WHAT TO TRUST, WHAT TO LEARN: A GROUNDED LEARNING CURRICULUM FOR SELF-EVOLVING LARGE MULTIMODAL MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

In the self-evolving paradigm of large multimodal models, a Proposer and a Solver derived from a shared backbone cooperate in a closed training loop: the Proposer generates questions from visual input, and the Solver's most frequent answer is treated as a self-defined "golden label". This avoids human labels but raises a fundamental question: **how is self-evolution steered, and can its self-rewarding signal be trusted?** We identify two failure modes: *Proposer bias*, where question rewards favor invalid or unhelpful tasks, and *Solver bias*, where agreement-based supervision reinforces shared errors as self-generated ground truth. Across seven checkpoints on POPE and HallusionBench, unanimous errors account for 14.5–76.7% of wrong majority answers, while entropy-targeted question rewards can further increase invalid questions. We propose **a grounded learning curriculum** that separates what to trust from what to learn: a frozen detector grounds candidate questions in visual evidence and enables the construction of supported answer sets in place of Solver consensus; the Solver learns from the full supported set, while the Proposer earns credit only when matched trial updates yield greater Solver progress on separate probe images. No task-specific question–answer labels, teacher model, or answer judge are used, preserving the self-evolving setting while grounding its signals in visual evidence. On eleven standard benchmarks, our curriculum raises Qwen3.5-9B's average accuracy from 75.58% to 78.61%, with the largest gains on CV-Bench ($+4.85$) and MMBench ($+4.80$). The same curriculum improves Qwen2.5-VL-7B and Qwen3-VL-8B, with average gains of $+2.88$ and $+2.60$ points, respectively, showing consistent gains across backbones. Evidence-grounded targets outperform majority-vote targets by 2.43 points, and a learned Proposer provides an additional 1.30-point gain over a frozen one. Code and models will be publicly released.

1 INTRODUCTION

Self-evolving multimodal models aim to improve their capabilities without annotations. One backbone plays two roles: a *Proposer* asks questions about unlabeled images, and a *Solver* learns to answer them (Thawakar et al., 2025; He et al., 2025). The loop needs two signals that no annotator provides: which answers are correct and which questions are worth teaching. Current methods take both signals from the Solver itself. The answer that most responses agree on becomes the training label, and questions on which the samples disagree are rewarded as usefully hard (Huang et al., 2025).

This design assumes that agreement tracks correctness and that uncertainty tracks teaching value; neither holds reliably. In text, R-Zero's majority-vote labels fall from 79% to 63% correct against GPT-4o answers over three iterations as its questions grow harder, while its Solver keeps training on them (Huang et al., 2025). We find the same failure in vision (Figure 1). Following EvoLMM's protocol (Thawakar et al., 2025), we draw five answers per question over seven checkpoints on POPE (Li et al., 2023) and HallusionBench (Guan et al., 2024); when the majority answer is wrong, all five agree on it in 14.5–76.7% of cases. These unanimous errors look maximally reliable, so agreement-based training rewards them; we call this *Solver bias* (Section 3.1). The question signal fails differently: an invalid or ambiguous question also produces disagreement, and training a Pro-

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

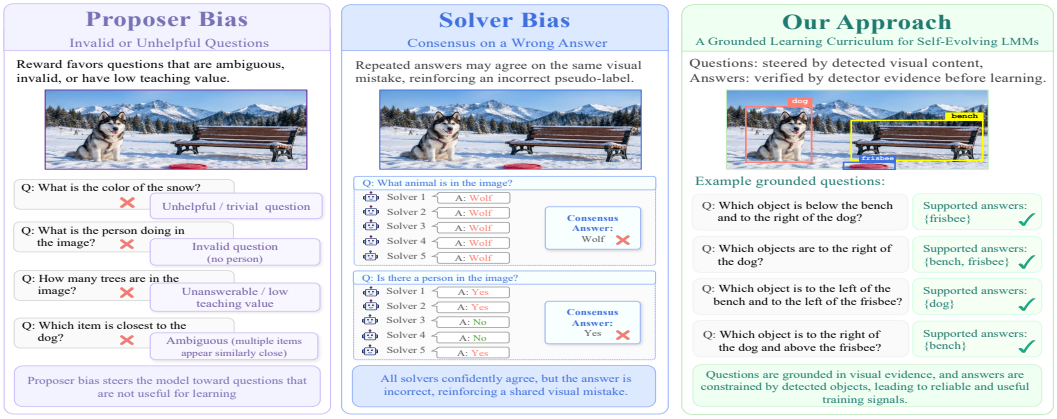


Figure 1: **Two biases in self-evolving training, and our proposed curriculum.** *Left:* Proposer bias. An uncertainty reward favors trivial, invalid or unhelpful questions. *Center:* Solver bias. Five answers agree on a wrong answer, which agreement-based training would reinforce. *Right:* Our curriculum grounds questions in detected objects and trains on the supported answer sets.

poser on an entropy reward raises the share of invalid questions from 20.8% to 36.5% on synthetic scenes with checkable answers (Section 3.2); we call this *Proposer bias*. Both failures come from asking the Solver's own answers to do two jobs. Our *grounded learning curriculum* separates them: it takes *what to trust* from the visual evidence of a frozen detector, which can flag a unanimous error that no vote count can, and *what to learn* from measured progress, with no labels, teacher or judge.

What to trust. A frozen object detector locates the objects in each image, and a fixed compiler turns these detections into spatial questions such as "Which object is entirely to the left of the chair and above the table?". Every category whose boxes satisfy both relations enters the *supported answer set*, and the Solver, which sees only the image and the question, learns to raise the total probability of this set rather than commit to one arbitrary answer.

What to learn. We briefly train two copies of the current Solver on two candidate question batches and credit the Proposer for the batch that improves the Solver more on separate probe images. At inference, only the Solver runs, without the detector.

We instantiate the curriculum with an RT-DETRv2 detector (Lv et al., 2024) and Qwen3.5-9B on natural images and validate it on five further models. Training uses no question–answer annotations, but the detector is pretrained with supervision, and about 11% of its supported claims are false under COCO annotations (Section 5.4). **Our contributions are:**

- **Diagnosing the two biases.** We show that no score computed only from vote counts can separate a unanimous error from a unanimous success, that such errors are common across model families, and that entropy rewards increase invalid question generation (Section 3).
- **A grounded learning curriculum.** The Solver learns detector-supported answer sets instead of its own agreement, and the Proposer, choosing among verified questions, is credited for the progress they produce, so an invalid or ambiguous question cannot earn credit by confusing the Solver (Section 4).
- **Evidence, not agreement, drives the Solver's gain.** On the same questions, evidence targets improve the eleven-benchmark average by 2.43 points more than majority-vote labels, which repeat every confident error of the base model (Section 5.3).
- **Learning what to teach adds a further gain.** A progress-credited Proposer improves the eleven-benchmark average over a frozen Proposer by 1.30 points and over an entropy-credited one by 1.40 points, with gains where the taught skill applies (Section 5).

2 Related Work

Self-evolving methods differ mostly in where their training signal gets its correctness, so we group prior work by that source, from signals the model produces for itself to evidence from outside it.

Agreement and uncertainty. R-Zero trains a text challenger and solver, rewarding questions on which the solver is uncertain and labeling answers by majority vote (Huang et al., 2025). EvoLMM brings this loop to images with a continuous entropy-targeted proposal reward (Thawakar et al., 2025), while VisPlay combines majority-frequency difficulty with diversity penalties and GRPO (He et al., 2025). Active Zero also learns which images to acquire based on vote consensus (He et al., 2026a), and iReasoner rewards agreement on intermediate steps (Sunil et al., 2026). MM-Zero, meanwhile, keeps majority-derived labels for hard questions (Li et al., 2026c). Across these approaches, supervision is primarily derived from the model's own responses rather than independently grounded evidence. Vision-Zero is an exception: players vote on which of them saw an edited copy of the image, so the reward is known by construction (Wang et al., 2026), but the model learns to spot edits rather than to answer questions about unedited images. Our Solver never trains on agreement, learning instead from detector-supported answer sets, while our Proposer is credited for measured progress rather than uncertainty, selecting from fixed, detector-supported menus that make it less expressive, so we compare learned and frozen selection directly.

Self-judging. RISE uses the current Solver as its own supervisor (Xu et al., 2026). CoVerRL trains answering, verification and correction in one policy (Pan et al., 2026). When Models Judge Themselves freezes the initial model as judge; SeeQ uses it to rewrite and filter questions (Wu et al., 2026; Liang et al., 2026). J-Zero (Chu et al., 2026) and Agent0-VL (Liu et al., 2025) instead start from an external reward model and a teacher-assisted cold start, respectively. A judge that shares the Solver's backbone can share its visual mistakes, which are exactly the unanimous errors we measure in Section 3.1; this is a risk rather than a demonstrated failure. Our targets come from a perception model with separate weights, whose errors need not coincide with the Solver's but are not absent.

Consistency across views. VISE combines geometric consistency with self-judged visibility (Venkatraman et al., 2026b); S^2VOPD distills a clean-image teacher into a degraded-image student (Li et al., 2026b), CVPD distills from informative crops and ghost views (Venkatraman et al., 2026a), and TTIQ rewards image–question dependence and confidence (He et al., 2026b). These methods report broad benchmark gains, but consistency alone cannot distinguish a shared error from a correct prediction. We instead study a narrower setting where detector evidence can check answers.

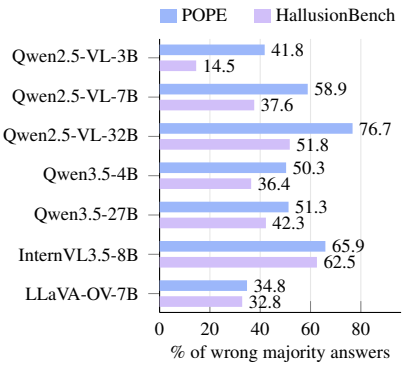
Structured evidence and executable tasks. GQA builds questions from annotated scene graphs, SpatialVLM and SpatialRGPT derive spatial supervision from pretrained perception, and Pseudo-Q generates region–query pairs from detector outputs (Hudson & Manning, 2019; Chen et al., 2024a; Cheng et al., 2024; Jiang et al., 2022); SpatialEvo co-trains a questioner and solver with geometric checks from 3D annotations (Li et al., 2026a), and Absolute Zero and EVE verify constructed tasks by execution (Zhao et al., 2025; Heng et al., 2026). We share the premise that structure can verify an answer, but ground that structure in a detector over unlabeled natural images and learn which questions to teach. Like SpatialVLM, we inherit supervision from the perception model, so our evidence is only as correct as the detections.

Progress credit and answer sets. Learning progress is an established signal for curriculum design (Graves et al., 2017): Learning by Asking selects visual questions by past accuracy gains (Misra et al., 2018), and SOAR, SEAL, INFUSER and LESS reward or select data by post-update performance or optimizer-aware influence against reference examples (Sundaram et al., 2026; Zweiger et al., 2025; Chen et al., 2026; Xia et al., 2024). We keep this idea but measure progress on held-out probe images against evidence-supported targets. Training on a candidate answer set follows partial-label learning (Feng et al., 2020), whose guarantees assume the set contains the true label; ours can be incomplete or wrong. Table 4 (Appendix A) contrasts the closest mechanisms.

