Model / Decode	IoU 0.5		IoU 0.6		IoU 0.7		IoU 0.8		IoU 0.9		Overall	
	F1	R	F1	R	F1	R	F1	R	F1	R	mF1	mR
Closed-source VLMs												
Gemini-3.7-Flash	34.66	22.66	33.58	21.91	31.62	20.64	27.09	17.46	17.13	10.68	27.01	17.51
GPT-5.6-Luna	44.53	42.05	35.53	33.22	26.42	24.51	17.27	15.93	6.46	5.93	23.66	22.09
GPT-5.6-Sol	66.85	66.19	58.35	57.61	47.53	46.82	32.78	32.22	12.72	12.50	40.11	39.57
Qwen-3.8-Max	79.00	76.96	76.62	74.69	72.17	70.61	64.51	63.24	42.92	42.16	**63.08**	**61.67**
Open-source and Dedicated Models												
Qwen-3.8-27B	69.54	68.93	65.84	65.13	60.73	60.04	51.90	51.23	32.94	32.47	52.73	52.12
SAM3	67.64	64.87	66.15	63.40	63.51	60.83	58.07	55.54	39.50	37.95	55.34	53.04
SAM3.1	67.61	64.78	66.11	63.30	63.39	60.65	58.07	55.50	39.31	37.67	55.29	52.92
RexOmni	74.43	72.82	71.24	69.68	65.55	64.13	53.13	51.88	28.95	28.03	54.75	53.49
LAM Family												
LAM AR	68.46	66.44	65.90	64.02	62.28	60.44	55.19	53.53	39.19	37.99	55.24	53.62
LAM Hybrid	71.33	68.01	68.18	65.08	63.03	60.18	54.93	52.49	37.08	35.57	55.59	53.09
LAM++ Hybrid	87.22	84.73	85.04	82.63	81.55	79.28	74.78	72.81	52.84	51.60	71.88	69.93
LAM++ GeoSpec	88.66	86.22	86.73	84.33	83.39	81.12	76.71	74.75	54.00	52.84	**73.43**	**71.51**

Effect of reinforcement learning. AR-GRPO further improves the high-resolution SFT model, reaching 66.21 overall mF1 when trained on full LVIS. Importantly, the failure-focused LVIS-p subset uses only about 40% of the training data yet reaches a comparable 65.87 mF1, while slightly improving the dense and small subsets to 45.59 and 49.57, respectively. This supports the use of targeted difficult examples for RL rather than simply increasing rollout volume. AR-GRPO also improves MTP decoding without directly optimizing the MTP policy, increasing its overall mF1 from 62.48 after SFT to 63.85, proving clear cross-policy transfer through the shared parameters.

