# BAS-OPD: BUDGET-AWARE SELECTIVE ON-POLICY SELF-DISTILLATION FOR FINE-GRAINED MULTIMODAL PERCEPTION

**Anonymous authors**
Paper under double-blind review

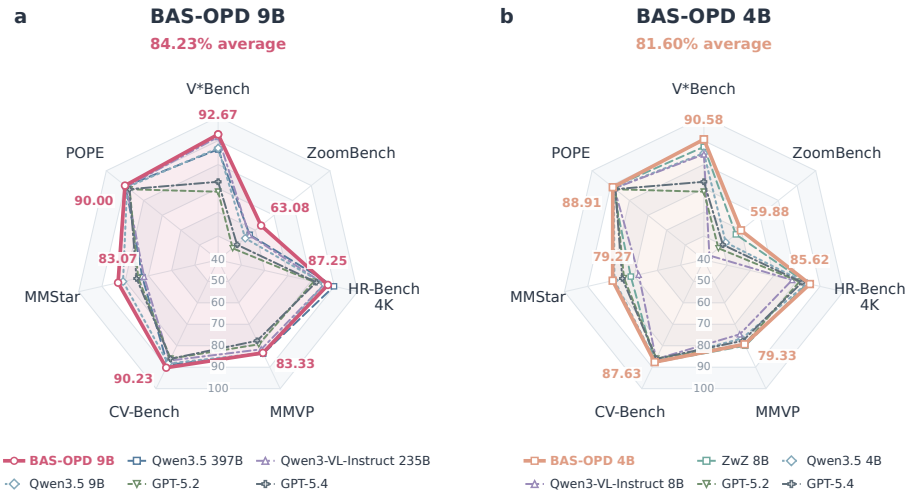



Figure 1: **Accuracy (%) across seven visual benchmarks.** In both panels, BAS-OPD achieves the highest seven-benchmark average among the compared models.

## ABSTRACT

Multimodal large language models (MLLMs) often struggle with fine-grained visual perception when critical evidence appears in local regions. On-policy distillation (OPD) can transfer privileged regional knowledge into a single-pass full-image policy, but formulations typically supervise every eligible rollout. This implicitly treats teacher feedback as equally valuable, even though the student policy evolves during on-policy training and the supervision value of different rollouts need not remain uniform. We therefore formulate privileged OPD as an online rollout-selection problem to study which rollouts merit supervision and introduce Budget-Aware Selective On-Policy Self-Distillation (**BAS-OPD**). The student generates a full candidate pool, while **BAS-OPD** selectively assigns crop-teacher supervision using random, uncertainty-based, or learned utility-based selection. Because the target utility is observable only after teacher scoring, we adopt a lightweight predictor to estimate it beforehand from detached student-side rollout statistics. Only selected rollouts enter the crop-teacher forward pass during training, while inference remains single-pass and selector-free. Across seven benchmarks, learned selection outperforms random and entropy-based selection at the same ratio, showing that how teacher supervision is allocated matters in addition to how much is used. With Qwen3.5-4B/9B, our method achieves 81.60%/84.23% average accuracy, exceeding full querying by 1.84/2.04 percentage while using only 25% of eligible rollouts and scoring 15.78%/16.02% of teacher-scored tokens. Thus, supervising more rollouts does not necessarily improve downstream performance.

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.

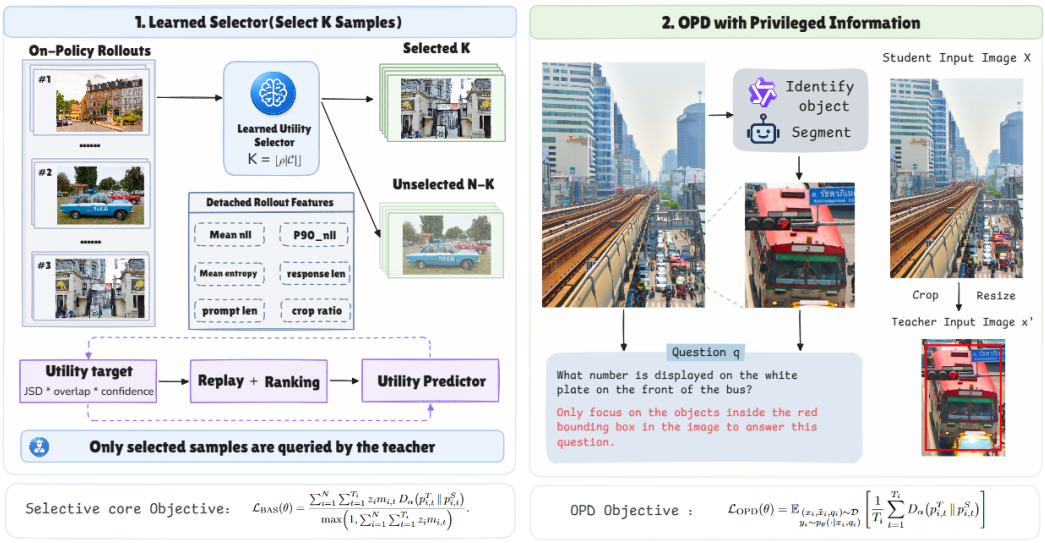



Figure 2: **Overview of BAS-OPD.** The student generates the full rollout pool, and the selector chooses $K$ rollouts for crop-teacher supervision. The selector is removed at inference.

## 1 Introduction

Multimodal large language models (MLLMs) have become general-purpose interfaces for combining visual perception with language reasoning (Liu et al., 2023; Dai et al., 2023). Their performance, however, remains brittle when an answer depends on a small object, a local attribute, or a spatial relation embedded in a high-resolution scene (Guan et al., 2024). V*Bench and HR-Bench expose this limitation by requiring models to locate and interpret fine visual evidence in large images (Wu & Xie, 2024; Wang et al., 2025). The underlying difficulty is not only local recognition. Processing the full image can obscure these details through resizing or limited spatial resolution, making the evidence needed to answer the question difficult to recover.

Existing methods improve access to local visual evidence through high-resolution processing or inference-time visual search. Model-side approaches retain more pixels through high-resolution encoders or dynamic tiling; for example, InternVL 1.5 processes an image using up to 40 tiles to support 4K inputs (Chen et al., 2024b; Li et al., 2024). Inference-time approaches instead decide where to look. V* performs language-guided visual search, while DC$^2$ recursively partitions an image, describes its patches, and retrieves relevant regional information (Wu & Xie, 2024; Wang et al., 2025). These approaches often require larger visual-token budgets or additional inference steps. A complementary approach is to use focused regional views during training, so that a model can learn from their clearer visual evidence while using only the full image at inference.

On-policy distillation supervises a student's own generated responses conditioned on the prefixes the student produces (Agarwal et al., 2024; Ko et al., 2024; Kim & Rush, 2016). In multimodal perception, Vision-OPD applies this procedure to regional supervision: a full-image student generates responses, while a crop-conditioned teacher scores the same response prefixes using clearer local evidence (Yuan et al., 2026). Existing formulations typically supervise every eligible rollout, implicitly treating teacher feedback as equally valuable across student trajectories. However, the value of teacher supervision need not be uniform: some rollouts may already exhibit close student-teacher agreement, whereas others may benefit more from privileged regional supervision. The relative value of these rollouts may also change as the student learns. This makes supervision allocation an inherently online selection problem: which student-generated rollouts should receive teacher feedback?

We introduce Budget-Aware Selective On-Policy Self-Distillation (**BAS-OPD**), which selects the student-generated rollouts that receive crop-teacher supervision. The student generates a full candidate pool, and a lightweight predictor ranks the rollouts by predicted utility, a proxy for the potential benefit of teacher supervision. We define the target utility by combining student–teacher disagree-

ment, teacher confidence, and token-support overlap. Because computing this target requires teacher scoring, the predictor estimates it beforehand from rollout statistics and crop metadata to guide selection. Only selected rollouts receive crop-teacher supervision and contribute to the distillation objective; their observed target utility values are then used to update the predictor online. The selector is used only during training; at inference, the model processes the full image in a single pass without crops or selection.

Across seven benchmarks and both Qwen3.5 scales, BAS-OPD gives the strongest aggregate result among the evaluated selection policies and full querying (Table 3). With the 4B backbone, it reaches 81.60%, 1.36 points above Entropy-25 and 1.84 points above full querying. Scaling to 9B raises the average to 84.23%, ahead of P90-NLL by 1.32 points and full querying by 2.04 points. This performance is obtained with 25% of full-query teacher calls and 15.78%/16.02% of their respective full-pool scored-token totals. The comparison suggests that where teacher supervision is allocated matters alongside how much supervision is provided. Our contributions are:

- **On-Policy Rollout Selection.** We formulate privileged on-policy distillation as an online rollout-selection problem: deciding which student-generated rollouts are most worth supervising.
- **Learned Supervision-Value Estimation.** We define utility as a proxy for teacher supervision value and use a lightweight predictor to estimate it from detached student-side rollout statistics before teacher scoring, enabling top-$K$ selection.
- **Selection Effectiveness and Supervision Efficiency.** Learned selection outperforms random and student-entropy selection and achieves higher average accuracy than full querying while reducing teacher calls and scored tokens. Utility concentration, ranking, and gradient analyses further characterize how selection concentrates supervision on high-value rollouts.

