EDIT, DON'T IMITATE: MINIMAL-EDIT ON-POLICY DISTILLATION FOR MULTIMODAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual mathematical reasoning can fail before reasoning begins: when a vision-language model (VLM) misreads a diagram, subsequent reasoning proceeds from false premises. This perceptual bottleneck is especially pronounced in smaller models, while accurate textual descriptions can substantially narrow the gap to larger models. Because obtaining such descriptions at inference requires additional annotation or model calls, we propose minimal-edit corrective on-policy distillation. Using reference figure descriptions and verified reference solutions as training-only information, a teacher scores incorrect student rollouts under a minimal-edit objective, and only teacher-disfavored tokens contribute to the corrective loss. Correct rollouts receive task-reward reinforcement learning. This preserves on-policy supervision while avoiding full-response imitation. Across model scales, the method requires no descriptions at test time and meets or exceeds the corresponding base model supplied with a reference figure description at inference. It also improves perception-oriented performance, particularly low-level geometric perception, and transfers to external mathematics and perception benchmarks. Because the minimal-edit teacher evaluates corrections rather than generating its own solution, the method remains effective with a frozen same-size teacher, although its gains are smaller than with a larger teacher on most metrics.

1 INTRODUCTION

Visual mathematics benchmarks are commonly treated as tests of reasoning, yet many failures occur before reasoning begins: when a vision-language model (VLM) misreads the diagram, even valid subsequent reasoning proceeds from false premises. This perceptual bottleneck is especially pronounced in smaller models. Giving a 2B VLM a faithful textual description of the diagram raises its Geometry3K (Lu et al., 2021) pass@1 from 0.32 to 0.46, approaching the 0.49 attained by an 8B model roughly four times its size (Figure 2). Although reasoning capacity still matters, this result identifies perception as a major source of smaller models' disadvantage. Obtaining an accurate description at inference, however, requires additional annotation, prompting, or model calls. We therefore ask whether this privileged perceptual signal can be used during training to improve a model that ultimately solves directly from the image and question.

To internalize this training-only perceptual signal, the model needs feedback on how its own solutions go wrong. Reinforcement learning with verifiable final-answer rewards provides only outcome-level feedback, which does not identify the source of an incorrect rollout (Shao et al., 2024). Process reward models could provide finer-grained signals, but require a separately trained evaluator with process-level supervision (Lightman et al., 2024). On-policy distillation instead treats the model being trained as a student and lets a teacher score the student's own token sequence, providing dense supervision at the states the student visits (Agarwal et al., 2024). A natural first attempt is privileged on-policy distillation: a frozen teacher sees the reference figure description while the student receives only the image and question.

Standard on-policy distillation (OPD), however, applies the divergence at every token, including positions where the teacher's preference can reflect phrasing or boilerplate rather than a substantive error (Wu et al., 2026). We therefore propose minimal-edit corrective on-policy distillation. For each incorrect rollout, the minimal-edit teacher receives the reference figure description, a verified reference solution, and the student's response. The conditioning prompt specifies a minimal-edit

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

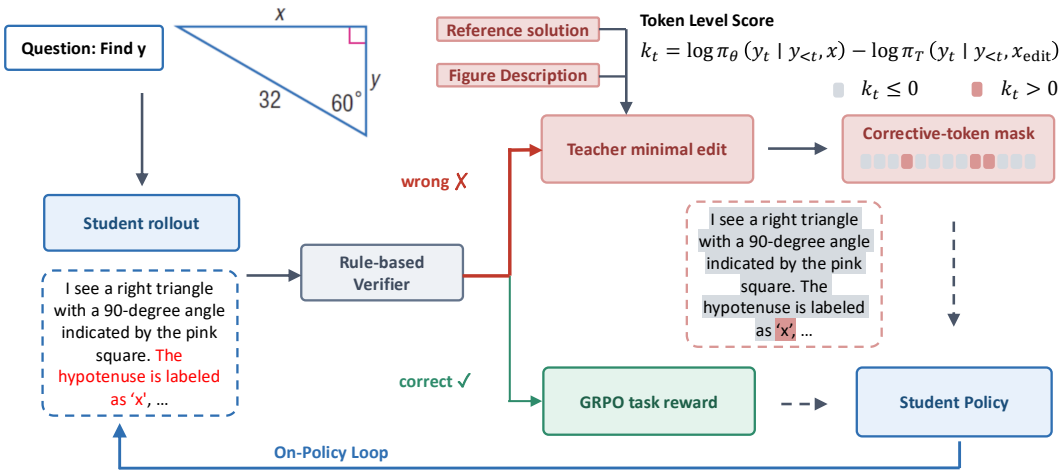


Figure 1: Overview of minimal-edit corrective on-policy distillation. A rule-based verifier routes correct rollouts to task reward; incorrect rollouts are scored by the minimal-edit teacher, and only teacher-disfavored tokens are used to update the student. Dashed arrows indicate loss calculation.

objective: correct all errors while preserving the portions of the response that are already correct. The teacher does not generate a revised solution. Instead, it scores the student's existing token sequence. Only positions where the teacher assigns the emitted token lower probability than the student contribute to the distillation loss and correct rollouts instead receive task-reward RL. Figure 1 summarizes this per-rollout routing, teacher-scoring, and token selection procedure.

Scoring the student's response under a minimal-edit objective, rather than generating an independent solution, makes the minimal-edit teacher a response-specific critic. This less demanding role reduces the capability expected of the teacher. Experiments at 2B and 8B show that, in the minimal-edit setting, a frozen copy of the student remains an effective teacher. Without a test-time description, minimal-edit distillation meets or exceeds the corresponding base model evaluated with a reference figure description at inference (Section 3.1), improves performance on perception-oriented evaluations, particularly low-level geometric perception, and generalizes to external, out-of-domain multimodal mathematics problems.

In summary, we make three contributions. First, we introduce minimal-edit corrective on-policy distillation, which uses reference figure descriptions and verified reference solutions to select teacher-disfavored positions in incorrect rollouts while reinforcing correct rollouts with task reward. Second, we show that response-specific correction enables effective distillation from a frozen same-size teacher. Third, we demonstrate that minimal-edit distillation improves reasoning without reference figure descriptions at inference, improves perception-oriented performance, and transfers to unseen visual-mathematics benchmarks.

2 Related Work

On-policy and corrective distillation. On-policy distillation is commonly optimized with a reverse KL related to KL-regularized reinforcement learning (Agarwal et al., 2024; Gu et al., 2024). Multimodal variants distill text-only teachers, steer the gradient toward visual grounding, normalize token losses, or allocate more weight to tokens affected by fine image detail (Bousselham et al., 2026; Yoon et al., 2026; Hao et al., 2026; Liu et al., 2026). Across modalities, however, token importance is nonuniform, raw disagreement need not be learnable, and cross-teacher disagreement can reflect style rather than task content (Xu et al., 2026; Wang et al., 2026b; Wu et al., 2026). Selective objectives consequently detect unrecoverable prefixes or use token advantages (Xin et al., 2026; Jia et al., 2026). Closest to our motivation, ROSD reflects on a rollout to locate its first erroneous span, then distills the suffix under a corrective idea (Zhao et al., 2026b); Correction2Mastery instead transfers a generated earliest-error correction through text and a scalar reward (Lyu et al., 2026). RLCSD contrasts correct- and incorrect-hint distributions to cancel privilege-induced style drift (Pan et al., 2026). Our minimal-edit teacher neither generates a replacement nor defines an er-

ror suffix: it scores the student's unchanged response under a minimal-edit instruction, and the sign of the sampled-token log-ratio selects disfavored tokens. This differs also from methods that train an explicit self-correction turn (Cheng et al., 2025; Ding & Zhang, 2025; Ding et al., 2026), whose reliability remains limited without external feedback (Huang et al., 2024). More broadly, work on model self-improvement formalizes the extent to which a model can provide useful supervision by evaluating its own generations as the generation–verification gap (Song et al., 2025).

Reward routing and privileged visual supervision. Methods combining task reward with distillation mix the signals at response level or weight them by group difficulty (Xu et al., 2025; Tan & Hong, 2026). SRPO instead routes correct samples to reward optimization and failed samples to self-distillation (Li et al., 2026). Our method shares this high-level partition but arrives at it from the minimal-edit objective: only incorrect rollouts require correction, while applying the task reward to correct rollouts avoids discarding useful samples. Our failed-rollout objective is further distinguished by minimal-edit conditioning and token-level sign selection. A separate line of work conditions teachers on privileged reference solutions, trajectories, or internalized contexts (Lopez-Paz et al., 2016; Zhao et al., 2026a; Penaloza et al., 2026; Ye et al., 2026). Vision-OPD uses evidence-centered crops (Yuan et al., 2026); PTD-PO constructs spatial and reasoning hints for failed rollouts (Xiang et al., 2026); ViCuR uses recoverable visual cues and a cue-recovery module (Tian et al., 2026); and ViGOS separately supervises explicit description and reasoning spans (Wang et al., 2026a). MaskingKD instead hides reasoning prefixes to force reliance on the image (Yu et al., 2026). These methods share the goal of avoiding shortcuts from privileged text: oracle-conditioned targets may be unreachable (Zhang, 2026), and when instance-specific privileged information is not recoverable from the student's input, the student may learn an aggregate of incompatible privileged-teacher behaviors rather than internalize the missing information (Zhu et al., 2026). Our reference figure description, by contrast, is recoverable from the image and is used only to condition a frozen teacher; the student requires no description, cue module, or structured perception stage at inference.

Perception in vision-language mathematical reasoning. Prior work identifies perception as a primary constraint for smaller models on diagram problems (Zhang et al., 2024b; Rahmanzadehgervi et al., 2024; Zhang et al., 2024a; Endo & Yeung-Levy, 2026). Perception-aware reinforcement learning derives training signals from image ablation (Wang et al., 2026c), consistency with visual annotations mined from correct traces (Xiao et al., 2026), rollouts on perturbed images (Liu et al., 2025), or a preliminary perception stage (Chen et al., 2025). Our signal instead comes from a teacher's textual reading of the diagram; we compare against released checkpoints from this line of work on a shared backbone (Section 4).