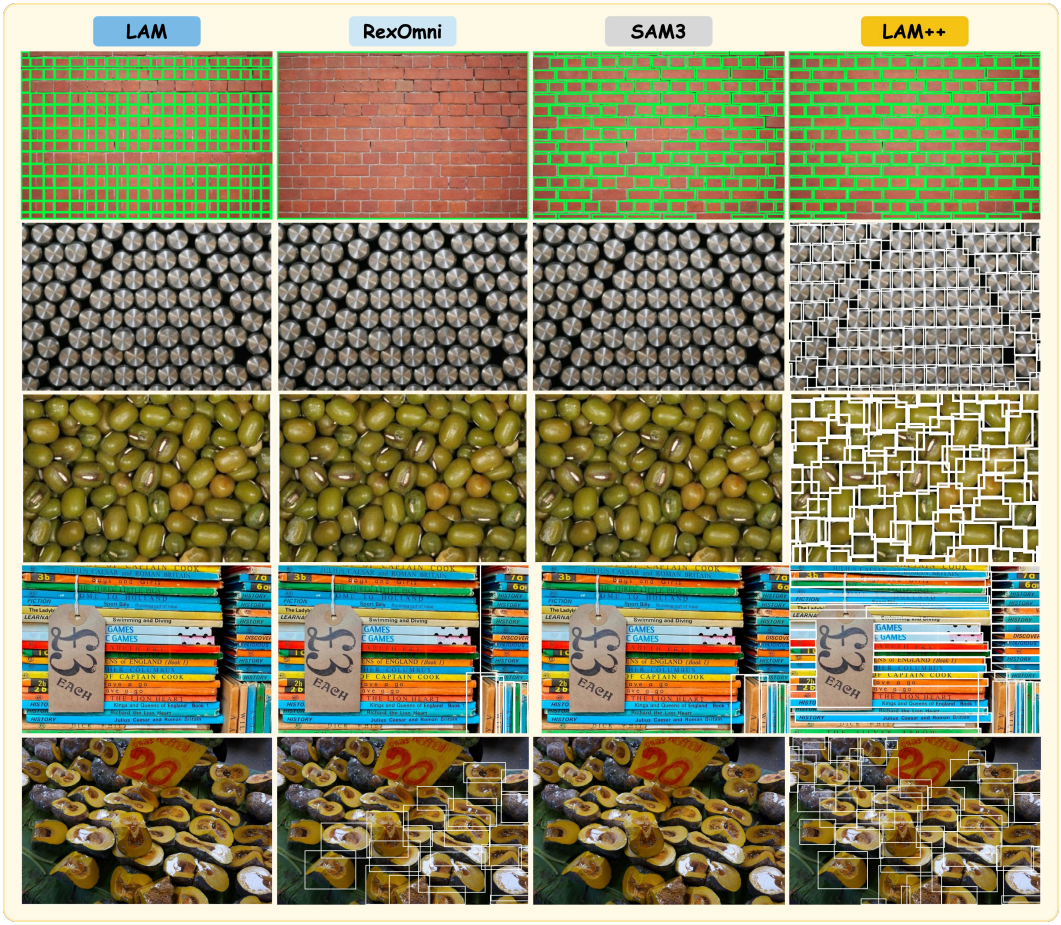

Figure A4: Visualization results on Dense200.

Direct MTP-GRPO exhibits the reverse transfer as well: optimizing MTP raises AR mF1 from 64.16 after SFT to 65.71. However, the resulting policy is noticeably more conservative. For example, dense-case AR recall drops from 42.01 with AR-GRPO on LVIS-p to 36.09 with MTP-GRPO, and overall recall decreases from 65.18 to 62.15. Thus, although MTP-GRPO confirms bidirectional transfer between the two decoding policies, it does not resolve the coverage errors that dominate dense localization. We consequently use AR-GRPO as the final RL post-training strategy.

F Detailed Training Recipe Ablations

We conduct controlled training ablations on LVIS (Gupta et al., 2019), which provides high-quality annotations and substantial variation in target size, scene density, and category frequency. The following experiments analyze the individual training choices summarized in Fig. 6.

Failure-driven data selection. We partition LVIS cases into four subsets: *dense*, with more than eight ground-truth boxes per query; *small*, with mean target-box area below 0.5% of the image area; *long-tail*, containing queries marked as rare; and *normal*, covering the remaining cases. As shown in Tab. A11, LAM is relatively competitive with SAM3 (Carion et al., 2026) on normal and long-tail cases but exhibits much larger gaps on dense scenes and small targets.

From the silver-label pool constructed in Sec. 3.1, we build two controlled 450K training sets: **Random Sample Silver Labels (RSL)**, sampled uniformly from the pool, and **Selected Silver Labels (SSL)**, biased toward dense and small-target examples. Both substantially improve LAM, while SSL produces larger gains on the targeted dense and small subsets. However, because normal cases dominate the LVIS validation set, this specialization slightly reduces the overall metric relative to

Table A10: **Detailed results on ReasonSeg.** We report positive-query F1 and recall (%) at representative IoU thresholds. mF1 and mR are averaged over all IoU thresholds from 0.50 to 0.95.