3 Why Self-Evolving Signals Fail

For an image I, a Proposer π_ϕ selects a question q and a Solver p_θ answers it. Agreement-based methods take both training signals from the histogram $h(a) = N^{-1} \sum_{i=1}^{N} \mathbf{1}[a_i = a]$ of N sampled answers $a_1, \ldots, a_N \sim p_\theta(\cdot \mid I, q)$: the modal answer $\hat{a} = \arg\max_a h(a)$ becomes the label, and a function of h, such as its entropy, rewards the Proposer. We call a mismatch between agreement and answer correctness *Solver bias*, and a mismatch between proposal reward and a question's validity or teaching value (the improvement it produces in the Solver on other images) *Proposer bias*.

(a) Solver bias: unanimous errors



(b) Proposer bias: invalid questions

	Untrained reference	Entropy	Gated entropy	Validity only
Invalid ↓	20.83	36.46 ±2.08	18.75 ±6.51	**0.35** ±0.60
Malformed ↓	4.17	4.86 ±1.59	4.86 ±2.62	**0.35** ±0.60
Semantic ↓	16.67	31.60 ±1.59	13.89 ±6.10	**0.00** ±0.00
Unique valid ↑	45.83	39.93 ±1.59	**47.57** ±3.35	43.40 ±2.62
Difficult ↑	27.08	24.31 ±2.62	**28.12** ±2.08	24.65 ±1.59
Empty pools ↓	4.17	18.06 ±2.41	1.39 ±2.41	**0.00** ±0.00
Mean entropy reward during training: invalid 0.56, valid 0.26				

Figure 2: **Evidence for the two biases.** (a) Solver bias: the share of wrong majority answers on which all five sampled answers agree, for seven checkpoints on POPE and HallusionBench (counts in Table 5). (b) Proposer bias: share (%) of the questions a Qwen3.5-4B Proposer writes on held-out synthetic scenes after 64 updates with each reward (mean ± SD over three seeds; the untrained reference is one run); bold marks the best value per row, and Appendix C defines each row.

3.1 Solver bias: agreement cannot certify its own labels

Vote counts cannot flag unanimous errors. Any score computed only from vote counts gives two unanimous questions the same value whether the shared answer is right or wrong, so its tie-aware AUC for separating unanimous errors from unanimous successes is exactly $1/2$. This covers majority voting, consensus-fraction rewards and entropy, but not verifiers that use the answer's identity or other evidence from the image.

Such errors are common. In a *unanimous error*, all $N = 5$ answers parse, agree and are wrong. Across seven checkpoints from Qwen2.5-VL, Qwen3.5, InternVL3.5 and LLaVA-OneVision on POPE and HallusionBench (Bai et al., 2025b; Qwen Team, 2026; Wang et al., 2025; Li et al., 2024; 2023; Guan et al., 2024), unanimous errors make up 14.5–76.7% of wrong majority answers (Figure 2a; Appendix B); conversely, on our 1,839 in-domain questions (Section 5.1), 69 of the 454 unanimous Qwen3.5-9B answers (15.2%) are wrong. VisPlay reports the resulting drift: its majority-vote pseudo-labels fall from 72% to 61% accuracy over three iterations (He et al., 2025).

Some estimators reinforce unanimous errors. GRPO's group-centered advantage is zero for a flat reward (Shao et al., 2024), but the examined EvoLMM update (Thawakar et al., 2025) scores $s_i = h(a_i)^{0.7}(1 - 0.1\ell_i)$, where ℓ_i is the answer length, against a running baseline. Thus, a constant reward maintains a positive advantage while it exceeds the baseline:

$$A_i = s_i - b_i, \quad b_{i+1} = \rho b_i + (1-\rho)s_i \quad \Longrightarrow \quad A_i = \rho^i(s - b_0) > 0 \ \text{ for } s > b_0, \tag{1}$$

and reinforces a unanimous answer, whether or not it is correct. In controlled single updates (Qwen3-VL-8B, three seeds), agreement-GRPO on four wrong answers and one correct one raises the wrong-minus-correct margin by 7.94 ± 0.12 nats more than a shuffled-reward placebo, and a running-baseline update on strict POPE errors raises it by 7.16 ± 0.03 nats although no decoded answer changes (Appendix B.2); Section 5.3 measures the accuracy cost of continued training.

3.2 Proposer bias: disagreement is not teaching value

Entropy does not measure validity. EvoLMM rewards questions whose answer entropy $\widehat{H}_N = -\sum_a h(a) \log h(a)$ is intermediate, $\widehat{R}_N = \exp[-(\widehat{H}_N - 0.9)^2/(2 \cdot 0.35^2)]$. A valid hard question, an invalid premise, several acceptable answers, and a plain Solver error can all produce the same answer distribution. With five samples, the reward pays the questions with the most scattered answers 5.5 times the reward they would receive under the true answer distribution (Appendix C.1).

Entropy training raises invalidity. On rendered scenes where validity is known exactly (a valid question parses and has exactly one answer), training a Qwen3.5-4B Proposer on this reward for 64

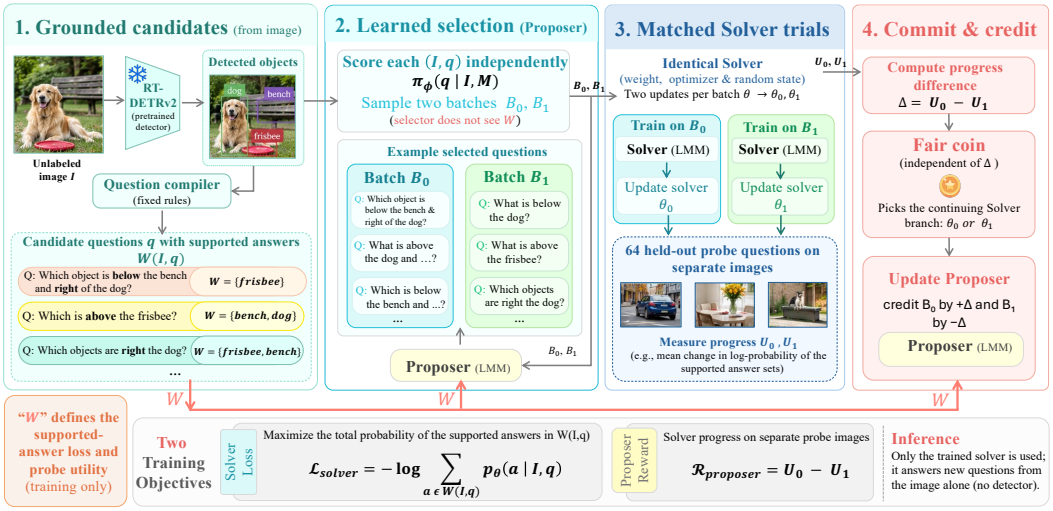


Figure 3: **Overview of the grounded learning curriculum.** (1) A frozen RT-DETRv2 detector finds objects in an unlabeled image I, and a fixed compiler turns them into candidate questions q, each with its set $W(I, q)$ of detector-supported answers. (2) The Proposer π_ϕ scores each candidate from the image and question alone, never seeing W, and samples two batches B_0, B_1. (3) Two copies of the Solver with identical weights, optimizer and random state each take two updates on one batch; U_0, U_1 measure the resulting gain in log-probability of the supported answer sets on separate probe images. (4) The Proposer is credited with $\Delta = U_0 - U_1$, while an independent fair coin picks the Solver branch that continues. *Bottom:* the Solver maximizes the total probability of the answers in W; W is used only in training, and only the Solver runs at inference.

updates raises the share of invalid questions on 24 held-out scenes from 20.8% to 36.5%, almost all due to semantic errors, while gating the reward on validity lowers it to 18.8% (Figure 2b; three seeds, Appendix C). This synthetic evidence shows the risk, not its size on photos: the seeds share evaluation scenes and were replicated after the first result was inspected.

4 Method: Grounded Learning Curriculum

Our curriculum separates the two decisions that agreement conflates. *What to trust* comes from visual evidence: a frozen detector supplies a set of supported answers, which becomes the Solver's target. *What to learn* comes from measured progress: the Proposer is credited by how much a question batch improves the Solver on separate probe images. Figure 3 shows the loop.

4.1 Questions with positive visual evidence

For each unlabeled image I, a frozen RT-DETRv2 detector (Lv et al., 2024) returns object categories and boxes, and a deterministic compiler builds questions from a relational grammar. Each question asks for the category of an object that satisfies two relations, one horizontal and one vertical, to one or two named anchors, for example, an object entirely left of the chair and above the table, where both relations must hold for the *same* object. The compiler checks the relations against the detections and collects every category with a complete witness (a detected box that satisfies both relations) into the supported set $W(I, q)$; questions without support are dropped, and questions with several supported answers keep all of them. The Solver sees only the unchanged image and question and answers with a category name; it never sees the boxes, witnesses or W (Appendix D gives the grammar and thresholds, and Figures 7 and 8 in the appendix show compiled questions on real images).

The check has a precise scope: with $C(I, q)$ the set of all valid answers, $W(I, q) \subseteq C(I, q)$ whenever the retained witnesses are sound. A missed target detection cannot invalidate a positive witness; however, a missed or incorrect anchor detection can. Therefore, we ask only positive, existential questions, never questions of absence, uniqueness, or counts. The detector never sees the question,

the Solver's answer, or the supported set; its inherited supervision comes only from its pretraining on annotated boxes.

4.2 SOLVER: SUPPORTED-SET LIKELIHOOD

Let $\ell_\theta(a \mid I, q)$ be the log-probability of the complete answer a including its end-of-message token. The Solver maximizes the probability of the supported event,

$$\mathcal{L}_S(I, q) = -s_\theta(I, q), \qquad s_\theta(I, q) = \log \sum_{a \in W(I,q)} \exp \ell_\theta(a \mid I, q), \tag{2}$$

so any supported answer satisfies the target, and none is chosen arbitrarily. This is the marginal likelihood of partial-label learning (Feng et al., 2020), with a candidate set from executable visual checks instead of the Solver's own agreement. An erroneous supported answer still receives credit, and because support is incomplete, lowering $\mathcal{L}_S$ need not raise the probability of valid answers outside W (Appendix D), so we always evaluate the Solver's raw decoded answers.