## 2 METHOD

### 2.1 ON-POLICY ROLLOUT SELECTION

Figure 2 summarizes the training and inference paths. Algorithm 1 in Appendix A.2 details the learned-selection training loop. BAS-OPD augments regional-to-global OPD with an online selection mechanism over student-generated rollouts. Consider a training batch $\mathcal{B} = \{(x_i, \tilde{x}_i, q_i)\}_{i=1}^{N}$, where $x_i$ is the full image, $\tilde{x}_i$ is its privileged evidence-centered crop, and $q_i$ is the language prompt. The full-image policy first samples an on-policy response $y_i = (y_{i,1}, \ldots, y_{i,T_i})$ for every example in the batch. For response position $t$, we denote the student distribution conditioned on the full image by $p^S_{i,t}$ and the crop-teacher distribution over the same response prefix by $p^T_{i,t}$:

$$p^S_{i,t} = p_\theta(\cdot \mid x_i, q_i, y_{i,<t}), \tag{1}$$

$$p^T_{i,t} = p_{\bar{\theta}}(\cdot \mid \tilde{x}_i, q_i, y_{i,<t}). \tag{2}$$

Here, $\theta$ denotes the trainable full-image policy and $\bar{\theta}$ denotes the crop-conditioned teacher maintained by the underlying OPD procedure. Standard OPD minimizes the expected token-level divergence along responses sampled from the student:

$$\mathcal{L}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{\substack{(x_i,\tilde{x}_i,q_i)\sim\mathcal{D} \\ y_i\sim p_\theta(\cdot|x_i,q_i)}} \left[ \frac{1}{T_i} \sum_{t=1}^{T_i} D_\alpha\big(p^T_{i,t} \,\|\, p^S_{i,t}\big) \right], \tag{3}$$

where $\mathcal{D}$ is the training distribution and $D_\alpha$ is the distillation divergence; our experiments use Jensen–Shannon divergence with $\alpha = 0.5$. The teacher scores the same response prefixes generated by the student. BAS-OPD does not change how responses are generated or how the teacher is regularized; it changes which rollouts are allowed to enter the crop-teacher forward pass.

Let $\mathcal{C} \subseteq \{1, \ldots, N\}$ be the samples for which privileged crops are available. Given a query ratio $\rho \in (0, 1]$, we impose the sample budget

$$K = \lfloor \rho |\mathcal{C}| \rfloor, \qquad \mathcal{S} \subseteq \mathcal{C}, \quad |\mathcal{S}| \le K, \tag{4}$$

where $\mathcal{S}$ is chosen before any crop-teacher computation. The full-image policy therefore produces rollouts for the complete batch, while the crop teacher scores only the selected subset. This ordering

is important: the budget reduces the actual teacher input batch rather than masking losses after an already completed teacher forward pass. For $i \notin \mathcal{S}$, no crop-teacher distribution or OPD target is constructed.

Selection aims to allocate teacher supervision to rollouts with higher estimated supervision value under the given budget. Because utility is observed only after teacher scoring, we predict it from statistics already produced by student rollouts (Algorithm 1, lines 3–9).

### 2.2 Learned Utility Selection

We use utility as a proxy for rollout-level teacher supervision value. Let $P^S_{i,t}$ and $P^T_{i,t}$ denote student and teacher probabilities evaluated on the student's top-$k$ token indices and renormalized over this shared support. We measure their disagreement with the Jensen–Shannon divergence

$$d_{i,t} = \mathrm{JSD}\big(P^S_{i,t}, P^T_{i,t}\big)\,. \tag{5}$$

Divergence alone can be uninformative when the teacher is diffuse or favors different tokens. We define $o_{i,t}$ as the fraction of student top-$k$ indices also in the teacher's own top-$k$ set, and compute the normalized teacher confidence

$$c_{i,t} = 1 - \frac{H(P^T_{i,t})}{\log k}, \tag{6}$$

We then define the detached per-sample utility target as

$$u_i = \frac{1}{\sum_t m_{i,t}} \sum_t m_{i,t}\, d_{i,t}\, o_{i,t}\, c_{i,t}, \tag{7}$$

where $m_{i,t}$ masks padding and invalid response positions. This target is high when student–teacher disagreement is large, the teacher is confident on the shared support, and their top-$k$ token sets overlap.

Computing this target utility requires teacher scoring, but rollouts must be selected beforehand. BAS-OPD therefore uses a lightweight predictor to estimate utility from features available before teacher scoring. For each rollout, Algorithm 1 (line 4) constructs a detached feature vector

$$\phi_i = \big[\overline{\ell}_i, Q_{0.9}(\ell_i), \overline{H}_i, T_i, L_i, a_i\big]^\top, \tag{8}$$

which summarizes rollout likelihood and entropy, response and prompt lengths, and crop area. These quantities are available before teacher scoring and require no additional MLLM forward pass. BAS-OPD predicts utility from normalized pre-query features:

$$\hat{u}_i = f_\psi(\mathrm{Normalize}(\phi_i)). \tag{9}$$

Feature definitions, running normalization, and network architecture are given in Appendix B.2. Algorithm 1 (lines 5–9) selects $K$ candidates uniformly before the first predictor update. Thereafter, it reserves $K_\mathrm{e} = \lceil \epsilon K \rceil$ queries for random exploration and assigns the remaining $K - K_\mathrm{e}$ queries to the highest-scoring candidates. Exploration samples from the remaining candidates to avoid a self-confirming selection loop while preserving the budget.

After teacher scoring, target utility values are computed for selected rollouts (Algorithm 1, line 12). Only queried samples yield $(\phi_i, u_i)$ pairs; unselected samples are never assigned synthetic utility labels. Observed feature–utility pairs train the predictor through replay using weighted Huber regression and pairwise ranking:

$$\mathcal{L}_\mathrm{sel} = \mathcal{L}_\mathrm{reg} + \lambda_\mathrm{rank}\mathcal{L}_\mathrm{rank}, \tag{10}$$

Regression calibrates utility values, while ranking learns their ordering. Algorithm 1 (line 14) updates only the selector from detached features and targets; the replay procedure and loss definitions are given in Appendix A.3.

### 2.3 Selective Distillation and Inference

**Selective distillation.** Only selected rollouts receive crop-teacher supervision and contribute to the distillation objective. Let $z_i = \mathbf{1}[i \in \mathcal{S}]$; we optimize the selective OPD objective in Eq. 11.

$$\mathcal{L}_\mathrm{BAS}(\theta) = \frac{\sum_{i=1}^{N}\sum_{t=1}^{T_i} z_i m_{i,t}\, D_\alpha\big(p^T_{i,t} \,\|\, p^S_{i,t}\big)}{\max\Big(1, \sum_{i=1}^{N}\sum_{t=1}^{T_i} z_i m_{i,t}\Big)}. \tag{11}$$

When $\rho = 1$, all eligible rollouts are selected and the objective recovers full-query OPD. At inference, all selection components are removed and the model retains single-pass full-image inference.

# 3 EXPERIMENTS

## 3.1 EXPERIMENTAL SETUP

**MLLM and selector training.** We evaluate 4B and 9B models; policy and cost comparisons cover both, while other analyses use 4B. Training uses 3,120 preprocessed full-image–crop pairs from the released training set (Yuan et al., 2026), JSD with $\alpha = 0.5$, top-100 distillation, crop-teacher EMA 0.05, and 8,192/1,024-token prompt/response limits. The MLLM uses AdamW (lr $2 \times 10^{-6}$; weight decay 0.01), and the selector uses Adam (lr $1 \times 10^{-3}$) in Learned-10/25/50. Each run uses 28 prompts per step, 8 rollouts per prompt, one epoch, and four NVIDIA A100-SXM4 80 GB GPUs. All runs use seed 42. Appendices B.2–B.3 and Table 5 report selector settings, paired inputs, and run lengths. Unless otherwise stated, all selection policies generate the same full student rollout pool; selection only determines which rollouts receive crop-teacher supervision.

**Evaluation benchmarks.** We evaluate on seven benchmarks spanning high-resolution perception, general visual understanding, and object hallucination. V*Bench (Wu & Xie, 2024), ZoomBench (Wei et al., 2026), and HR-Bench 4K (Wang et al., 2025) assess fine-grained perception; MMVP (Tong et al., 2024b), CV-Bench (Tong et al., 2024a), and MMStar (Chen et al., 2024a) measure broader visual capabilities. POPE (Li et al., 2023) evaluates object hallucination across adversarial, popular, and random splits.