3 METHOD

3.1 HOW MUCH DO REFERENCE FIGURE DESCRIPTIONS IMPROVE VISUAL REASONING?

To identify what limits small VLMs on geometry problems, we investigate how much of their performance gap arises from diagram perception rather than downstream reasoning. We probe this distinction on Geometry3K (Lu et al., 2021) by evaluating Qwen3-VL models (Bai et al., 2025b) at 2B, 4B, and 8B in two settings: with the image and question alone, and with the same input augmented at inference by an offline reference figure description. We generate each description with Muse Spark (Meta AI, 2026), which receives the diagram, question, and ground-truth answer to check completeness. Its prompt prohibits revealing the answer, intermediate reasoning, or derived values, and we screen exact answer matches for leakage. Appendix E provides the full generation and screening protocol. Figure 2 shows substantial gains at every scale. With the description, the 2B model reaches 0.462 pass@1, matching the 4B model without it and approaching the 8B model's 0.491. The 4B model reaches 0.672, exceeding the 8B model without a description. A faithful textual reading of the diagram is therefore comparable to a two- to four-fold increase in parameter count in these comparisons. Our objective is to internalize the description's perceptual advantage so that the student approaches performance under that condition using only the image and question.

To formalize this transfer, under on-policy distillation the student samples a rollout and the teacher scores the same token sequence, so training remains on the student's distribution. Let π_θ denote the student, π_T the teacher, and x and x_T their respective inputs. For $y \sim \pi_\theta(\cdot \mid x)$, the reverse-KL objective is

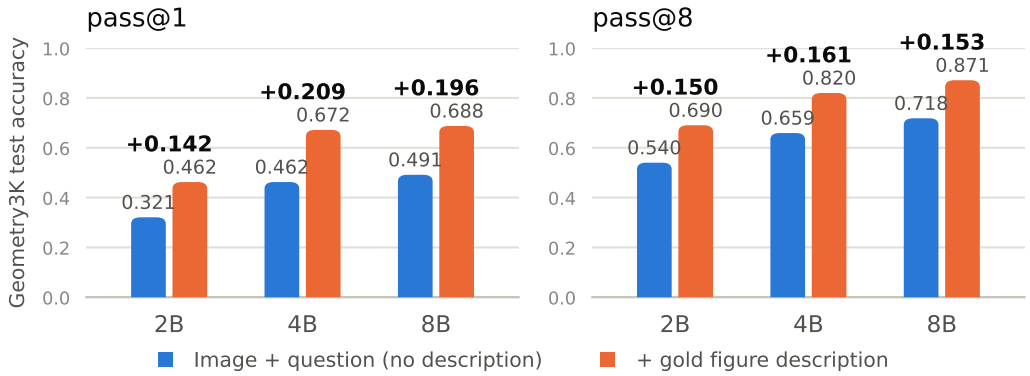


Figure 2: Geometry3K performance with image-only input or an added reference figure description. Descriptions narrow the size gap at every scale, identifying perception as a major bottleneck.

$$\mathcal{L}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{y\sim\pi_\theta(\cdot|x)} \sum_{t=1}^{T} \big[\log \pi_\theta(y_t \mid y_{<t}, x) - \log \pi_T(y_t \mid y_{<t}, x_T)\big],$$

where each log-ratio is the single-sample k_1 estimator of the conditional reverse KL at token t (Agarwal et al., 2024), and their sum estimates $\mathrm{KL}(\pi_\theta \| \pi_T)$ over the rollout. Unlike a scalar rollout reward, this objective supplies a signal at each token, and its gradient coincides with KL-regularized reinforcement learning toward the teacher. Under naive OPD, the teacher receives the same image and question as the student and differs only in having a larger parameter count. To use information from the reference figure description during training, we additionally give the teacher a reference figure description d while withholding it from the student, so $x_T = (d, x)$. We refer to this asymmetric setup as privileged OPD. The student samples a rollout from the image and question alone, and the teacher scores that same rollout with the description. The per-token signal is the difference between the log probabilities assigned by the student and teacher to the same token. Because the student never receives the description during training or inference, the signal is transferred under the input conditions the student will face at test time. This privileged information is recoverable from the student's input: the description is a textual reading of the image rather than an external fact. The teacher therefore supplies a capability the student can in principle acquire from its own observation.

3.2 Minimal-edit corrective on-policy distillation

An incorrect rollout is rarely wrong throughout: its error is often confined to a few spans, such as a misread label, measure, or geometric relation. Applying the divergence to every token also trains on positions where the reasoning is already correct and teacher disagreement may reflect phrasing rather than correctness, diluting the response-specific corrective signal. For each training problem, let A denote a verified reference solution. Given an incorrect on-policy rollout y, the minimal-edit teacher receives the reference figure description, reference solution, and student rollout. It is instructed to correct all errors using the smallest possible edits while preserving the portions that are already correct and the student's style. This instruction defines the edit-conditioned input $x_{\mathrm{edit}} = (d, A, y)$, but we do not generate the corrected response. Instead, the teacher scores the student's existing sequence, yielding $\pi_T(y_t \mid y_{<t}, x_{\mathrm{edit}})$. Tokens compatible with a minimal correction should retain high probability, whereas conflicting tokens should receive lower probability. The procedure thus requires one teacher scoring pass and no online teacher generation. Let

$$k_t = \log \pi_\theta(y_t \mid y_{<t}, x) - \log \pi_T(y_t \mid y_{<t}, x_{\mathrm{edit}})$$

be the token-level k_1 estimate under edit conditioning, corresponding to the quantity summed in the OPD objective above. If $k_t \le 0$, the minimal-edit teacher assigns at least as much probability to the emitted token as the student does, consistent with preserving it. If $k_t > 0$, the teacher assigns less probability, which we interpret as evidence that the token conflicts with the minimal correction. We

Privileged Teacher

Internal Monologue:I see a right triangle here, with a 60-degree angle indicated by the pink square. The hypotenuse is labeled as 'x', and the side adjacent to the 60-degree angle is 32. The angle opposite the side labeled 'y' is 60 degrees. I need to find the length of side 'y'.

Since it's a right triangle with a 60-degree angle, the other acute angle must be 30 degrees because the angles in a triangle add up to 180 degrees. I can use the properties of a 30-60-90 triangle here. In a 30-60-90 triangle, the sides are in a specific ratio: the side opposite the 30-degree angle is half the hypotenuse, and the side opposite the 60-degree angle is half the hypotenuse times the square root of 3.

But wait, in this triangle, I don't know if the side opposite the 60-degree angle is the one labeled 'y'. Let me look again at the diagram. The angle at the bottom is 60 degrees. So the side opposite the 60-degree angle is the one labeled 'y'. The hypotenuse is 'x', and the side adjacent to the 60-degree angle is 32. Since it's a right triangle, I can use trigonometric ratios.

I can use the cosine function because the side adjacent to the angle (32) and the hypotenuse (x) are known. So, cos(60°) = adjacent/hypotenuse = 32/x. I know that cos(60°) is equal to 0.5. So, 0.5 = 32/x. Solving for x, I get x = 64. Now that I have the hypotenuse, I can use the sine function to find 'y'. Sin(60°) = opposite/hypotenuse = y/x. So, y = x * sin(60°) = 64 * sin(60°). Since sin(60°) is equal to √3/2, I get y = 64 * (√3/2) = 32√3. So the value of 'y' is 32√3.

Final Answer:\boxed{32\sqrt{3}}

Minimal Edit Teacher

Internal Monologue:I see a right triangle here, with a 60-degree angle indicated by the pink square. The hypotenuse is labeled as 'x', and the side adjacent to the 60-degree angle is 32. The angle opposite the side labeled 'y' is 60 degrees. I need to find the length of side 'y'.

Since it's a right triangle with a 60-degree angle, the other acute angle must be 30 degrees because the angles in a triangle add up to 180 degrees. I can use the properties of a 30-60-90 triangle here. In a 30-60-90 triangle, the sides are in a specific ratio: the side opposite the 30-degree angle is half the hypotenuse, and the side opposite the 60-degree angle is half the hypotenuse times the square root of 3.

But wait, in this triangle, I don't know if the side opposite the 60-degree angle is the one labeled 'y'. Let me look again at the diagram. The angle at the bottom is 60 degrees. So the side opposite the 60-degree angle is the one labeled 'y'. The hypotenuse is 'x', and the side adjacent to the 60-degree angle is 32. Since it's a right triangle, I can use trigonometric ratios.

I can use the cosine function because the side adjacent to the angle (32) and the hypotenuse (x) are known. So, cos(60°) = adjacent/hypotenuse = 32/x. I know that cos(60°) is equal to 0.5. So, 0.5 = 32/x. Solving for x, I get x = 64. Now that I have the hypotenuse, I can use the sine function to find 'y'. Sin(60°) = opposite/hypotenuse = y/x. So, y = x * sin(60°) = 64 * sin(60°). Since sin(60°) is equal to √3/2, I get y = 64 * (√3/2) = 32√3. So the value of 'y' is 32√3.

Final Answer:\boxed{32\sqrt{3}}

Figure 3: Token-level k_1 estimates for the incorrect response to the question in Figure 1, under privileged (top) and minimal-edit (bottom) conditioning. Red tokens ($k_t > 0$ are tokens that teacher disfavors relative to student) are retained and green tokens are discarded. Minimal-edit conditioning more precisely targets the leg–hypotenuse error, yielding a more interpretable corrective signal.

define the binary corrective mask $m_t = \mathbf{1}[k_t > 0]$ and its selected-token set $M = \{t : m_t = 1\}$; all other positions are discarded. Figure 3 illustrates the effect. The student misreads the hypotenuse of a right triangle as a leg. The conventional privileged teacher assigns a positive k_1 estimate to 224 of the 515 response tokens, including many in correct passages. Under minimal-edit conditioning, the mask instead selects a position in the phrase asserting the incorrect hypotenuse.

