Self-Improving Small Object Grounding in LVLMs

Tianze Yang Yucheng Shi Ruitong Sun Ninghao Liu Jin Sun
University of Georgia

Project page: https://groundvlm.github.io/

Can internal attention patterns in Large Vision Language Models (LVLMs) identify reliable small-object boxes without fine-tuning? In this work, we provide an affirmative answer. Attention structure in LVLMs encodes grounding quality—a lightweight IoU regressor trained solely on attention maps achieves strong IoU prediction (Pearson $r > 0.67$). This regressor powers the regressor-based variant of our **Attention-based Candidate Selection (ACS)** framework, called **ACS-Learned**, which selects the best box from multiple sampled candidates to improve object grounding. By analyzing what the regressor learns, we reveal which transformer layers and heads are most critical and derive **ACS-Free**: a training-free selector that ranks candidates by attention entropy on these discriminative heads, with no learned component at inference. Experiments on COCO and Objects365 demonstrate up to 19% self-improvement on small object localization, with ACS-Free ranking best among all training-free methods, demonstrating that useful attention structure improves both localization reliability and interpretability in LVLMs.

1. Introduction

Large Vision Language Models (LVLMs) [2, 4, 10, 21, 34] have shown remarkable capabilities in vision-language joint reasoning. Recently, researchers explored using LVLMs for direct object localization of bounding box coordinates [11, 41] without specialized detection heads (e.g., Qwen2.5-VL [2] and InternVL-3.5 [34]). This capability is crucial for autonomous navigation, robotics, AR, and visual assistance.

Despite the notable progress, we found that SOTA LVLMs struggle with grounding *small* objects: as illustrated in Figure 1, they achieve **strong performance on large objects but not on small objects**. This is particularly concerning for safety-critical deployment, where small objects such as distant pedestrians are common.

To improve general localization performance, prior works [13, 28, 41, 42] use additional fine-tuning, external localization modules, or heuristic decoding strategies, yet they require substantial computational resources or architectural modifications. Instead, we ask the following fundamental and intriguing question:

Can internal representations identify reliable small-object boxes without fine-tuning LVLMs?

In this work, we provide an affirmative answer. We propose a novel framework that leverages the intrinsic attention patterns of LVLMs to estimate bounding box quality and select the best localization output from multiple sampled responses, with a focus on small objects. Our key insight is that the spatial attention structures across layers and heads correlate strongly with grounding quality.

To validate this idea, we introduce our **Attention-based Candidate Selection (ACS)** framework. First, we perform multi-response sampling to collect a set of candidate bounding boxes and their associated attention maps. Then, we train a lightweight IoU regressor to predict bounding box quality directly from attention patterns, demonstrating that attention alone carries sufficient signal for quality estimation. The regressor serves dual purposes: (1) as the regressor-based selector **ACS-Learned** that directly scores each candidate; and (2) as an analytical tool—through gradient attribution and entropy analysis of the trained regressor, we identify which layers and heads are most critical for localization and discover that specific attention heads exhibit strong entropy-IoU correlations. This finding enables **ACS-Free**, a training-free variant that selects boxes purely from entropy patterns in those discriminative heads, eliminating the need for any learned component at inference.

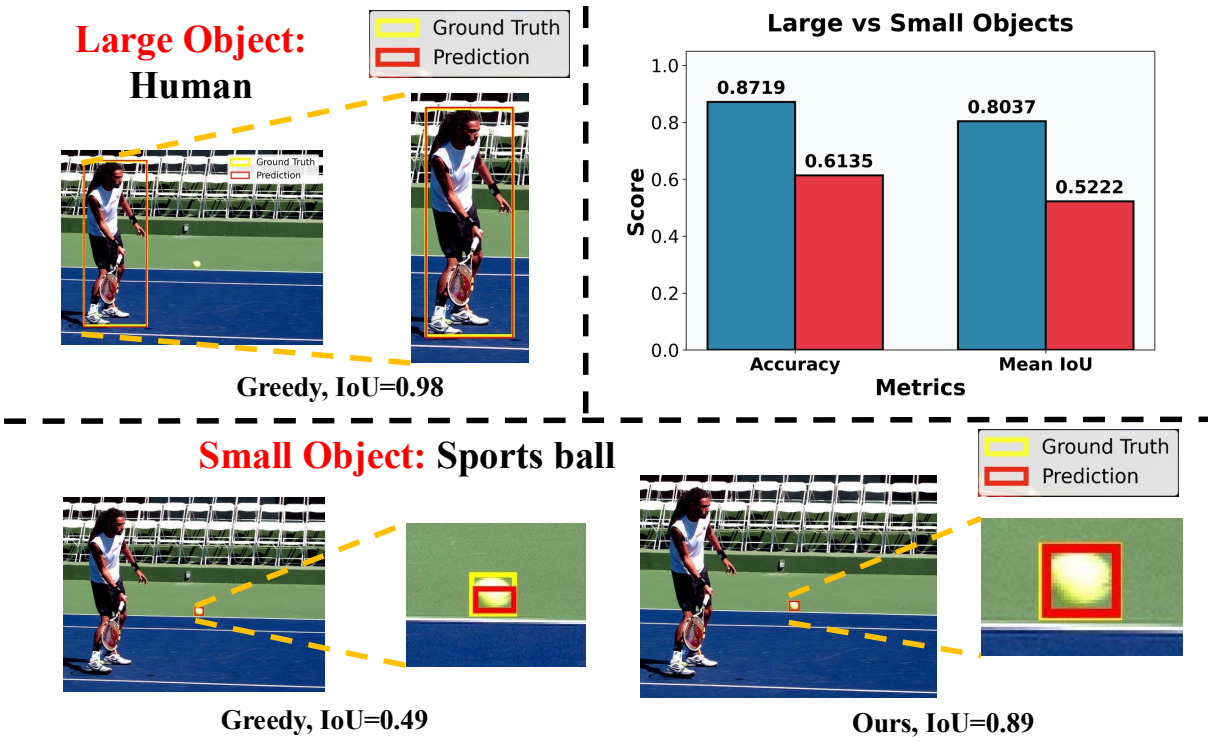


Figure 1: Performance comparison between large and small object grounding in Qwen2.5-VL. The significant performance gap highlights the challenge of small object detection. Our self-improvement framework helps LVLMs to find small targets.

ACS requires no modification, fine-tuning, or auxiliary supervision to LVLMs, apart from a small regression module used for analyzing attention patterns and selecting high-quality candidates. This design achieves *self-improvement* and allows our method to be applied to different LVLMs, serving as a plug-and-play enhancement to those models.

Our main contributions are:

1) Discovery. We identify a strong connection between internal attention patterns of LVLMs and grounding quality: attention structure encodes whether a predicted small-object box is reliable.

2) Validation and exploitation. A lightweight IoU regressor trained on attention maps confirms this with high correlation, and ACS-Learned uses it directly for candidate selection, achieving consistent gains across different LVLMs and datasets.

3) Interpretability and distillation. Gradient and entropy analysis of the regressor reveals which transformer heads matter for localization. ACS-Free operationalizes this as a parameter-free entropy rule, achieving the best performance among all training-free baselines.

4) Results and insights. Our framework achieves up to 19% self-improvement on small object localization without LVLM fine-tuning or external detectors. The localization-critical layers we identify provide interpretable insight into how LVLMs process spatial information in grounding objects.

2. Related Works

Object localization/grounding with VLMs. Vision Language Models have demonstrated strong ability in general visual understanding tasks. In localization-related tasks, VLMs are commonly used for reference grounding [4, 17, 41, 42]. Segmentation is also related to localization [13, 14]. Reasoning-based approach has been proposed to further enhance the visual understanding performance of VLMs [24, 28]. To improve localization performance of VLMs, prior work [11, 14, 20, 22, 27] mainly uses external models such as SAM [12] or object detectors [45]. In this work, we study LVLMs such as Qwen2.5-VL [2] and

InternVL-3.5 [34] that can directly output bounding box coordinates, and improve this ability without external models or finetuning.

Attention in LLMs. Recent work [3, 44] shows that attention mechanisms are key to understanding how LLMs process and ground information from image tokens [5, 36]. Building on this insight, many studies [11, 20, 37] further exploit the structure of attention patterns, using attention maps as effective localization priors and feeding them into lightweight localizers [12] for downstream prediction. Zhang et al. [43] use cross-modal attention to crop question-relevant regions, notably improving zero-shot VQA on small details. In this work, we reveal new insights into attention patterns and their connection to localization quality in LVLMs.

Inference-time sampling. Test-time scaling is a powerful training-free strategy for improving model performance. Early work shows that sampling multiple responses and aggregating them improves accuracy [35], with self-consistency [35] using majority voting over diverse reasoning paths. Subsequent extensions explore weighted voting [16], confidence-based selection [38], and minimum Bayes risk decoding [33]. Recent advances in compute allocation [1, 32] demonstrate that strategic sampling can rival larger models, with Best-of-N sampling [8, 18] and tree-search methods [15, 23, 40] achieving strong gains in math reasoning. In vision-language settings, diverse sampling benefits visual chain-of-thought prompting [6] and fine-grained reasoning [29]. However, its use in object localization remains underexplored. Token-level vocabulary entropy has been explored as a proxy for model confidence [38], but it captures a fundamentally different signal from spatial attention patterns used in our work.

3. Preliminary

Object grounding with LVLMs. Given an image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$ and a text query q for object localization, LVLMs predict a bounding box of the queried object, represented as $b = [x_1, y_1, x_2, y_2]$ when the object is present in the image. For challenging small objects, the boxes are inaccurate.

Inference-time sampling for localization. One way to improve localization is to generate multiple responses [7, 23, 40]. As the number of responses increases, the chance that at least one box is good also increases. We can generate diverse boxes through temperature-controlled sampling: $\mathcal{R} = \{r_1, r_2, \ldots, r_N\} \sim p_{\text{LVLMs}}(\cdot|\mathbf{I}, q, \tau)$, where there are N responses and τ is the temperature. Each response r_i generates M_i bounding boxes, forming a *candidate set* $\mathcal{B} = \{b_1, b_2, \ldots, b_T\}$ where $T = \sum_{i=1}^{N} M_i$. The key challenge in inference-time sampling is to *select the best bounding box from* $\mathcal{B}$. The bottleneck is not generating candidates but knowing which one is reliable—and we show that internal attention patterns are the key.

4. Methodology

4.1 Overview

To select high-quality bounding boxes from the candidate set $\mathcal{B}$, we exploit the fact that coordinates are generated through autoregressive decoding where the LVLM attends to visual regions. These attention patterns of visual regions should reflect the model's spatial understanding, distinguishing good predictions from bad ones. However, which specific attention characteristics are most informative for grounding quality, and whether they can be used without a learned model, remains unclear.

We address this through a three-stage investigation, formalized as the **Attention-based Candidate Selection (ACS)** framework (Figure 2). Stage 1: we train a lightweight IoU regressor on attention maps to test the hypothesis that attention encodes localization quality (§4.2); we deploy this regressor directly as **ACS-Learned**. Stage 2: we analyze the trained regressor via gradient attribution and entropy analysis to discover which layers and heads drive localization quality (§4.3). Stage 3: we distill this understanding into **ACS-Free**, a training-free variant that ranks candidates by entropy on the identified discriminative heads, requiring no learned component at inference (§4.4).

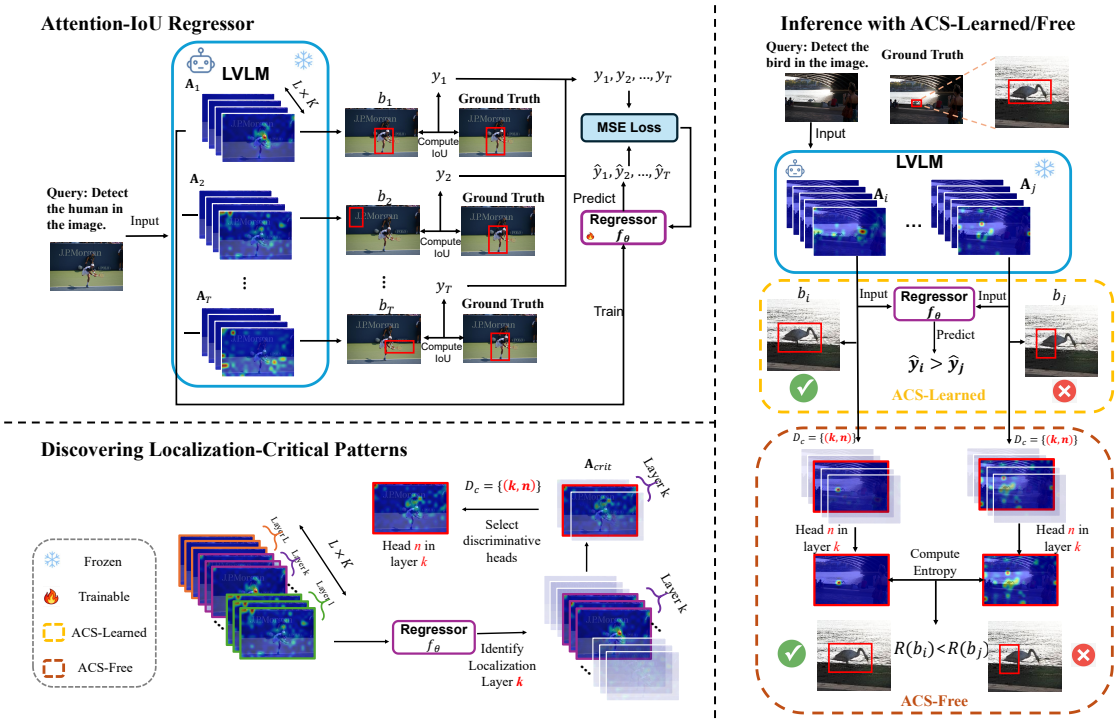


Figure 2: Our pipeline follows a discovery arc in three stages: (1) training a lightweight **IoU regressor** on attention maps to test the hypothesis that attention encodes localization quality and deploying as **ACS-Learned**; (2) analyzing the trained regressor via gradient attribution and entropy analysis to identify localization-critical layers and discriminative heads; and (3) distilling this understanding into **ACS-Free**, a training-free selector that ranks candidates by attention entropy on the identified heads, requiring no learned component at inference.