Model / Decode	IoU 0.5		IoU 0.6		IoU 0.7		IoU 0.8		IoU 0.9		Overall	
	F1	R	F1	R	F1	R	F1	R	F1	R	mF1	mR
Closed-source VLMs												
Gemini-3.7-Flash	67.55	70.01	63.89	65.64	61.64	63.06	56.59	57.53	48.66	49.29	**57.61**	**58.92**
GPT-5.6-Luna	58.39	59.59	52.28	53.02	42.66	43.11	33.06	33.33	22.10	22.27	39.22	39.70
GPT-5.6-Sol	63.18	64.48	58.40	59.07	51.73	51.99	42.41	42.60	30.82	30.89	46.54	46.96
Open-source and Dedicated Models												
Qwen-3.8-27B	68.48	69.50	65.34	66.02	61.93	62.42	55.62	55.98	43.81	44.02	56.73	57.22
SAM3	7.74	8.62	6.80	7.59	5.87	6.56	5.87	6.56	5.21	5.79	6.06	6.76
RexOmni	46.91	48.52	42.76	44.02	39.86	40.93	35.46	36.42	28.55	29.21	36.99	38.04
LAM Family												
LAM AR	46.28	48.01	43.48	44.66	40.89	41.83	36.83	37.58	31.56	32.18	38.43	39.42
LAM Hybrid	46.10	48.13	42.84	44.40	40.23	41.44	35.67	36.68	28.16	28.96	37.33	38.60
LAM++ Hybrid	62.43	65.38	59.29	61.78	56.19	58.30	51.07	52.90	45.21	46.72	53.12	55.19
LAM++ GeoSpec	63.28	65.51	59.49	61.26	56.74	58.30	53.02	54.31	46.82	47.88	**54.38**	**55.91**

Table A11: **Ablation of the LAM++ post-training recipe on LVIS.** We report mF1 and mRecall for different LVIS subsets under both autoregressive (AR) and multi-token prediction (MTP) decoding. Best results under AR and MTP decoding are highlighted in yellow and blue, respectively.

Training / CKPT	Decode	Normal		Dense		Small		Long-Tail		Overall	
		mF1	mR	mF1	mR	mF1	mR	mF1	mR	mF1	mR
SAM3 reference	–	69.35	69.89	41.92	36.35	41.48	39.23	36.65	36.34	59.61	58.98
LAM	AR	61.48	58.74	18.41	14.13	24.71	22.26	36.06	34.44	48.21	45.38
	MTP	57.29	55.20	12.61	9.83	21.72	20.02	34.31	33.37	44.23	42.16
SFT on RSL	AR	67.35	66.47	27.14	22.62	29.97	27.78	47.42	46.77	54.13	52.60
	MTP	63.12	62.89	20.86	17.99	27.51	26.03	39.62	39.64	50.25	49.47
SFT on SSL	AR	65.48	65.27	28.82	25.30	30.57	28.59	43.72	44.44	53.17	52.23
	MTP	60.36	60.90	21.87	19.81	27.78	26.61	37.62	37.87	48.61	48.49
Single Query	AR	66.62	66.25	29.80	25.96	31.19	29.06	45.28	45.10	54.14	53.03
	MTP	64.75	65.05	24.56	22.12	30.54	29.07	43.06	43.99	52.43	52.05
High Resolution	AR	73.53	73.18	43.16	38.24	47.13	44.31	58.57	58.23	64.16	62.84
	MTP	71.91	72.35	38.36	35.27	45.89	44.17	55.94	56.17	62.48	62.11
AR-GRPO on LVIS	AR	**75.55**	**75.01**	45.44	41.15	49.29	47.04	60.17	59.83	**66.21**	64.98
	MTP	**73.56**	**74.60**	**40.48**	37.82	46.56	45.96	57.40	**58.86**	**63.85**	64.24
AR-GRPO on LVIS-p	AR	74.88	74.91	**45.59**	**42.01**	**49.57**	**47.72**	60.07	**60.25**	65.87	**65.18**
	MTP	72.57	74.56	40.33	**38.39**	**46.67**	**46.59**	57.03	58.59	63.23	**64.43**
MTP-GRPO on LVIS-p	AR	74.95	72.38	43.76	36.09	48.91	44.02	**60.19**	57.70	65.71	62.15
	MTP	73.26	72.84	38.56	33.52	**46.67**	44.11	**57.73**	57.38	63.64	62.33

RSL. This suggests a coverage–precision trade-off: stronger dense supervision encourages more exhaustive predictions but can also increase over-generation on ordinary scenes.