Table 1: **Comparison with state-of-the-art MLLMs across seven benchmarks.** Among open-source single-forward-pass results, the best result is bold and the second-best is underlined.

| Model | Param. Size | V*Bench | ZoomBench | HR-Bench 4K | MMVP | CV-Bench | MMStar | POPE |
|---|---|---|---|---|---|---|---|---|
| ***"Thinking-with-Images" Agentic Models*** | | | | | | | | |
| DeepEyes (Zheng et al., 2025) | 7B | 79.58 | 46.04 | 74.25 | 72.00 | 76.81 | 61.33 | 88.42 |
| Thyme (Zhang et al., 2025) | 7B | 81.68 | 46.39 | 76.75 | 70.33 | 75.81 | 60.53 | 88.58 |
| DeepEyesV2 (Hong et al., 2025) | 7B | 80.63 | 46.27 | 75.00 | 72.33 | 80.48 | 61.07 | 89.08 |
| ***Closed-Source Models (Single Forward Pass)*** | | | | | | | | |
| GPT-5.2 (OpenAI, 2025) | – | 68.59 | 47.81 | 81.12 | 79.33 | 85.99 | 74.40 | 87.51 |
| GPT-5.4 (OpenAI, 2026) | – | 72.77 | 50.18 | 82.25 | 77.67 | 86.08 | 75.13 | 87.58 |
| Gemini-3.1-Pro (Google DeepMind, 2026a) | – | 88.48 | 60.83 | 89.62 | 91.00 | 89.98 | 83.67 | 89.72 |
| Gemini-3.5-Flash (Google DeepMind, 2026b) | – | 90.05 | 60.95 | 89.50 | 94.33 | 88.83 | 84.20 | 89.88 |
| ***Open-Source Models (Single Forward Pass)*** | | | | | | | | |
| Qwen3.5 (Qwen Team, 2026) | 4B | 84.82 | 51.72 | 84.38 | 76.67 | 87.13 | 78.53 | 88.28 |
| Qwen3.5 (Qwen Team, 2026) | 9B | 86.91 | 54.56 | 85.75 | <u>83.33</u> | 88.29 | <u>80.93</u> | 88.88 |
| Qwen3-VL-Instruct (Bai et al., 2025) | 8B | 84.29 | 42.96 | 78.00 | 74.67 | 86.22 | 68.07 | 88.12 |
| MiMo-VL-RL (Team et al., 2025) | 7B | 81.68 | 43.20 | 73.25 | 75.67 | 80.45 | 69.87 | 88.36 |
| ZwZ (Wei et al., 2026) | 8B | 87.43 | 57.16 | 84.50 | 80.00 | 87.77 | 71.47 | 88.97 |
| MiniCPM-V-4.5 (Yu et al., 2025b) | 9B | 69.63 | 42.84 | 68.75 | 78.00 | 79.88 | 67.53 | 87.51 |
| Qwen3-VL-Instruct (Bai et al., 2025) | 235B | <u>91.62</u> | 56.69 | 85.75 | 81.67 | 86.90 | 71.87 | 89.26 |
| Qwen3.5 (Qwen Team, 2026) | 397B | 86.39 | 56.92 | **89.88** | **83.67** | <u>88.92</u> | 72.60 | <u>89.63</u> |
| **BAS-OPD (Ours)** | 4B | 90.58 | <u>59.88</u> | 85.62 | 79.33 | 87.63 | 79.27 | 88.91 |
| **BAS-OPD (Ours)** | 9B | **92.67** | **63.08** | <u>87.25</u> | <u>83.33</u> | **90.23** | **83.07** | **90.00** |

**Evaluation protocol.** Evaluation uses only the original full image, with privileged crops and all selection modules disabled. We use deterministic decoding with temperature 0 and seed 42. Benchmark-specific parsers and exact-answer equivalence matching score responses when possible; only unresolved responses are judged by GPT-5.5 at temperature 0. Appendix B.4 provides the complete prompt and configuration. We report accuracy or the official aggregate score for each benchmark.

### 3.2 Experimental Results

**Comparison with state-of-the-art MLLMs.** Table 1 compares BAS-OPD with agentic, closed-source, and open-source MLLMs. BAS-OPD 9B averaged 84.23%, exceeding Qwen3.5-9B, the strongest non-BAS open-source single-forward-pass model by average score, by 3.00 points. It ranked first on five open-source benchmarks and second on two, and outperformed Qwen3-VL-Instruct 235B on all seven. BAS-OPD 4B averaged 81.60%, exceeding Qwen3.5-9B and Qwen3.5-397B by 0.37 and 0.46 points, respectively. Against closed-source models, BAS-OPD 9B led on four benchmarks.

Table 2: **Accuracy (%) of vanilla inference, SFT, GRPO, DAPO, OPSD, and BAS-OPD on matched Qwen3.5 backbones.** Best results within each backbone are bold.

| | Fine-Grained Visual Tasks | | | Holdout Tasks | | | |
|---|---|---|---|---|---|---|---|
| **Method** | **V*Bench** | **ZoomBench** | **HR-Bench 4K** | **MMVP** | **CV-Bench** | **MMStar** | **POPE** |
| *Qwen3.5-4B Backbone* | | | | | | | |
| Vanilla (Qwen Team, 2026) | 84.82 | 51.72 | 84.38 | 76.67 | 87.13 | 78.53 | 88.28 |
| SFT (Ouyang et al., 2022) | 78.53 | 54.44 | 80.00 | 78.00 | 85.78 | 68.27 | 87.44 |
| GRPO (Shao et al., 2024) | 83.25 | 55.62 | 82.50 | **79.33** | 87.16 | 70.67 | 86.39 |
| DAPO (Yu et al., 2025a) | 85.86 | 55.38 | 84.38 | **79.33** | 87.07 | 72.53 | 86.59 |
| OPSD (Zhao et al., 2026) | 83.77 | 54.32 | 82.00 | 79.00 | 87.19 | 74.87 | 88.88 |
| **BAS-OPD (Ours)** | **90.58** | **59.88** | **85.62** | **79.33** | **87.63** | **79.27** | **88.91** |
| *Qwen3.5-9B Backbone* | | | | | | | |
| Vanilla (Qwen Team, 2026) | 86.91 | 54.56 | 85.75 | **83.33** | 88.29 | 80.93 | 88.88 |
| SFT (Ouyang et al., 2022) | 81.68 | 58.82 | 83.13 | 81.33 | 88.05 | 73.53 | 87.50 |
| GRPO (Shao et al., 2024) | 85.86 | 57.40 | 87.00 | 80.67 | 87.70 | 73.33 | 87.72 |
| DAPO (Yu et al., 2025a) | 87.43 | 56.09 | 86.00 | 79.67 | 87.39 | 75.80 | 87.51 |
| OPSD (Zhao et al., 2026) | 90.58 | 57.75 | 84.50 | 80.33 | 87.37 | 78.93 | 87.50 |
| **BAS-OPD (Ours)** | **92.67** | **63.08** | **87.25** | **83.33** | **90.23** | **83.07** | **90.00** |

Table 3: **Selection-policy accuracy and teacher cost.** Budgeted policies use 25%; $28 \times 2$ matches Learned's call count, and token ratios use each run's full pool.

| | Teacher Cost | | Fine-Grained Visual Tasks | | | Holdout Tasks | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Policy** | **Query Ratio** | **Token Ratio** | **V*Bench** | **Zoom Bench** | **HR-Bench 4K** | **MMVP** | **CV-Bench** | **MMStar** | **POPE** | **Avg.** |
| *Qwen3.5-4B Backbone* | | | | | | | | | | |
| Full | 100.00 | 100.00 | 87.43 | 58.46 | 82.25 | 78.33 | 85.56 | 77.73 | 88.58 | 79.76 |
| Full ($28 \times 2$) | 100.00 | 100.00 | 86.91 | 58.46 | 78.25 | 77.67 | 85.38 | 77.20 | 88.97 | 78.98 |
| Random | 25.00 | 26.28 | 86.39 | 56.57 | 82.88 | 78.33 | 85.77 | 77.00 | 87.83 | 79.25 |
| Entropy | 25.00 | 27.65 | 88.48 | 58.58 | 83.62 | 77.33 | 86.41 | 77.87 | 89.40 | 80.24 |
| Mean-NLL | 25.00 | 29.01 | 87.96 | 59.64 | 80.12 | 76.33 | 86.57 | 78.47 | 89.24 | 79.76 |
| P90-NLL | 25.00 | 41.18 | 89.01 | 59.17 | 80.75 | 78.67 | 86.15 | 77.60 | **89.47** | 80.12 |
| Linear | 25.00 | **13.82** | **90.58** | 58.11 | 81.38 | 77.00 | 85.68 | 76.60 | 89.09 | 79.78 |
| **Learned** | 25.00 | 15.78 | **90.58** | **59.88** | **85.62** | **79.33** | **87.63** | **79.27** | 88.91 | **81.60** |
| *Qwen3.5-9B Backbone* | | | | | | | | | | |
| Full | 100.00 | 100.00 | 89.53 | 60.71 | 84.50 | 81.67 | 88.62 | 81.00 | 89.31 | 82.19 |
| Full ($28 \times 2$) | 100.00 | 100.00 | 89.01 | 60.24 | 81.88 | 81.00 | 88.30 | 80.53 | 89.54 | 81.50 |
| Random | 25.00 | 26.11 | 88.48 | 59.88 | 84.88 | 81.67 | 88.51 | 80.80 | 89.07 | 81.90 |
| Entropy | 25.00 | 27.92 | 91.10 | 61.42 | 85.62 | 81.33 | 89.20 | 81.47 | 89.67 | 82.83 |
| Mean-NLL | 25.00 | 25.73 | 90.58 | 62.01 | 83.38 | 80.67 | 89.42 | 82.27 | 89.58 | 82.56 |
| P90-NLL | 25.00 | 30.28 | 91.62 | 61.66 | 84.00 | 82.67 | 89.08 | 81.53 | 89.82 | 82.91 |
| Linear | 25.00 | **13.57** | 92.15 | 61.07 | 84.75 | 81.67 | 88.84 | 80.93 | 89.73 | 82.73 |
| **Learned** | 25.00 | 16.02 | **92.67** | **63.08** | **87.25** | **83.33** | **90.23** | **83.07** | **90.00** | **84.23** |

**Comparison with alternative training objectives.** Table 2 compares BAS-OPD with vanilla inference, SFT, GRPO, DAPO, and OPSD on matched Qwen3.5 backbones. BAS-OPD was best or tied-best on all seven tasks at both scales. Its largest gains over the strongest alternatives were 4.72 points on V*Bench and 4.26 on ZoomBench at 4B, and 4.26 on ZoomBench and 2.14 on MMStar at 9B; MMVP was tied at both scales. Relative to OPSD, BAS-OPD improved every benchmark and raised the seven-task average by 3.03 and 3.24 points at 4B and 9B, respectively, under a 25% teacher-query budget. Matched training settings are detailed in Appendix B.1.