4.2 Attention-IoU Regressor & ACS-Learned

To identify which attention patterns correlate with localization quality, we directly train a lightweight *Attention-IoU Regressor* $f_\theta : \mathbf{A} \to \hat{Y}$ that predicts IoU scores from attention patterns $\mathbf{A}$, where $Y \in [0, 1]$ denotes the Intersection over Union (IoU) between prediction and ground truth.

Attention extraction. For each candidate bounding box $b_j \in \mathcal{B}$, we extract their attention maps from all layers and heads. Specifically, we consider an LVLM with L layers and K attention heads per layer. When generating each coordinate token $c \in \{x_1, y_1, x_2, y_2\}$, we extract the attention weights over vision tokens: $\mathcal{A}_{c,j}^{(l,h)} \in \mathbb{R}^{H_v \times W_v}$, where $l \in [1, L]$ is the layer index, $h \in [1, K]$ is the head index, and (H_v, W_v) are the spatial dimensions of the vision token grid. We aggregate attention across all layers and heads for each coordinate: $\mathbf{A}_{c,j} = [\mathcal{A}_{c,j}^{(1,1)}, \mathcal{A}_{c,j}^{(1,2)}, \ldots, \mathcal{A}_{c,j}^{(L,K)}] \in \mathbb{R}^{L \times K \times H_v \times W_v}$. The representation for b_j is: $\mathbf{A}_j = \{\mathbf{A}_{x_1,j}, \mathbf{A}_{y_1,j}, \mathbf{A}_{x_2,j}, \mathbf{A}_{y_2,j}\}$.

Regressor training. The IoU regressor f_θ predicts IoU scores through a neural network parameterized by θ: $\hat{y}_j = f_\theta(\mathbf{A}_j) \in [0, 1]$. Given a training dataset $\mathcal{D}$ of (bounding box, IoU) pairs collected from LVLM inference, we train f_θ with the mean squared error:

$$\theta^* = \arg\min_\theta \frac{1}{|\mathcal{D}|} \sum_{(b_j, y_j) \in \mathcal{D}} (y_j - \hat{y}_j)^2. \tag{1}$$

Under the MSE loss, the optimal predictor learns the conditional expectation $f_\theta(\mathbf{A}) \approx \mathbb{E}[Y|\mathbf{A}]$.

MSE training also has a principled information-theoretic justification [9]. The mutual information $I(f_\theta(\mathbf{A}); Y) = H(Y) - H(Y|f_\theta(\mathbf{A}))$ quantifies how much uncertainty about IoU is reduced by observing the regressor's predictions. When we model the regression error as Gaussian noise, i.e., $Y = f_\theta(\mathbf{A}) + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, the conditional entropy becomes $H(Y|f_\theta(\mathbf{A})) = \frac{1}{2}\log(2\pi e\sigma^2)$, where $\sigma^2 =$

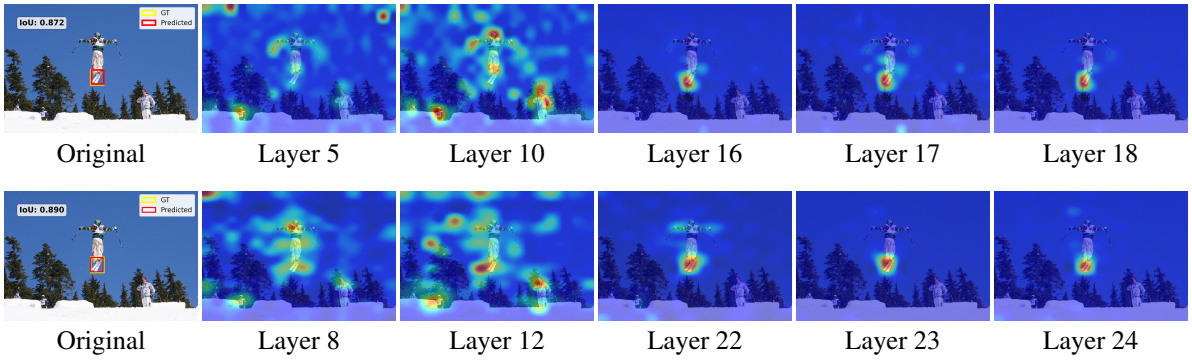


Figure 3: Attention maps across important layers for Qwen2.5-VL (first row) and InternVL-3.5 (second row). The first column shows the original images with bounding boxes, followed by attention visualizations from five different layers.

$\mathbb{E}[(Y - \hat{Y})^2]$ is precisely the MSE (detailed derivation in Appendix B). Since $H(Y)$ is constant for a given dataset, minimizing MSE is equivalent to minimizing $H(Y|f_\theta(\mathbf{A}))$ as well as maximizing $I(f_\theta(\mathbf{A}); Y)$. This result shows that the IoU regressor not only predicts IoU accurately, but also *preserves localization-relevant information* contained in attention patterns.

Inference with ACS-Learned. At inference time, the trained regressor is a natural solution for bounding box selection. Given the candidate set $\mathcal{B}$, we apply the IoU regressor to predict the IoU score for each candidate based on its attention patterns. We refer to this regressor-based inference strategy as **ACS-Learned**: $b^* = \arg\max_{b_j \in \mathcal{B}} f_\theta(\mathbf{A}_j)$. It ranks all candidates by their predicted IoUs and selects the one with the highest estimated quality.

4.3 Discovering Localization-Critical Patterns

While ACS-Learned is effective for object grounding, it requires the learned model during inference. To remove this dependency, we analyze what the trained IoU regressor has learned and translate those findings into training-free selection rules. Our analysis proceeds in two steps. First, we identify the transformer layers that contribute most to localization prediction through gradient attribution. Second, we examine the attention characteristics within these layers that correlate with localization to obtain usable quantities for ACS-Free (§4.4).

Table 1: Top-10 localization-critical layers.

Model	Layer Index
Qwen2.5-VL-7B	14, 15, 16, 17, 18, 19, 20, 21, 24, 25
InternVL-3.5-8B	17, 18, 19, 20, 21, 22, 23, 24, 26, 35

Localization-critical layers. To assess the contribution of different layers to localization-related signals, we analyze gradients from the trained IoU regressor [25, 31, 39]. Specifically, for each layer l and head h in an LVLM, we compute the importance score for coordinate $c \in \{x_1, y_1, x_2, y_2\}$:

$$I_c^{(l,h)} = \max_{i,j} \left| \frac{\partial \mathcal{L}_{\text{MSE}}(f_\theta(\mathbf{A}), y)}{\partial \mathcal{A}_{c,i,j}^{(l,h)}} \right|. \tag{2}$$

The per-coordinate layer importance $I_c^{(l)} = \frac{1}{K} \sum_{h=1}^{K} I_c^{(l,h)}$ captures which layers most affect IoU predictions.

The top-10 important layers for both Qwen2.5-VL-7B and InternVL-3.5-8B are reported in Table 1. For Qwen2.5-VL, layers 14–21 contribute the most, which we identify as *localization-critical layers*

$\mathbf{A}_{\text{crit}}$. This finding is confirmed by attention visualizations in Figure 3: layers 16-18 focus precisely on target objects, while earlier layers (≤ 12) show dispersed patterns, suggesting a global-to-object transition (more in Appendix C).

The concentration of gradient magnitudes in layers 14–21 indicates that mutual information $I(\mathbf{A}; Y)$ is predominantly captured by these layers. In other words, if $\mathbf{A}_{\text{crit}}$ denotes attention from these critical layers, then $I(\mathbf{A}_{\text{crit}}; Y) \approx I(\mathbf{A}; Y)$. This naturally leads to a hypothesis: within localization-critical layers, accurate bounding boxes correspond to concentrated (low-entropy) attention, while poor predictions yield diffuse (high-entropy) attention.

Discovering the entropy-IoU correlation. To test the hypothesis that attention concentration reflects localization quality, we examine how attention spatial entropy within the localization-critical layers correlates with IoU scores. Specifically, we compute the entropy of attention maps for bounding boxes of varying quality and compare their distributions. The entropy of an attention map measures its spatial concentration:

$$H(\mathcal{A}) = -\sum_{u=1}^{H_v}\sum_{v=1}^{W_v} p_{u,v} \log p_{u,v}, \tag{3}$$

where $p_{u,v} = \mathcal{A}_{u,v} / \sum_{k,l} \mathcal{A}_{k,l}$ normalizes the attention map into a probability distribution. We partition training samples into three groups based on IoU: $\mathcal{G}_{\text{high}} = \{b_j : \text{IoU}(b_j) > \tau_{\text{high}}\}, \mathcal{G}_{\text{low}} = \{b_j : 0 < \text{IoU}(b_j) \leq \tau_{\text{low}}\}, \mathcal{G}_{\text{zero}} = \{b_j : \text{IoU}(b_j) = 0\}$, where $\tau_{\text{low}} < \tau_{\text{high}}$ are predefined thresholds. For each (layer l, head h) and each coordinate c, we compute the per-head mean entropy across samples in IoU group g: $\bar{H}_{c,g}^{(l,h)} = \frac{1}{|\mathcal{G}_g|} \sum_{b_j \in \mathcal{G}_g} H\left(\mathcal{A}_{c,j}^{(l,h)}\right)$, and aggregate to the layer level by averaging over the K heads of layer l: $\bar{H}_{c,g}^{(l)} = \frac{1}{K} \sum_{h=1}^{K} \bar{H}_{c,g}^{(l,h)}$, which is what Figure 4 visualizes across the three IoU groups: In localization-critical layers, *high-IoU samples consistently exhibit lower entropy than low/zero-IoU samples*, making entropy a strong discriminator of localization accuracy in these layers. Not all heads within these layers contribute equally to this separation. To identify which heads are most discriminative, we measure the *entropy difference* between high- and low-IoU groups: $\Delta H_c^{(l,h)} = \bar{H}_{c,\text{low}}^{(l,h)} - \bar{H}_{c,\text{high}}^{(l,h)}$. Heads with large $\Delta H_c^{(l,h)}$ are selected as the key indicators.

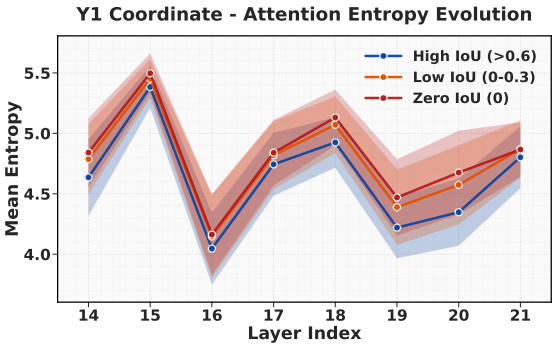


Figure 4: Attention entropy in localization-critical layers. High-IoU predictions have lower entropy than low/zero-IoU predictions.

4.4 ACS-Free: Training-Free Distillation

Given the findings in §4.3, ACS-Free distills the knowledge learned by the IoU regressor into a parameter-free rule based on spatial entropy ranking. Note that ACS-Free is not an independent method. Its localization-critical attention head selection is derived from the regressor's gradient and entropy analysis and would not be discoverable without first training and analyzing the regressor.