Single-query SFT. Using the same SSL data, we replace LAM's multi-query training format with the single-query formulation introduced in Sec. 3.2. This consistently improves both AR and MTP decoding across LVIS subsets and also narrows the AR–MTP performance gap. The result supports our hypothesis that separating localization concepts reduces cross-concept interference and encourages more complete localization of each query.

Table A12: **Distribution of diagnostic query subsets in LVIS.** Dense, small, and long-tail subsets may overlap, while normal contains queries belonging to none of these failure subsets.

Subset	Definition	Train		Validation	
		Cases	Ratio	Cases	Ratio
Dense	GT boxes/query ≥ 8	31,619	8.79%	6,037	8.6%
Small	Mean box area / image area < 0.005	46,912	13.04%	22,831	32.6%
Long-tail	LVIS rare category (`frequency = r`)	1,489	0.41%	329	0.5%
Normal	None of the above	280,342	77.91%	44,291	63.3%

High-resolution SFT. We next apply the high-resolution visual processing described in Sec. 3.2, resizing the shorter image side to 1024 pixels under MoonViT's (Kimi, 2025) native-resolution scheme. This produces the largest gain among the SFT ablations. Improvements are especially pronounced on dense scenes and small targets, indicating that insufficient visual granularity is a major bottleneck for localization. The resulting SFT model also surpasses SAM3 (Carion et al., 2026) on the overall LVIS metric.

Failure-focused RL. Starting from the high-resolution single-query SFT checkpoint, we apply the localization reward and AR-GRPO formulation from Sec. 3.3. To test whether RL benefits more from difficult examples than from additional data volume, we construct **LVIS-p**, a failure-focused subset containing dense, small-target, and long-tail cases together with a small fraction of normal examples. Although LVIS-p contains only about 40% of the full LVIS training data, AR-GRPO on LVIS-p achieves gains comparable to training on the full set. This suggests that failure-focused sampling provides a more efficient source of RL supervision than uniform scaling.

Cross-policy transfer. AR-GRPO improves not only AR decoding but also MTP decoding, even though the MTP policy receives no direct RL loss. We attribute this behavior to the shared model parameters underlying the two decoding policies. Conversely, direct MTP-GRPO also changes AR performance, demonstrating bidirectional cross-policy transfer. However, MTP-GRPO shifts both policies toward a more conservative operating point: precision increases while recall decreases, particularly on dense scenes. Since missed instances are the dominant failure mode in dense localization, we adopt **AR-GRPO** as the final RL post-training strategy. Additional details of the MTP rollout formulation and optimization behavior are provided in Appendix Sec. C.

G Additional GeoSpec Analysis

We provide additional analysis of GeoSpec, including a detailed coordinate-tolerance sweep, its dependence on target size, and several decoding details concerning box ordering, termination, and draft rejection.

Detailed coordinate-tolerance ablation. To isolate the effect of coordinate tolerance from the verification interval, we fix $K = 1$ and evaluate different values of ϵ on the dense LVIS subset, containing 6,037 image–query cases and 118,355 ground-truth boxes. The AR verifier uses greedy decoding, and coordinate acceptance follows exactly the criterion defined in Sec. 3.4; structural tokens require exact agreement.