4.3 PROPOSER: PAIRED-PROGRESS CREDIT

The Proposer has its own parameters ϕ and scores each candidate question independently by its YES-minus-NO log-odds under a fixed usefulness prompt that contains only the image and that question; a softmax over the compiled menu M gives the policy $\pi_\phi(q \mid I, M)$. Each training window draws four teaching images with their compiled menus, and we sample two batches B_0, B_1 of one question per image. Both branches start from identical Solver, optimizer and random states, take two Solver updates, and are scored on the same 64 held-out probes $\mathcal{P}$, one supported question on each of 64 separate photographs:

$$U_b = \frac{1}{|\mathcal{P}|} \sum_{(I,q) \in \mathcal{P}} \big[s_{\theta_b}(I, q) - s_\theta(I, q) \big]. \tag{3}$$

With $\Delta = U_0 - U_1$ treated as a constant, the Proposer minimizes

$$\mathcal{L}_P = -\frac{\Delta}{2} \Big[\sum_{q \in B_0} \log \pi_\phi(q \mid I_q, M_q) - \sum_{q \in B_1} \log \pi_\phi(q \mid I_q, M_q) \Big], \tag{4}$$

whose negative gradient is an unbiased estimate of the gradient of expected two-update probe progress given the current Solver (Appendix D; the applied step is clipped AdamW). A fair coin, not the higher utility, decides which branch the Solver continues from (the other is discarded), so training follows the sampled policy. Credit therefore depends on whether a batch improves the Solver on *other* images, not on how much it confuses the Solver: an invalid or ambiguous question earns nothing unless it produces progress. Learning-progress teachers are established (Graves et al., 2017; Sundaram et al., 2026); our contribution is to measure progress against evidence-supported targets, so the Solver never trains on agreement and the Proposer is never paid for disagreement.

5 EXPERIMENTS

We report the main results (Section 5.2), ablate the Solver target, the Proposer credit and the set objective (Section 5.3), and analyze why evidence targets and progress credit help (Section 5.4).

5.1 EXPERIMENTAL SETUP

Models and arms. The main backbone is Qwen3.5-9B; Qwen2.5-VL-7B and Qwen3-VL-8B (Qwen Team, 2026; Bai et al., 2025b;a) (three seeds each), and InternVL3.5-8B, LLaVA-OneVision-7B, and Gemma3-12B-IT (Wang et al., 2025; Li et al., 2024; Gemma Team, 2025) (one seed each) test generality. Solver and Proposer are separate LoRA roles on one backbone, trained for 8,000 Solver updates on 851 detector-supported questions from 256 photographs, with 64 separate photographs as probes; no benchmark answer enters targets or credit. On Qwen3.5-9B, a frozen-Proposer arm keeps the initial selection policy, starts from identical weights, optimizer, and random

Table 1: **Comparison with self-evolving methods on eleven benchmarks.** Base models, published self-evolving methods on the same backbone (†: original-paper results under their own protocols) and ours after 8,000 Solver updates (mean ± SD, three seeds; Qwen3.5-9B with thinking disabled). The best score for each backbone is in bold. RWQA: RealWorldQA, HallB: HallusionBench. MMMU (Yue et al., 2024), A-OKVQA (Schwenk et al., 2022), MMStar (Chen et al., 2024b), MMBench (Liu et al., 2024), RWQA (xAI & visheratin, 2024), AI2D (Kembhavi et al., 2016), ChartQA (Masry et al., 2022), MathVista (Lu et al., 2024), POPE (Li et al., 2023), HallusionBench (Guan et al., 2024), CV-Bench (Tong et al., 2024).

		General						Chart	Math	Hallucination		Spatial
Model	Method	MMMU	A-OKVQA	MMStar	MMBench	RWQA	AI2D	ChartQA	MathVista	POPE	HallB	CV-Bench
Qwen2.5 VL-7B	Base	51.78	66.57	62.33	83.08	70.07	82.90	86.40	69.00	88.16	69.51	80.17
	EvoLMM†	52.01	–	–	–	69.9	83.41	86.70	70.52	88.4	–	–
	VisPlay†	38.27	–	–	–	69.67	–	–	–	–	**92.32**	–
	Ours	**54.85** ±0.92	**68.95** ±0.37	**65.38** ±0.26	**87.54** ±0.47	**73.16** ±0.51	**85.40** ±0.11	**86.84** ±0.65	**72.27** ±0.29	**89.57** ±0.31	72.84 ±0.15	**84.87** ±0.26
Qwen3 VL-8B	Base	60.56	65.64	65.33	85.14	70.59	83.74	84.92	60.60	88.63	71.82	86.72
	MM-Zero†	58.3	–	–	–	–	–	79.6	**67.2**	–	74.1	–
	CVPD†	–	–	–	86.10	71.07	84.63	–	–	–	–	88.27
	VISE†	52.69	–	–	85.46	70.03	84.10	**85.41**	–	**90.32**	–	–
	Ours	**63.81** ±1.40	**67.49** ±0.46	**68.33** ±0.29	**89.29** ±0.51	**72.98** ±0.28	**86.32** ±0.09	85.24 ±0.34	62.93 ±0.32	90.21 ±0.27	**74.52** ±0.18	**91.22** ±0.33
Qwen3.5 9B	Base	64.11	64.50	70.87	83.51	75.56	85.36	86.12	53.80	89.28	72.13	86.16
	Ours	**67.52** ±1.03	**66.69** ±0.51	**74.07** ±0.25	**88.31** ±0.42	**78.47** ±0.35	**88.01** ±0.08	**86.52** ±0.56	**57.93** ±0.30	**90.88** ±0.21	**75.29** ±0.18	**91.01** ±0.42

states, and trains the Solver in the same way, but never updates the Proposer. It is therefore supervised fine-tuning on detector-derived answers to a fixed question distribution, like SpatialVLM's data (Chen et al., 2024a), and our gain over it is what self-evolution adds.

Evaluation. *In-domain* accuracy measures the taught skill: the same grammar compiles 1,839 held-out questions from randomly sampled 267 images from COCO val2017 (teaching and probes use COCO unlabeled2017), and we score the Solver's raw answers against COCO's human annotations, counting an answer wrong when they cannot resolve it. We report mean ± SD over three seeds; paired effects count unresolved answers against us, so they sit slightly below differences of mean accuracy (Appendix D.5). Benchmarks use deterministic parsers, no model judge and, for Qwen3.5-9B, thinking disabled, so its scores sit below the official thinking-mode results (Qwen Team, 2026). Of the reported sets, only in-domain, MMMU and A-OKVQA were seen during development (Appendix E).

5.2 Main results

In-domain. Training on detector-supported answers raises in-domain accuracy from 69.8% to 84.6% after 8,000 updates ($+14.78 \pm 1.02$ points, in every seed; Table 8), with most of the gain in the first 1,000 updates.

Standard benchmarks. On Qwen3.5-9B, the eleven-benchmark average increases from 75.58 to 78.61 (Table 1). The largest gains occur on CV-Bench ($+4.85$) and MMBench ($+4.80$), whose spatial and object-relation questions resemble the taught ones, whereas ChartQA, which the question grammar never covers, remains unchanged; part of the MathVista gain reflects fewer truncated answers (Appendix E). The gains exceed the published gains of prior self-evolving methods on most shared benchmarks, each under its own setup (Table 10), and Figure 5 shows questions the Solver answers correctly only after training.

Other models. The curriculum also improves Qwen2.5-VL-7B and Qwen3-VL-8B by $+2.88$ and $+2.60$ points on average, and three further model families in single-seed runs (Table 2).

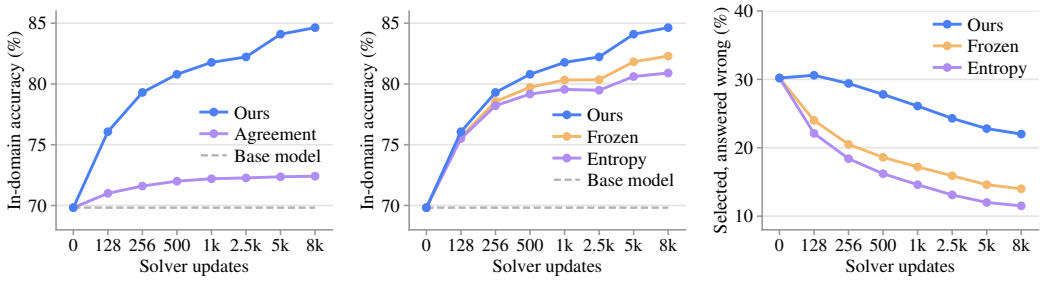
5.3 Ablation study

Table 3 crosses the Solver target with the Proposer credit on Qwen3.5-9B, and Figure 4 follows each arm throughout the training.

Solver target. From the same question pool and with the learned Proposer, agreement targets (the majority of five base-model answers) reach 72.4% after 8,000 updates, against 84.6% with evidence,

Table 2: **Generality across model families.** The same curriculum, unchanged, applied to three further model families (Wang et al., 2025; Li et al., 2024; Gemma Team, 2025) for 8,000 Solver updates with one seed; Base is the released instruction-tuned model. Every family improves on every benchmark, most on MMBench.

Model	MMMU	MMStar	MMBench	RWQA	AI2D	MathVista	HallB
InternVL3.5-8B	73.4	69.3	79.5	67.5	84.0	78.4	54.5
+ Ours	**76.00**	**72.07**	**83.23**	**69.93**	**85.95**	**81.20**	**57.10**
LLaVA-OneVision-7B	48.8	61.7	80.8	66.3	81.4	63.2	42.7
+ Ours	**52.67**	**65.00**	**86.19**	**69.41**	**84.52**	**66.80**	**45.95**
Gemma3-12B-IT	59.6	56.1	71.8	59.8	84.2	62.9	47.2
+ Ours	**63.22**	**59.20**	**76.16**	**62.88**	**86.92**	**66.20**	**50.26**



(a) Solver target (b) Proposer credit (c) What the Proposer selects

Figure 4: **Evidence targets and progress credit each add to self-evolution** (Qwen3.5-9B, in-domain accuracy, three seeds). (a) Evidence vs. agreement targets, both trained from the base model with our learned Proposer. (b) A Proposer credited by progress (ours), by answer entropy as in EvoLMM, or never updated (frozen), all with evidence targets. (c) The share of each Proposer's selections that the Solver answers outside the supported set (higher: more unsolved).

and gain 2.43 points less on the benchmark average (Table 3, Figure 4a); a control continued from a shared checkpoint with a frozen Proposer agrees in every seed (Appendix F): the target gives most of the gain.

Proposer credit. Progress credit improves on the frozen Proposer by $+2.30 \pm 0.90$ points in-domain and $+1.30 \pm 0.20$ on the benchmark average after 8,000 updates; the difference is within noise early and increases with training (Figure 4b). Entropy credit, as in EvoLMM, performs below the frozen Proposer, and six alternative designs tested during development did not outperform frozen or uniform selection (Appendix F.7). Learning what to teach is therefore beneficial only when credited by measured progress, and it contributes less than the target.

Set objective. Learning one sampled supported answer instead of the full set changes accuracy by less than one point (Appendix F), since few teaching questions have several supported answers.

5.4 Analysis

Why evidence targets help. Agreement training keeps the base model's unanimous errors, whereas evidence training can correct them where the detector supports another answer (Figure 8, Appendix F). Agreement cannot detect unanimous errors (Section 3.1); external evidence can, although 10.9% of detector-supported answers are wrong under COCO annotations (Appendix F.4).

What the Proposer selects. Figure 4c tracks how often the current Solver answers the selected questions incorrectly. This share falls least under progress credit, which keeps selecting unsolved questions, whereas entropy credit favors questions with dispersed answers, which on verified questions can arise from several acceptable answers (Section 3.2).