### 3.3 Selection and Ablation Analysis

**Selection effectiveness under a fixed query ratio.** Table 3 compares accuracy and teacher cost, while Figure 7 visualizes both model scales. At the shared 25% query ratio on 4B, Random-25, Entropy-25, Mean-NLL, P90-NLL, Linear-25, and Learned-25 averaged 79.25%, 80.24%, 79.76%, 80.12%, 79.78%, and 81.60%, respectively (definitions in Appendix A.1). Learned-25 outperformed every alternative selector by 1.36–2.35 points and full querying by 1.84 points. At 9B, it averaged 84.23%, exceeding the strongest alternative selector, P90-NLL, by 1.32 points and full querying by 2.04 points. Learned-25 led full querying on all seven benchmarks at both scales. The 4B and 9B runs used 15.78% and 16.02% of full-pool scored tokens; Appendix C reports detailed costs for both scales.

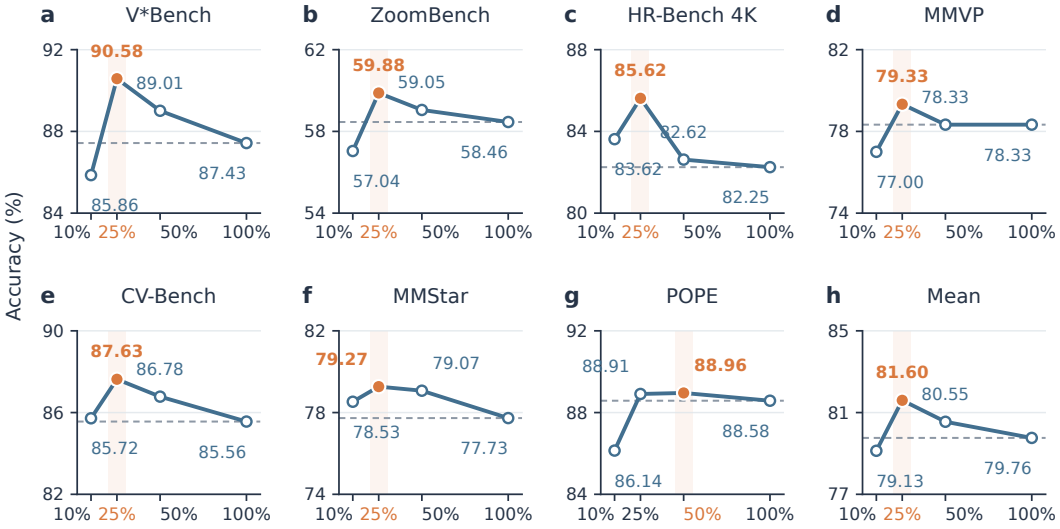



Figure 3: **Qwen3.5-4B accuracy across query budgets.** Panel (h) is the seven-benchmark score. Orange marks maxima; dashed lines mark full querying.

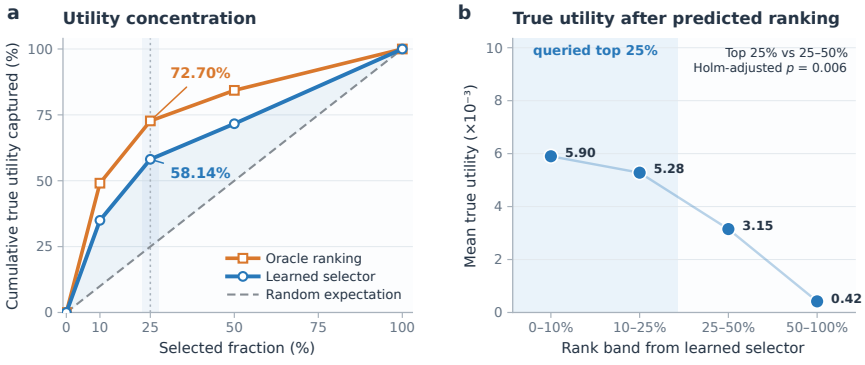



Figure 4: **Qwen3.5-4B BAS-OPD utility diagnostics over 2,240 candidates.** (a) Cumulative utility capture. (b) Mean utility by predicted rank band; $p$ is Holm-adjusted over three one-sided paired sign-flip tests.

**Effect of query budget.** Figure 3 compares four budgets on seven benchmarks. The 25% budget leads six sweeps, while 50% leads on POPE by 0.05 points. At 10%, 25%, 50%, and 100%, the seven-benchmark scores are 79.13%, 81.60%, 80.55%, and 79.76%, confirming non-monotonic returns.

**Utility concentration and ranking behavior.** Figure 4 audits 2,240 rollouts across ten batches. At 25%, learned ranking captures 58.14% of utility, versus 72.70% for the oracle and 25% for random ranking (Figure 4a). Utility decreases across rank bands; the top-quartile mean is 1.75 times the next quartile ($5.52 \times 10^{-3}$ versus $3.15 \times 10^{-3}$; Holm-adjusted $p = 0.006$; Figure 4b). The audit freezes the trained selector, ranks candidates before teacher scoring, and then scores all candidates to test whether the queried subset is enriched for observed utility. Appendix D provides representative cases.

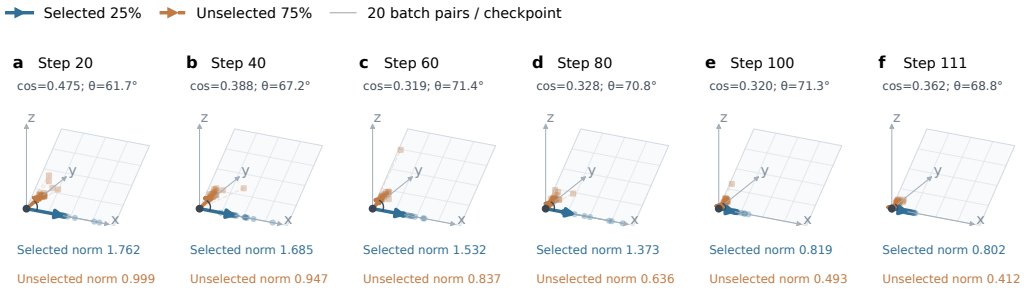



Figure 5: **Qwen3.5-4B BAS-OPD gradient-pair geometry for Learned-25.** Six checkpoints are shown. Arrow lengths encode the selected and unselected mean gradient norms, and the included angle is the arccosine of their mean cosine similarity; faint vectors show 20 batch pairs per checkpoint. The display coordinates are not a projection of the high-dimensional model gradients.

**Gradient geometry.** Across six checkpoints, the selected 25% had 1.66–2.16× larger mean OPD gradient norms than the unselected 75% (Figure 5). Cosine similarities of 0.319–0.475 correspond to angles of 61.7°–71.4°. Thus, the selected and unselected groups induce substantially different training directions, separated by at least 61.7°, although these one-trajectory measurements (20 batch pairs/checkpoint) do not imply that unselected rollouts are harmful.

Table 4: **Ablations of learned-selector design and input features.** Both blocks share the Full selector control; Avg. is the unweighted seven-benchmark mean.

| | Fine-Grained Visual Tasks | | | Holdout Tasks | | | | |
|---|---|---|---|---|---|---|---|---|
| **Selector variant** | **V*Bench** | **Zoom Bench** | **HR-Bench 4K** | **MMVP** | **CV-Bench** | **MMStar** | **POPE** | **Avg.** |
| *Utility-target and selector-design ablations* | | | | | | | | |
| No overlap ($d\,c$) | 86.39 | 57.04 | 82.25 | 76.33 | 86.09 | 70.80 | 88.97 | 78.27 |
| No confidence ($d\,o$) | 88.48 | 55.50 | 81.00 | 77.33 | 84.99 | 69.40 | 88.87 | 77.94 |
| No ranking loss | 89.53 | 56.69 | 79.38 | 76.33 | 85.46 | 70.53 | **89.11** | 78.15 |
| No exploration | 85.86 | 58.70 | 83.38 | 77.00 | 86.23 | 71.47 | 89.03 | 78.81 |
| *Leave-one-feature-out ablations* | | | | | | | | |
| Without mean NLL | 87.96 | 57.40 | 78.12 | 78.67 | 85.87 | 76.27 | 88.96 | 79.04 |
| Without P90 NLL | 85.86 | 57.28 | 77.25 | 78.33 | 85.93 | 75.93 | 89.12 | 78.53 |
| Without token entropy | 83.77 | 58.34 | 81.50 | 78.00 | 83.99 | 77.07 | 88.83 | 78.79 |
| Without response length | 85.34 | 56.57 | 80.88 | 77.33 | 85.78 | 76.27 | 88.97 | 78.73 |
| Without prompt length | 86.39 | 56.57 | 79.75 | 77.67 | 85.24 | 75.40 | 89.02 | 78.58 |
| Without crop-area ratio | 87.96 | 59.29 | 81.00 | 78.00 | 86.24 | 78.53 | 89.07 | 80.01 |
| **Full selector** | **90.58** | **59.88** | **85.62** | **79.33** | **87.63** | **79.27** | 88.91 | **81.60** |

**Utility-target and selector-design ablations.** Table 4 separates design-component and input-feature ablations. Its upper block isolates top-$k$ overlap, teacher confidence, ranking supervision, and random exploration. The full selector achieved the highest seven-benchmark average of 81.60%.