The minimal-edit teacher is applied only to incorrect rollouts, since correct rollouts require no correction. Instead of discarding these useful on-policy samples, we optimize correct rollouts with the GRPO loss. Computing advantages from only the correct rollouts would yield no relative reward signal. We therefore first assign every rollout a binary reward using the rule-based final-answer verifier and compute group-relative advantages over the full rollout group. We then route correct rollouts to the GRPO loss and incorrect rollouts to the minimal-edit distillation loss. The corrective mask retains roughly 2% of the tokens in incorrect rollouts throughout training (Figure 5). Averaging over all response tokens would therefore shrink the corrective term and make its magnitude depend on the number of selected positions. We instead normalize over the selected tokens:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{task}}(\theta) + \frac{\lambda}{|M|} \sum_{t \in M} \left[\log \pi_\theta(y_t \mid y_{<t}, x) - \log \pi_T(y_t \mid y_{<t}, x_{\text{edit}}) \right],$$

where $\mathcal{L}_{\text{task}}$ is the GRPO policy-gradient objective on correct rollouts and λ weights the corrective term. The computational overhead relative to standard OPD depends on the incorrect-rollout fraction and sequence length; it decreases as training produces more correct rollouts. Appendix B quantifies this tradeoff, and Appendix A provides the complete training pseudocode.

4 Experiments

Dataset construction We use the Geometry3K dataset, where each problem pairs a diagram with a question and one ground-truth answer. We use the reference figure descriptions introduced in Section 3.1 for every training and test diagram. In privileged and minimal-edit distillation, the student receives only the original image and question, while the teacher additionally receives the reference figure description. Minimal-edit training additionally requires one offline reference solution per training problem. We sample 32 solutions from the privileged teacher, conditioned on the description, figure, and question, and retain the first whose boxed final answer passes the rule-based

Table 1: Main results for the Qwen3-VL-2B student. Per column, the best result is ***bold*** and the second best is underlined; our minimal-edit rows are shaded.

	Geometry3K		MathVerse	Fig-desc	Geoperception	
Method	pass@1	pass@8	(ALL)	recall	pass@1	pass@4
Qwen3-VL-2B (base)	0.321	0.540	37.8	0.567	0.179	0.311
SFT (rejection sampling)	0.392	0.675	35.4	0.605	0.244	0.433
SFT (describe-then-solve)	0.475	0.791	37.9	0.670	0.362	0.580
SFT (describe-figure)	0.214	0.598	24.3	0.770	0.359	0.617
SFT (mix: rejection + describe)	0.406	0.688	33.5	**0.784**	0.289	0.473
GRPO (plain prompt)	0.447	0.625	42.1	0.552	0.279	0.420
GRPO (figure-desc prompt)	0.173	0.361	34.9	0.415	0.278	0.474
GRPO (mix: plain + figure-desc)	0.430	0.693	43.3	0.560	0.308	0.485
naive OPD (8B teacher)	0.404	0.628	41.5	0.655	0.310	0.464
privileged OPD (8B teacher)	0.487	0.707	40.0	0.742	0.320	0.462
minimal-edit (8B teacher, ours)	**0.554**	**0.794**	45.8	0.633	**0.455**	**0.635**
minimal-edit (2B self-teacher, ours)	0.498	0.756	**46.9**	0.588	0.438	0.634

checker. We discard problems for which none of the 32 candidates passes; for example, this filtering reduces the 2B training corpus from approximately 2.1K to 1.8K problems.

Training setup We evaluate three student backbones with larger teachers from the same model family: Qwen3-VL-2B with Qwen3-VL-8B and Qwen3-VL-8B with Qwen3-VL-32B (Bai et al., 2025b), and Qwen2.5-VL-7B with Qwen2.5-VL-32B (Bai et al., 2025a). The Qwen2.5 pair enables comparison with published checkpoints on a shared backbone. For the Qwen3-VL-2B and Qwen3-VL-8B students, we also run self-distillation. All larger and self-teachers are frozen and used only for scoring. We sample $n = 8$ on-policy rollouts per problem at a 2048-token response limit. The policy-gradient term uses clipping bounds of 0.2 in both directions. We train for 30 epochs at maximum with a learning rate of 1×10^{-6} and a optimization minibatch of 96 problems per step. The corrective weight is $\lambda = 0.01$. Training is implemented in verl (Sheng et al., 2025).

Baselines We compare four baseline families: (1) supervised fine-tuning on rejection-sampled correct traces, describe-then-solve traces, the figure-description objective, or a mixture of rejection-sampled and description traces; (2) GRPO (Shao et al., 2024) with the plain image-and-question prompt, the same prompt augmented with the reference figure description, or a mixture of both; (3) naive OPD (Agarwal et al., 2024), whose teacher receives the same image and question as the student; and (4) our privileged-OPD baseline, which additionally provides the figure reference figure description to the teacher. No method receives the description at inference, so any benefit must be internalized in the model's parameters. On Qwen2.5-VL-7B, we additionally compare with five published checkpoints trained by reinforcement learning or a perception-oriented objective: Perception-R1 (Xiao et al., 2026), NoisyRollout (Liu et al., 2025), PAPO (Wang et al., 2026c), DeepEyes (Zheng et al., 2026), and Vision-R1 (Huang et al., 2026b).

Evaluation setup. We evaluate mathematical reasoning on Geometry3K (Lu et al., 2021) and MathVerse (Zhang et al., 2024b), and perception with figure-description recall and Geoperception (Zhang et al., 2024a). For mathematical reasoning, we sample 32 rollouts per Geometry3K problem at temperature 0.8 with top-p 1.0 and a 2048-token response limit, and report pass@k. On MathVerse, we use greedy decoding with one response per problem, score responses with the benchmark's standard GPT-4o mini judge (OpenAI, 2024), and report the average over its five splits. For perception, figure-description recall measures the percentage of facts in the reference figure description recovered by a model-generated description. We independently sample 3 descriptions per test diagram at temperature 0.8 and report their mean recall; Appendix E provides the complete Gemini 3.1 Pro judge protocol (Google DeepMind, 2026), and Appendix F reports the corresponding standard deviations. Geoperception tests low-level geometric relationships organized into predicate categories of unequal size. We balance each evaluation subset by sampling the same number of examples from every category, decode 8 rollouts per problem at temperature 0.8 with top-p 1.0 and a

Table 2: Main results for the Qwen3-VL-8B student. Per column, the best result is ***bold*** and the second best is underlined; our minimal-edit rows are shaded.

	Geometry3K		MathVerse	Fig-desc	Geoperception	
Method	pass@1	pass@8	(ALL)	recall	pass@1	pass@4
Qwen3-VL-8B (base)	0.491	0.718	51.1	0.687	0.437	0.551
SFT (rejection sampling)	0.599	0.840	47.8	0.781	0.424	0.574
SFT (describe-then-solve)	0.691	**0.911**	43.5	0.798	0.500	**0.686**
SFT (describe-figure)	0.489	0.841	42.1	0.839	0.461	0.669
SFT (mix: rejection + describe)	0.624	0.849	50.9	0.846	0.484	0.622
GRPO (plain prompt)	0.664	0.823	55.1	0.676	0.530	0.648
GRPO (figure-desc prompt)	0.609	0.831	54.9	0.668	0.506	0.632
GRPO (mix: plain + figure-desc)	0.633	0.841	56.8	0.708	0.500	0.611
naive OPD (32B teacher)	0.604	0.790	57.8	0.750	0.466	0.557
privileged OPD (32B teacher)	**0.708**	0.837	58.2	**0.878**	0.496	0.565
minimal-edit (32B teacher, ours)	0.699	0.899	**59.0**	0.727	**0.553**	0.673
minimal-edit (8B self-teacher, ours)	0.693	0.897	53.0	0.745	0.550	0.679

2048-token response limit, and average pass@k over three random seeds. Appendix F provides the predicate-level pass@1 breakdown. Tables 1 and 2 report results for the Qwen3-VL models, while Table 3 compares methods on a shared Qwen2.5-VL-7B backbone.

4.1 Main results

Multimodal mathematics. Minimal-edit distillation provides consistent gains across Geometry3K and MathVerse. Using only the image and question at inference, both privileged and minimal-edit distillation exceed the corresponding base model supplied with a reference figure description, whose pass@1 is 0.462 at 2B and 0.688 at 8B. SFT remains competitive in domain but does not transfer reliably to MathVerse, whereas GRPO transfers better but remains below minimal-edit distillation. On the shared Qwen2.5-VL-7B backbone, minimal-edit distillation leads all multimodal-mathematics metrics while using one of the smallest training sets.

Perception-oriented evaluations. Description-oriented SFT provides a strong reference for figure-description recall because it directly optimizes that output. Minimal-edit distillation improves both perception-oriented evaluations over the base models at 2B and 8B, whereas GRPO with reference figure descriptions does not reliably improve perception over the base model. On Qwen2.5-VL-7B, minimal-edit distillation also performs competitively with published external models on the perception-oriented metrics. Perception-R1 leads figure-description recall and Geoperception pass@1, which confirms the construct validity of our perception oriented evaluations. Together, the SFT and GRPO results show that direct figure-description supervision does not consistently transfer across reasoning and perception. Most clearly for 2B figure-description-prompt GRPO, which underperforms the base model on every metric except Geoperception.