Transfer. On benchmarks, the gains concentrate on the skills the evidence component can verify, spatial and object relations rather than charts, and they hold across model families; the Solver also generalizes to relation–anchor pairs never seen in teaching (Appendix F.4).

Table 3: **Ablation of the Solver target and the Proposer credit** (Qwen3.5-9B, 8,000 Solver updates, mean over three seeds). Targets come from agreement (the majority of five base-model answers) or detector evidence; the Proposer is credited by answer entropy, by measured progress, or not at all (frozen); every arm trains from the base model. In-domain: the 1,839 held-out questions of Section 5.1; Avg.: the mean over the eleven benchmarks of Table 1.

Solver target	Proposer credit	In-domain	Avg.	CV-Bench	MMBench	MathVista	POPE
Base model	—	69.82	75.58	86.16	83.51	53.80	89.28
Agreement	Entropy (EvoLMM)	72.00	75.99	87.47	85.09	56.07	88.66
Agreement	Progress	72.41	76.18	87.78	85.34	55.90	88.87
Evidence	Frozen	82.29	77.31	88.81	86.31	56.83	89.86
Evidence	Entropy	80.90	77.21	88.58	86.36	56.67	89.74
Evidence	Progress (ours)	**84.63**	**78.61**	**91.01**	**88.31**	**57.93**	**90.88**



(a) Where is the car (red) relative to the truck? above ✗ → below ✓

(b) Where is the dining table relative to the couch? above ✗ → below ✓

(c) Closer to the books (red): table (blue) or bookcase (green)? bookcase ✗ → table ✓

(d) Closer to the camera: chair (red) or bookcase (blue)? bookcase ✗ → chair ✓

Figure 5: **Spatial questions answered correctly after grounded teaching.** Answers of the base model → the trained Solver after 5,000 updates (majority of six runs: three seeds, frozen and learned Proposer) on benchmark questions with a visible answer. The curriculum teaches 2D relations between detected objects, as in (a) and (b); distance (c) and depth (d) show qualitative transfer beyond them. Boxes are thickened for visibility. More examples in the appendix (Figures 6, 7 and 9).

6 Limitations and Discussion

The evidence inherits the detector's annotated pretraining and is fallible and incomplete, and the grammar covers only existential spatial conjunctions over object categories, which limits gains to the skills it teaches. COCO validation was used in developing the detector, and our development cohorts are not fresh confirmation data. The bias studies measure unanimous errors, executable invalidity and local update effects, not long-run harm. Most of the gain comes from the evidence targets, which detector-supervised fine-tuning with a frozen Proposer already obtains; the learned Proposer adds less (+1.30 on the benchmark average). Three-seed standard deviations are rough, and we do not compare against a matched self-judging baseline.

Beyond object detection. The curriculum needs only a component outside the Solver that can tell which answers the image supports. The detector witnesses spatial relations, which is why chart reading barely moves; other skills call for OCR, a chart parser, a depth estimator for 3D relations (Chen et al., 2024a), a segmenter for counting and absent objects (which POPE tests but our positive-only grammar cannot ask about), or a program executor for geometry (Zhao et al., 2025). Several could run side by side, the paired-progress Proposer learning which teaches most; the error rate of each must be measured, as for the detector. We plan to test these extensions as future work.

7 Conclusion

Self-evolving multimodal models take both training labels and question rewards from their own answers, and both signals fail: unanimous errors, which no vote count can flag, make up 14.5–76.7% of wrong majority answers, and entropy rewards favor invalid questions. Our curriculum separates the two decisions, teaching the answers that detector evidence supports and crediting questions by the progress they produce. On Qwen3.5-9B, evidence targets give most of the gain (+12.22 points in-domain over agreement), and progress credit adds +2.30 in-domain and +1.30 on eleven benchmarks over a frozen Proposer. Self-evolution needs a correctness signal from outside the model's answers, and the component that supplies it decides which skills the model can teach itself.

ETHICS STATEMENT

This work uses only public data: COCO images (Lin et al., 2014) for teaching, probes and in-domain evaluation, and the public evaluation splits of the benchmarks in Table 1, under their original terms. It involves no human subjects, no crowdsourcing and no new data collection, and we release no images. COCO photographs show people, so the detector-derived questions ask only about object categories and their positions, never about identities or personal attributes. The supervision inherits the detector's pretraining on annotated boxes, including its category coverage and its errors (10.9% of supported claims are false; Appendix F.4), so a model trained this way can learn those errors. Benchmark answers never enter training targets or rewards, and development benchmarks that were exposed during method design are marked as such (Appendix E). Our findings concern how self-training loops can reinforce their own mistakes; we see no direct path to harmful use beyond that of the underlying models.

REPRODUCIBILITY STATEMENT

Section 4 and Appendix D specify the question grammar, detector settings, both losses, the paired-progress credit, adapter and optimizer settings, the training schedule and the conservative estimand. Appendix E lists every evaluation population, metric, parser and answer limit, and states which populations were used during development. Appendices B and C give the protocols and full counts behind the two biases, and Appendix F gives the full results behind Section 5, including the Proposer designs that failed.

AI USE STATEMENT

In this work, we used AI tools, including Claude (Anthropic) as a coding agent, to help with coding and implementing the method, give feedback on the design of experiments, and help analyze and interpret results. We have strictly not used AI tools to generate data sets, formulate mathematical claims or write proofs; translation, dataset cleaning and qualitative data analysis are not applicable to this work. We additionally used AI tools for literature search, code editing, and polishing the text. The authors designed the study, reviewed all AI-assisted work, checked every experiment, reported number and citation, and take full responsibility for the content of this paper, including text, claims and artifacts produced with generative AI.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a. URL https://arxiv.org/abs/2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b. URL https://arxiv.org/abs/2502.13923.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. SpatialVLM: Endowing Vision-Language Models with Spatial Reasoning Capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14455–14465, 2024a. URL https://openaccess.thecvf.com/content/CVPR2024/html/Chen_SpatialVLM_Endowing_Vision-Language_Models_with_Spatial_Reasoning_Capabilities_CVPR_2024_paper.html.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024b.

Siyu Chen, Miao Lu, Beining Wu, Heejune Sheen, Fengzhuo Zhang, Shuangning Li, Zhiyuan Li, Jose Blanchet, Tianhao Wang, and Zhuoran Yang. INFUSER: Influence-guided self-evolution improves reasoning. *arXiv preprint arXiv:2606.09052*, 2026. URL https://arxiv.org/abs/2606.09052v4. Version 4.

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. SpatialRGPT: Grounded spatial reasoning in vision-language models. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.01584.

Gyouk Chu, Myeongho Jeon, and Eunho Yang. J-Zero: Unified Challenger–Solver–Judge Co-Evolution from Zero Data. *arXiv preprint arXiv:2608.26582*, 2026. URL https://arxiv.org/abs/2608.26582v1. Version 1.

Lei Feng, Jiaqi Lv, Bo Han, Miao Xu, Gang Niu, Xin Geng, Bo An, and Masashi Sugiyama. Provably consistent partial-label learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/file/7bd28f15a49d5e5848d6ec70e584e625-Paper.pdf.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Alex Graves, Marc G. Bellemare, Jacob Menick, Rémi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1311–1320. PMLR, 2017. URL https://proceedings.mlr.press/v70/graves17a.html.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14375–14385. IEEE, 2024.

Jinghan He, Junfeng Fang, Feng Xiong, Zijun Yao, Fei Shen, Haiyun Guo, Jinqiao Wang, and Tat-Seng Chua. Active Zero: Self-Evolving Vision-Language Models through Active Environment Exploration. *arXiv preprint arXiv:2602.11241*, 2026a. URL https://arxiv.org/abs/2602.11241v1. Version 1.

Xinrui He, Ting-Wei Li, Junting Wang, Mengting Ai, Xinyu He, Hanghang Tong, and Jingrui He. Harnessing image question dependence for better vlm test-time reinforcement learning. *arXiv preprint arXiv:2609.13296*, 2026b.

Yicheng He, Chengsong Huang, Zongxia Li, Jiaxin Huang, and Yonghui Yang. VisPlay: Self-Evolving Vision-Language Models from Images. *arXiv preprint arXiv:2511.15661*, 2025. URL https://arxiv.org/abs/2511.15661v2. Version 2.

Yongrui Heng, Chaoya Jiang, Han Yang, Shikun Zhang, and Wei Ye. Eve: Verifiable self-evolution of mllms via executable visual transformations. *arXiv preprint arXiv:2604.18320*, 2026.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-Zero: Self-Evolving Reasoning LLM from Zero Data. *arXiv preprint arXiv:2508.05004*, 2025. URL https://arxiv.org/abs/2508.05004v4. Version 4.

Drew A. Hudson and Christopher D. Manning. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. URL https://arxiv.org/abs/1902.09506.

Haojun Jiang, Yuanze Lin, Dongchen Han, Shiji Song, and Gao Huang. Pseudo-Q: Generating pseudo language queries for visual grounding. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15492–15502. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01507.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images, 2016.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. URL https://arxiv.org/abs/2408.03326.

Dingming Li, Yingxiu Zhao, Xinrui Cheng, Kangheng Lin, Hongbo Peng, Hongxing Li, Zixuan Wang, Yuhong Dai, Haodong Li, Jia Wang, Yukang Shi, Liang Zhao, Jianjian Sun, Zheng Ge, Xiangyu Zhang, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. SpatialEvo: Self-Evolving Spatial Intelligence via Deterministic Geometric Environments. *arXiv preprint arXiv:2604.14144*, 2026a. URL https://arxiv.org/abs/2604.14144v1. Version 1.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, 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. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.20. URL https://aclanthology.org/2023.emnlp-main.20/.

Yijiang Li, Yijun Liang, Yunjie Tian, Bingyang Wang, Ke Zhang, Zhenfei Yin, Di Fu, Philip Torr, and Nuno Vasconcelos. Self-Supervised Visual On-Policy Distillation. *arXiv preprint arXiv:2608.14144*, 2026b. URL https://arxiv.org/abs/2608.14144v1. Version 1.

Zongxia Li, Hongyang Du, Chengsong Huang, Xiyang Wu, Lantao Yu, Yicheng He, Jing Xie, Xiaomin Wu, Zhichao Liu, Jiarui Zhang, and Fuxiao Liu. MM-Zero: Self-Evolving Multi-Model Vision Language Models From Zero Data. *arXiv preprint arXiv:2603.09206*, 2026c. URL https://arxiv.org/abs/2603.09206v1. Version 1.

Yijun Liang, Hengguang Zhou, Ming Li, Lichen Li, Cho-Jui Hsieh, and Tianyi Zhou. Self-Evolving Visual Questioner. *arXiv preprint arXiv:2606.13929*, 2026. URL https://arxiv.org/abs/2606.13929v1. Version 1.

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 (ECCV)*, pp. 740–755, 2014.

Jiaqi Liu, Kaiwen Xiong, Peng Xia, Yiyang Zhou, Haonian Ji, Lu Feng, Siwei Han, Mingyu Ding, and Huaxiu Yao. Agent0-VL: Exploring Self-Evolving Agent for Tool-Integrated Vision-Language Reasoning. *arXiv preprint arXiv:2511.19900*, 2025. URL https://arxiv.org/abs/2511.19900v2. Version 2.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*, 2024.