For each coordinate $c \in \{x_1, y_1, x_2, y_2\}$, we first identify the n most discriminative heads:

$$\mathcal{D}_c = \{(l,h) \in \mathcal{U} \mid (l,h) \text{ in Top-}n(\Delta H_c^{(l,h)})\},$$

where $\mathcal{U} = \{(l,h) : 1 \leq l \leq L, 1 \leq h \leq K\}$ denotes all layer-head pairs. During inference, ACS-Free evaluates each candidate b_j by computing its average entropy across $\mathcal{D}_c$:

$$\bar{H}_c(b_j) = \frac{1}{|\mathcal{D}_c|} \sum_{(l,h) \in \mathcal{D}_c} H\left(\mathcal{A}_{c,j}^{(l,h)}\right). \tag{4}$$

To combine information from all four coordinate tokens, ACS-Free converts entropy values into ranks. For each coordinate c, we define $r_c(b_j) = \text{rank}(\bar{H}_c(b_j))$, where lower entropy receives better (smaller)

rank. The overall score is: $R(b_j) = r_{x_1}(b_j) + r_{y_1}(b_j) + r_{x_2}(b_j) + r_{y_2}(b_j)$. Finally, ACS-Free selects the candidate with the lowest overall rank: $b^* = \arg\min_{b_j \in \mathcal{B}} R(b_j)$. The detailed ranking procedure is provided in Appendix E.

5. Experiments

5.1 Setup

Table 2: Object detection performance on COCO and Objects365 (single-object). We compare two variants of our **Attention-based Candidate Selection (ACS)** framework—**ACS-Learned**, the regressor-based selector, and **ACS-Free**, the training-free variant derived from analyzing the regressor's discriminative heads—against greedy decoding and six sampling-based baselines that select from N=10 sampled responses. Best method per column in **bold**; best training-free method per column (excluding ACS-Learned) is underlined.

		Qwen2.5-VL-7B				InternVL-3.5-8B			
Method	τ	**COCO**		**Objects365**		**COCO**		**Objects365**	
		Acc@0.5	**mIoU**	**Acc@0.5**	**mIoU**	**Acc@0.5**	**mIoU**	**Acc@0.5**	**mIoU**
Greedy	–	61.4	52.2	43.0	36.9	49.1	42.9	21.9	21.9
Pass@1	1.0	52.3	45.3	30.3	26.8	38.8	36.1	13.3	15.1
	0.7	59.3	50.3	36.5	32.1	45.0	40.4	16.0	17.3
	0.5	59.6	50.9	39.9	34.6	47.1	41.9	19.8	19.8
FirstValid	1.0	54.2	47.0	32.4	28.8	42.2	39.8	15.6	18.0
	0.7	60.3	51.2	38.1	33.5	48.8	44.1	18.3	20.4
	0.5	60.9	52.0	41.7	36.1	50.8	45.5	21.9	22.3
TokEntropy	1.0	51.7	43.8	32.0	28.1	39.1	37.2	11.9	14.7
	0.7	55.9	47.2	35.3	30.5	47.1	43.1	15.3	17.8
	0.5	58.1	49.0	37.7	32.8	49.1	44.2	20.3	20.5
MajVote	1.0	56.0	48.5	32.4	30.0	50.2	45.0	18.6	20.5
	0.7	59.2	51.0	37.6	33.5	51.7	46.4	21.8	22.7
	0.5	60.4	51.6	39.4	34.6	53.8	46.8	23.2	23.4
MeanBBox	1.0	32.7	32.1	14.1	15.2	36.7	35.3	10.2	14.1
	0.7	45.4	41.5	22.9	22.7	41.8	39.0	14.2	17.2
	0.5	50.1	44.8	29.8	27.7	45.4	41.1	16.6	19.1
Smallest	1.0	27.0	25.5	18.6	18.3	18.3	24.1	6.2	10.5
	0.7	42.7	37.7	26.2	24.7	28.0	30.2	10.8	13.9
	0.5	49.3	42.7	32.1	29.5	34.3	34.2	13.6	16.6
ACS-Free	1.0	59.4	50.2	39.4	34.6	47.9	44.5	20.6	21.7
	0.7	62.1	52.3	42.6	37.1	51.3	46.5	23.3	23.5
	0.5	63.4	53.5	43.0	37.9	53.1	47.4	24.8	24.4
ACS-Learned	1.0	64.0	53.0	41.5	36.5	55.0	48.3	21.9	22.5
	0.7	64.9	54.7	45.0	**39.2**	58.4	50.4	23.5	23.6
	0.5	**65.3**	**55.2**	**45.1**	39.1	**58.6**	**50.6**	**25.5**	**24.9**

Datasets and protocol. Our experiments focus on *small objects* (0.1%–1% of image area), evaluating on 2,225 instances each from MS COCO [19] (all qualifying cases in the validation set) and Objects365 [26]. For the multi-object setting, we evaluate 759 COCO cases with multiple target small objects. Cases with no valid prediction are assigned IoU = 0.

LVLMs and sampling details. We evaluate on Qwen2.5-VL-7B and InternVL-3.5-8B. The IoU regressor is trained on the COCO training split; evaluation uses held-out instances from both COCO and Objects365. We generate training/validation data using a temperature of $\tau = 1.0$, and all analyses of attention layers and heads are conducted on the validation subset. For inference, we sample N=10 responses using

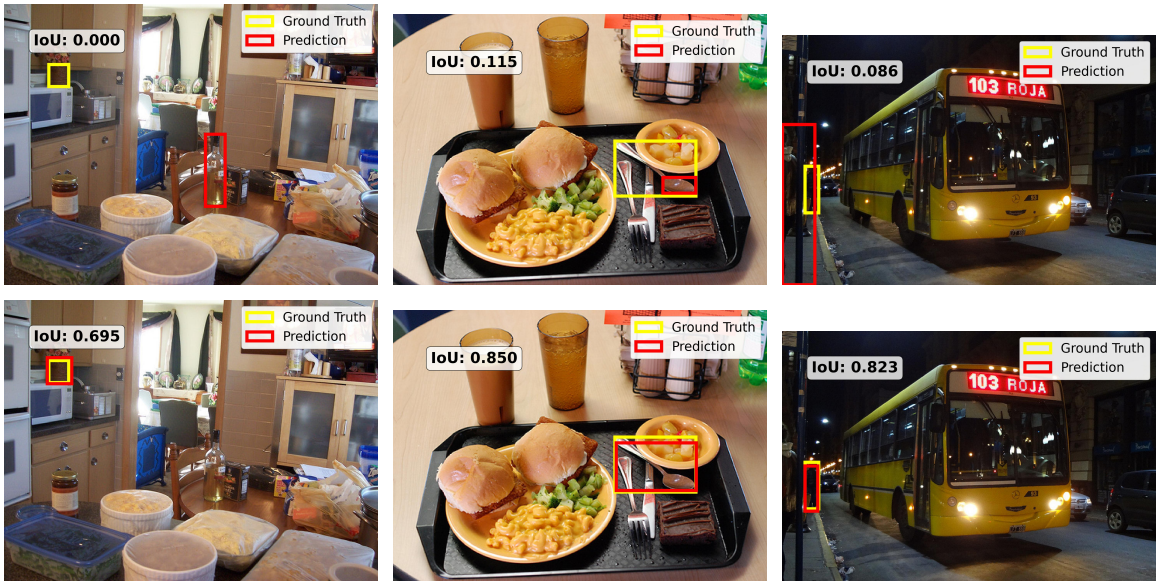


Figure 5: Qualitative comparison of localization. Yellow boxes denote the ground-truth bounding box, and red boxes denote the predicted bounding box. Each column corresponds to a different object category. Top: Greedy Decoding; Bottom: Ours.

$\tau \in \{0.5, 0.7, 1.0\}$ with top-$p = 0.9$ and top-$k = 100$. Based on gradient analysis, we focus on layers 14-21 for Qwen2.5-VL and layers 17-26 for InternVL-3.5 as localization-critical layers. Our ACS-Free uses n=3 discriminative attention heads per coordinate.

Attention extraction. LVLMs generates tokens in raw outputs. For each predicted box $[x_1, y_1, x_2, y_2]$, we locate the token for each coordinate and use these four attention maps as the input of f_θ. We collect the attention assigned to the corresponding visual tokens of LVLMs during generation. More details are in Appendix A.

Baselines. Since our approach performs self-improvement using only the LVLM's internal representations without retraining or external localization modules, we compare against baselines that similarly rely on the model's native outputs. We start with (1) **Greedy Decoding (Greedy)**: standard deterministic decoding without sampling. The remaining baselines all draw N sampled responses and differ in how they pick a single output: (2) **Pass@1** always returns response 1 with no validity check, capturing raw single-sample quality; (3) **FirstValid** selects the first response containing a parseable bounding box; (4) **TokEntropy** [30, 38] ranks responses by the summed vocabulary entropy of the four bbox coordinate tokens and picks the lowest; (5) **MajVote** synthesizes a bounding box by per-coordinate majority vote across candidates; (6) **MeanBBox** synthesizes a bounding box from the per-coordinate arithmetic mean; (7) **Smallest** selects the smallest-area candidate, designed to verify that our entropy-based selection is not simply favoring smaller boxes.

Metrics. For single-object localization, we report Acc@0.5 [11, 34] and Mean IoU (mIoU). For the IoU regressor, we report MAE and Pearson r. For the multi-object setting, we evaluate Precision, Recall, and F1.

5.2 Regressor Performance

Table 3 reports IoU regressor performance. The strong correlation (Pearson $r > 0.67$ across all settings) validates that attention alone encodes a meaningful localization signal—a proof-of-concept before we examine downstream localization gains.

Table 3: IoU regression performance on COCO val.

Temp	Qwen2.5-VL		InternVL-3.5	
	MAE	Pearson r	MAE	Pearson r
0.5	0.165	0.737	0.180	**0.682**
0.7	0.159	0.751	0.178	0.680
1.0	**0.148**	**0.780**	**0.176**	0.677

5.3 Self-Improved Localization Results

Table 2 presents results on COCO and Objects365. We discuss results in three parts.

ACS-Learned: attention-based selection works. ACS-Learned consistently outperforms all baselines across configurations, confirming that the learned regressor captures a genuine localization-quality signal. On COCO, ACS-Learned improves Acc@0.5 from 61.4% to 65.3% for Qwen2.5-VL (+6.35%) and from 49.1% to 58.6% for InternVL-3.5 (+19.35%) over Greedy decoding. On the more challenging Objects365, ACS-Learned maintains stable improvements: +4.88% for Qwen2.5-VL and +16.44% for InternVL-3.5, demonstrating robustness across diverse object types.

ACS-Free: best training-free method. ACS-Free, which distills the regressor's gradient analysis into a parameter-free entropy rule, has the strongest overall performance among all training-free methods, producing the underlined best results in 7 out of 8 column comparisons. At τ=0.5, ACS-Free improves Acc@0.5 over greedy by +3.26% for Qwen2.5-VL and +8.15% for InternVL-3.5 on COCO, demonstrating that entropy on regressor-identified heads captures most of the localization-quality signal without any learned component. The one exception—the weaker InternVL COCO Acc@0.5 at τ=0.5, where MajVote slightly exceeds ACS-Free— reflects that performance gain is affected by the base performance of a model.

TokEntropy: a negative result. TokEntropy, which ranks candidates by the summed vocabulary entropy of bbox coordinate tokens, consistently underperforms even Pass@1 across most settings. This is a clear negative result: token-level confidence does not capture spatial grounding quality. The signal that matters for localization is not how uncertain the model is about which token to generate, but how its spatial attention is organized—precisely the signal ACS-Free exploits.

Greedy vs. Sampling Trade-offs. While Greedy Decoding is simple, it often produces invalid or unusable outputs such as "cannot see the target object" or incomplete bounding boxes. This problem is particularly severe for InternVL, where greedy decoding fails to generate any valid bounding box for many cases, leading to reduced localization accuracy. Temperature sampling increases the chance of producing a valid box, which explains why FirstValid can outperform greedy decoding on InternVL, but it remains suboptimal at higher temperatures without a principled attention-based selector.

5.4 Effect of Temperature and Sampling Size

Temperature and sampling size jointly control the diversity-quality tradeoff. Lower temperatures ($\tau = 0.5$) yield more conservative and accurate boxes, while higher temperatures introduce more noisy candidates, causing FirstValid to degrade substantially (e.g., 60.9% to 54.2% Acc@0.5 on COCO for Qwen2.5-VL). As shown in Tables 4 and 5, increasing N quickly improves ACS-Learned, but performance saturates once N>9; even N=3–5 gives substantial gains across temperatures. Thus, N=10 provides a strong balance between candidate diversity, accuracy, and computational cost.

Table 4: Performance saturation across different N (Qwen2.5-VL, COCO, $\tau = 1.0$).

N	3	5	10	15	20	30
Acc@0.5	61.2	62.5	**64.0**	63.4	63.1	63.0
mIoU	46.3	51.4	**53.0**	52.9	52.6	52.6

Table 5: Effect of sampling size N across temperatures on Qwen2.5-VL (COCO).

N	$\tau = 1.0$		$\tau = 0.7$		$\tau = 0.5$	
	Acc@0.5	mIoU	Acc@0.5	mIoU	Acc@0.5	mIoU
1	53.6	46.3	60.0	51.0	60.2	51.6
3	61.2	51.4	64.3	54.1	63.7	54.2
5	62.5	52.3	**65.2**	54.5	64.8	54.8
10	**64.0**	**53.0**	64.9	**54.7**	**65.3**	**55.2**

5.5 Multiple and Medium/Large Objects

ACS also works on images with multiple objects and medium/large objects settings.