Minimal-edit distillation reduces dependence on teacher scale because the teacher need not generate an independent solution. Conditioned on the student's rollout and a verified reference, it is instructed to make the smallest edits needed to correct the rollout. Correcting an existing solution in this way is more verification-like than generating a solution from scratch. This interpretation is consistent with work on the generation–verification gap, which shows that models can derive useful supervision by evaluating their own generations once they have sufficient verification capability (Song et al., 2025). Empirically, the self-teacher remains close to the larger teacher at both scales. Their gap is negligible at 8B but slightly more pronounced at 2B, consistent with the smaller model's more limited ability to provide its own corrective signal. Even at 2B, however, minimal-edit self-distillation still outperforms naive and privileged distillation with the larger teacher.

Table 3: Published Qwen2.5-VL-7B checkpoints and our method on the same backbone. Metrics follow Table 1; per column the best result is ***bold*** and the second-best value or tie is underlined.

		Geometry3K		MathVerse	Fig-desc	Geoperception	
Model	Train	pass@1	pass@8	(ALL)	recall	pass@1	pass@4
Qwen2.5-VL-7B	–	0.373	0.683	45.1	0.431	0.330	0.575
Perception-R1-7B	1.4K	0.504	0.690	44.9	**0.541**	**0.420**	0.570
NoisyRollout-7B	2.1K+6.4K	0.522	0.709	47.9	0.500	0.410	0.556
PAPO-7B	38.8K	0.407	0.640	48.9	0.470	0.351	0.540
DeepEyes-7B	47K	0.393	0.693	46.1	0.447	0.363	**0.588**
Vision-R1-7B	200K+10K	0.436	0.637	47.7	0.486	0.392	0.552
minimal-edit	1.8K	**0.543**	**0.718**	**49.3**	0.477	0.410	0.568

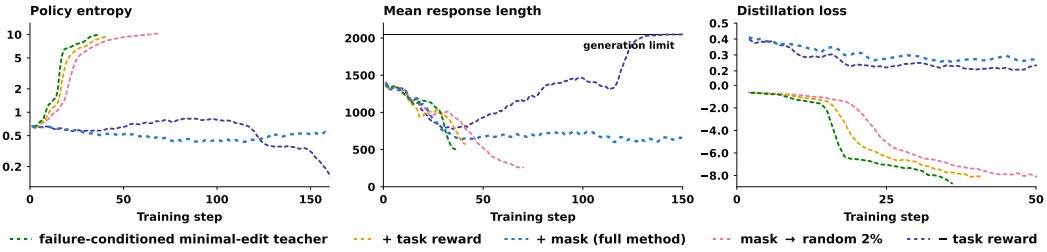


Figure 4: Training dynamics for Qwen3-VL-2B component ablations: policy entropy (left), mean response length (center), and distillation loss (right). Only the full method remains stable.

4.2 Ablation Studies

Table 5 tests the sensitivity of minimal-edit distillation to the corrective weight λ and to the verified reference solution. Using the Qwen3-VL-2B student and Qwen3-VL-8B teacher, we vary λ from 0.001 to 0.1 while holding the rest of the training configuration fixed; the final row removes the reference solution from the selected $\lambda = 0.01$ configuration. Every swept weight improves on the untrained base across all measurements. The corrective weight has a nonmonotonic effect: an intermediate weight performs consistently best, whereas either weakening or strengthening the distillation term reduces accuracy. This suggests that minimal-edit supervision is most effective when it complements rather than overwhelms the task-reward signal. Removing the verified reference solution also degrades all six measures. The reference solution therefore serves as an additional anchor for identifying how an incorrect rollout should be corrected. This ablation does not separately isolate the figure description because reference solutions are themselves generated with it to maximize verified-solution yield. Removing it only during teacher scoring leaves this dependence intact, while regenerating references without descriptions also changes solution yield and quality.

Table 4: Component configurations used in the ablation study.

	ME Teacher	Reward	Mask
privileged OPD	✗	✗	✗
+ ME teacher	✓	✗	✗
+ task reward	✓	✓	✗
+ random 2% mask	✓	✓	R
full method	✓	✓	✓
− task reward	✓	✗	✓

Table 4 defines the component combinations. ME Teacher denotes minimal-edit conditioning for scoring incorrect rollouts, Reward applies task reward to correct rollouts, and Mask refers to the corrective mask that retains tokens with $k_t > 0$; R instead denotes a uniformly random token position mask that matches the corrective mask's empirical keep rate of roughly 2% (Figure 5). Components are added cumulatively through task reward, after which the random-mask control and the full method differ only in how masked tokens are selected. The final row removes task reward from the full configuration. Runs without task reward contain no combined loss and therefore require no corrective weight; all other ablation runs use $\lambda = 0.01$. Figure 4 shows the corresponding training dynamics. The incomplete configurations become unstable through two distinct failure modes. Under minimal-edit conditioning, the teacher sees the student's rollout and is explicitly instructed to preserve its cor-

Table 5: Corrective-weight sweep and reference-solution ablation for the Qwen3-VL-2B student with a Qwen3-VL-8B teacher. Performance peaks at an intermediate weight and declines without the reference solution. Per column, the best result is ***bold*** and the second best is underlined; the configuration used in our main experiments is shaded.

	Geometry3K		MathVerse	Fig-desc	Geoperception	
λ	pass@1	pass@8	(ALL)	recall	pass@1	pass@4
0.001	0.442	0.736	42.5	0.584	0.373	0.550
0.005	0.432	0.721	39.0	0.589	0.374	0.569
0.01	**0.554**	**0.794**	**45.8**	**0.633**	**0.455**	**0.635**
0.05	0.407	0.697	42.3	0.609	0.396	0.594
0.1	0.366	0.655	41.1	0.610	0.357	0.562
0.01 (no ref. solution)	0.471	0.752	43.8	0.618	0.433	0.625

rect portions (Appendix G.2). Higher teacher probability on these already supplied tokens therefore largely reflects copying from the rollout rather than corrective information that the student should learn. Because preserved tokens constitute most of the response, their negative token-level k_1 estimates dominate the unmasked objective (Figure 3). The distillation term consequently becomes strongly negative and policy entropy explodes. Replacing the corrective mask with the random 2% mask produces the same failure mode. Matching the mask's sparsity is therefore insufficient and validates that the tokens selected by the corrective mask carry meaningful corrective information.

Without task reward on correct rollouts, the opposite failure occurs: entropy collapses and responses grow to the generation limit. This suggests that task reward does more than retain otherwise unused correct samples: by reinforcing successful on-policy behavior, it anchors the policy to solution patterns already known to succeed. This interpretation is consistent with the long-horizon stability obtained by routing correct samples to reward-aligned RL in SRPO (Li et al., 2026), and with evidence that verified-correct reference rollouts can prevent policy drift (Huang et al., 2026a). Both failure modes drive a marked decline in validation accuracy and result in optimization collapse. By contrast, the full method maintains stable entropy and distillation loss throughout training.

5 Conclusion and Future Perspectives

In this work, we identify diagram perception as a major source of smaller VLMs' disadvantage in visual mathematical reasoning and introduce minimal-edit corrective on-policy distillation to use privileged perceptual information for response-specific corrective supervision without requiring it at inference. For each incorrect rollout, a privileged minimal-edit teacher scores the student's existing tokens, and only positions the teacher disfavors are used to update the policy; correct rollouts instead receive task reward. Across model scales, this response-specific supervision improves reasoning and performance on perception-oriented evaluations, particularly low-level geometric perception, meets or exceeds the corresponding base model evaluated with a reference figure description at inference, and remains effective with a frozen same-size teacher. By comparison, supervised fine-tuning, description-conditioned reinforcement learning, and full-token privileged distillation do not transfer the benefit of figure descriptions as consistently.

More broadly, our findings suggest that recoverable privileged information can be used as a training-time diagnostic rather than an additional input or generation target. Disagreement with a privileged critic can expose where information available in a complex visual input fails to enter the model's reasoning. One limitation is that the sign test used by the reverse-KL mask is not guaranteed to identify every token occurrence that reflects an underlying error. In Figure 3, for example, some occurrences of the student's leg–hypotenuse misreading do not receive positive k_1 estimates and are therefore not selected by the mask. Nevertheless, this partial localization provides an effective corrective signal empirically in our experiments. A future direction of this work may extend beyond geometry to charts, scientific figures, tables, and other domains in which an additional textual reading makes visual evidence easier to verify. This will require reliable ways to construct privileged views that are faithful to the original input, aligned with the task, and informative about the specific errors a model can learn to correct.

AI USE STATEMENT

We used generative AI tools in the experimental pipeline. Muse Spark generated the reference figure descriptions used as privileged training information and as references for evaluation. We screened these descriptions for answer leakage and manually inspected exact answer matches. Frozen vision-language models generated candidate reference solutions; we retained only solutions whose final answers passed the rule-based verifier. We also used model-based evaluators: the standard GPT-4o mini judge for MathVerse and a Gemini 3.1 Pro judge for figure-description recall, under the fixed evaluation protocols and prompts described in the paper. Claude Code was used to assist with implementing the proposed research idea. We manually examined the code itself and performed both smoke tests and full tests of the resulting implementation to verify its correctness. These uses supported research execution, synthetic-data generation, and evaluation.

Generative AI assistants also aided literature discovery; editing, and organizing parts of the manuscript. We reviewed all AI-assisted text and verified citations against the cited sources. We did not rely on generative AI to establish any formal proof. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the information needed to reproduce the method and experiments throughout the paper and appendix. Section 3 defines the learning objective, and Appendix A gives the complete training pseudocode, including rollout routing, teacher scoring, mask construction, normalization, and empty-branch handling. The dataset-construction paragraph describes figure-description screening, reference-solution generation, and rule-based filtering, while Section 4 and Appendix D report the model configurations, optimization hyperparameters, sampling settings, sequence limits, checkpoint-selection procedure, and baseline training details. The exact privileged and minimal-edit teacher prompts are provided in Appendix G; the figure-description generation and evaluation protocol, including the judge prompt, is specified in Appendix E. Finally, Appendix B states the assumptions and measured inputs used in the computational-cost analysis.

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. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2306.13649.

Shuai Bai et al. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923*, 2025a. URL https://arxiv.org/abs/2502.13923.