Wenyu Lv, Yian Zhao, Qinyao Chang, Kui Huang, Guanzhong Wang, and Yi Liu. RT-DETRv2: Improved Baseline with Bag-of-Freebies for Real-Time Detection Transformer. *arXiv preprint arXiv:2407.17140*, 2024. URL https://arxiv.org/abs/2407.17140.

Ahmed Masry, Jia Qing Tan, Shafiq Joty, Enamul Hoque, et al. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pp. 2263–2279, 2022.

Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 15630–15649. PMLR, 2022. URL https://proceedings.mlr.press/v162/mindermann22a.html.

Ishan Misra, Ross Girshick, Rob Fergus, Martial Hebert, Abhinav Gupta, and Laurens van der Maaten. Learning by asking questions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. URL https://arxiv.org/abs/1712.01238.

Teng Pan, Yuchen Yan, Zixuan Wang, Ruiqing Zhang, Guiyang Hou, Wenqi Zhang, Weiming Lu, Jun Xiao, and Yongliang Shen. CoVerRL: Breaking the consensus trap in label-free reasoning via generator-verifier co-evolution. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29833–29853, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1376. URL https://aclanthology.org/2026.acl-long.1376/.

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

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European conference on computer vision*, pp. 146–162. Springer, 2022.

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. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Shobhita Sundaram, John Quan, Ariel Kwiatkowski, Kartik Ahuja, Yann Ollivier, and Julia Kempe. Teaching Models to Teach Themselves: Reasoning at the Edge of Learnability. *arXiv preprint arXiv:2601.18778*, 2026. URL https://arxiv.org/abs/2601.18778v3. Version 3.

Meghana Sunil, Manikandarajan Venmathimaran, and Muthu Subash Kavitha. iReasoner: Trajectory-Aware Intrinsic Reasoning Supervision for Self-Evolving Large Multimodal Models. *arXiv preprint arXiv:2601.05877*, 2026. URL https://arxiv.org/abs/2601.05877v3. Version 3.

Omkar Thawakar, Shravan Venkatraman, Ritesh Thawkar, Abdelrahman Shaker, Hisham Cholakkal, Rao Muhammad Anwer, Salman Khan, and Fahad Khan. EvoLMM: Self-evolving large multimodal models with continuous rewards. *arXiv preprint arXiv:2511.16672*, 2025. URL https://arxiv.org/abs/2511.16672v4. Version 4, revised June 9, 2026.

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, 2024.

Shravan Venkatraman, Omkar Thawakar, Ritesh Thawkar, Abdelrahman Shaker, and Rao Muhammad Anwer. Perception Before Supervision: Self-Contained Visual Distillation from Counterfactual Blind Spots. *arXiv preprint arXiv:2608.09931*, 2026a. URL https://arxiv.org/abs/2608.09931v1. Version 1.

Shravan Venkatraman, Ritesh Thawkar, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Salman Khan, and Fahad Khan. Paying More Attention to Visual Tokens in Self-Evolving Large Multimodal Models. *arXiv preprint arXiv:2606.27373*, 2026b. URL https://arxiv.org/abs/2606.27373v1. Version 1.

Qinsi Wang, Bo Liu, Tianyi Zhou, Jing Shi, Yueqian Lin, Yiran Chen, Hai Helen Li, Kun Wan, and Wentian Zhao. Vision-Zero: Scalable VLM self-evolution via multi-agent self-play. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.25541.

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. URL https://arxiv.org/abs/2508.18265.

Zhengxian Wu, Kai Shi, Chuanrui Zhang, Zirui Liao, Jun Yang, Ni Yang, Qiuying Peng, Luyuan Zhang, Hangrui Xu, Tianhuang Su, Zhenyu Yang, Haonan Lu, and Haoqian Wang. When Models Judge Themselves: Unsupervised Self-Evolution for Multimodal Reasoning. *arXiv preprint arXiv:2603.21289*, 2026. URL https://arxiv.org/abs/2603.21289v2. Version 2.

xAI and visheratin. Realworldqa. https://huggingface.co/datasets/visheratin/realworldqa, 2024. URL https://huggingface.co/datasets/visheratin/realworldqa.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2402.04333.

Chaoran Xu, Yingmao Miao, Pengfei Zhang, Hao Dou, Lei Sun, and Xiangxiang Chu. RISE: Reliable Improvement in Self-Evolving Vision-Language Models. *arXiv preprint arXiv:2605.20914*, 2026. URL https://arxiv.org/abs/2605.20914v2. Version 2.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*, 2024.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute Zero: Reinforced Self-play Reasoning with Zero Data. *arXiv preprint arXiv:2505.03335*, 2025. URL https://arxiv.org/abs/2505.03335v3. Version 3.

Adam Zweiger, Jyothish Pari, Han Guo, Ekin Akyürek, Yoon Kim, and Pulkit Agrawal. Self-adapting language models. *arXiv preprint arXiv:2506.10943*, 2025. URL https://arxiv.org/abs/2506.10943v2. Version 2.

APPENDIX

The appendix follows the order of the main paper. Appendix A compares the closest methods in detail (Section 2). Appendices B and C give the protocols and full results behind the two biases (Section 3). Appendix D specifies question construction, optimization and the estimand (Section 4), Appendix E the evaluation protocol, and Appendix F the additional results of Section 5.

A CLOSEST METHODS IN DETAIL

Table 4 contrasts how the closest methods obtain their questions, the signal that rewards them and the resources that verify their answers, based on each primary paper. It compares information sources, not matched reproductions or downstream performance.

Table 4: **Closest self-evolving and structured-evidence methods.** How each method obtains its questions, the signal that rewards them and the resources that verify its answers; the table compares mechanisms, not performance.

Method	Question policy	Reward or learning signal	Resources and verification scope
EvoLMM (Thawakar et al., 2025)	Learned question generation	Entropy-targeted proposal reward; running-baseline RL	Solver answer frequencies supply pseudo-supervision.
RISE (Xu et al., 2026)	Learned question generation	Difficulty, validity and skill balance	Current solver judges questions and answers; the external quality judge is diagnostic only.
VISE (Venkatraman et al., 2026b)	Self-generated localization queries; grounding is trained	Geometric box consistency and self-judged visibility changes	Transformed regions supply grounding feedback, without a direct question-teaching utility reward.
CVPD (Venkatraman et al., 2026a)	Offline question generation; no proposer learning	Crop/full and ghost/full distributional contrasts; distillation	Same-backbone EMA teachers and a reference anchor.
Absolute Zero (Zhao et al., 2025)	Learned generation of program-based tasks	Executor validity and solver-success-based proposal reward	Program execution checks the constructed reasoning tasks.
MM-Zero (Li et al., 2026c)	Rendered visual tasks	Solvability and difficulty; pseudo-label grading	Rendering supports construction, while hard-question labels retain majority-based supervision.
SOAR (Sundaram et al., 2026)	Learned training-data generation	Post-update change in reference-graded mathematical accuracy	Known hard-problem answers supply the outer teaching reward.
Ours	Learned selection from compiled finite question menus	Paired probe-logmass changes after actual trial updates; independent branch commitment	Frozen external detector with supervised pretraining supplies positive answer sets. No task-specific QA labels or additional answer-writing VLM; witness soundness is conditional.

B CONSENSUS ERRORS AND THEIR UPDATES

This appendix supports Section 3.1: how often unanimous answers are wrong, and what an agreement-based update does to them.

B.1 PREVALENCE OF UNANIMOUS ERRORS

For every item and checkpoint we draw five independent answers. Modal accuracy counts an item whenever one draw parses; strict consensus requires all five draws to parse and agree before the shared answer is scored, so a malformed draw is never absorbed into a unanimous decision. Every

retained model–dataset cell meets a prespecified mean parse-rate floor of 0.90. The study covers seven checkpoints from four model families: POPE contributes 1,500 questions on 483 images and HallusionBench 939–951 items, depending on parsing. "Right" and "wrong" refer to the benchmarks' reference answers. From Qwen2.5-VL-3B to 32B, the strict-wrong share rises from 41.78% to 76.71% on POPE and from 14.55% to 51.75% on HallusionBench (paired differences of 34.93 $[26.24, 43.54]$ and 37.21 $[29.98, 44.52]$ points over 10,000 whole-image-cluster resamples); these describe the observed checkpoints, not a causal effect of scale.

Table 5: **Consensus errors across seven checkpoints on POPE and HallusionBench.** For each model we draw five answers per question and report modal accuracy, the number of wrong majority answers on which all five draws agree (unanimous / wrong), and that count as a share of all items and of wrong majority answers. Unanimous errors are common and, within Qwen2.5-VL, rise with scale (on POPE, $41.78\% \rightarrow 76.71\%$ of wrong majority answers from Qwen2.5-VL-3B to 32B); the largest share of wrong majority answers per benchmark is in bold. Hallu.: HallusionBench.

Model	Benchmark	Items	Accuracy	Unanimous / wrong	% of items	% of wrong
Qwen2.5-VL-3B	POPE	1500	85.80	89/213	5.93	41.78
Qwen2.5-VL-7B	POPE	1500	82.00	159/270	10.60	58.89
Qwen2.5-VL-32B	POPE	1500	85.40	168/219	11.20	**76.71**
Qwen3.5-4B	POPE	1500	88.73	85/169	5.67	50.30
Qwen3.5-27B	POPE	1500	89.47	81/158	5.40	51.27
InternVL3.5-8B	POPE	1500	88.27	116/176	7.73	65.91
LLaVA-OV-7B	POPE	1500	89.07	57/164	3.80	34.76
Qwen2.5-VL-3B	Hallu.	950	65.26	48/330	5.05	14.55
Qwen2.5-VL-7B	Hallu.	951	68.14	114/303	11.99	37.62
Qwen2.5-VL-32B	Hallu.	950	72.95	133/257	14.00	51.75
Qwen3.5-4B	Hallu.	951	76.03	83/228	8.73	36.40
Qwen3.5-27B	Hallu.	951	79.60	82/194	8.62	42.27
InternVL3.5-8B	Hallu.	939	73.27	157/251	16.72	**62.55**
LLaVA-OV-7B	Hallu.	950	57.37	133/405	14.00	32.84

B.2 Update consequences

For a constant reward s, the running baseline $b_{i+1} = \rho b_i + (1-\rho)s$ gives the advantage $s - b_i = \rho^i(s - b_0)$, which proves Equation 1; with $b_0 = 0$, $s = 1$ and $\rho = 0.9$ the first five advantages are $1, 0.9, 0.81, 0.729, 0.6561$, so a unanimous answer keeps receiving positive credit. Controlled updates on Qwen3-VL-8B (rank-8 LoRA, three seeds; parameters, optimizer, reference policy, baseline and random state restored for every comparison) show that agreement-GRPO on four wrong answers and one correct one raises the wrong-minus-correct log-likelihood margin by 7.94 ± 0.12 nats more than a balanced shuffled-reward placebo ($+518.11 \pm 5.28$ projected onto frozen gradients), and on 61 strict errors from a POPE replay a running-baseline update raises it by 7.16 ± 0.03 nats relative to no update, while all 600 strict-state checks give exactly zero coefficient, gradient norm and projection. In a replay on 483 POPE-adversarial groups (5,796 rows, three seeds), shuffling rewards still moves the wrong-answer margin by $+7.14$ nats, because every shuffled response carries the same answer; decoded accuracy is unchanged (412/483 in every arm), so the replay shows local reinforcement, not harm on held-out data.