Removing individual components lowered average accuracy by 2.79–3.66 points, with teacher confidence causing the largest drop, consistent with its role in downweighting diffuse teacher distributions. On MMStar, the ablations caused larger drops of 7.80–9.87 points, showing that the gains extended to holdout evaluation, although their magnitude varied by benchmark.

**Selector-feature ablations.** The lower block of Table 4 removes one of six input features while keeping the rollout pool, teacher budget, exploration, utility target, replay, and OPD objective fixed. Every removal reduced the seven-benchmark average from 81.60% to 78.53–80.01%. Removing P90 NLL caused the largest drop (3.07 points), whereas removing crop-area ratio caused the smallest (1.59 points).

# 4 Related Work

## 4.1 On-Policy Distillation

Conventional distillation supervises a student on a fixed data distribution, whereas on-policy distillation evaluates the teacher along student-generated trajectories to reduce the train–inference mismatch. GKD trains on student-generated sequences, while MiniLLM and DistiLLM combine student rollouts with alternative divergence objectives or adaptive on- and off-policy mixtures (Agarwal et al., 2024; Gu et al., 2024; Ko et al., 2024). Recent self-distillation methods condition the same model on privileged reasoning traces or demonstrations, extending on-policy learning to mathematical reasoning and continual learning (Zhao et al., 2026; Shenfeld et al., 2026). Complementary studies identify compatibility, novel teacher capability, and dense-supervision cost as central OPD factors (Li et al., 2026), and expose sampled-token failures from imbalanced supervision and unreliable teacher guidance (Fu et al., 2026a). Recent evidence further shows that OPD can recover much of its full-data gain from small query sets by inducing broad state coverage through student rollouts (Fu et al., 2026b). Privileged regional supervision has also been explored for multimodal perception (Yuan et al., 2026). BAS-OPD instead addresses online rollout selection: it determines which student-generated trajectories should receive teacher supervision before the teacher forward pass, subject to a fixed supervision budget.

## 4.2 Fine-Grained Visual Understanding for MLLMs

Fine-grained MLLM perception is limited when small but decisive evidence is lost during image encoding. High-resolution architectures retain more detail through dynamic tiling, variable visual-token counts, or adaptive image slicing, as in InternVL 1.5, Qwen2-VL, and LLaVA-UHD (Chen et al., 2024b; Wang et al., 2024; Guo et al., 2024). Inference-time methods such as V* and $DC^2$ instead search, partition, or retrieve relevant regions for each query (Wu & Xie, 2024; Wang et al., 2025). These approaches improve access to local evidence, but increase visual processing or introduce additional inference steps. BAS-OPD is complementary: it changes neither the visual architecture nor the inference procedure, but allocates privileged regional supervision selectively during training and retains single-pass full-image inference.

# 5 Conclusion and Limitations

We introduced BAS-OPD, which formulates privileged on-policy distillation as an online rollout-selection problem and uses estimated supervision value to determine which student-generated rollouts receive crop-teacher supervision. Learned-25 with Qwen3.5-4B improved the full-query result from 79.76% to 81.60% while using 25% of teacher calls and 15.78% of full-query teacher-scored tokens. With Qwen3.5-9B, it reached 84.23% using 25% of calls and 16.02% of scored tokens. These results show that selectively allocating teacher supervision to high-value rollouts can improve supervision effectiveness while reducing teacher-scoring cost.

**Limitations.**

All runs use seed 42. Policy and cost comparisons cover 4B/9B; other analyses are 4B-only. End-to-end speedups depend on implementation because student rollout generation and optimization remain unchanged. Validation across seeds and model families remains future work.

## AI USE STATEMENT

In accordance with conference policies, we disclose that OpenAI's GPT models were used in two limited roles. First, GPT-5.5 served as an automatic judge at temperature 0 for benchmark responses that could not be scored through benchmark-native option parsing or exact-answer matching. Second, GPT-based tools assisted with grammatical refinement, language polishing, and preparation of the LaTeX manuscript. GPT models were not used as the student, crop teacher, or learned selector, and did not participate in training-time supervision allocation or model inference. The BAS-OPD formulation, experimental design, result interpretation, and core intellectual contributions originated from the human authors.

## ETHICS STATEMENT

All scientific ideas, methodological innovations, experimental designs, analyses, and interpretations originated from the human authors and were verified by them. The authors take full responsibility for the manuscript.

## REPRODUCIBILITY STATEMENT

The appendix provides implementation settings, selector and training configurations, evaluation details, and teacher-cost accounting. Tables 6 and 5 specify the selector and policy training configurations. All reported training results use seed 42; the diagnostic audit is described in the main text.

## REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes, 2024. URL https://arxiv.org/abs/2306.13649.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv (Cornell University)*, 2025.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In *Advances in Neural Information Processing Systems*, volume 37, 2024a.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, Ji Ma, Jiaqi Wang, Xiaoyi Dong, Hang Yan, Hewei Guo, Conghui He, Botian Shi, Zhenjiang Jin, Chao Xu, Bin Wang, Xingjian Wei, Wei Li, Wenjian Zhang, Bo Zhang, Pinlong Cai, Licheng Wen, Xiangchao Yan, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. How far are we to GPT-4V? closing the gap to commercial multimodal models with open-source suites, 2024b. URL https://arxiv.org/abs/2404.16821.

Wenliang Dai, Junnan Li, DONGXU LI, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 49250–49267. Curran Associates, Inc., 2023. doi: 10.52202/075280-

2142. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9a6a435e75419a836fe47ab6793623e6-Paper-Conference.pdf.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes, 2026a. URL https://arxiv.org/abs/2603.25562.

Zixuan Fu, Bingxiang He, Yuxin Zuo, Haohuan Huang, Jinqian Zhang, Ruhang Xiao, Cheng Qian, Qinyu Luo, Huan-ang Gao, Yudong Wang, Zhiyuan Liu, Ning Ding, and Chaojun Xiao. Rethinking on-policy distillation of large language models II: One training example. *arXiv preprint arXiv:2609.04172*, 2026b.

Google DeepMind. Gemini 3.1 Pro model card. Technical report, Google DeepMind, February 2026a. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-1-Pro-Model-Card.pdf.

Google DeepMind. Gemini 3.5 Flash model card. Technical report, Google DeepMind, May 2026b. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-5-Flash-Model-Card.pdf.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. HallusionBench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14375–14385, 2024.

Zonghao Guo, Ruyi Xu, Yuan Yao, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, and Gao Huang. LLaVA-UHD: An LMM perceiving any aspect ratio and high-resolution images. In *European Conference on Computer Vision*, 2024.

Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, Guohai Xu, and Xing Yu. DeepEyesV2: Toward agentic multimodal model. *arXiv preprint arXiv:2511.05271*, 2025.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, Austin, Texas, 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 24872–24895, 2024.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 292–305, 2023.

Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26763–26773, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916, 2023. doi: 10.52202/075280-1516.

OpenAI. Update to GPT-5 system card: GPT-5.2. Technical report, OpenAI, December 2025. URL https://cdn.openai.com/pdf/3a4153c8-c748-4b71-8e31-aecbde944f8d/oai_5_2_system-card.pdf.

OpenAI. GPT-5.4 Thinking system card. Technical report, OpenAI, March 2026. URL https://deploymentsafety.openai.com/gpt-5-4-thinking.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. doi: 10.52202/068431-2011. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

Core Team, Zihao Yue, Zhenru Lin, Yifan Song, Weikun Wang, Shuhuai Ren, Shuhao Gu, Shicheng Li, Peidian Li, Liang Zhao, Lei Li, Kainan Bao, Hao Tian, Hailin Zhang, Gang Wang, Dawei Zhu, Cici, Chenhong He, Bowen Ye, Bowen Shen, Zihan Zhang, Zihan Jiang, Zhixian Zheng, Zhichao Song, Zhenbo Luo, Yue Yu, Yudong Wang, Yuanyuan Tian, Yu Tu, Yihan Yan, Yi Huang, Xu Wang, Xinzhe Xu, Xingchen Song, Xing Zhang, Xing Yong, Xin Zhang, Xiangwei Deng, Wenyu Yang, Wenhan Ma, Weiwei Lv, Weiji Zhuang, Wei Liu, Sirui Deng, Shuo Liu, Shimao Chen, Shihua Yu, Shaohui Liu, Shande Wang, Rui Ma, Qiantong Wang, Peng Wang, Nuo Chen, Menghang Zhu, Kangyang Zhou, Kang Zhou, Kai Fang, Jun Shi, Jinhao Dong, Jiebao Xiao, Jiaming Xu, Huaqiu Liu, Hongshen Xu, Heng Qu, Haochen Zhao, Hanglong Lv, Guoan Wang, Duo Zhang, Dong Zhang, Di Zhang, Chong Ma, Chang Liu, Can Cai, and Bingquan Xia. MiMo-VL technical report, 2025. URL https://arxiv.org/abs/2506.03569.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 87310–87356. Curran Associates, Inc., 2024a. doi: 10.52202/079017-2771. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/9ee3a664ccfeabc0da16ac6f1f1cfe59-Paper-Conference.pdf.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9568–9578, 2024b.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7907–7915, 2025.