Table 6: ACS for Qwen2.5-VL on COCO multi-object.

Setting	Prec. (%)	Rec. (%)	F1 (%)
Greedy (w/o NMS)	47.21	40.19	43.42
Greedy + NMS (0.5)	**57.13**	40.01	47.06
Greedy + NMS (0.8)	56.29	40.10	46.83
ACS ($\tau = 0.5$)	49.10	**50.18**	49.63
ACS ($\tau = 0.7$)	50.93	50.13	**50.53**
ACS ($\tau = 1.0$)	50.22	47.09	48.60

Multiple objects. Sampled bounding boxes are grouped per image via IoU-based Union–Find clustering, and the candidate with the highest predicted IoU from each cluster is selected (global best if no cluster forms). Table 6 shows Greedy decoding often produces many near-duplicate boxes that reduce recall; NMS suppresses redundancy and improves precision but cannot recover recall. In contrast, ACS-Learned uses temperature-based sampling to produce more diverse hypotheses, increasing recall at a modest cost to precision—a natural precision–recall trade-off. Our method achieves the highest F1 score, showing that candidate selection is an effective solution for multi-object localization.

Medium/Large objects. We further validate on medium/large objects (occupying $> 5\%$ of image area, roughly with width and height both $> 22\%$) using an equal number of cases as the small-object setting. For Qwen2.5-VL, both Greedy and ACS-Learned reach 87% Acc@0.5—the model is already near-optimal here, leaving little headroom for selection. For InternVL-3.5, ACS-Learned still improves Acc@0.5 from 64% to 70%, showing the method can help even on larger objects when the base model has weaker localization capability. Since LVLMs generally perform better on large than small objects, we focus on the more challenging small-object setting, where attention-based selection has the most impact.

6. Conclusion

We discovered that internal attention structure in LVLMs encodes object grounding quality, validated by an IoU regressor from attention maps alone. Building on this finding, our Attention-based Candidate Selection (ACS) framework improves small-object localization through two variants: ACS-Learned, which exploits the learned regressor signal, and ACS-Free, which distills it into a parameter-free entropy rule for interpretable and deployment-friendly use. Both variants achieve substantial gains with only 3–5 samples and minimal overhead. Our findings also advance understanding of how LVLMs process spatial reasoning.

7. Limitations

Our approach relies on internal attention maps and therefore requires white-box access to the LVLM. Closed-API models such as GPT-4V or Gemini do not expose attention activations, and so ACS is not directly applicable in those settings. This trade-off is inherent to attention-based methods and is what enables our mechanistic analysis.

As a sampling-based method, ACS incurs additional inference cost relative to greedy decoding. Performance saturates at 3–5 samples and the overhead is modest—attention is already computed during the forward pass and the regressor is lightweight—but latency-sensitive deployments may still prefer single-pass decoding.

The Attention-IoU Regressor in ACS-Learned is trained per LVLM and is not directly transferable across architectures with different attention shapes. Training is lightweight (3.6–4.4M parameters; minutes on a single GPU), and once a regressor is trained, the resulting head analysis enables ACS-Free, which requires no learned component at inference.

8. Ethical Considerations

Data and models. All experiments use public benchmarks (MS COCO [19] and Objects365 [26]) and open-source LVLMs (Qwen2.5-VL and InternVL-3.5) under their respective licenses. No new data was collected and no personally identifiable information is involved.

Intended use and misuse. Our method improves small-object localization in LVLMs and is intended for benign applications such as assistive vision, robotics, and autonomous-driving safety. As with any object-detection technique, the improved reliability could in principle be repurposed for surveillance; the contribution is methodological and is not tailored toward such use cases.

Use of LLMs. Large language models were used for language polishing (grammar and phrasing) and minor coding assistance (e.g., shell scripts, plotting). All research ideas, methodology, experimental design, analyses, and claims are the authors' own.

References

[1] Zeyuan Allen-Zhu. ICML 2024 Tutorial: Physics of Language Models, July 2024. Project page: https://physics.allen-zhu.com/.

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[3] Jing Bi, Junjia Guo, Yunlong Tang, Lianggong Bruce Wen, Zhang Liu, Bingjie Wang, and Chenliang Xu. Unveiling visual perception in language models: An attention head analysis approach. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4135–4144, 2025.

[4] 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*, 2023.

[5] Shiqi Chen, Tongyao Zhu, Ruochen Zhou, Jinghan Zhang, Siyang Gao, Juan Carlos Niebles, Mor Geva, Junxian He, Jiajun Wu, and Manling Li. Why is spatial reasoning hard for vlms? an attention mechanism perspective on focus areas. *arXiv preprint arXiv:2503.01773*, 2025.

[6] Zhenfang Chen, Qinhong Zhou, Yikang Shen, Yining Hong, Zhiqing Sun, Dan Gutfreund, and Chuang Gan. Visual chain-of-thought prompting for knowledge-based visual reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 1254–1262, 2024.

[7] Kanzhi Cheng, Li YanTao, Fangzhi Xu, Jianbing Zhang, Hao Zhou, and Yang Liu. Vision-language models can self-improve reasoning via reflection. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8876–8892, 2025.

[8] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[9] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[10] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 91–104, 2025.

[11] Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. Your large vision-language model only needs a few attention heads for visual grounding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9339–9350, 2025.

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

[13] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9579–9589, 2024.

[14] Mengcheng Lan, Chaofeng Chen, Yue Zhou, Jiaxing Xu, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. Text4seg: Reimagining image segmentation as text generation. *arXiv preprint arXiv:2410.09855*, 2024.

[15] Jiaxi Li, Yucheng Shi, Jin Lu, and Ninghao Liu. Mits: Enhanced tree search reasoning for llms via pointwise mutual information. *arXiv preprint arXiv:2510.03632*, 2025.

[16] Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333, 2023.

[17] Yiwei Li, Yikang Liu, Jiaqi Guo, Lin Zhao, Zheyuan Zhang, Xiao Chen, Boris Mailhe, Ankush Mukherjee, Terrence Chen, and Shanhui Sun. Rau: Reference-based anatomical understanding with vision language models. *arXiv preprint arXiv:2509.22404*, 2025.

[18] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[20] Zhiwei Lin, Yongtao Wang, and Zhi Tang. Training-free open-ended object detection and segmentation via attention as prompts. *Advances in Neural Information Processing Systems*, 37:69588–69606, 2024.

[21] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[22] Yuqi Liu, Bohao Peng, Zhisheng Zhong, Zihao Yue, Fanbin Lu, Bei Yu, and Jiaya Jia. Seg-zero: Reasoning-chain guided segmentation via cognitive reinforcement. *arXiv preprint arXiv:2503.06520*, 2025.

[23] Zhenting Qi, Mingyuan Ma, Jiahang Xu, Li Lyna Zhang, Fan Yang, and Mao Yang. Mutual reasoning makes smaller llms stronger problem-solvers. *arXiv preprint arXiv:2408.06195*, 2024.

[24] Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. Pixellm: Pixel reasoning with large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26374–26383, 2024.

[25] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: visual explanations from deep networks via gradient-based localization. *International journal of computer vision*, 128:336–359, 2020.

[26] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019.

[27] Haozhan Shen, Tiancheng Zhao, Mingwei Zhu, and Jianwei Yin. Groundvlp: Harnessing zero-shot visual grounding from vision-language pre-training and open-vocabulary object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4766–4775, 2024.

[28] Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, Ruochen Xu, and Tiancheng Zhao. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.

[29] Yucheng Shi, Quanzheng Li, Jin Sun, Xiang Li, and Ninghao Liu. Enhancing cognition and explainability of multimodal foundation models with self-synthesized data. *arXiv preprint arXiv:2502.14044*, 2025.

[30] Yucheng Shi, Tianze Yang, Canyu Chen, Quanzheng Li, Tianming Liu, Xiang Li, and Ninghao Liu. Searchrag: Can search engines be helpful for llm-based medical question answering? In *2025 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 4051–4056. IEEE, 2025.

[31] Karen Simonyan. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[32] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

[33] Mirac Suzgun, Luke Melas-Kyriazi, and Dan Jurafsky. Follow the wisdom of the crowd: Effective text generation via minimum bayes risk decoding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4265–4293, 2023.

[34] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[35] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[36] Mingrui Wu, Xinyue Cai, Jiayi Ji, Jiale Li, Oucheng Huang, Gen Luo, Hao Fei, Guannan Jiang, Xiaoshuai Sun, and Rongrong Ji. Controlmllm: Training-free visual prompt learning for multimodal large language models. *Advances in Neural Information Processing Systems*, 37:45206–45234, 2024.

[37] Size Wu, Sheng Jin, Wenwei Zhang, Lumin Xu, Wentao Liu, Wei Li, and Chen Change Loy. F-lmm: Grounding frozen large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24710–24721, 2025.

[38] Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*, 2023.

[39] Tianze Yang, Yucheng Shi, Mengnan Du, Xuansheng Wu, Qiaoyu Tan, Jin Sun, and Ninghao Liu. Concept-centric token interpretation for vector-quantized generative models. *arXiv preprint arXiv:2506.00698*, 2025.

[40] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

[41] Heng Yin, Yuqiang Ren, Ke Yan, Shouhong Ding, and Yongtao Hao. Rod-mllm: Towards more reliable object detection in multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14358–14368, June 2025.

[42] Yufei Zhan, Shurong Zheng, Yousong Zhu, Hongyin Zhao, Fan Yang, Ming Tang, and Jinqiao Wang. Griffon v2: Advancing multimodal perception with high-resolution scaling and visual-language co-referring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22947–22957, 2025.

[43] Jiarui Zhang, Mahyar Khayatkhoei, Prateek Chhikara, and Filip Ilievski. Towards perceiving small visual details in zero-shot visual question answering with multimodal llms. *arXiv preprint arXiv:2310.16033*, 2023.

[44] Zhisong Zhang, Yan Wang, Xinting Huang, Tianqing Fang, Hongming Zhang, Chenlong Deng, Shuaiyi Li, and Dong Yu. Attention entropy is a key factor: An analysis of parallel context encoding with full-attention-based pre-trained language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9840–9855, 2025.

[45] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *European conference on computer vision*, pages 350–368. Springer, 2022.

APPENDIX

Appendix Table of Contents

Appendix Section	Page
A. Training Setup for the Attention-IoU Regressor	15
B. Differential Entropy of a Gaussian Random Variable	19
C. Complete Attention Map Visualizations	19
D. Prompt for Object Localization	24
E. Rank-based Entropy Selection	24
F. Entropy Analysis in Localization-Critical Layers	25
G. Additional Qualitative Results	25

A. Training Setup for the Attention-IoU Regressor

We focus on Qwen2.5-VL and InternVL-3.5 because both LVLMs possess strong object-localization capabilities and, critically, can directly output bounding box coordinates in their responses. The training data for the IoU regressor are constructed from 5,000 image-category pairs. For each pair, each LVLM was queried with temperature $\tau = 1.0$ to produce 10 sampled responses, and all valid predicted bounding boxes in these responses are collected. Each candidate box is paired with its four coordinate-specific attention maps based on (x_1, y_1, x_2, y_2), extracted from all layers and heads of the underlying LVLM. For a coordinate value such as $[15,\ 240,\ 35,\ 460]$, we extract the attention maps by selecting the *first token* of each coordinate. Concretely, for example, for a predicted box $[15,\ 240,\ 35,\ 460]$, we take the first token of each coordinate: the first coordinate uses the attention of token 1 in 15 for A_{x_1}, the second coordinate uses the attention of token 2 in 240 for A_{y_1}, the third uses token 3 in 35 for A_{x_2}, and the fourth uses token 4 in 460 for A_{y_2}. Thus, although the coordinate value is 240, its associated attention map comes from token 2.

The IoU regressor takes these attention maps as input and predicts the IoU between the candidate box and the ground-truth box.

A.1 Model configuration

For Qwen2.5-VL, attention maps contain 28 layers with 28 heads per layer, while InternVL-3.5 provides 36 layers and 32 heads. Each coordinate branch uses a CNN encoder with hidden dimensions $\{64, 128, 256\}$ and dropout rate 0.3, followed by a fusion MLP of dimension 256. A sigmoid activation constrains the output to $[0, 1]$.

A.2 Optimization

All IoU regressors are trained with Adam as the optimizer, a learning rate of 3e-4, a weight decay of 1e-4, and a batch size of **64**. Training runs for up to **100 epochs**, using MSE loss and cosine learning-rate scheduling. Early stopping is applied with patience 15 and minimum improvement 0.001.