C Proposer Rewards on Executable Questions

This appendix supports Section 3.2: why a sampled entropy reward overpays scattered answers, and how an entropy-trained Proposer behaves when question validity is known exactly.

C.1 Finite-sample entropy

For a histogram with counts n_j summing to N, $\widehat{H}_N = \log N - \frac{1}{N}\log\prod_j n_j^{n_j}$; at $N = 5$ only seven entropy values are possible. For the band-pass reward $R(H) = \exp[-(H - 0.9)^2/(2 \cdot 0.35^2)]$, the expected reward of a finite sample differs from the reward of the population distribution: a uniform

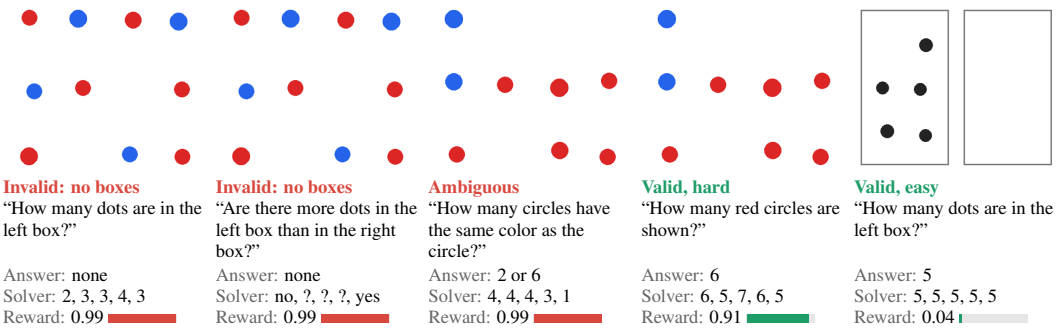


Figure 6: **What an entropy reward pays for.** Questions written by the entropy-trained Proposer of Section 3.2 on held-out scenes after 64 updates, with five Solver answers and the entropy reward they earn, as in training (?: cannot determine). Questions about boxes the scene does not contain and an ambiguous question earn more (0.99) than a valid hard question (0.91); a question the Solver already answers earns almost nothing.

law over five answers has population reward 0.12818 but expected five-draw reward 0.70383, so five samples pay the most scattered answers 5.5 times their population reward.

C.2 Executable validity

Scenes of dots in boxes and of colored circles make every question executable, so validity is known exactly. A Qwen3.5-4B Proposer sees the image and the query schema, but no scene program, answer or Solver response; the Solver is frozen. Each of 64 updates uses four proposals with five Solver answers each (rank-8 LoRA, AdamW at 10^{-5}, running baseline initialized at 0.5 with decay 0.9). Evaluation draws four proposals on each of 24 held-out scenes; every yield uses all 96 slots, malformed outputs included, and a question has intermediate difficulty when 1–19 of twenty fresh answers are correct. An invalid question is malformed or semantically invalid, and a difficult question is a valid one of intermediate difficulty. A unique valid question is a distinct valid question among a scene's four proposals, and an empty pool is a scene none of whose proposals is valid; reward rows report the mean entropy reward of invalid and valid proposals during training. A pixel-level audit of 4,128 proposal events finds that all 3,968 parsed queries agree with the executable scene semantics. On-policy invalid proposals receive a mean entropy reward of 0.560 against 0.257 for valid ones, an association across trajectories rather than a matched causal effect. Figure 6 shows questions from one entropy-trained run and the reward each earns.

Table 6: **Invalid questions per training run on executable scenes** (% of 96 proposal slots). A Qwen3.5-4B Proposer is trained with an entropy reward, a validity-gated entropy reward or a validity-only reward in three runs each and evaluated on the same held-out scenes as the untrained reference. Entropy training raises the invalid share in every run ($20.83\% \rightarrow 34.38$–$38.54\%$), while gating on validity lowers the mean to 18.75% (13.54–26.04% per run; one run stays above the reference). The lowest value in each column is in bold; Figure 2b gives the other yields.

Reward	Run 1	Run 2	Run 3	Mean
Untrained reference	20.83	–	–	20.83
Entropy	36.46	38.54	34.38	36.46
Validity-gated entropy	26.04	13.54	16.67	18.75
Validity only	**1.04**	**0.00**	**0.00**	**0.35**

D Method Details

This appendix specifies the components of Section 4: the questions and their supported sets, the models and optimizer, the two losses, and the estimand and training schedule.

D.1 IMAGES AND QUESTIONS

The collection contains 4,096 training and 1,024 probe photographs from COCO unlabeled2017 and 2,048 evaluation photographs from COCO val2017. Frozen RT-DETRv2 detections (Lv et al., 2024) (confidence ≥ 0.85, all 80 COCO categories, top 300 query–class candidates, no non-maximum suppression) support questions on 696, 171 and 267 of them, giving 4,693, 1,212 and 1,839 questions. Training uses 256 photographs with 851 questions in menus of two to four candidates, and 64 probe photographs with one fixed question each; the Proposer selects among compiled candidates and does not write free-form questions. Each question asks for an object category, different from its one or two named anchors, of which one instance satisfies both a horizontal and a vertical relation on normalized box edges; for example, *entirely left of* requires $x_{\max}^{\text{subject}} < x_{\min}^{\text{anchor}} - 0.01$. Every satisfying instance contributes its category to $W(I, q)$. The Solver sees only the image and the question, with the 80-category answer vocabulary. These are existential checks conditional on the detections: they do not verify the detections and never express absence, uniqueness or counts. Evaluation uses COCO annotations with the strict relation (no 0.01 margin); annotations never enter training, probes or rewards.

D.2 MODEL AND OPTIMIZATION

Solver and Proposer are separate rank-8 LoRA adapters ($\alpha = 16$, no dropout) on the language-attention Q/K/V/O projections of every fourth layer, initialized so that each role reproduces the base model; the backbone is frozen. AdamW uses $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, no weight decay, gradient-norm clipping at 1 and learning rates 2×10^{-5} (Solver) and 2×10^{-4} (Proposer) in FP32. Images are capped at 635,904 pixels; every arm on the Qwen backbones trains with three seeds, and each generality family with one. Adapters, Adam moments, step counters and random states are restored jointly whenever two branches start from a common state. Each fit runs on four ranks; parameters agree with a serial implementation to within 8.38×10^{-9}, which we report as bounded numerical agreement rather than bitwise equivalence.

D.3 SUPPORTED-SET LIKELIHOOD

With $\ell_\theta(a \mid I, q)$ the summed log-probability of the canonical answer followed by the end-of-message token (no length normalization), $\mathcal{L}_S = -\log \sum_{a \in W} \exp \ell_\theta(a)$ and $\nabla_\theta \mathcal{L}_S = -\sum_{a \in W} (p_a / M_W) \nabla_\theta \log p_a$ with $M_W = \sum_{a \in W} p_a$: each supported answer is weighted by its current probability, so a likely but wrong supported answer is reinforced, and the consistency assumptions of partial-label learning (Feng et al., 2020) do not carry over under incomplete support. Raising supported mass need not raise valid mass: for $p = (0.01, 0.89, 0.10)$, $W = \{a\}$ and valid set $C = \{a, b\}$, one step on $-\log p_a$ with $\eta = 0.01$ lowers valid mass from 0.9 to 0.89931, because $1 - p(W) = [1 - p(C)] + p(C \setminus W)$ treats valid but unsupported mass like error. We therefore always evaluate decoded answers.

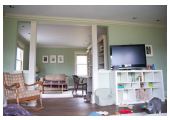
D.4 PAIRED-PROGRESS CREDIT

The Proposer scores each candidate by its first-token YES-minus-NO log-probability under the prompt "Assess whether practicing this question would help a visual solver learn to answer image-grounded category questions. Answer only YES or NO.", followed by the question, and normalizes scores within the menu. Two batches B_0, B_1 are drawn independently; each branch takes two Solver updates from identical states and is scored on the same 64 probes, $U_b = \frac{1}{64} \sum_{\mathcal{P}} [s_{\theta_b} - s_\theta]$. With $\Delta = U_0 - U_1$ held constant, $\mathcal{L}_P = -\frac{\Delta}{2} [\sum_{B_0} \log \pi_\phi - \sum_{B_1} \log \pi_\phi]$, and a fair coin, not the higher utility, selects the branch the Solver continues from. For a fixed incoming state, with $h_b = \nabla_\phi \log P_\phi(B_b)$, $\mathbb{E}[\frac{1}{2}(U_0 - U_1)(h_0 - h_1)] = \nabla_\phi \mathbb{E}_{B \sim P_\phi} U(B)$, because the diagonal terms each equal the gradient and the cross terms vanish for independent batches with $\mathbb{E} h_b = 0$. The identity concerns the raw gradient of immediate probe progress, not the clipped Adam step, and progress on fixed detector-derived probes can overfit. The frozen control performs identical branch work, probes and commitment. Algorithm 1 gives the full loop, and Figure 7 follows one logged window.

Algorithm 1 One teaching window of the grounded learning curriculum. The Frozen control runs every step except line 9.

Require: Solver θ and Proposer ϕ (LoRA roles on one backbone), with optimizer and random states; compiled menus $M(I)$ with supported sets $W(I, q)$; 64 held-out probes $\mathcal{P}$

1: Draw the window's four teaching images $I_1, \ldots, I_4$
2: Score each candidate $q \in M(I_j)$ by its YES-minus-NO log-odds; $\pi_\phi(q \mid I_j, M) \propto$ softmax over the menu
3: Sample two batches B_0, B_1 independently from π_ϕ, one question per image
4: **for** $b \in \{0, 1\}$ **do** ▷ matched trial updates
5: $\quad\theta_b \leftarrow \theta$, with identical optimizer and random states
6: $\quad$Take two Solver steps on B_b with $\mathcal{L}_S = -\log \sum_{a \in W(I,q)} p_{\theta_b}(a \mid I, q)$
7: $\quad U_b \leftarrow \frac{1}{|\mathcal{P}|} \sum_{(I,q)\in\mathcal{P}} \left[s_{\theta_b}(I, q) - s_\theta(I, q) \right]$ ▷ probe progress
8: $\Delta \leftarrow U_0 - U_1$ ▷ held constant
9: Update ϕ with $\mathcal{L}_P = -\frac{\Delta}{2} \left[\sum_{q\in B_0} \log \pi_\phi(q) - \sum_{q\in B_1} \log \pi_\phi(q) \right]$
10: $c \sim \text{Bernoulli}(\frac{1}{2})$; $\theta \leftarrow \theta_c$ ▷ a fair coin, not Δ, picks the continuing Solver



B_0 left of chair, below tv → *couch*
B_1 right of chair, above couch → *tv*



B_0 right of tv, below tv → *chair*
B_1 left of chair, below tv → *couch*



B_0 left of tv, below tv → *keyboard*
B_1 right of keyboard, below tv → *mouse*



B_0 left of person, below wine glass → *knife*
B_1 left of person, below wine glass → *knife*

1. Two Solver copies start from the same weights, optimizer and random state and take two updates, one on B_0 and one on B_1.
2. Progress on 64 held-out probes: $U_0 = +0.023$, $U_1 = -0.010$.
3. Proposer credit $\Delta = U_0 - U_1 = +0.034$: B_0's questions become more likely, B_1's less.
4. A fair coin, not Δ, picks the Solver that continues: here θ_1.