Lai Wei, Liangbo He, Jun Lan, Lingzhong Dong, Yutong Cai, Siyuan Li, Huijia Zhu, Weiqiang Wang, Linghe Kong, Yue Wang, Zhuosheng Zhang, and Weiran Huang. Zooming without zooming: Region-to-image distillation for fine-grained multimodal perception. *arXiv preprint arXiv:2602.11858*, 2026.

Penghao Wu and Saining Xie. $V^*$: Guided visual search as a core mechanism in multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13084–13094, 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Weinan Dai, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025a.

Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Yuanqian Zhao, Bokai Xu, Junbo Cui, Yingjing Xu, Liqing Ruan, Luoyuan Zhang, Hanyu Liu, Jingkun Tang, Hongyuan Liu, Qining Guo, Wenhao Hu, Bingxiang He, Jie Zhou, Jie Cai, Ji Qi, Zonghao Guo, Chi Chen, Guoyang Zeng, Yuxuan Li, Ganqu Cui, Ning Ding, Xu Han, Yuan Yao, Zhiyuan Liu, and Maosong Sun. MiniCPM-V 4.5: Cooking efficient MLLMs via architecture, data, and training recipe, 2025b. URL https://arxiv.org/abs/2509.18154.

Qianhao Yuan, Jie Lou, Xing Yu, Hongyu Lin, Le Sun, Xianpei Han, and Yaojie Lu. Vision-OPD: Learning to see fine details for multimodal LLMs via on-policy self-distillation. *arXiv preprint arXiv:2605.18740*, 2026.

Yi-Fan Zhang, Xingyu Lu, Shukang Yin, Chaoyou Fu, Wei Chen, Xiao Hu, Bin Wen, Kaiyu Jiang, Changyi Liu, Tianke Zhang, Haonan Fan, Kaibing Chen, Jiankang Chen, Haojie Ding, Kaiyu Tang, Zhang Zhang, Liang Wang, Fan Yang, Tingting Gao, and Guorui Zhou. Thyme: Think beyond images, 2025. URL https://arxiv.org/abs/2508.11630.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. DeepEyes: Incentivizing "thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025.

# A Implementation Details

## A.1 Selection baselines and matched controls

Random and student-entropy selection use the same query-budget interface as learned selection, but neither trains a utility predictor.

**Random-25** samples $K = \lfloor 0.25|\mathcal{C}| \rfloor$ eligible rollouts uniformly without replacement at each training step. Sampling uses a CPU random permutation with a seed derived from the selector seed and training step. Only these selected rollouts are sent to the crop teacher. The policy has no trainable selector and requires no additional student forward pass.

**Entropy-25** is a fixed student-uncertainty heuristic. In all reported Entropy-25 experiments, token entropy is available directly from the rollout path, and each candidate is scored by its mean entropy over valid response tokens. The top-$K$ rows are queried. This policy adds no trainable parameters and requires no additional student forward pass.

**Full querying** ($28\times2$) uses 28 prompts and two rollouts per prompt, yielding 56 candidates per step, and sends all 56 to the crop teacher. This matches the 56 teacher calls made by Learned-25 from its full pool of $28\times8 = 224$ candidates while retaining a 100% query ratio within the smaller pool. It uses the same 111 training steps and has no selector, predictor update, or random exploration.

**Mean-NLL-25** and **P90-NLL-25** retain the full 224-candidate student rollout pool and the 25% teacher budget. They rank candidates by the response-level mean or 90th percentile, respectively, of sampled-token NLL. With $K = 56$ and exploration ratio $\epsilon = 0.1$, each policy queries the 50 highest-scoring candidates and six candidates sampled uniformly from the remainder. These heuristics train no predictor and perform no replay updates.

**Linear-25** uses the same normalized six rollout features, online utility targets, replay buffer, regression and ranking losses, exploration, and 25% query budget as Learned-25. Its only architectural change is replacing the layer-normalized hidden MLP with a single $\mathrm{Linear}(6, 1)$ utility predictor, isolating the effect of predictor capacity.

### A.2 COMPLETE TRAINING ALGORITHM

**Algorithm 1** BAS-OPD with learned utility selection

**Input:** Policy $\theta$; query ratio $\rho$; exploration ratio $\epsilon$; replay threshold $M$; EMA rate $\tau$.
**Output:** Trained full-image policy $\theta$ for inference without crops or a selector.

1: Initialize $\bar{\theta} \leftarrow \theta$, predictor $f_\psi$, feature normalizer, and FIFO buffer $\mathcal{R} \leftarrow \varnothing$.
2: **for** each training step $s$ with rollout batch $\mathcal{B}$ **do**
3: Generate $y_i \sim p_\theta(\cdot \mid x_i, q_i)$ for all $i \in \mathcal{B}$; identify crop-eligible candidates $\mathcal{C}$.
4: Form detached features $\phi_i$ (Eq. 8) and update normalization on $\mathcal{C}$; set $K \leftarrow \lfloor \rho|\mathcal{C}| \rfloor$.
5: **if** $f_\psi$ has not yet been updated **then**
6: $\mathcal{S} \leftarrow \mathrm{Uniform}(\mathcal{C}, K)$, sampling without replacement.
7: **else**
8: Predict $\hat{u}_i \leftarrow f_\psi(\mathrm{Normalize}(\phi_i))$ for $i \in \mathcal{C}$; set $K_\mathrm{e} \leftarrow \lceil \epsilon K \rceil$.
9: Form $\mathcal{S}$ from the top $K - K_\mathrm{e}$ candidates by $\hat{u}_i$ plus $K_\mathrm{e}$ candidates sampled uniformly without replacement from the remainder.
10: **end if**
11: Run the standard OPD student forward for $p^S$; obtain detached crop-teacher $p^T$ on the same tokens **only for** $i \in \mathcal{S}$.
12: Compute detached utility labels $u_i$ for $i \in \mathcal{S}$ (Eq. 7).
13: Update $\theta$ using $\mathcal{L}_\mathrm{BAS}$ (Eq. 11); after a valid update, set $\bar{\theta} \leftarrow (1-\tau)\bar{\theta} + \tau\theta$.
14: Add valid detached $(\phi_i, u_i, s)$ to bounded replay $\mathcal{R}$; if new pairs were added and $|\mathcal{R}| \geq M$, update **only** $\psi$ from replay using $\mathcal{L}_\mathrm{sel}$ (Eq. 10).
15: **end for**

### A.3 SELECTOR TRAINING AND UPDATE HANDLING

**Implementation and update handling.** For the selected subset $\mathcal{S}$, the crop teacher produces detached distributions on the same response tokens evaluated by the student. Under a reduced budget, unselected rows are excluded before the crop-teacher forward pass and do not contribute to the selective distillation loss. The MLLM parameters $\theta$ and selector parameters $\psi$ are optimized separately by $\mathcal{L}_\mathrm{BAS}$ and $\mathcal{L}_\mathrm{sel}$, respectively. Selection baselines and supervision-cost definitions are given in Appendices A.1 and C.

Algorithm 1 (line 14) stores observed feature–utility pairs in a bounded first-in, first-out replay buffer together with their collection steps. Replay updates require newly observed valid pairs and the minimum buffer size specified in Table 6. Each update samples a minibatch uniformly without replacement, capped by the current buffer size. At training step $s$, an item collected at $s_i$ receives the recency weight $w_i = 2^{-(s-s_i)/h}$, where $h$ is the age half-life.

For each replay update, we normalize stored features using the current statistics without changing those statistics, and recompute $\hat{u}_i$ with the current predictor while retaining gradients to $\psi$. The weighted Huber term in Eq. 10 is

$$\mathcal{L}_\mathrm{reg} = \frac{\sum_i w_i \ \mathrm{Huber}(\hat{u}_i - u_i)}{\sum_i w_i}. \tag{12}$$

The ranking term $\mathcal{L}_\mathrm{rank}$ trains the ordering of labeled samples. Each pair $(i, j)$ with $|u_i - u_j| > 10^{-6}$ contributes $\mathrm{softplus}[-\mathrm{sgn}(u_i - u_j)(\hat{u}_i - \hat{u}_j)]$, weighted by $\sqrt{w_i w_j}$. Features and targets are detached, and the selector has a separate optimizer, so $\mathcal{L}_\mathrm{sel}$ does not update the MLLM.

The denominator of Eq. 11 is clamped to at least one. When $\mathcal{S} = \varnothing$, crop-teacher scoring is skipped and the distillation term is zero; the remaining updates follow the standard OPD optimizer and teacher-update rules. The teacher EMA follows a valid student update (Algorithm 1, line 13). An empty selected set provides no new utility labels and therefore does not trigger a replay update.