A.3 Attention extraction

For Qwen2.5-VL, attention maps are computed over all image tokens produced by the vision encoder, and all heads are stacked along the channel dimension; the maps are then reshaped to a spatial size of 24×24. For InternVL-3.5, we compute attention over the last 256 vision tokens, because these tokens correspond to the model's thumbnail representation; all heads are similarly stacked as channels, while the native 16×16 resolution is preserved.

A.4 Model Architecture and Computational Efficiency

The detailed architecture and computational cost of the regressor are summarized in Tables 7–10. Specifically, Tables 7 and 8 present the layer-wise structure and parameter statistics for the regressor trained on Qwen2.5-VL attention features, while Tables 9 and 10 provide the corresponding details for InternVL-3.5.

Despite having 3.6M and 4.4M parameters of the regressors for Qwen2.5-VL and InternVL-3.5 respectively, regressor inference is highly efficient: processing 2,225 test cases takes only 5–6 minutes on a single A6000 GPU, which is negligible compared to the LVLM's own inference time. This efficiency makes the IoU regressor a practical and scalable solution for real-time bounding box selection in production environments.

Table 7: **Layer Summary of the Attention-IoU Regressor trained on Qwen2.5-VL attention maps.**

Layer (type:depth-idx)	**Output Shape**	**Param #**
Attention-IoU Regressor	[1, 1]	–
CoordinateAttentionCNN Branch (repeated 4×):		
Conv2d: 1-1	[1, 64, 24, 24]	451,648
BatchNorm2d: 1-2	[1, 64, 24, 24]	128
ReLU: 1-3	[1, 64, 24, 24]	–
MaxPool2d: 1-4	[1, 64, 12, 12]	–
Conv2d: 1-5	[1, 128, 12, 12]	73,856
BatchNorm2d: 1-6	[1, 128, 12, 12]	256
ReLU: 1-7	[1, 128, 12, 12]	–
MaxPool2d: 1-8	[1, 128, 6, 6]	–
Conv2d: 1-9	[1, 256, 6, 6]	295,168
BatchNorm2d: 1-10	[1, 256, 6, 6]	512
ReLU: 1-11	[1, 256, 6, 6]	–
AdaptiveAvgPool2d: 1-12	[1, 256, 1, 1]	–
Linear: 1-13	[1, 128]	32,896
Fusion MLP:		
Linear: 2-1	[1, 256]	131,328
BatchNorm1d: 2-2	[1, 256]	512
ReLU: 2-3	[1, 256]	–
Linear: 2-4	[1, 128]	32,896
ReLU: 2-5	[1, 128]	–
Linear: 2-6	[1, 1]	129

Table 8: **Parameters and Memory Usage of the Attention-IoU Regressor trained on Qwen2.5-VL attention maps.**

Description	**Value**
Total parameters	3,582,721
Trainable parameters	3,582,721
Non-trainable parameters	0
Total mult-adds (G)	1.13
Input size (MB)	7.23
Forward/backward pass size (MB)	4.14
Params size (MB)	14.33
Estimated total size (MB)	25.69

Table 9: **Layer Summary of the Attention-IoU Regressor trained on InternVL-3.5 attention maps.**

Layer (type:depth-idx)	**Output Shape**	**Param #**
Attention-IoU Regressor	[1, 1]	–
CoordinateAttentionCNN Branch (4× identical branches):		
Conv2d: 1-1	[1, 64, 16, 16]	663,616
BatchNorm2d: 1-2	[1, 64, 16, 16]	128
ReLU: 1-3	[1, 64, 16, 16]	–
Dropout2d: 1-4	[1, 64, 16, 16]	–
MaxPool2d: 1-5	[1, 64, 8, 8]	–
Conv2d: 1-6	[1, 128, 8, 8]	73,856
BatchNorm2d: 1-7	[1, 128, 8, 8]	256
ReLU: 1-8	[1, 128, 8, 8]	–
Dropout2d: 1-9	[1, 128, 8, 8]	–
MaxPool2d: 1-10	[1, 128, 4, 4]	–
Conv2d: 1-11	[1, 256, 4, 4]	295,168
BatchNorm2d: 1-12	[1, 256, 4, 4]	512
ReLU: 1-13	[1, 256, 4, 4]	–
AdaptiveAvgPool2d: 1-14	[1, 256, 1, 1]	–
Flatten: 1-15	[1, 256]	–
Dropout: 1-16	[1, 256]	–
Linear: 1-17	[1, 128]	32,896
Fusion MLP:		
Linear: 2-1	[1, 256]	131,328
ReLU: 2-2	[1, 256]	–
BatchNorm1d: 2-3	[1, 256]	512
Dropout: 2-4	[1, 256]	–
Linear: 2-5	[1, 128]	32,896
ReLU: 2-6	[1, 128]	–
Dropout: 2-7	[1, 128]	–
Linear: 2-8	[1, 1]	129

Table 10: **Parameters and Memory Usage of the Attention-IoU Regressor trained on InternVL-3.5 attention maps.**

Description	**Value**
Total parameters	4,430,593
Trainable parameters	4,430,593
Non-trainable parameters	0
Total mult-adds (M)	717.64
Input size (MB)	4.72
Forward/backward pass size (MB)	1.84
Params size (MB)	17.72
Estimated total size (MB)	24.29

B. Differential Entropy of a Gaussian Random Variable

This section supplements the result used in Section 3.2 of the main paper by providing the full derivation of the conditional Gaussian entropy.

We derive the standard result used in the main text:

$$H(Y \mid f_\theta(\mathbf{A})) = \frac{1}{2}\log(2\pi e\sigma^2),$$

under the standard regression noise model where

$$Y \mid f_\theta(\mathbf{A}) = z \sim \mathcal{N}(z, \sigma^2).$$

That is, the conditional distribution of Y given the prediction $z = f_\theta(\mathbf{A})$ is Gaussian with mean z and variance σ^2.

Derivation. By definition, the conditional differential entropy is

$$H(Y \mid Z) = -\int p(y \mid z)\, \log p(y \mid z)\, dy,$$

where $Z = f_\theta(\mathbf{A})$. If $Y \mid Z = z \sim \mathcal{N}(z, \sigma^2)$, then

$$p(y \mid z) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-z)^2}{2\sigma^2}\right),$$
$$\log p(y \mid z) = -\tfrac{1}{2}\log(2\pi\sigma^2) - \frac{(y-z)^2}{2\sigma^2}.$$

Substituting into the entropy integral:

$$\begin{aligned} H(Y \mid Z = z) &= -\int p(y \mid z)\, \log p(y \mid z)\, dy \\ &= \int p(y \mid z)\Big[\tfrac{1}{2}\log(2\pi\sigma^2) + \tfrac{(y-z)^2}{2\sigma^2}\Big] dy \\ &= \tfrac{1}{2}\log(2\pi\sigma^2) + \tfrac{1}{2\sigma^2}\int p(y \mid z)(y-z)^2 dy. \end{aligned}$$

Because $\int p(y \mid z)dy = 1$ and $\int p(y \mid z)(y-z)^2 dy = \sigma^2$,

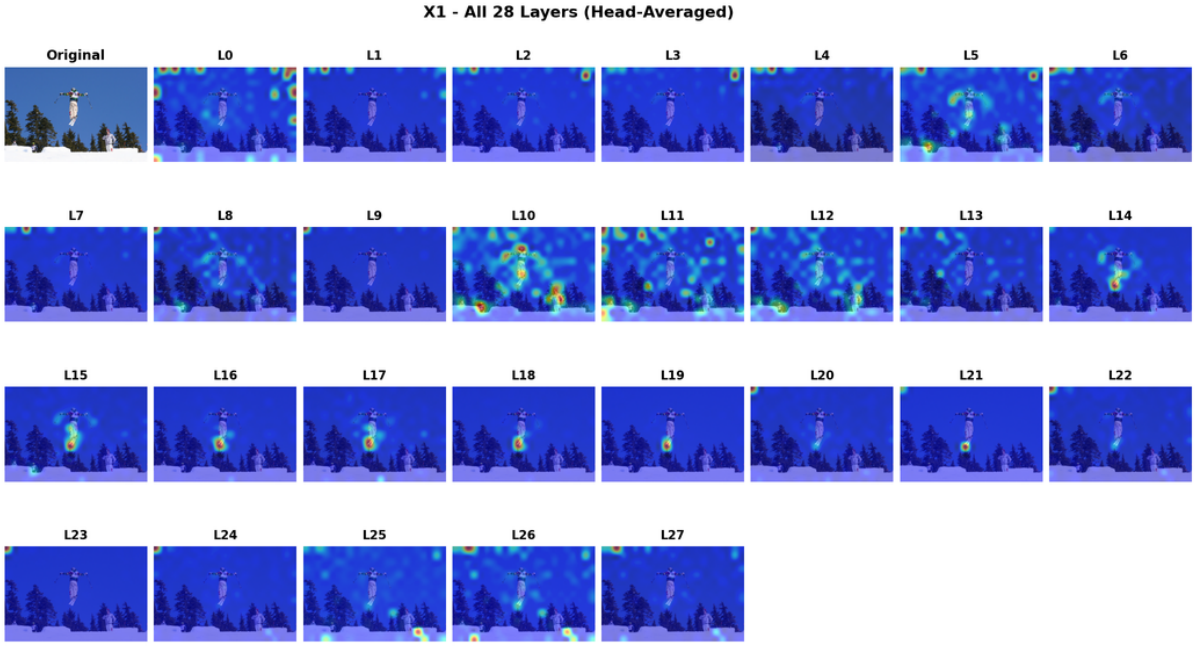
$$H(Y \mid Z = z) = \tfrac{1}{2}\log(2\pi\sigma^2) + \tfrac{1}{2} = \tfrac{1}{2}\log(2\pi e\sigma^2).$$

Since the expression does not depend on z, the conditional entropy is

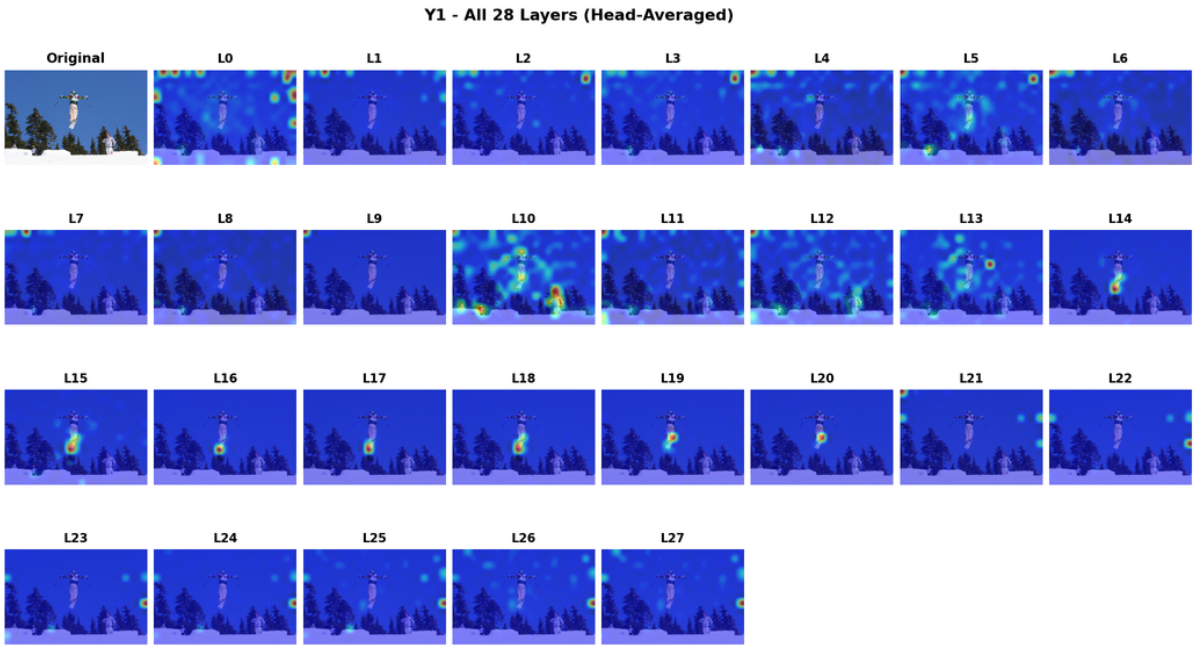
$$H(Y \mid Z) = \tfrac{1}{2}\log(2\pi e\sigma^2).$$

C. Complete Attention Map Visualizations

Below we provide the complete attention map visualizations of Qwen2.5-VL and InternVL-3.5 corresponding to Figure 3 in the main paper. These figures show all layers and coordinate-specific attention maps used in our analysis.