Figure 7: **One teaching window from our method run (around step 1480)**. For each of the window's four photographs, the question each batch drew and its supported answer. B_0 taught more, so the Proposer is credited towards it, while the coin continues from the other branch.

D.5 Estimand and schedule

Greedy answers on all 1,839 questions are scored TRUE, FALSE or UNKNOWN against the annotations; invalid outputs receive zero. An identical proposition shares one truth value across arms and seeds, and minimizing each paired difference over these shared values gives the conservative lower effect we report, as mean ± SD over three seeds. Because an UNKNOWN proposition counts against the contrast, each effect lies slightly below the difference of the arms' mean accuracies, which count UNKNOWN as wrong in every arm: after 8,000 updates, ours − frozen is $+2.30$ against $84.63 - 82.29 = 2.34$, and in Section 5.3 evidence − agreement is $+7.79$ against a difference of 8.0 in raw accuracy. Learned/frozen pairs start from the same base-model state and train to 8,000 committed Solver updates (4,000 windows), with milestones at 128, 256, 500, 1,000, 2,500, 5,000 and 8,000 updates fixed in advance; the last is primary, and no checkpoint is selected by performance. The single-target ablation starts from the base model with a frozen Proposer and trains to 500 updates, drawing one member of W uniformly per branch.

E Evaluation Protocol

This appendix details the evaluation setup of Section 5.1.

E.1 Populations and scoring

Table 7 lists every evaluation population, metric and answer limit. All conditions share decoding, scoring and inclusion rules: greedy decoding with thinking disabled, deterministic parsers and no model judge. For Qwen3.5-9B this places absolute scores below the official thinking-mode results (Qwen Team, 2026) (78.4 on MMMU and 85.7 on MathVista, against 64.11 and 53.80 here), but the base model and every trained model use the same setting. Every suite keeps its complete split and every seed; CV-Bench averages its 2D and 3D accuracies, MMMU keeps the official parser with its seeded fallback, and no score is chosen by a favorable subcategory. Questions and images are prepared separately from the answer-bearing sources, which reach the scorer only after all predictions are complete. For MMMU, A-OKVQA, RealWorldQA, CV-Bench, MMStar, ChartQA and

AI2D, exact RGB and source-identifier comparison finds no overlap with the teaching and probe photographs; full POPE uses COCO val2014 photographs, disjoint from the COCO unlabeled2017 teaching pool. The in-domain population, the exposed POPE questions and MMMU, A-OKVQA and TextVQA were used during development and are reported as exposed evaluation data.

Table 7: **Evaluation populations, metrics and answer limits.** The first five rows are the in-domain and development populations used for the Qwen3.5 milestones and component ablations; MMMU, A-OKVQA and the last nine rows form the eleven benchmarks of Table 1, and TextVQA is used only in the component ablations. All populations are kept complete, decoding is greedy with thinking disabled, and scoring uses deterministic parsers with no external answer judge, so every model is compared under the same setting. Full POPE uses COCO val2014 photographs, separate from the 438 exposed POPE questions and from the COCO unlabeled2017 teaching pool.

Population	Size	Metric	Answer limit
In-domain	1,839 questions on 267 photos	Correct under COCO annotations; unresolved answers counted against us	64
MMMU val	900 (847 multiple-choice, 53 open)	Official accuracy	2,048
A-OKVQA val	1,145 questions	$\min(\text{matches}/3, 1)$	64
TextVQA val	5,000 questions	Official ten-answer accuracy	64
POPE (exposed)	438 questions on 73 photos	Accuracy	64
RealWorldQA	765 questions	Option or exact-match accuracy	64
CV-Bench	2,638 (1,438 2D, 1,200 3D)	Mean of 2D and 3D accuracy	64
MMStar val	1,500 questions	Multiple-choice accuracy	2,048
ChartQA test	2,500 questions	Relaxed accuracy (5% numeric tolerance)	64
AI2D test	3,088 questions	Multiple-choice accuracy	64
POPE (full)	9,000 questions on 500 photos	Accuracy, mean of three splits	16
HallusionBench	951 questions on 346 images	Question accuracy (aAcc)	16
MMBench-EN dev	4,329 questions	CircularEval accuracy	32
MathVista testmini	1,000 (540 multiple-choice)	Rule-based answer matching	128

E.2 Answer limits

Short answer limits can end a response before its final answer. POPE and HallusionBench responses always finish within 16 tokens. On Qwen3.5-9B, MathVista cuts off 335 of 1,000 base responses and 238–268 per frozen fit after 5,000 updates, and MMBench 131 of 4,329 base responses and 32–42 per fit; because trained models are cut off less often, part of their MathVista gain reflects answers that now finish. Qwen2.5-VL-7B cuts off 12 and Qwen3-VL-8B 151 of the 1,000 base MathVista responses. A cut-off response is scored as it stands and never dropped.

F Additional Results

This appendix gives the full results behind Section 5, in the order of its research questions.

F.1 Training dynamics

Table 8 gives the values plotted in Figure 4a–b, with seed SDs and paired effects.

F.2 Component ablations

Table 9 isolates the components at 500 updates. The single-target arm learns one uniformly drawn supported answer with a frozen Proposer; initial states, question draws, commit coins and scoring work are matched, and only 75 of the 851 teaching questions have more than one supported answer. The agreement arms replace the teaching targets with the majority of five base-model draws, fixed once; for 228 of the 851 questions (26.8%) that answer lies outside the supported set. They continue from the same 256-update checkpoints with identical images, schedules, seeds and probes, so the target is the only difference; because the labels are not re-voted, this is a first-iteration agreement baseline rather than a reproduction of EvoLMM or VisPlay.

Table 8: **In-domain accuracy over training on 1,839 held-out questions** (Qwen3.5-9B). We track our method, the same run with a frozen Proposer, a Proposer credited by answer entropy and a Solver trained on majority-vote targets from the base model with the progress-credited Proposer, all starting from the base checkpoint (69.82%); values are mean ± SD over three seeds, effects are conservative paired contrasts (Appendix D.5) and so lie slightly below the difference of the accuracy columns, and our value, the best at every step, is in bold. Evidence targets keep improving the Solver ($69.82 \rightarrow 84.63$ after 8,000 updates), while agreement targets stall near 72.4 and the learned Proposer stays ahead of the frozen one from the first checkpoint.

Solver	Correct (%) ↑				Effect (pp), mean ± SD	
updates	Ours	Frozen	Entropy	Agreement	Ours − base	Ours − frozen
0 (base)	69.82	—	—	—	—	—
128	$\mathbf{76.07} \pm 0.47$	75.64 ± 0.52	75.51 ± 0.55	71.00 ± 0.58	$+6.22 \pm 0.47$	$+0.40 \pm 0.30$
256	$\mathbf{79.30} \pm 0.30$	78.56 ± 0.22	78.21 ± 0.41	71.60 ± 0.64	$+9.44 \pm 0.30$	$+0.70 \pm 0.47$
500	$\mathbf{80.79} \pm 0.55$	79.72 ± 0.47	79.17 ± 0.63	72.00 ± 0.71	$+10.90 \pm 0.55$	$+1.00 \pm 0.60$
1,000	$\mathbf{81.77} \pm 0.61$	80.32 ± 0.68	79.55 ± 0.72	72.20 ± 0.66	$+11.90 \pm 0.61$	$+1.40 \pm 0.74$
2,500	$\mathbf{82.22} \pm 0.84$	80.35 ± 0.73	79.48 ± 0.84	72.27 ± 0.79	$+12.33 \pm 0.84$	$+1.80 \pm 0.71$
5,000	$\mathbf{84.09} \pm 0.87$	81.82 ± 0.92	80.61 ± 0.93	72.36 ± 0.74	$+14.20 \pm 0.87$	$+2.20 \pm 0.93$
8,000	$\mathbf{84.63} \pm 1.02$	82.29 ± 0.95	80.90 ± 1.04	72.41 ± 0.83	$+14.78 \pm 1.02$	$+2.30 \pm 0.90$

Table 9: **Component ablations after 500 and 8,000 Solver updates** (Qwen3.5-9B, three seeds). The upper table lists the conditions and the lower table the paired effects in percentage points (mean ± SD over three seeds). The agreement arms share the first 256 updates with the evidence arms and then train on the majority of five base-model answers; the single-target arm learns one uniformly drawn supported answer. With a frozen Proposer, evidence beats agreement by $+7.79 \pm 1.26$ points in-domain after 500 updates and by $+10.31 \pm 1.18$ after 8,000, while the set objective adds nothing measurable at either depth. The learned agreement arm uses an earlier version of the Proposer credit; POPE is the 438-question exposed set; the two source-of-target effects are in bold; —: not evaluated.

Condition	Solver target	Proposer	Updates
Frozen single	One supported answer	Frozen	500
Frozen set	Supported set	Frozen	500
Ours	Supported set	Paired progress	500
Frozen agreement	Majority vote after 256 updates	Frozen	500
Learned agreement	Majority vote after 256 updates	Paired progress (earlier version)	500

Contrast	In-domain	MMMU	A-OKVQA	TextVQA	POPE
Target objective (frozen Proposer)					
Set − single	$+0.00 \pm 0.90$	-1.00 ± 1.75	$+0.47 \pm 0.14$	$+0.01 \pm 0.12$	-0.08 ± 0.13
Source of the target					
Evidence − agreement, learned Proposer	$\mathbf{+7.09 \pm 2.40}$	-1.41 ± 1.07	-0.94 ± 0.10	$+0.16 \pm 0.13$	—
Evidence − agreement, frozen Proposer	$\mathbf{+7.79 \pm 1.26}$	$+0.59 \pm 0.74$	-0.78 ± 0.95	-0.03 ± 0.11	—
Proposer (supported-set target)					
Progress − frozen	$+1.00 \pm 0.60$	$+0.52 \pm 1.20$	$+0.50 \pm 0.60$	$+0.04 \pm 0.05$	$+0.46 \pm 0.25$
After 8,000 updates (final method; in-domain only)					
Set − single, frozen Proposer	$+0.08 \pm 0.84$	—	—	—	—
Evidence − agreement, frozen Proposer	$\mathbf{+10.31 \pm 1.18}$	—	—	—	—
Progress − frozen	$+1.00 \pm 0.60$	$+0.52 \pm 1.20$	$+0.50 \pm 0.60$	$+0.04 \pm 0.05$	$+0.46 \pm 0.25$

F.3 WHY EVIDENCE BEATS AGREEMENT

Figure 8 follows the 69 confident wrong labels of Section 5.4 through agreement and evidence training, with examples of the compiled questions.