# B Training and Evaluation Details

## B.1 Training configuration

**Matched training-objective baselines.** The comparisons in Table 2 use the same 3,120-pair preprocessed split without method-specific subsampling. GRPO, DAPO, OPSD, and BAS-OPD generate eight rollouts per prompt, whereas SFT applies supervised updates directly and generates none.

Each 111-step run processes 3,108 examples (28 per batch), dropping the final 12. Table 5 reports the shared training and policy-specific rollout settings.

Table 5: **Training configurations.** Shared settings apply to both backbones; Full ($28 \times 2$) uses two rollouts per prompt.

*(a) Shared model-training settings*

| | | | |
|---|---|---|---|
| Backbones | Qwen3.5-4B / 9B | Training examples | 3,120 (3,108 processed) |
| Epochs / steps | 1 / 111 | Max. model length | 9,216 tokens |
| Prompt / response limits | 8,192 / 1,024 | MLLM optimizer | AdamW |
| Learning rate / weight decay | $2 \times 10^{-6}$ / 0.01 | Distillation | JSD ($\alpha = 0.5$), top-100 |
| Crop-teacher EMA rate | 0.05 | Hardware | 4× A100-SXM4 80 GB |
| Training seed | 42 | Training schedule | One epoch |

*(b) Rollout and selection settings*

| **Policy** | **Prompt batch** | **Rollouts per prompt** | **Candidates per step** | **Selected per step** | **Steps** | **Query ratio** | **Seed** |
|---|---|---|---|---|---|---|---|
| Full querying | 28 | 8 | 224 | 224 | 111 | 100% | 42 |
| Full ($28 \times 2$) | 28 | 2 | 56 | 56 | 111 | 100% | 42 |
| Random-25 | 28 | 8 | 224 | 56 | 111 | 25% | 42 |
| Entropy-25 | 28 | 8 | 224 | 56 | 111 | 25% | 42 |
| Mean-NLL-25 | 28 | 8 | 224 | 56 | 111 | 25% | 42 |
| P90-NLL-25 | 28 | 8 | 224 | 56 | 111 | 25% | 42 |
| Linear-25 | 28 | 8 | 224 | 56 | 111 | 25% | 42 |
| **Learned-25** | 28 | 8 | 224 | 56 | 111 | 25% | 42 |

### B.2 PREDICTOR CONFIGURATION AND FEATURE HANDLING

Algorithm 1 (line 4) forms the detached feature vector in Eq. 8. The sampled-token negative log-likelihood is $\ell_{i,t} = -\log p_{\text{old}}(y_{i,t} \mid x_i, q_i, y_{i,<t})$; $\overline{\ell}_i$ and $Q_{0.9}(\ell_i)$ are its response-level mean and 90th percentile. The remaining features are mean token entropy $\overline{H}_i$ when available, response length $T_i$, prompt length $L_i$, and crop-to-full-image area ratio $a_i$.

The predictor $f_\psi$ consists of layer normalization, one hidden linear layer with a SiLU activation, and a scalar output. The operator $\mathrm{Normalize}(\cdot)$ uses running feature statistics, updated on crop-eligible candidates during feature extraction and held fixed during replay updates.

Table 6 gives the selector settings shared by Learned-10, Learned-25, and Learned-50. The predictor runs on the CPU and ranks candidates by predicted utility, without dividing by response-token cost. Running feature normalization uses exponential mean and variance with numerical constant $10^{-6}$. Unavailable entropy, prompt-length, or crop-area values are represented by zero.

Table 6: **Learned-selector configuration shared across query budgets.** Learned-10, Learned-25, and Learned-50 use the same predictor, replay, ranking, and exploration settings.

| **Hyperparameter** | **Value** |
|---|---|
| Feature dimension / hidden dimension | 6 / 32 |
| Selector optimizer / learning rate | Adam / $1 \times 10^{-3}$ |
| Running-normalizer momentum | 0.99 |
| Replay capacity / start threshold | 512 / 96 labels |
| Replay minibatch size | 256 |
| Selector updates per training step | 2 |
| Replay age half-life | 10 steps |
| Pairwise ranking weight | 0.2 |
| Minimum utility gap for a pair | $1 \times 10^{-6}$ |
| Exploration ratio | 0.10 |
| Selector seed | 0 |
| Score mode | predicted utility |

### B.3 TRAINING INPUT EXAMPLES

Figure 6 shows three records from the training split. For each record, the student receives the full scene with the dataset-provided red bounding-box overlay, while the crop teacher receives the paired evidence crop. Both branches use the displayed prompt and the same student-generated response prefix during OPD.

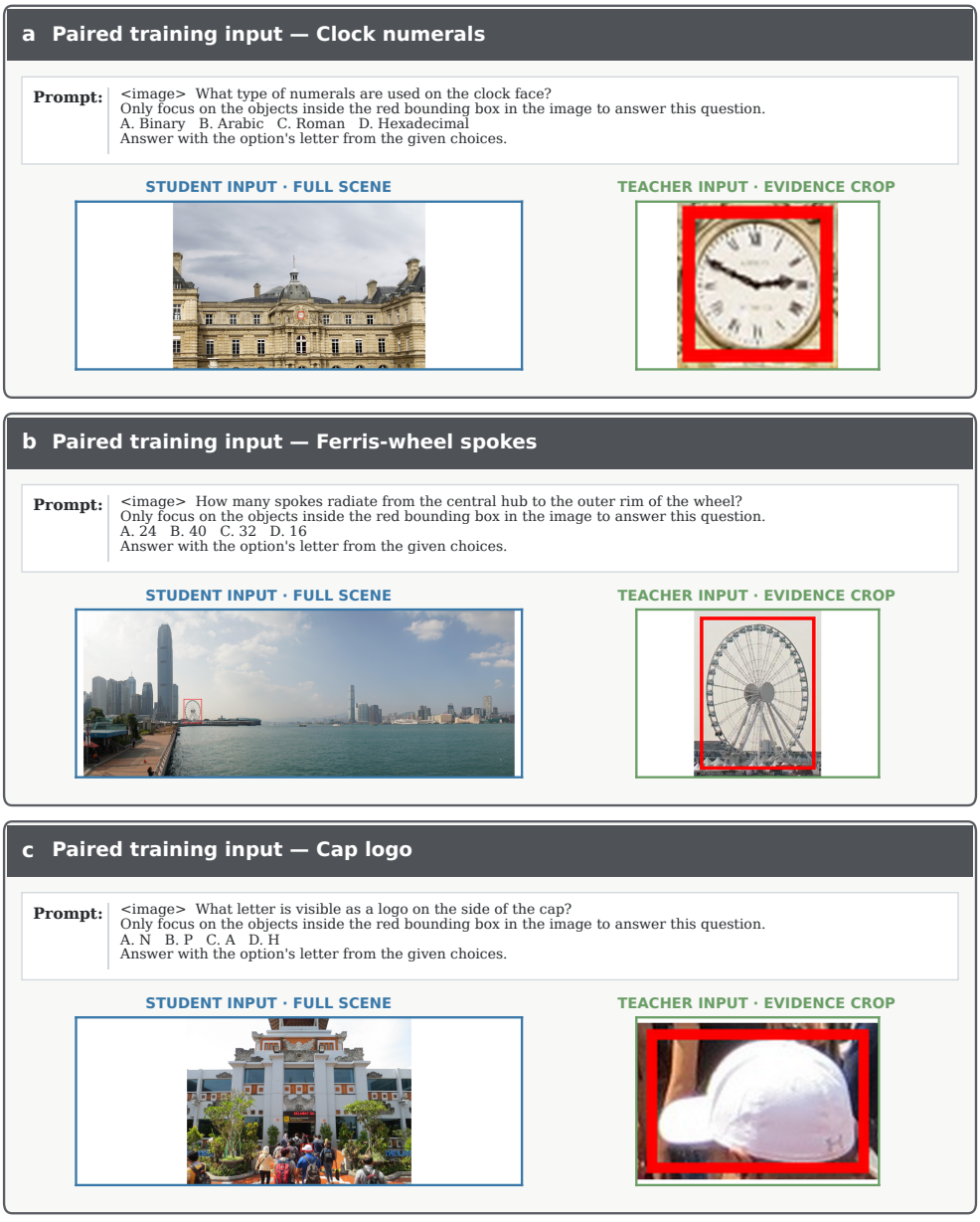



Figure 6: **Examples of paired training inputs and prompts.** Each row shows the recorded prompt, the full-scene student input, and the paired evidence crop used by the teacher. Images and prompts are reproduced from the training split; only display size and prompt line breaks are adjusted.

### B.4 Evaluation protocol

Generation uses temperature 0 and seed 42. We first apply benchmark-specific deterministic parsers and exact-answer equivalence matching. Only responses unresolved by these rules are judged by GPT-5.5, using temperature 0 and a maximum output length of 2,048 tokens. The first valid *Yes* or *No* label in the response is used as the verdict.

***GPT-5.5 judging prompt***

Your task is to judge whether the response expresses the same meaning as the answer of a question.
The question is: `{question}`
The answer is: `{answer}`
The response is: `{response}`
Please check and compare them and then judge. If the response is correct, your output should be *Yes*. Otherwise, your output should be *No*. Directly give me your output.

CV-Bench is reported as the macro-average of its 2D and 3D accuracy categories, and POPE is reported by accuracy in the policy comparison. The POPE evaluation contains 9,000 samples.