Shuai Bai et al. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631*, 2025b. URL https://arxiv.org/abs/2511.21631.

Walid Bousselham, Hilde Kuehne, and Cordelia Schmid. VOLD: Reasoning Transfer from LLMs to Vision-Language Models via On-Policy Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26209–26218, 2026. URL https://arxiv.org/abs/2510.23497.

Guizhen Chen, Weiwen Xu, Hao Zhang, Hou Pong Chan, Deli Zhao, Anh Tuan Luu, and Yu Rong. GeoPQA: Bridging the Visual Perception Gap in MLLMs for Geometric Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 25680–25688, 2025. URL https://arxiv.org/abs/2509.17437.

Kanzhi Cheng, Yantao Li, Fangzhi Xu, Jianbing Zhang, Hao Zhou, and Yang Liu. Vision-Language Models Can Self-Improve Reasoning via Reflection. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 8876–8892, 2025. URL https://arxiv.org/abs/2411.00855.

Yi Ding and Ruqi Zhang. Sherlock: Self-Correcting Reasoning in Vision-Language Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2505.22651.

Yi Ding, Ziliang Qiu, Bolian Li, and Ruqi Zhang. Learning Self-Correction in Vision-Language Models via Rollout Augmentation. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2602.08503.

Mark Endo and Serena Yeung-Levy. Downscaling Intelligence: Exploring Perception and Reasoning Bottlenecks in Small Multimodal Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 778–788, 2026. URL https://arxiv.org/abs/2511.17487.

Google DeepMind. Gemini 3.1 Pro. Google DeepMind, 2026. URL https://deepmind.google/models/gemini/pro/.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge Distillation of Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2306.08543.

Dongze Hao, Zhiwei Jin, Chen Chen, and Haonan Lu. Stabilizing On-Policy Distillation for MLLM Reasoning with Global Normalization. *arXiv preprint arXiv:2606.09091*, 2026. URL https://arxiv.org/abs/2606.09091.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training Compute-Optimal Large Language Models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30016–30030, 2022. doi: 10.52202/068431-2176. URL https://arxiv.org/abs/2203.15556.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large Language Models Cannot Self-Correct Reasoning Yet. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.01798.

Kexin Huang, Junkang Wu, Jinda Lu, Shuo Yang, Chiyu Ma, Jiancan Wu, Xiang Wang, Xiangnan He, Guoyin Wang, and Jingren Zhou. ARMOR: Stabilizing On-Policy LLM RL with Off-Policy Anchor Samples. *arXiv preprint arXiv:2607.10481*, 2026a. URL https://arxiv.org/abs/2607.10481.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Xu Tang, Yao Hu, and Shaohui Lin. Vision-R1: Incentivizing Reasoning Capability in Multimodal Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2026b. URL https://arxiv.org/abs/2503.06749.

Nan Jia, Haojin Yang, Xing Ma, Jiesong Lian, Shuailiang Zhang, Weipeng Zhang, Ke Zeng, Xunliang Cai, and Zequn Sun. Asymmetric On-Policy Distillation: Bridging Exploitation and Imitation at the Token Level. *arXiv preprint arXiv:2605.06387*, 2026. URL https://arxiv.org/abs/2605.06387.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling Laws for Neural Language Models. *arXiv preprint arXiv:2001.08361*, 2020. URL https://arxiv.org/abs/2001.08361.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165. URL https://arxiv.org/abs/2309.06180.

Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. Unifying Group-Relative and Self-Distillation Policy Optimization via Sample Routing. *arXiv preprint arXiv:2604.02288*, 2026. URL https://arxiv.org/abs/2604.02288.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/aca97732e30bcf1303bc22ac3924fd16-Abstract-Conference.html.

Ruiqi Liu, Xiaolei Lv, Gengsheng Li, Ximo Zhu, Zhiheng Wang, Zhengbo Zhang, Junkai Chen, Zhiheng Li, Bo Li, Jun Gao, and Shu Wu. Visual-Advantage On-Policy Distillation for Vision-Language Models. *arXiv preprint arXiv:2605.21924*, 2026. URL https://arxiv.org/abs/2605.21924.

Xiangyan Liu, Jinjie Ni, Zijian Wu, Chao Du, Longxu Dou, Haonan Wang, Tianyu Pang, and Michael Qizhe Shieh. NoisyRollout: Reinforcing Visual Reasoning with Data Augmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2504.13055.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations (ICLR)*, 2016. URL https://arxiv.org/abs/1511.03643.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://arxiv.org/abs/1711.05101.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-GPS: Interpretable Geometry Problem Solving with Formal Language and Symbolic Reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP)*, pp. 6774–6786, 2021. URL https://arxiv.org/abs/2105.04165.

Yuanjie Lyu, Chengyu Wang, Jun Huang, and Tong Xu. Student-Centered Distillation Narrows the Agentic Gap Between Small and Large LLMs. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2509.14257.

Meta AI. Introducing Muse Spark 1.3. Meta AI Research, 2026. URL https://research.meta.ai/blog/introducing-muse-spark-1-3.

OpenAI. GPT-4o Mini Model. OpenAI API Documentation, 2024. URL https://developers.openai.com/api/docs/models/gpt-4o-mini.

Leyi Pan, Shuchang Tao, Yunpeng Zhai, Lingzhe Zhang, Zhaoyang Liu, Bolin Ding, Aiwei Liu, and Lijie Wen. RLCSD: Reinforcement Learning with Contrastive On-Policy Self-Distillation. *arXiv preprint arXiv:2606.11709*, 2026. URL https://arxiv.org/abs/2606.11709.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged Information Distillation for Language Models. *arXiv preprint arXiv:2602.04942*, 2026. URL https://arxiv.org/abs/2602.04942.

Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. Vision language models are blind: Failing to translate detailed visual features into words. In *Asian Conference on Computer Vision (ACCV)*, pp. 293–309, 2024. URL https://arxiv.org/abs/2407.06581.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory Optimizations Toward Training Trillion Parameter Models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16, 2020. doi: 10.1109/SC41405.2020.00024. URL https://doi.org/10.1109/SC41405.2020.00024.

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.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A Flexible and Efficient RLHF Framework. In *Proceedings of the Twentieth European Conference on Computer Systems (EuroSys)*, pp. 1279–1297, 2025. URL https://arxiv.org/abs/2409.19256.

Yuda Song, Hanlin Zhang, Carson Eisenach, Sham M. Kakade, Dean Foster, and Udaya Ghai. Mind the Gap: Examining the Self-Improvement Capabilities of Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2412.02674.

Zhiquan Tan and Yinrong Hong. Self-Supervised On-Policy Distillation for Reasoning Language Models. *arXiv preprint arXiv:2605.17497*, 2026. URL https://arxiv.org/abs/2605.17497.

Kanghui Tian, Siyuan Liu, Ziang Yan, Sheng Xia, Shuai Dong, and Yi Wang. ViCuR: Visual Cues as Recoverable Privilege for Multimodal On-Policy Distillation. *arXiv preprint arXiv:2606.05718*, 2026. URL https://arxiv.org/abs/2606.05718.

Sihan Wang, Xiyao Liu, Lianqing Liu, and Zhi Han. Seeing Before Reasoning: Decoupling Perception and Reasoning for Shortcut-Resilient Multimodal On-Policy Self-Distillation. *arXiv preprint arXiv:2606.19120*, 2026a. URL https://arxiv.org/abs/2606.19120.

Yuanyi Wang, Su Lu, Yanggan Gu, Pengkai Wang, Yifan Yang, Zhaoyi Yan, Congkai Xie, Jianmin Wu, and Hongxia Yang. Not All Disagreement Is Learnable: Token Teachability in On-Policy Distillation. *arXiv preprint arXiv:2605.26844*, 2026b. URL https://arxiv.org/abs/2605.26844.

Zhenhailong Wang, Xuehang Guo, Sofia Stoica, Haiyang Xu, Hongru Wang, Hyeonjeong Ha, Xiusi Chen, Yangyi Chen, Ming Yan, Fei Huang, and Heng Ji. Perception-Aware Policy Optimization for Multimodal Reasoning. In *International Conference on Learning Representations (ICLR)*, 2026c. URL https://arxiv.org/abs/2507.06448.

Yecheng Wu, Song Han, and Han Cai. Lightning OPD 2.0: Mitigating Style Bias in Cross-Teacher On-Policy Distillation for Large Reasoning Models. *arXiv preprint arXiv:2607.28449*, 2026. URL https://arxiv.org/abs/2607.28449.

Shizhe Xiang, Ke An, Wenlong Yu, Yue Liu, Jian Luan, Pei Fu, and Qilong Wang. Teaching the Way, Not the Answer: Privileged Tutoring Distillation for Multimodal Policy Optimization. *arXiv preprint arXiv:2606.07000*, 2026. URL https://arxiv.org/abs/2606.07000.

Tong Xiao, Xin Xu, Zhenya Huang, Hongyu Gao, Quan Liu, Qi Liu, and Enhong Chen. Perception-R1: Advancing Multimodal Reasoning Capabilities of MLLMs via Visual Perception Reward. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2506.07218.

Haoran Xin, Anhao Zhao, Ying Sun, Jin Li, Xiaoyu Shen, and Hui Xiong. Escaping the KL Agreement Trap in On-Policy Distillation. *arXiv preprint arXiv:2606.09471*, 2026. URL https://arxiv.org/abs/2606.09471.

Hongling Xu, Qi Zhu, Heyuan Deng, Jinpeng Li, Lu Hou, Yasheng Wang, Lifeng Shang, Ruifeng Xu, and Fei Mi. KDRL: Post-Training Reasoning LLMs via Unified Knowledge Distillation and Reinforcement Learning. *arXiv preprint arXiv:2506.02208*, 2025. URL https://arxiv.org/abs/2506.02208.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. TIP: Token Importance in On-Policy Distillation. *arXiv preprint arXiv:2604.14084*, 2026. URL https://arxiv.org/abs/2604.14084.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-Policy Context Distillation for Language Models. *arXiv preprint arXiv:2602.12275*, 2026. URL https://arxiv.org/abs/2602.12275.