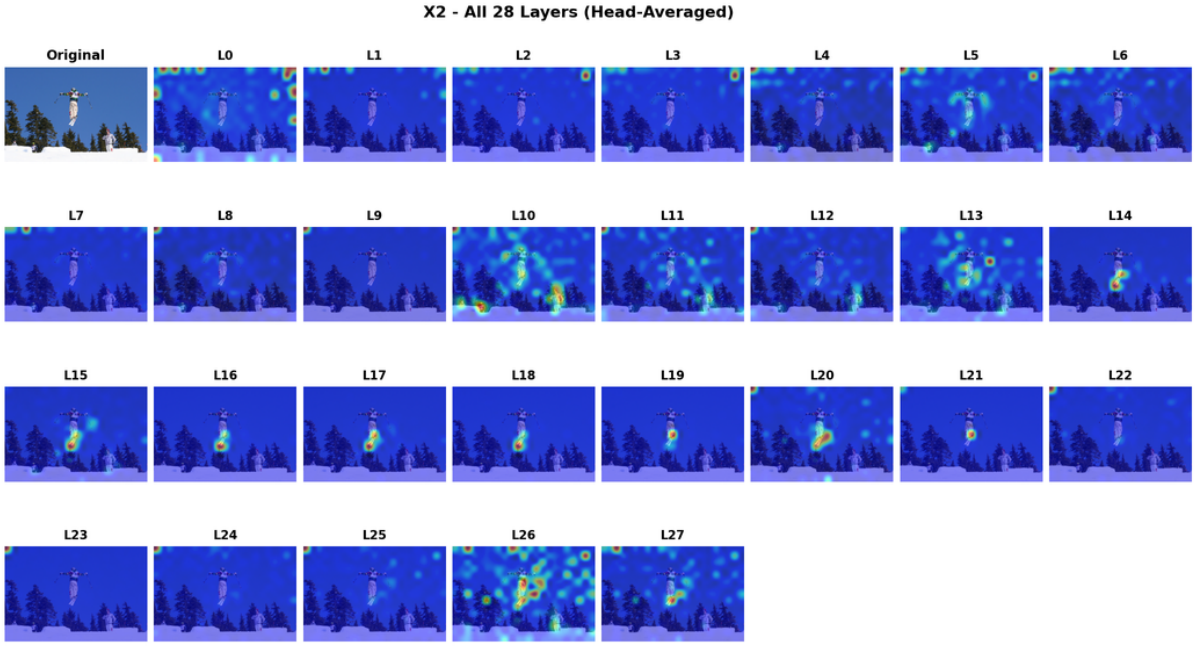


(a) Attention maps across all layers for x1 coordinate prediction.

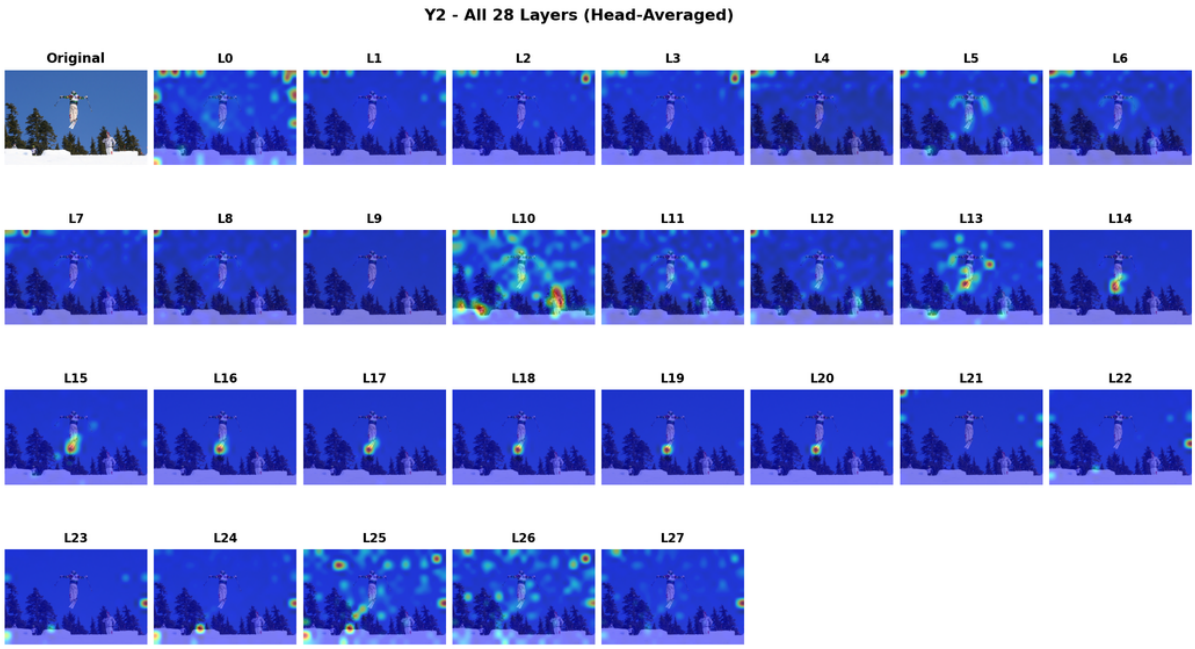


(b) Attention maps across all layers for y1 coordinate prediction.

Figure 6: **Qwen2.5-VL attention visualization for object localization (Part 1).** Early layers exhibit diffuse and globally distributed attention, while mid-to-late layers progressively focus on the target object for both (a) x_1 and (b) y_1 coordinate prediction. These coordinate-specific attention maps serve as the input to the regressor for estimating the IoU of each candidate bounding box. For each layer, the visualization represents the attention map averaged across all heads. This example is randomly selected, and other samples exhibit similar patterns.

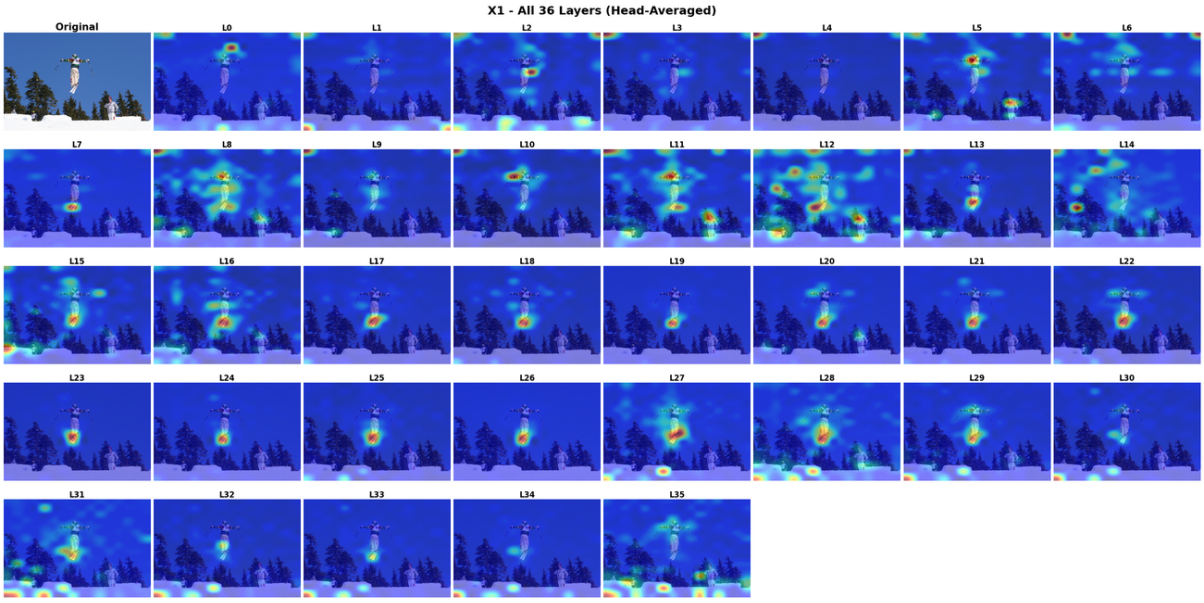


(a) x2 coordinate prediction.

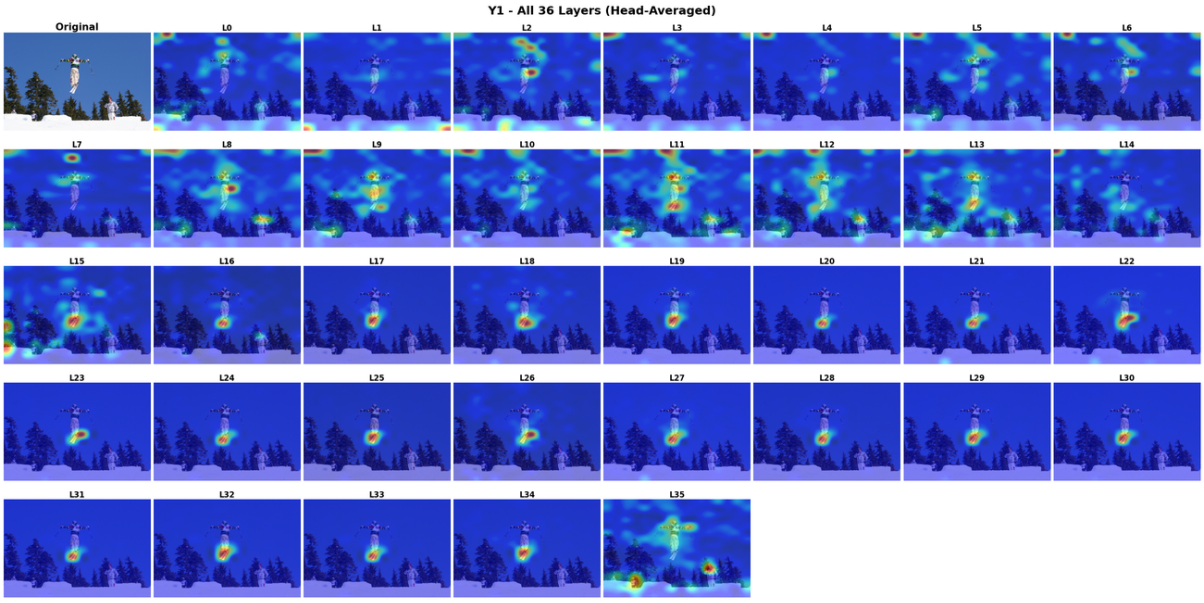


(b) y2 coordinate prediction.

Figure 7: **Qwen2.5-VL attention visualization for object localization (Part 2).** Early layers exhibit diffuse and globally distributed attention, while mid-to-late layers progressively focus on the target object for both (a) x_2 and (b) y_2 coordinate prediction. These coordinate-specific attention maps serve as the input to the regressor for estimating the IoU of each candidate bounding box. For each layer, the visualization represents the attention map averaged across all heads. This example is randomly selected, and other samples exhibit similar patterns.

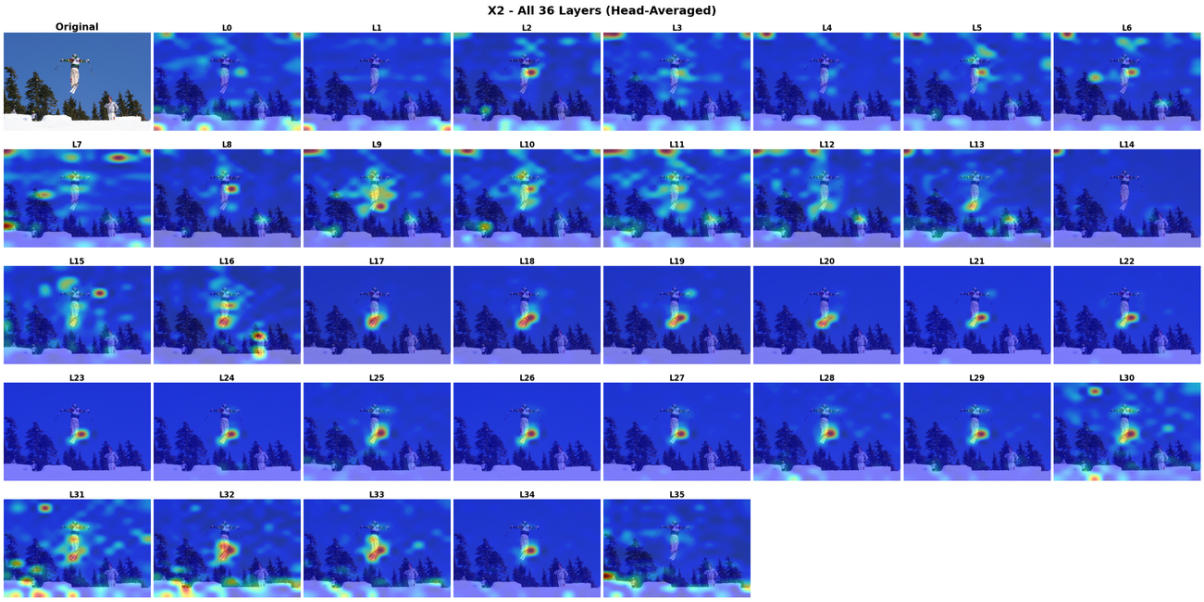


(a) x1 coordinate prediction.