## C Teacher-Supervision Cost

Training cost is recorded directly as the number of selected teacher calls, $|\mathcal{S}|$, and the number of response tokens scored by the teacher, $\sum_{i\in\mathcal{S}} T_i$.

Figure 7 summarizes the seven-benchmark score and teacher cost at both model scales. Table 7 reports the corresponding supervision counts for both backbones.

**Teacher-scored tokens.** At 4B, Random-25, Entropy-25, Mean-NLL-25, P90-NLL-25, Linear-25, and Learned-25 used 26.28%, 27.65%, 29.01%, 41.18%, 13.82%, and 15.78% of their respective full-pool token totals (Table 3). Thus, Learned-25 achieved the highest aggregate accuracy while scoring fewer tokens than the random, entropy, Mean-NLL, and P90-NLL selectors.

**Scaling to 9B.** The same comparison at 9B used scored-token ratios of 26.11%, 27.92%, 25.73%, 30.28%, 13.57%, and 16.02%, respectively. Learned-25 again combined the highest aggregate accuracy with fewer scored tokens than the random, entropy, Mean-NLL, and P90-NLL selectors.

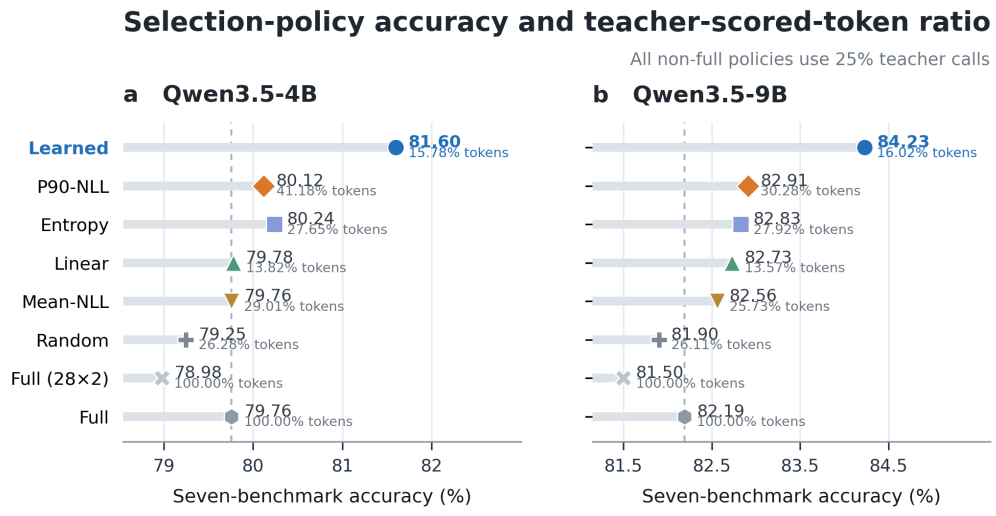



Figure 7: **Selection-policy accuracy and scored-token ratios.** Panels compare all Table 3 policies on Qwen3.5-4B and Qwen3.5-9B.

Table 7: **Teacher-supervision costs for the policies in Table 3.** Full-pool tokens are the recorded totals used as the denominator for each token ratio.

| Policy | Candidates | Teacher calls | Query ratio | Scored tokens | Full-pool tokens | Token ratio |
|---|---|---|---|---|---|---|
| Full querying | 24,864 | 24,864 | 100.00% | 2,167,850 | 2,167,850 | 100.00% |
| Full ($28\times2$) | 6,216 | 6,216 | 100.00% | 546,370 | 546,370 | 100.00% |
| Random-25 | 24,864 | 6,216 | 25.00% | 569,674 | 2,167,709 | 26.28% |
| Entropy-25 | 24,864 | 6,216 | 25.00% | 599,321 | 2,167,526 | 27.65% |
| Mean-NLL-25 | 24,864 | 6,216 | 25.00% | 562,924 | 1,940,751 | 29.01% |
| P90-NLL-25 | 24,864 | 6,216 | 25.00% | 651,755 | 1,582,597 | 41.18% |
| Linear-25 | 24,864 | 6,216 | 25.00% | 365,134 | 2,642,069 | 13.82% |
| **Learned-25** | 24,864 | 6,216 | 25.00% | 342,089 | 2,167,864 | 15.78% |
| *Qwen3.5-9B Backbone* | | | | | | |
| Full querying | 24,864 | 24,864 | 100.00% | 2,118,437 | 2,118,437 | 100.00% |
| Full ($28\times2$) | 6,216 | 6,216 | 100.00% | 533,286 | 533,286 | 100.00% |
| Random-25 | 24,864 | 6,216 | 25.00% | 549,547 | 2,104,731 | 26.11% |
| Entropy-25 | 24,864 | 6,216 | 25.00% | 585,863 | 2,098,421 | 27.92% |
| Mean-NLL-25 | 24,864 | 6,216 | 25.00% | 543,571 | 2,112,693 | 25.73% |
| P90-NLL-25 | 24,864 | 6,216 | 25.00% | 632,679 | 2,089,517 | 30.28% |
| Linear-25 | 24,864 | 6,216 | 25.00% | 290,915 | 2,143,884 | 13.57% |
| **Learned-25** | 24,864 | 6,216 | 25.00% | 339,863 | 2,121,607 | 16.02% |

# D QUALITATIVE CASE STUDY

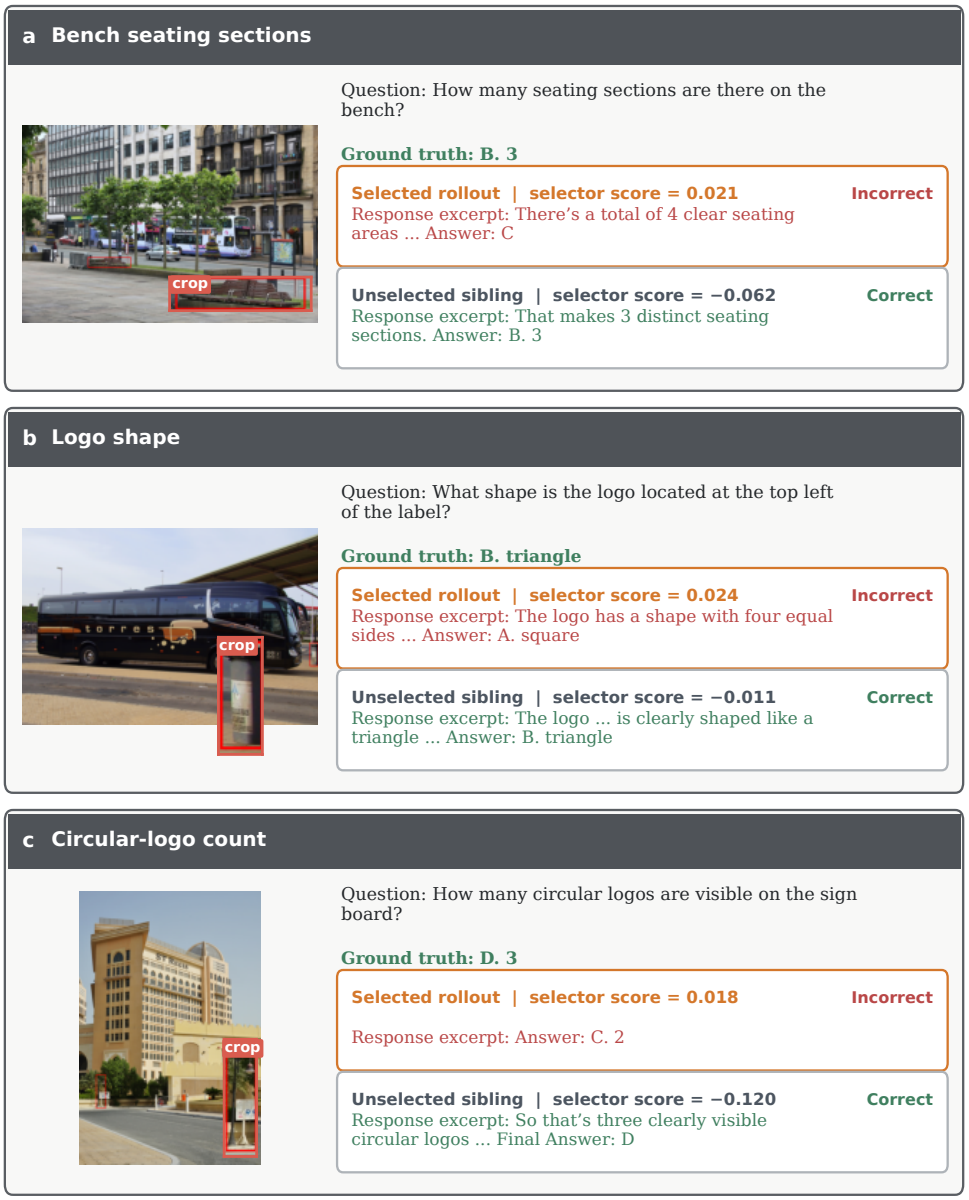



Figure 8: **Selected and unselected rollouts for matched prompts.** Each panel shows an erroneous rollout selected for crop-teacher supervision and a correct unselected sibling generated for the same prompt. Orange borders denote selection; green and red text indicate correctness.