Hee Suk Yoon, Eunseop Yoon, Jaehyun Jang, SooHwan Eom, Ji Woo Hong, Mark Hasegawa-Johnson, Qi Dai, Chong Luo, and Chang D. Yoo. Decomposed On-Policy Distillation for Vision-Language Reasoning: Steering Gradients for Visual Grounding. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2606.00564.

Seonghoon Yu, Dongjun Nam, Byung-Kwan Lee, and Jeany Son. Hide to See: Reasoning-prefix Masking for Visual-anchored Thinking in VLM Distillation. *arXiv preprint arXiv:2605.11651*, 2026. URL https://arxiv.org/abs/2605.11651.

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. URL https://arxiv.org/abs/2605.18740.

Jiarui Zhang, Ollie Liu, Tianyu Yu, Jinyi Hu, and Willie Neiswanger. Euclid: Supercharging Multimodal LLMs with Synthetic High-Fidelity Visual Descriptions. *arXiv preprint arXiv:2412.08737*, 2024a. URL https://arxiv.org/abs/2412.08737.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Peng Gao, and Hongsheng Li. MathVerse: Does Your Multi-modal LLM Truly See the Diagrams in Visual Math Problems? In *European Conference on Computer Vision (ECCV)*, pp. 169–186, 2024b. URL https://arxiv.org/abs/2403.14624.

Wenhao Zhang. Beyond Absolute Imitation: Anchored Residual Guidance for Privileged On-Policy Distillation. *arXiv preprint arXiv:2606.10385*, 2026. URL https://arxiv.org/abs/2606.10385.

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*, 2026a. URL https://arxiv.org/abs/2601.18734.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. PyTorch FSDP: Experiences on Scaling Fully Sharded Data Parallel. *Proceedings of the VLDB Endowment*, 16 (12):3848–3860, 2023. doi: 10.14778/3611540.3611569. URL https://www.vldb.org/pvldb/vol16/p3848-huang.pdf.

Ziqi Zhao, Xinyu Ma, Liu Yang, Yujie Feng, Daiting Shi, Jingzhou He, Xin Xin, Zhaochun Ren, and Xiao-Ming Wu. ROSD: Reflective On-Policy Self-Distillation for Language Model Reasoning across Domains. *arXiv preprint arXiv:2605.28014*, 2026b. URL https://arxiv.org/abs/2605.28014.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 400–410, 2024. doi: 10.18653/v1/2024.acl-demos.38. URL https://aclanthology.org/2024.acl-demos.38/.

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. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2505.14362.

Siqi Zhu, Xuyan Ye, Hongyu Lu, Weiye Shi, and Ge Liu. The Many Faces of On-Policy Distillation: Pitfalls, Mechanisms, and Fixes. *arXiv preprint arXiv:2605.11182*, 2026. URL https://arxiv.org/abs/2605.11182.

A MINIMAL-EDIT CORRECTIVE OPD PSEUDOCODE

Algorithm 1 gives the complete training procedure. Each training example contains the student input x_i (image and question), a training-only figure description d_i, a verified reference solution A_i, and the ground-truth answer g_i. The verifier V maps a rollout and g_i to a binary correctness reward. We first sample all K rollouts from a frozen snapshot of the current student and compute their task advantages from the K verifier rewards. A correct rollout enters only the task-reward branch. For an incorrect rollout, the teacher receives the complete rollout inside the minimal-edit prompt and scores a second, teacher-forced occurrence of the same rollout. Thus, TEACHERFORCE is a scoring pass, not a decoding call.

The corrective mask is computed from detached student and teacher scores and is therefore fixed during the subsequent policy update. We normalize the corrective term by the number Z of selected tokens across the minibatch; if $Z = 0$, its contribution is zero. In the experiments, TASKADVANTAGES and $\widehat{\mathcal{L}}_{\mathrm{PG}}$ are the group-relative advantages and clipped policy-gradient objective described in Section 4. The task objective is evaluated only on correct rollouts, although its group statistics are computed from all K verifier rewards. Consequently, every sampled rollout contributes through exactly one branch.

Algorithm 1 Minimal-edit corrective on-policy distillation (ME-OPD)

Require: Dataset $\mathcal{D} = \{(x_i, d_i, A_i, g_i)\}$; student π_θ; frozen teacher π_T; verifier V; rollouts per example K; corrective weight λ

for each optimization step **do**
  Sample a minibatch $\mathcal{B} \subset \mathcal{D}$ and set $\theta_{\mathrm{old}} \leftarrow \theta$
  Initialize correct-rollout buffer $\mathcal{C} \leftarrow \varnothing$ and incorrect-rollout buffer $\mathcal{I} \leftarrow \varnothing$
  for each $(x_i, d_i, A_i, g_i) \in \mathcal{B}$ **do**
    for $j = 1, \ldots, K$ **do**
      $y_{ij} \sim \pi_{\theta_{\mathrm{old}}}(\cdot \mid x_i)$
      $r_{ij} \leftarrow V(y_{ij}, g_i) \in \{0, 1\}$
    $(\widehat{a}_{i1}, \ldots, \widehat{a}_{iK}) \leftarrow \textsc{TaskAdvantages}(r_{i1}, \ldots, r_{iK})$
    for $j = 1, \ldots, K$ **do**
      if $r_{ij} = 1$ **then**
        Append $(x_i, y_{ij}, \widehat{a}_{ij})$ to $\mathcal{C}$
      else
        $x^{ij}_{\mathrm{edit}} \leftarrow \textsc{EditPrompt}(x_i, d_i, A_i, y_{ij})$
        $\ell^S_{ij} \leftarrow \textsc{LogProbs}(\pi_{\theta_{\mathrm{old}}}, x_i, y_{ij})$ ▷ student scores from the rollout-policy
        $\ell^T_{ij} \leftarrow \textsc{TeacherForce}(\pi_T, x^{ij}_{\mathrm{edit}}, y_{ij})$ ▷ teacher scores the same rollout
        $m_{ij} \leftarrow \mathrm{stopgrad}\big(\mathbf{1}[\ell^S_{ij} - \ell^T_{ij} > 0]\big)$ ▷ compute the corrective mask
        Append $(x_i, y_{ij}, \ell^T_{ij}, m_{ij})$ to $\mathcal{I}$
  if $|\mathcal{C}| > 0$ **then**
    $\widehat{\mathcal{L}}_{\mathrm{task}} \leftarrow \widehat{\mathcal{L}}_{\mathrm{PG}}(\theta; \theta_{\mathrm{old}}, \mathcal{C})$
  else
    $\widehat{\mathcal{L}}_{\mathrm{task}} \leftarrow 0$ ▷ no correct rollouts
  $Z \leftarrow \sum_{(x,y,\ell^T,m)\in\mathcal{I}} \sum_t m_t$ ▷ number of selected corrective tokens
  if $Z > 0$ **then**
    $\widehat{\mathcal{L}}_{\mathrm{corr}} \leftarrow \dfrac{1}{Z} \sum_{(x,y,\ell^T,m)\in\mathcal{I}} \sum_t m_t \left[\log \pi_\theta(y_t \mid y_{<t}, x) - \ell^T_t\right]$
  else
    $\widehat{\mathcal{L}}_{\mathrm{corr}} \leftarrow 0$ ▷ avoid division by zero when no token is selected
  $\widehat{\mathcal{L}} \leftarrow \widehat{\mathcal{L}}_{\mathrm{task}} + \lambda \widehat{\mathcal{L}}_{\mathrm{corr}}$
  Update θ by descending $\nabla_\theta \widehat{\mathcal{L}}$

At inference, the trained student receives only x; the reference figure description, reference solution, teacher, verifier, and edit prompt are all training-only components.

B Computational Cost Analysis

We compare the per-step floating-point operation (FLOP) cost of minimal-edit corrective on-policy distillation with naive OPD, holding student behavior fixed at the sequence lengths and incorrect-rollout fractions observed in the minimal-edit runs. Let N be the number of student parameters and let the teacher contain rN parameters. Let P_s and R denote the mean student-prompt and response lengths, respectively, and define the complete student sequence length as $D = P_s + R$. Each problem produces K student rollouts, of which a fraction w are incorrect. Finally, P_t denotes the fixed portion of the minimal-edit teacher prompt in Section G.2, including the edit instruction, figure description, reference solution, and image tokens, measured with an empty student response.

Kaplan et al. (2020) give a leading-order forward cost of $2N$ FLOPs per token, apart from a context-length-dependent attention term; over D tokens, this yields $2ND$. Hoffmann et al. (2022) use $6ND$ as the corresponding forward–backward training cost. Our runs use activation recomputation (gradient checkpointing), which re-evaluates forward activations during backpropagation and raises the update cost to $8ND$.

Both methods share the same student computation: rollout generation over R response tokens, a separate post-rollout forward pass that evaluates behavior-policy log probabilities over D tokens, and a policy update over D tokens. Their respective costs are

$$C_{\text{gen}} \simeq 2KNR, \qquad C_{\text{logprob}} \simeq 2KND, \qquad C_{\text{train}} \simeq 8KND, \tag{1}$$

and hence

$$C_{\text{student}} \simeq KN(2R + 10D). \tag{2}$$

In naive OPD, the teacher scores every student rollout over the complete sequence of length D. The teacher cost is $2rKND$, giving

$$\begin{aligned} C_{\text{naive}} &\simeq KN\left[2R + (10 + 2r)D\right] \\ &\simeq KND\left[\frac{2R}{D} + 10 + 2r\right]. \end{aligned} \tag{3}$$

The minimal-edit teacher scores only the fraction w of rollouts that are incorrect. For each teacher-scored rollout, the on-policy student response, with mean length R, occurs once in the conditioning prompt and once as the suffix whose log probabilities are evaluated, so the teacher processes $P_t + 2R$ tokens. One copy is embedded in the teacher prompt, allowing the teacher to condition on the complete student response; the other is the teacher-forced suffix whose log probabilities are evaluated. The resulting total is

$$\begin{aligned} C_{\text{edit}} &\simeq KN\left[2R + 10D + 2wr(P_t + 2R)\right] \\ &\simeq KND\left[\frac{2R}{D} + 10 + \frac{2wr(P_t + 2R)}{D}\right]. \end{aligned} \tag{4}$$

The corrective mask and correctness gate require no additional model evaluations.