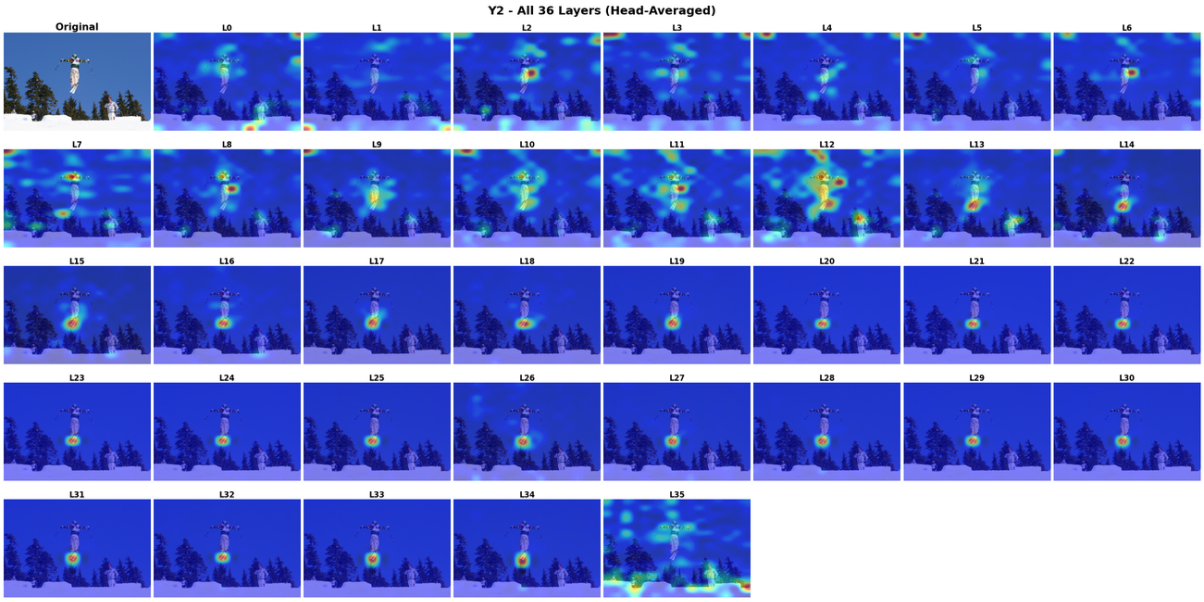


(b) y1 coordinate prediction.

Figure 8: **InternVL-3.5 attention visualization for object localization (Part 1).** Early layers show diffuse and broadly distributed attention, while mid-to-late layers gradually focus on the target object for both (a) x_1 and (b) y_1 coordinate prediction. These coordinate-specific attention maps serve as the input to the regressor for estimating the IoU of each candidate bounding box. For each layer, the visualization represents the attention map averaged across all heads. This example is randomly selected, and other samples exhibit similar patterns.



(a) x2 coordinate prediction.



(b) y2 coordinate prediction.

Figure 9: **InternVL-3.5 attention visualization for object localization (Part 2).** Early layers show diffuse and broadly distributed attention, while mid-to-late layers gradually focus on the target object for both (a) x_2 and (b) y_2 coordinate prediction. These coordinate-specific attention maps serve as the input to the regressor for estimating the IoU of each candidate bounding box. For each layer, the visualization represents the attention map averaged across all heads. This example is randomly selected, and other samples exhibit similar patterns.

D. Prompt for Object Localization

To ensure a fair comparison across models, both Qwen2.5-VL and InternVL-3.5 are prompted using the same instruction shown below. With this prompt, the LVLM produces responses that follow the example format and include bounding boxes in a consistent, structured output.

Prompt for Localization

```
Detect all instances of {object_name} in the image:

Your task is to detect any {object_name} objects that are visible in the image.

For each detected object, provide:
The bounding box coordinates [x1, y1, x2, y2]

Output format (as a JSON array):
[
  {"bbox_2d": [x1, y1, x2, y2], "label": "{object_name}"},
  {"bbox_2d": [x1, y1, x2, y2], "label": "{object_name}"}
]

Important: Focus on any {object_name} that might be present in the image,
especially those that are still visible and NOT masked out.

Example output:
[
  {"bbox_2d": [200, 150, 280, 220], "label": "{object_name}"},
  {"bbox_2d": [350, 180, 420, 270], "label": "{object_name}"}
]
```

E. Rank-based Entropy Selection

In the main paper, we describe ACS-Free using rank-based aggregation to select bounding boxes. Here we clarify the ranking procedure. Given a candidate set $\mathcal{B} = \{b_1, b_2, \ldots, b_T\}$ of T bounding boxes for an image, we first compute the average entropy $\bar{H}_c(b_j)$ for each coordinate $c \in \{x_1, y_1, x_2, y_2\}$ across the discriminative attention heads $\mathcal{D}_c$. Then, for each coordinate independently, we rank all candidates based on their entropy values in ascending order (lower entropy receives better rank), yielding coordinate-wise ranks $r_c(b_j) \in \{1, 2, \ldots, T\}$. The total rank sum $R(b_j) = r_{x_1}(b_j) + r_{y_1}(b_j) + r_{x_2}(b_j) + r_{y_2}(b_j)$ aggregates the quality across all four coordinates. Finally, we select the bounding box with the minimum rank sum: $b^* = \arg\min_{b_j \in \mathcal{B}} R(b_j)$. This rank-based approach is more robust than directly using mean entropy values, as it is invariant to entropy scale differences across coordinates and ensures balanced contribution from all four coordinate predictions.

F. Entropy Analysis in Localization-Critical Layers

Figure 10 presents the full version of Figure 4 in the main text, providing a comprehensive visualization of our localization-critical layer analysis, including the entropy patterns of attention maps for all four coordinate tokens. The figure shows that in the layers identified by the IoU regressor, high-IoU predictions exhibit clearly lower entropy than low- or zero-IoU cases, indicating stronger and more focused attention. This consistent separation confirms that these layers contain the dominant localization signal. The visualization also highlights which layers provide the most discriminative patterns, motivating the design of **ACS-Free**, the training-free variant of our framework that uses entropy on these discriminative heads for bounding box selection.

Implementation detail of ACS-Free. For ACS-Free, we construct $\mathcal{D}_c$ as the index set of the top-3 most discriminative heads for the attention map of coordinate c; during inference, the attention maps from these heads are used to compute the coordinate-wise entropy $\bar{H}_c(b_j)$ for ranking candidate boxes.

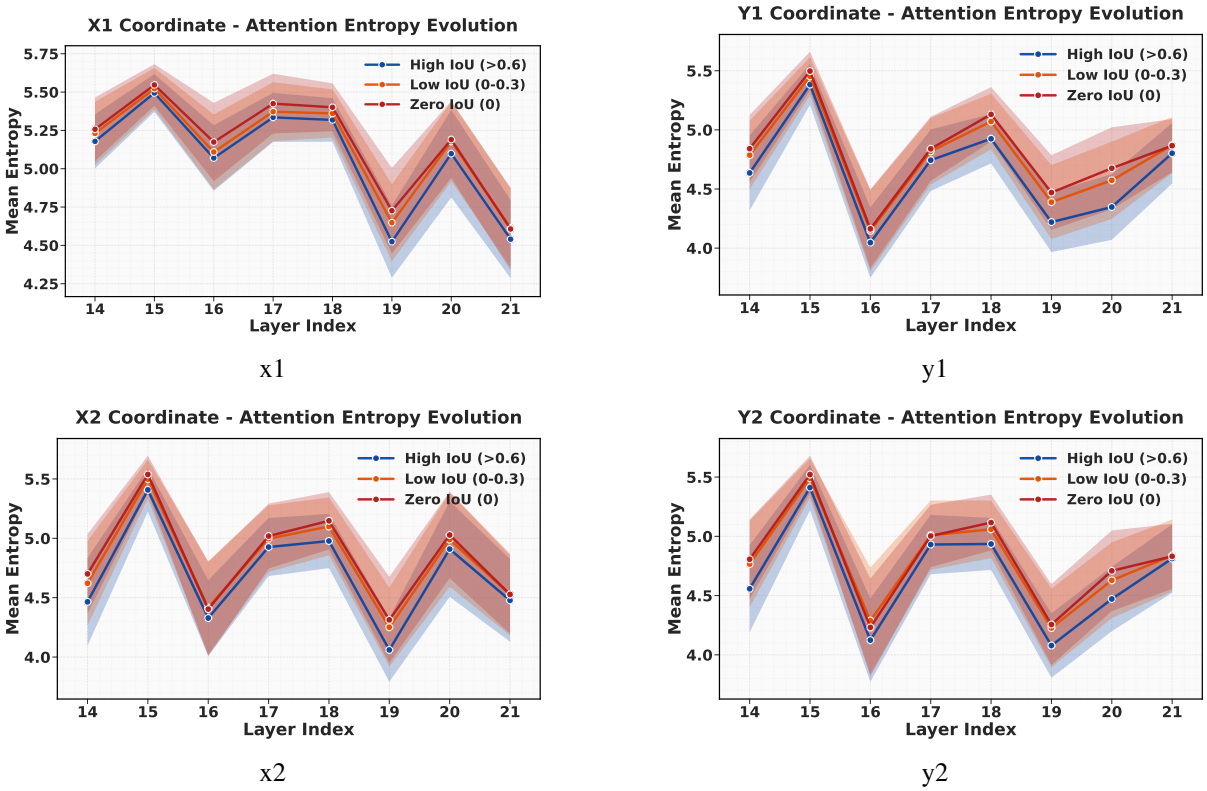


Figure 10: Important layers analysis across the attention maps of four bounding box coordinates.

G. Additional Qualitative Results

Figures 11 to 15 provide additional qualitative localization results that complement Figure 5 in the main paper. Each panel corresponds to a different object category. For each case, the top image shows the prediction from Greedy decoding and the bottom image shows the result from our method, following the same visualization protocol as the main text: yellow boxes denote ground-truth bounding boxes and red boxes denote predicted bounding boxes, with the IoU value annotated in the corner.

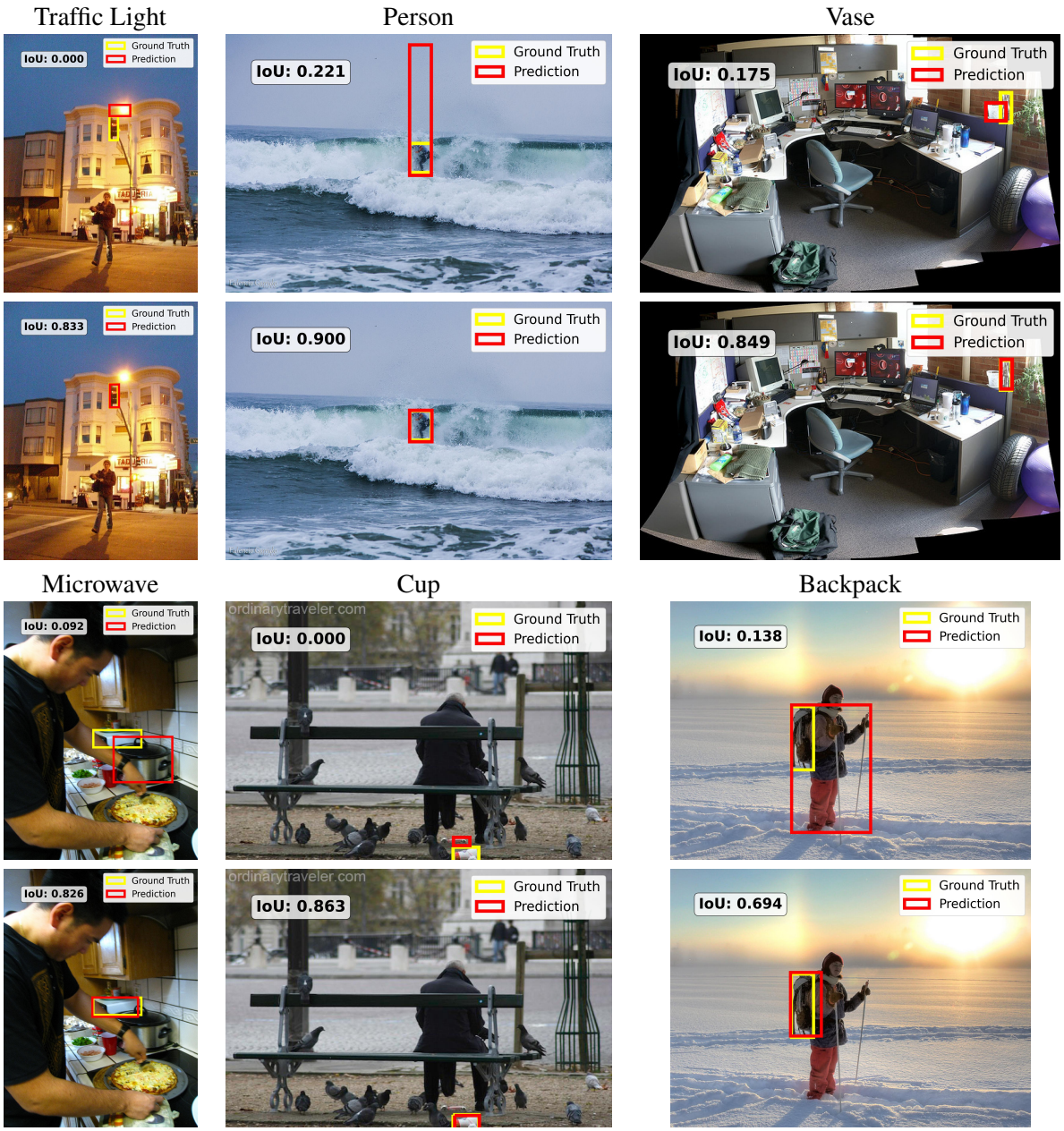


Figure 11: Additional qualitative comparisons on COCO using Qwen2.5-VL. Each case shows Greedy (top) and Ours (bottom).