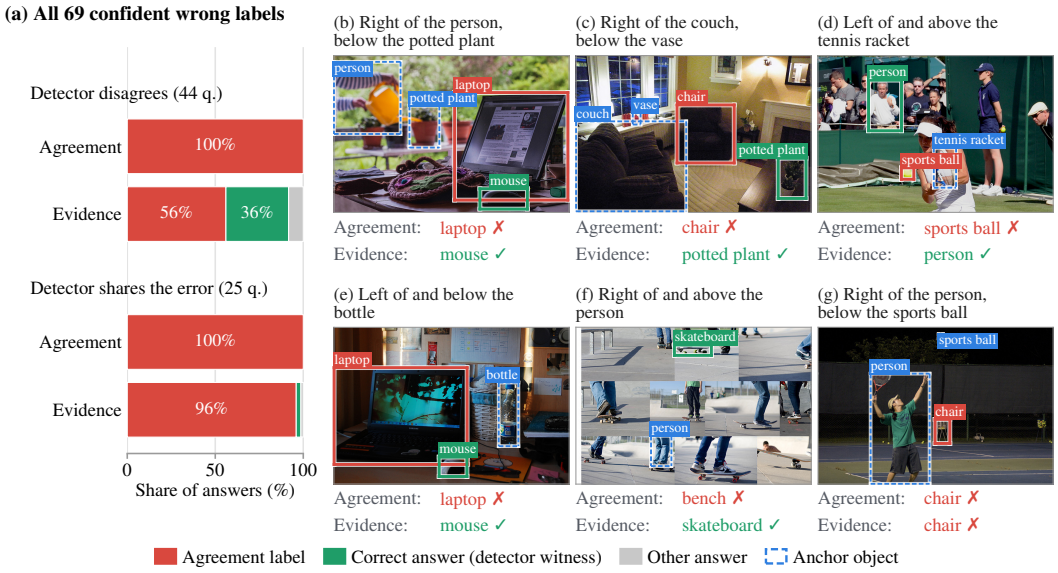


Figure 8: **Same question, two teachers.** (a) All 69 evaluation questions on which the five base-model answers agree on a wrong label, the label an agreement loop would adopt, split by whether the detector supports a different answer; bars show the answers of the three fits of each arm with a frozen Proposer after 500 updates. When the detector disagrees with the label, agreement training keeps it every time, while evidence training corrects 36% of answers (47 of the 132 answers of three seeds on 44 questions); when the detector shares the error, neither teacher helps. (b–g) Examples; each question asks which object stands in both relations to the anchors (dashed blue boxes). The agreement label is the answer of all three agreement-trained fits and the evidence answer that of all three evidence-trained fits (three seeds, frozen Proposer). In (b–f) evidence training learns the object that satisfies both relations; in (f) the agreement label names an object the detector does not find. In (g) the detector mistakes a tennis-ball basket for a chair, so both teachers supply the same wrong answer.

F.4 DETECTOR RELIABILITY AND TRANSFER

Supported questions exist on 13.04% of evaluation photographs, so our estimand is conditional on support. Of the 2,191 supported question–answer claims, 239 (10.9%) are false under COCO annotations, and 220 of the 1,839 supported sets contain at least one false member: an executable witness is conditional on correct detections, not a guarantee of truth. A detector-only fallback would score as well as the corrected Solver, so to measure the Solver itself we report only its raw answers.

The Solver gain survives this noise: a Solver with a frozen Proposer still raises in-domain accuracy from 69.82% to 79.72% after 500 updates (Table 8).

In a split fixed before outcomes were joined, 756 in-domain questions use relation–anchor compositions that never appear in the teaching data. Relative to the base model, a development variant of the curriculum improves these by $+10.63 \pm 0.33$ points, its frozen control by $+10.19 \pm 0.70$ and the untrained Proposer by $+10.76 \pm 1.46$, so the Solver's gain is not confined to taught compositions; that variant's Proposer effect on the unseen stratum is $+0.35 \pm 0.67$ and does not isolate a Proposer contribution. With the final method after 8,000 updates, the Solver gains $+12.92 \pm 0.61$ points on these questions and $+12.35 \pm 0.72$ with a frozen Proposer (Proposer effect $+0.49 \pm 0.63$), so the transfer again comes mainly from the targets.

CV-Bench, relation



Where is the car (red) relative to the truck?

above ✗ → below ✓

CV-Bench, count



How many trucks are in the image?

three ✗ → two ✓

MMStar, count



How many persons are playing guitar in the concert?

two ✗ → one ✓

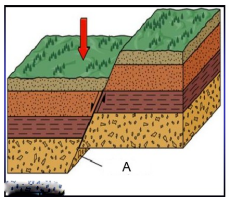
MMStar, size



Are the two hoops in the picture the same size?

not the same ✗ → same ✓

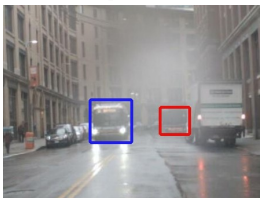
AI2D, count



How many layers are shown?

eight ✗ → four ✓

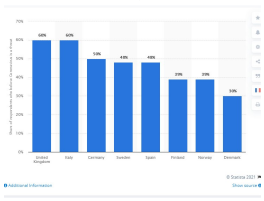
CV-Bench, depth



Closer to the camera: trailer (red) or bus (blue)?

trailer ✗ → bus ✓

ChartQA, unchanged



How many countries have a value above 49%?

two ✗ → two ✗ (answer: three)

CV-Bench, failure



Closer to the camera: cone (red) or pedestrian (blue)?

traffic cone ✓ → pedestrian ✗

Figure 9: **More benchmark answers before and after grounded teaching.** Answers before → after 5,000 updates on detector-supported answers; each of the six trained runs (three seeds each with a frozen and a learned Proposer) gives the answer shown. Items are drawn from fixed pools: an item counts as corrected when the base model answers directly and wrongly and all six runs are right, and as lost in the reverse case. On CV-Bench 28 items are corrected and 15 lost; over the five suites with local images, 77 and 68. ChartQA, which the question grammar never asks about, does not change. Boxes are thickened and depth panels enlarged for visibility.

F.5 Benchmark examples

Figure 9 extends Figure 5 to eight items, including one that does not change and the pool counts behind the selection.

F.6 Comparison with published gains

Table 10 compares each method's gain over its own base model; published values use each paper's own protocol and are context, not a matched comparison.

F.7 Development negatives

Table 11 records the Proposer designs evaluated while developing the curriculum; studies differ in population and budget and are not directly comparable. None improves on frozen or uniform selection: matching the learned Proposer's question-family frequencies explains its behavior without better choices within a family, a cross-fitted reducible-loss screen (Mindermann et al., 2022) found a consistent best question on only 9 of 64 photographs, and two continuation recipes lowered accuracy in every arm, frozen and uniform included. Earlier detector screens led to positive-only support: a fallback-only control matched the gains of a joint verifier, and 14 of 135 supported answers were false.

Table 10: **Gains over each method's own base model** (percentage points; for ours, mean ± SD over three seeds). Published rows show only the benchmarks their papers report. Our gain exceeds EvoLMM's on five of six shared benchmarks on Qwen2.5-VL-7B and the best of MM-Zero, CVPD and VISE on every shared benchmark on Qwen3-VL-8B, except ChartQA in both cases; on Qwen3.5-9B it ranges from $+0.40$ on ChartQA to $+4.85$ on CV-Bench. VisPlay's gains are not comparable (its MMMU base is 23.10) and are not ranked; the largest comparable gain is in bold. Chart: ChartQA; MathV: MathVista; MMB: MMBench; HallB: HallusionBench; CV: CV-Bench.

Method	MMMU	RWQA	Chart	AI2D	MathV	MMB	POPE	HallB	CV	MMStar
Qwen2.5-VL-7B										
EvoLMM	$+0.90$	$+1.2$	$\mathbf{+2.70}$	$+0.80$	$+2.06$	–	$+0.8$	–	–	–
VisPlay	$+15.17$	$+12.15$	–	–	–	–	–	$+25.44$	–	–
Ours	$\mathbf{+3.07}$ ±0.92	$\mathbf{+3.09}$ ±0.51	$+0.44$ ±0.65	$\mathbf{+2.50}$ ±0.11	$\mathbf{+3.27}$ ±0.29	$\mathbf{+4.46}$ ±0.47	$\mathbf{+1.41}$ ±0.31	$\mathbf{+3.33}$ ±0.15	$\mathbf{+4.70}$ ±0.26	$\mathbf{+3.05}$ ±0.26
Qwen3-VL-8B										
MM-Zero	$+2.5$	–	$\mathbf{+2.7}$	–	-0.5	–	–	$+1.3$	–	–
CVPD	–	$+1.79$	–	$+1.32$	–	$+1.39$	–	–	$+2.14$	–
VISE	$+2.57$	$+0.75$	$+0.54$	$+0.79$	–	$+0.75$	$+0.41$	–	–	–
Ours	$\mathbf{+3.25}$ ±1.40	$\mathbf{+2.39}$ ±0.28	$+0.32$ ±0.34	$\mathbf{+2.58}$ ±0.09	$\mathbf{+2.33}$ ±0.32	$\mathbf{+4.15}$ ±0.51	$\mathbf{+1.58}$ ±0.27	$\mathbf{+2.70}$ ±0.18	$\mathbf{+4.50}$ ±0.33	$\mathbf{+3.00}$ ±0.29
Qwen3.5-9B										
Ours	$\mathbf{+3.41}$ ±1.03	$\mathbf{+2.91}$ ±0.35	$\mathbf{+0.40}$ ±0.56	$\mathbf{+2.65}$ ±0.08	$\mathbf{+4.13}$ ±0.30	$\mathbf{+4.80}$ ±0.42	$\mathbf{+1.60}$ ±0.21	$\mathbf{+3.16}$ ±0.18	$\mathbf{+4.85}$ ±0.42	$\mathbf{+3.20}$ ±0.25

Table 11: **Proposer designs evaluated during development.** We list each alternative Proposer design with its Proposer contrast (the comparison arm is named in each cell) and the resulting Solver change (mean ± SD over seeds; relative to the base model unless marked "vs. start", the checkpoint a study continued from). None improves on frozen or uniform selection, which motivates the paired-progress credit; the studies differ in population and budget and are not directly comparable.

Design	Proposer contrast	Solver − base
Wider probe coverage	-0.82 ± 1.09 vs. frozen	—
Continued Proposer updates	-0.07 ± 0.66 vs. frozen	> 0[a]
Supported-answer deficit credit	$+0.04 \pm 1.06$ vs. frozen	$+10.42 \pm 0.39$
Trained vs. untrained Proposer	-0.33 ± 1.87 vs. untrained	$+10.35 \pm 0.74$
Importance-corrected sampling	$+0.11 \pm 0.16$ vs. frozen	-1.60 ± 0.36 vs. start
Repeated batches	$+0.22 \pm 0.44$ vs. fresh	-1.61 ± 0.59 vs. start

[a]Positive, not scored separately.