Dividing Eq. (4) by Eq. (3) yields

$$\frac{C_{\text{edit}}}{C_{\text{naive}}} = 1 + \frac{2r\left[w(P_t + 2R) - D\right]}{2R + (10 + 2r)D}. \tag{5}$$

Because the denominator is positive, minimal-edit distillation is less expensive than naive OPD whenever

$$w < w^* = \frac{D}{P_t + 2R}. \tag{6}$$

The right-hand side is the ratio of the sequence length processed by the naive teacher for each rollout, D, to the sequence length processed by the minimal-edit teacher for each scored rollout, $P_t + 2R$. Thus, at $w = w^*$, the expected teacher-processed tokens per rollout are equal: $w^*(P_t + 2R) = D$.

Table 6 reports the quantities used in the comparison. Response length, prompt length, and incorrect-rollout fraction are collected from the Weights & Biases (W&B) run logs and averaged over all training steps. Using the minimal-edit statistics for both methods isolates the training recipe from differences in response length and truncation behavior.

Table 6: Inputs to the computational-cost estimate. Sequence lengths are measured in tokens.

Quantity	8B student / 32B teacher	2B student / 8B teacher
Mean response length R	736.62	783.46
Mean student-prompt length P_s	209.32	209.35
Student sequence $D = P_s + R$	945.94	992.81
Incorrect-rollout fraction w	0.275	0.456
Parameter ratio r	4.00	4.00
Fixed edit-prompt length P_t	1369.2	1316.6

Table 7: Estimated per-step compute of minimal-edit distillation relative to naive OPD.

Configuration	D	$P_t + 2R$	w^*	C_{naive}	C_{edit}	$C_{\text{edit}}/C_{\text{naive}}$
8B student / 32B teacher	945.94	2842.44	0.333	$19.56\,KND$	$18.17\,KND$	**0.929**
2B student / 8B teacher	992.81	2883.52	0.344	$19.58\,KND$	$22.17\,KND$	**1.133**

Substituting the measured values into Eqs. (3)–(6) gives Table 7. We report each cost as a multiple of KND; this common factor cancels in the ratio. For the 8B student, $w = 0.275 < w^* = 0.333$, so the minimal-edit teacher processes $w(P_t + 2R) = 781.67$ tokens per rollout in expectation, fewer than the $D = 945.94$ tokens processed by the naive teacher. Minimal-edit distillation therefore requires 7.1% fewer FLOPs. For the 2B student, $w = 0.456 > w^* = 0.344$ and $w(P_t + 2R) = 1314.89 > D = 992.81$, yielding 13.3% more FLOPs.

Figure 5 shows that the correct-rollout fraction increases over the 2B run, so the complementary incorrect-rollout fraction w decreases. The minimal-edit teacher is therefore invoked less often as training progresses, reducing the cost of minimal-edit OPD relative to naive OPD, whose teacher scores every rollout.

C Training Dynamics of the Minimal-Edit Run

Figure 5 reports the two quantities that determine how supervision is allocated during the full Qwen3-VL-2B minimal-edit run. Among incorrect rollouts, the corrective mask retains only about 2% of response tokens. The keep fraction fluctuates within a narrow band over 250 steps and shows no sustained upward trend, even as the on-policy distribution changes. The teacher therefore continues to identify a small set of candidate corrections.

At the same time, the correct-rollout fraction rises from roughly one third early in training to more than one half near the end. Because correct and incorrect rollouts are handled by mutually exclusive branches, this increase changes the mixture of supervision: more sampled responses receive task-reward reinforcement, while the remaining failures continue to receive sparse token-level correction. The joint behavior is consistent with the intended division of labor in the objective. The model produces successful rollouts more often without losing the sparse character of the corrective signal on unsuccessful ones.

D Baseline Training Details

Supervised fine-tuning. We train all supervised baselines with LlamaFactory (Zheng et al., 2024) using full-parameter fine-tuning: the vision encoder, multimodal projector, and language model are all unfrozen. Training uses AdamW (Loshchilov & Hutter, 2019) in bfloat16 with DeepSpeed ZeRO-3 (Rajbhandari et al., 2020) for up to three epochs, a learning rate of 1×10^{-5} with cosine decay, a warmup ratio of 0.1, and an effective batch size of 12 sequences. The maximum combined prompt-and-target length is 4096 tokens. Input images are capped at 262144 pixels, corresponding to at most 334 visual tokens per figure.

Reinforcement learning. We train the reinforcement-learning baselines with GRPO (Shao et al., 2024) in verl (Sheng et al., 2025), using an FSDP2 actor (Zhao et al., 2023) and vLLM (Kwon et al.,

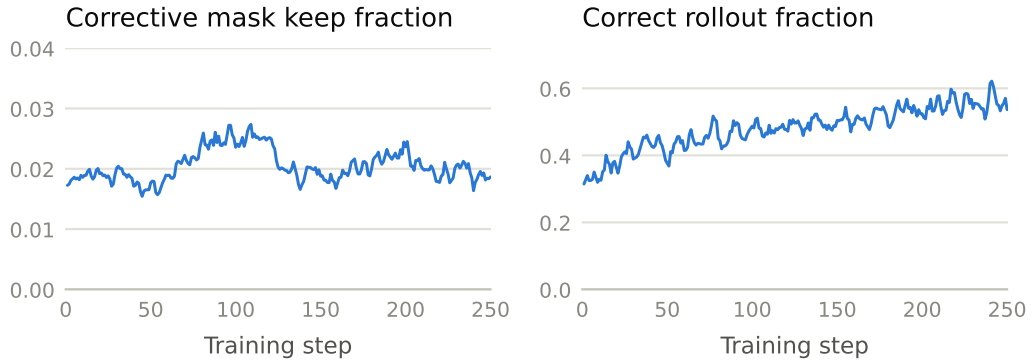


Figure 5: Training dynamics of the full minimal-edit run. Corrective-mask keep rate on incorrect rollouts (left) and correct-rollout fraction (right). The mask stays near 2% as accuracy rises, shifting training toward task reward without diffusing corrective supervision.

2023) for rollout generation. Each step samples eight rollouts per prompt at temperature 1.0 from a batch of 96 prompts. We use a minibatch of 96 prompts and one optimization epoch per rollout batch, yielding a single strictly on-policy update. Prompts and responses are limited to 1024 and 2048 tokens, respectively. The actor is optimized with AdamW (Loshchilov & Hutter, 2019) using a learning rate of 1×10^{-6}, a constant schedule without warmup, weight decay of 0.01, and gradient clipping at 1.0. We use a symmetric clipping range of 0.2, a dual-clip bound of 3.0, no entropy bonus, and token-mean loss aggregation. We apply neither an explicit KL loss nor a KL penalty in the reward and therefore retain no reference model. The reward is the rule-based Geometry3K correctness signal, and training runs for 30 epochs. For both supervised fine-tuning and GRPO, we select the checkpoint with the highest validation accuracy.

E Reference Figure Description Collection and Use

E.1 Figure-description collection

We use Muse Spark to collect the reference figure descriptions with the prompt below. For each Geometry3K example, the model receives the diagram, the original question, and the verified final answer. The generation prompt explicitly prohibits derived values, intermediate reasoning, or the answer itself. These collected descriptions are the privileged visual context provided to both the conventional privileged teacher and the minimal-edit teacher during training; they also serve as the references for the figure-description recall evaluation below. Figure 6 shows qualitative examples collected with this procedure.

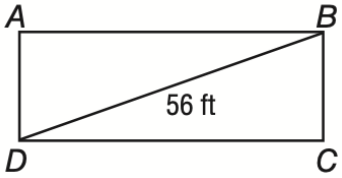
```
SYSTEM INSTRUCTION (SYSTEM_INSTRUCTION), passed via --system

You convert a geometry diagram into a precise textual description so
that a text-only language model, which cannot see the image, can
solve the problem.

The text-only solver will receive exactly:  YOUR figure description,
followed by the original question (verbatim).  Your job is to make
sure that combination contains every fact needed to solve it.

You are given the diagram image, the question, and the correct final
answer.  Use the answer ONLY to check completeness:
-- Actually solve the problem yourself from the figure and the
question.
-- Confirm the answer is derivable from [your figure description +
the question] ALONE. If you cannot derive it, your description is
missing a visual fact---find it in the figure and add it.
```

```
-- NEVER reveal the answer, the solution, intermediate steps, or any
computed/derived value in your description.  The solver must solve it
itself.

Describe ONLY what is visually shown in the figure, precisely:
-- Every labeled point and how the points are arranged (which are
collinear, which form the vertices of which shape).
-- Every line, ray, segment, arc, circle, polygon, and intersection
(intersection points, endpoints, which segments make up a path).
-- EVERY numeric measure or algebraic expression printed in the
figure:  side lengths, segment lengths, angle measures, arc measures,
coordinates, and labels like ``3x-1''; state which element each value
belongs to.
-- Every geometric mark:  right-angle squares (state which angle is
90 degrees), tick marks indicating congruent segments, arc marks
indicating congruent angles, parallel arrows, tangent points, and
center marks.
-- Relationships shown in the figure:  tangency, parallelism,
perpendicularity, midpoints, and bisectors---but only when the figure
itself marks them.

Note on the question text:  the question sometimes states constraints
or definitions (e.g., ``ZQ = 3a - 11'', ``BG is a median of triangle
ABC'', ``the radius is 4'').  That information stays in the final
prompt because the question is preserved verbatim, so you do not need
to repeat it---but DO account for it when checking solvability, and
make sure any fact that lives ONLY in the figure ends up in your
description.

Strict rules:
-- Describe only what is drawn or labeled.  Do not include any value
not explicitly shown in the figure that you computed during your own
solution.
-- Do NOT state the answer in the description.
-- If a value or label is unreadable, write ``unlabeled'' rather than
guessing.
-- Use plain text and standard math notation (e.g., ``angle ABC = 40
degrees'', ``AB = 3x - 1'').  Do not use markdown.

Respond with a single JSON object and nothing else:
{"figure_description":  "<the textual rendering>", "confidence":
"high" | "medium" | "low", "sufficient":  true | false}
where ``sufficient'' is true only if the correct answer is derivable
from your figure description plus the question alone.
```