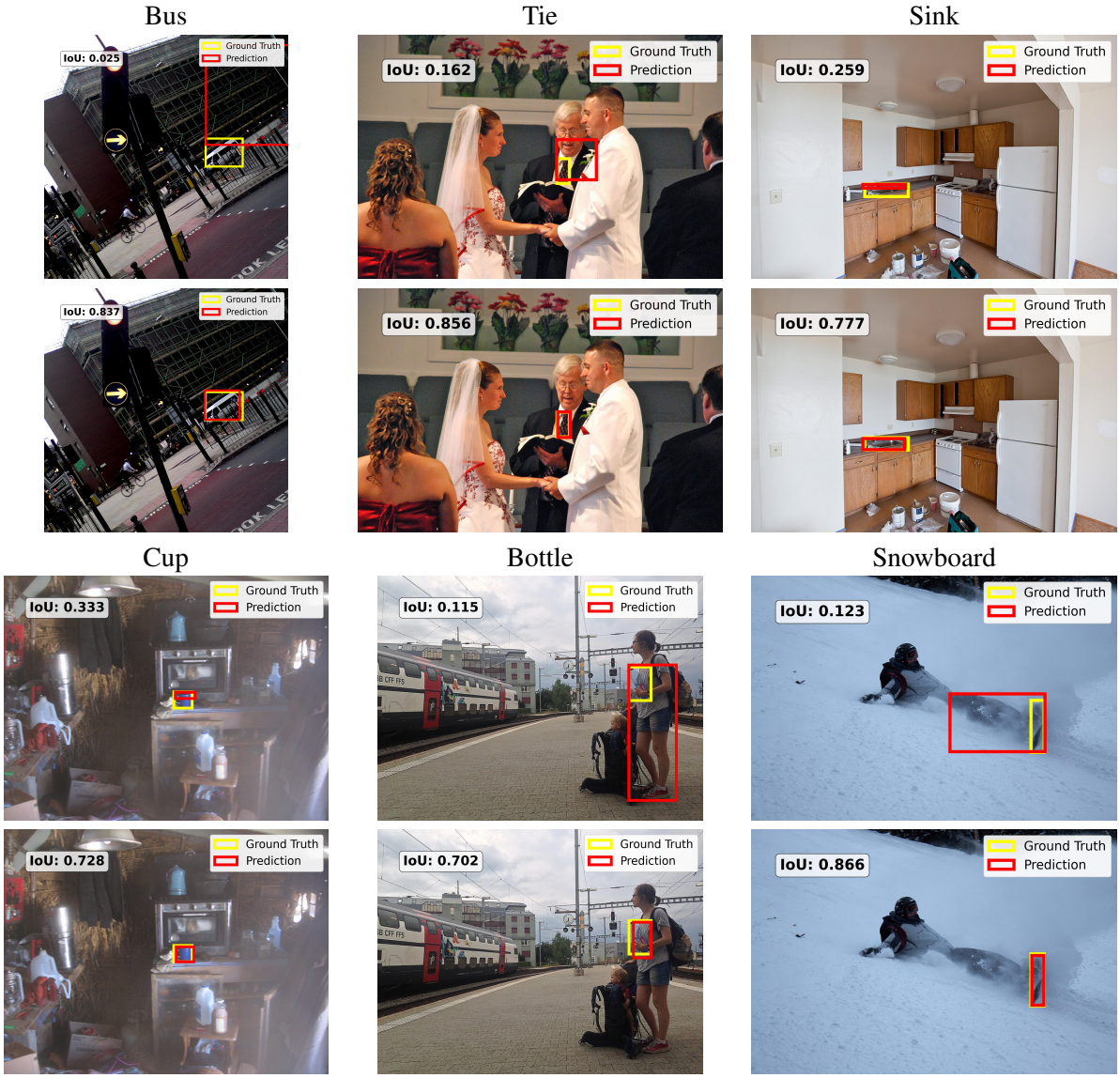


Figure 12: Additional qualitative comparisons on COCO using Qwen2.5-VL. Each case shows Greedy (top) and Ours (bottom).

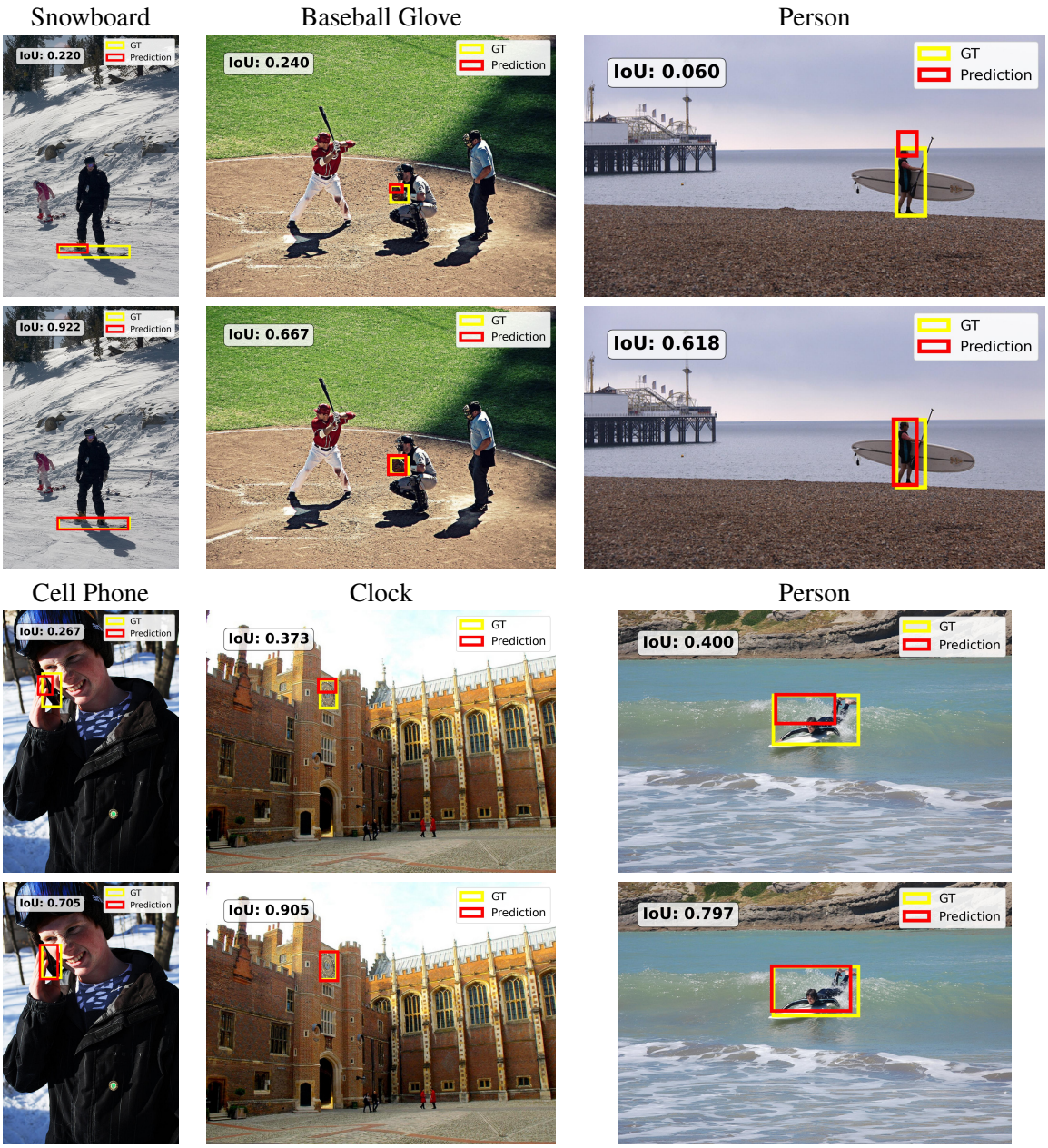


Figure 13: Additional qualitative comparisons on COCO using InternVL3.5. Each case shows Greedy (top) and Ours (bottom).

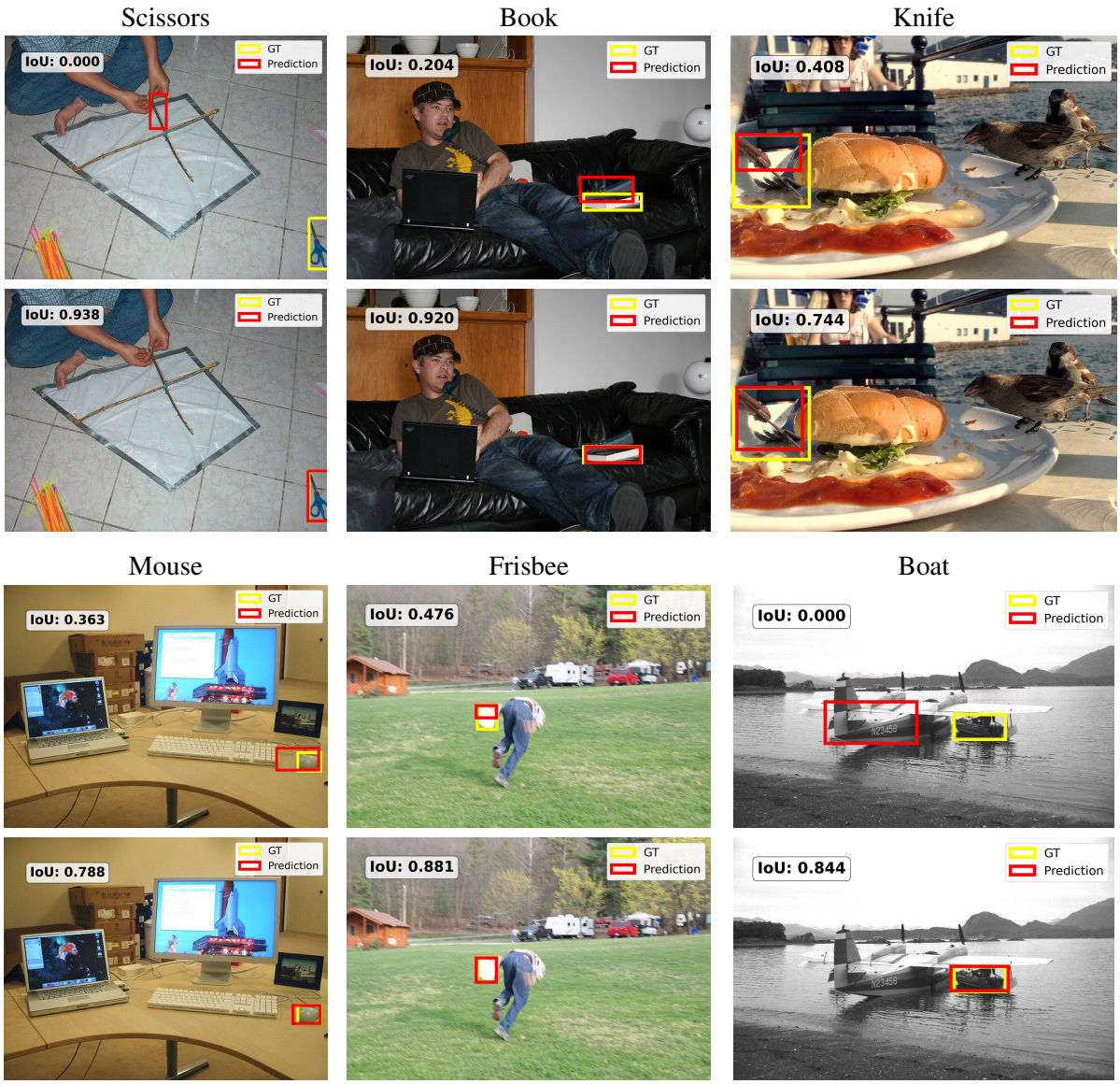


Figure 14: Additional qualitative comparisons on COCO using InternVL3.5. Each case shows Greedy (top) and Ours (bottom).

Figure 15: Additional qualitative comparisons on the Objects365 dataset using Qwen2.5-VL. Each case shows Greedy (top) and Ours (bottom).