Tab. A13 shows a smooth trade-off between verification tolerance and localization quality. When $\epsilon = 0$, coordinate tokens must exactly agree with the AR verifier, corresponding to the strict speculative-verification baseline. This setting nearly preserves AR accuracy, achieving 45.46 mF1 compared with 45.51 for AR, but only 37.94% of drafted boxes are fully accepted. Relaxing the tolerance substantially improves draft acceptance: the full-box acceptance rate increases to 74.06% at $\epsilon = 3$ and 76.58% at $\epsilon = 7$, while mF1 decreases by only 0.11 and 0.20 points, respectively. Beyond this range, acceptance improves only gradually, whereas localization quality continues to decrease. We therefore use $\epsilon = 7$ as a moderate operating point that captures most of the available acceptance gain without overly relaxing geometric verification.

Sensitivity to target size. A fixed coordinate tolerance has different geometric effects for targets of different sizes. For a small box, the same absolute coordinate displacement represents a larger fraction of its spatial extent than for a large box. We therefore partition ground-truth boxes according to their area relative to the full image and report the resulting localization quality in Tab. A14.

Table A13: **Detailed coordinate-tolerance ablation on dense LVIS.** We fix $K = 1$ to isolate the effect of ϵ. Recall and F1 are reported at representative IoU thresholds, while mR and mF1 average over IoU thresholds from 0.50 to 0.95 with a step of 0.05. $\epsilon = 0$ is the strict exact-coordinate verification baseline. Full Accept denotes the fraction of drafted boxes whose complete token sequence is accepted by the AR verifier.

Method	Overall		IoU = 0.5		IoU = 0.6		IoU = 0.7		IoU = 0.8		IoU = 0.9		Full Accept
	mR	mF1	R	F1	R	F1	R	F1	R	F1	R	F1	
AR	41.39	45.51	57.55	63.76	53.25	58.79	47.48	52.18	38.78	42.41	23.60	25.58	–
GeoSpec ($\epsilon = 0$)	41.36	45.46	57.53	63.74	53.21	58.75	47.39	52.08	38.78	42.40	23.58	25.56	37.94%
GeoSpec ($\epsilon = 1$)	41.32	45.46	57.56	63.82	53.24	58.83	47.37	52.10	38.70	42.35	23.47	25.46	62.82%
GeoSpec ($\epsilon = 3$)	41.26	45.40	57.53	63.80	53.19	58.77	47.29	52.03	38.62	42.26	23.44	25.44	74.06%
GeoSpec ($\epsilon = 5$)	41.23	45.35	57.51	63.77	53.13	58.69	47.24	51.96	38.58	42.20	23.42	25.41	75.68%
GeoSpec ($\epsilon = 7$)	41.20	45.31	57.47	63.73	53.08	58.63	47.20	51.92	38.55	42.17	23.40	25.38	**76.58%**
GeoSpec ($\epsilon = 10$)	41.16	45.26	57.40	63.64	53.04	58.57	47.15	51.86	38.52	42.13	23.37	25.35	77.76%
GeoSpec ($\epsilon = 15$)	41.06	45.14	57.27	63.48	52.95	58.45	47.03	51.70	38.41	42.00	23.31	25.27	78.91%
GeoSpec ($\epsilon = 20$)	40.93	44.97	57.08	63.24	52.73	58.18	46.90	51.51	38.28	41.83	23.25	25.20	79.95%
GeoSpec ($\epsilon = 30$)	40.70	44.72	56.83	62.96	52.44	57.85	46.59	51.17	38.03	41.56	23.12	25.07	81.58%
GeoSpec ($\epsilon = 50$)	40.33	44.25	56.42	62.39	52.00	57.28	46.13	50.60	37.64	41.09	22.88	24.78	84.84%

Table A14: **Effect of coordinate tolerance across ground-truth box sizes.** Boxes are grouped by normalized area, defined as ground-truth box area divided by image area. Entries report GT-box micro mIoU (%). Matching is one-to-one within each image–query case by maximum total IoU, with unmatched ground-truth boxes assigned IoU zero. The ϵ sweep uses $K = 1$.

Method	$< 0.01\%$ ($N = 7{,}538$)	0.01–0.05% ($N = 22{,}331$)	0.05–0.2% ($N = 32{,}602$)	0.2–1% ($N = 36{,}633$)	1–5% ($N = 16{,}083$)	$\geq 5\%$ ($N = 3{,}168$)	All
AR	10.87	23.26	40.24	59.42	69.70	75.51	46.05
GeoSpec ($\epsilon = 0$)	10.91	23.16	40.16	59.43	69.66	75.51	46.01
GeoSpec ($\epsilon = 1$)	10.79	23.21	40.35	59.45	69.62	75.42	46.06
GeoSpec ($\epsilon = 3$)	10.66	23.04	40.27	59.44	69.68	75.42	46.00
GeoSpec ($\epsilon = 5$)	10.61	22.98	40.16	59.51	69.73	75.31	45.98
GeoSpec ($\epsilon = 7$)	10.61	22.88	40.12	59.47	69.74	75.37	45.94
GeoSpec ($\epsilon = 10$)	10.53	22.78	39.87	59.48	69.79	75.49	45.87
GeoSpec ($\epsilon = 15$)	10.41	22.66	39.74	59.33	69.74	75.43	45.75
GeoSpec ($\epsilon = 20$)	10.41	22.52	39.52	59.10	69.76	75.43	45.59
GeoSpec ($\epsilon = 30$)	10.34	22.19	39.08	58.74	69.55	75.25	45.26
GeoSpec ($\epsilon = 50$)	10.08	21.59	38.61	58.39	69.53	75.03	44.88

For this diagnostic, predicted and ground-truth boxes are matched one-to-one within each image–query case by maximizing total IoU, and unmatched ground-truth boxes receive IoU zero. The reported GT-box micro mIoU therefore reflects both localization quality and missed instances. The effect of increasing ϵ is concentrated on smaller targets: boxes occupying 0.01%–0.2% of the image degrade progressively as the tolerance becomes large, whereas boxes occupying at least 1% remain comparatively stable. This behavior is consistent with a fixed coordinate deviation constituting a larger relative perturbation for small boxes, and provides another reason to avoid overly relaxed verification.

Sequential verification protocol. The final GeoSpec configuration uses $K = 2$ and $\epsilon = 7$. In each decoding cycle, MTP drafts up to K consecutive boxes, stopping draft construction early if it proposes EOS. The resulting draft sequence is passed through the AR verifier in a single causal teacher-forcing forward pass. Thus, target verification is parallel in computation, but draft acceptance remains sequential in logic. Tokens are checked from left to right, and GeoSpec commits only the longest prefix that satisfies the verification criterion. At the first mismatch, AR completes the current box and the next MTP drafting cycle starts from the corrected context.

Different MTP and AR box orderings. GeoSpec verifies the serialized box sequence directly and does not perform set-level matching or reordering between the MTP and AR predictions. For example, suppose the AR policy prefers the order (A, B, C) while MTP drafts (B, A, C). The initial structural token may agree, but the coordinates of the first box will typically fail verification. GeoSpec then retains only the accepted prefix, lets AR complete its preferred box A, and restarts MTP drafting from the resulting context.

We deliberately avoid post-hoc Hungarian matching between the K drafted boxes and hypothetical AR boxes. AR predictions for later draft positions are conditioned on the preceding draft tokens; once an earlier token is rejected and replaced by AR, this causal context changes. Consequently, later verification results computed under the rejected prefix cannot be safely reused. Ordering disagreement therefore mainly reduces the accepted draft length and decoding efficiency. For $\epsilon > 0$, two sufficiently close coordinate sequences may still satisfy the geometric acceptance criterion; this is the intended approximation introduced by GeoSpec.

Early MTP termination. An EOS proposed by MTP is treated as a draft token rather than an unconditional termination signal. Unlike coordinate tokens, EOS requires exact agreement with the AR verifier. If MTP proposes EOS while AR predicts another $\langle \mathrm{box} \rangle$, the draft EOS is rejected and AR generates the next box. If both policies predict EOS, it is accepted and generation terminates. Conversely, if MTP proposes another box while AR predicts EOS, the box-start token is rejected and AR terminates. Thus, an early MTP EOS cannot by itself truncate a GeoSpec output; disagreement only incurs additional fallback computation.

Draft rejection and suffix invalidation. GeoSpec does not treat a drafted box as an atomic accept-or-reject unit. If the first mismatch occurs inside a box, the matching token prefix of that box is retained and AR generates only its remaining tokens. However, all subsequent draft tokens are discarded.

More formally, let L denote the length of the consecutively accepted prefix across a draft of K boxes. Since each non-terminal box contains $B = 6$ tokens,

$$q = \left\lfloor \frac{L}{B} \right\rfloor, \qquad r = L \bmod B, \qquad B = 6, \tag{17}$$

where q is the number of completely accepted boxes and r is the number of accepted tokens in the next box. GeoSpec commits the q complete boxes and the r-token prefix, after which AR completes the current box. All remaining drafted boxes are discarded. In particular, if rejection occurs inside the first drafted box, all later $K - 1$ boxes are discarded.

This suffix invalidation follows directly from causal conditioning. After a rejected draft token is replaced by an AR token, every subsequent AR distribution must be conditioned on the corrected prefix rather than the rejected draft prefix. Previously computed verification results for later tokens are therefore no longer valid. The same effect explains the verification-interval trade-off observed in Fig. 7(d): larger K amortizes AR verification when early drafts are accepted, but wastes more speculative computation when an early mismatch invalidates a longer suffix.

H Detailed Accuracy–Efficiency Analysis

All experiments in Fig. 7 are conducted on the LVIS validation set or its subsets using the final AR-GRPO checkpoint selected in Sec. 4.3. GeoSpec requires no additional training: it directly uses the checkpoint's MTP policy for drafting and AR policy for verification. Thus, these experiments isolate decoding-time effects under fixed model weights.

Overall accuracy–efficiency trade-off. As shown in Fig. 7(a), GeoSpec preserves essentially the same localization accuracy as pure AR decoding while increasing decoding throughput by approximately 68%. It also substantially improves over the original LAM (Wang et al., 2026) while recovering much of the parallel-decoding efficiency of MTP. This shows that geometry-aware speculative verification can retain AR-level localization quality without paying the full cost of sequential decoding.

Verify interval and coordinate tolerance. We evaluate the two GeoSpec hyperparameters under settings chosen to expose their respective effects. The verify interval K controls the global verification schedule, so we evaluate it on the full LVIS validation set. As shown in Fig. 7(d), increasing K reduces verification frequency but increases wasted computation when inaccurate drafts require correction, with $K = 2$ providing the best overall efficiency.

The coordinate tolerance ϵ determines whether MTP-drafted coordinates are accepted by the AR verifier. We evaluate it on the dense LVIS subset, where longer multi-box sequences make changes in draft acceptance and correction behavior more pronounced. As shown in Fig. 7(e), increasing ϵ raises draft acceptance, but the efficiency gain quickly saturates while localization accuracy gradually decreases. We therefore use $K = 2$ and $\epsilon = 7$.

Scaling with scene density and query count. For both scaling analyses, we select the 5,000 LVIS validation images with the most queries and separately group the same images by target-box count and query count. As shown in Fig. 7(b,c), the original LAM degrades substantially as either factor increases, whereas LAM++ GeoSpec remains more robust.

As scene density increases, more decoded boxes amortize the fixed cost of high-resolution image encoding and prefix prefill, allowing GeoSpec to approach or exceed the end-to-end box throughput of LAM AR. As query count increases, the image-prefix KV states are computed once and reused across independent single-query decoding runs. This amortizes the additional cost of single-query inference and keeps LAM++ throughput close to LAM AR even in high-query regimes.