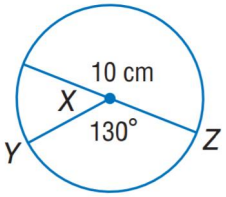
```
USER TURN (USER_TEMPLATE), passed as the positional prompt argument

Question that will be asked about this figure (kept verbatim in the
final prompt; it may itself contain givens you must account for):
{question}

Correct final answer (for your completeness check only; do not reveal
the answer, the solution, or any derived value in your description):
{answer}

Solve the problem yourself to confirm your figure description,
together with the question above, contains every fact needed to reach
that answer, then produce the JSON figure description now.
```

E.2 RECALL CALCULATION

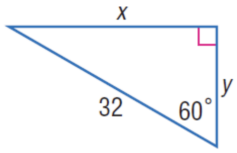
For each test diagram, we provide the Gemini 3.1 Pro judge (Google DeepMind, 2026) with only the reference figure description and one model-generated candidate description. The judge first decomposes the reference figure description into independently checkable visual facts, such as the existence of a shape, the placement of a labeled point, one printed measure, or one marked geometric



The figure shows a rectangle ABCD. The vertices are arranged as follows: A is at the top left, B is at the top right, C is at the bottom right, and D is at the bottom left. A diagonal line segment connects vertex D to vertex B. The diagonal DB is labeled '56 ft'.



A circle is shown with its center marked by a solid dot. A line segment passes through the center, forming a diameter, with one endpoint on the circle labeled Z. The length of this diameter is indicated by the label '10 cm' placed above it. A radius is drawn from the center to a point Y on the circle. The central angle formed by the radius to Y and the radius to Z is labeled 130 degrees. The adjacent central angle, formed by the radius to Y and the other half of the diameter, is labeled X.



A right triangle is shown. The top horizontal side has length x. The right vertical side has length y. The angle between the top horizontal side and the right vertical side is marked with a right-angle square, indicating it is 90 degrees. The hypotenuse has length 32. The angle between the hypotenuse and the right vertical side is 60 degrees.

Figure 6: Qualitative examples of reference figure descriptions used for the figure-description recall evaluation. Each row pairs a Geometry3K diagram with its reference figure description, whose objects, labels, measures, and geometric relations are decomposed into atomic facts by the Gemini 3.1 Pro judge.

relation. Compound statements are split so that each numeric value or relationship contributes a separate fact. It then marks each reference fact as covered only when the candidate conveys the same information with a matching value or relationship. Equivalent wording receives credit, whereas an omission, a contradictory value, or an incompatible relationship does not.

Let g_{ij} be the jth atomic fact returned for diagram i, and let $c_{ij} = \mathbf{1}[g_{ij}$ is correctly covered$]$. The per-diagram score is

$$R_i = \frac{\sum_{j=1}^{G_i} c_{ij}}{G_i}, \qquad R_{\text{macro}} = \frac{1}{M} \sum_{i=1}^{M} R_i, \tag{7}$$

where G_i is the number of atomic reference facts and M is the number of test diagrams. Thus each diagram contributes equally regardless of how many facts its reference figure description contains.

E.3 Gemini 3.1 Pro judge prompt

The prompt below is used for every candidate, with {reference} and {candidate} replaced by the corresponding descriptions.

```
SYSTEM INSTRUCTION

You evaluate the FACTUAL RECALL of a CANDIDATE description of a
geometry diagram against a REFERENCE FIGURE DESCRIPTION of the same
diagram.  Treat every fact in the REFERENCE FIGURE DESCRIPTION as
correct ground truth.

Step 1---Decompose the REFERENCE FIGURE DESCRIPTION into a list of
atomic, independently-checkable visual facts.  Each fact is ONE
self-contained piece of information, for example:
-- the existence of one shape (a circle, a triangle ABC, ...),
-- one labeled point and where it sits (P is the intersection of the
two chords; A, B, C are collinear),
```

```
-- ONE numeric measure or algebraic label printed on one element
(segment AB = 6; angle ABC = 40 degrees; an arc labeled ``3x-1''),
-- ONE geometric relationship marked in the figure (AB is parallel to
CD; the circle is tangent to line L at T; M is the midpoint of AB; a
right angle at C).
Split compound statements into separate facts.  Do NOT invent facts
the reference does not state.  Do NOT include the question, the task,
or any solving/answer.

Step 2---For EACH reference fact, decide whether the CANDIDATE
description conveys the SAME information CORRECTLY:
-- covered = true:  the candidate asserts this fact with a matching
value/relationship (wording, order, and phrasing may differ;
equivalent restatements count).
-- covered = false:  the candidate OMITS the fact, OR states it with
a DIFFERENT or CONTRADICTORY value (e.g., the reference says a
segment is 6 but the candidate says it is 8---a different value never
counts as covered).

Step 3---Count how many atomic visual facts the CANDIDATE contains
(every distinct fact it states, including any that are absent from or
contradict the reference).

Output a single JSON object and nothing else:
{"reference_facts":  [{"fact":  "<short statement of the reference
fact>", "covered":  true | false, "candidate_evidence":  "<short
quote from the candidate, or empty if omitted>"}],
"candidate_fact_count":  <integer>,
"notes":  "<one short sentence>"}
List every reference fact exactly once.  candidate_fact_count must be
an integer.
```

```
USER TURN

REFERENCE FIGURE DESCRIPTION:
{reference}

CANDIDATE description (to be scored against the reference):
{candidate}

Decompose the reference into atomic facts, mark which ones the
candidate conveys correctly, and count the candidate's total facts.
Respond with the JSON object now.
```

F Additional Experimental Results

F.1 Figure-description recall standard deviations

The main tables report mean figure-description recall over three candidate descriptions sampled independently for every test diagram at temperature 0.8 using seeds 1001, 1002, and 1003. Each candidate is scored against the same reference figure description using the judge in Section E. Because space permits only the mean in the main tables, Table 8 and Table 9 repeat those means with their standard deviations across the three sampled-description draws.

F.2 Geoperception breakdown by predicate

Tables 10–12 decompose the aggregate Geoperception pass@1 reported in the main tables by predicate category, with eight rollouts per example. All three tables report means only so that every row and model scale uses a consistent format. The predicate columns are angle classification (Angle), line comparison (Line), point on line (OnLine), equality (Equal), point on circle (OnCirc), parallelism (Parallel), and perpendicularity (Perp).

Table 8: Figure-description recall for Qwen3-VL-2B and Qwen3-VL-8B students. Entries are mean $\pm$ standard deviation across three independently sampled descriptions per diagram; the main tables report the corresponding means. Per column, the best result is ***bold*** and the second best is underlined.

Method	2B student	8B student
Base	0.567 ± 0.007	0.687 ± 0.002
SFT (rejection sampling)	0.605 ± 0.009	0.781 ± 0.008
SFT (describe-then-solve)	0.670 ± 0.010	0.798 ± 0.011
SFT (describe-figure)	$\underline{0.770 \pm 0.005}$	0.839 ± 0.007
SFT (mix: rejection + describe)	$\mathbf{0.784 \pm 0.006}$	$\underline{0.846 \pm 0.003}$
GRPO (plain prompt)	0.552 ± 0.009	0.676 ± 0.007
GRPO (figure-desc prompt)	0.415 ± 0.005	0.668 ± 0.011
GRPO (mix: plain + figure-desc)	0.560 ± 0.003	0.708 ± 0.005
Naive OPD (larger teacher)	0.655 ± 0.005	0.750 ± 0.013
Privileged OPD (larger teacher)	0.742 ± 0.004	$\mathbf{0.878 \pm 0.003}$
Minimal-edit (larger teacher, ours)	0.633 ± 0.003	0.727 ± 0.005
Minimal-edit (self-teacher, ours)	0.588 ± 0.004	0.745 ± 0.003

Table 9: Figure-description recall on the Qwen2.5-VL-7B backbone. Entries are mean $\pm$ standard deviation across three independently sampled descriptions per diagram; the main table reports the corresponding means. The best result is ***bold*** and the second best is underlined.

Model	Recall
Qwen2.5-VL-7B	0.431 ± 0.007
Perception-R1-7B	$\mathbf{0.541 \pm 0.002}$
NoisyRollout-7B	$\underline{0.500 \pm 0.014}$
PAPO-7B	0.470 ± 0.005
DeepEyes-7B	0.447 ± 0.003
Vision-R1-7B	0.486 ± 0.007
Minimal-edit (ours)	0.477 ± 0.009

G Teacher Prompts

G.1 Privileged teacher prompt

The conventional privileged teacher is conditioned on the diagram image, the collected figure description, and the original question. The student's rollout is then appended as the teacher-forced sequence whose token probabilities are scored.

```
<image>
{figure description}
{question}
```

G.2 Minimal-edit teacher prompt

For each incorrect student rollout, the frozen teacher is conditioned on the following prompt. During training, the teacher does not generate the requested rewrite; the prompt conditions a teacher-forced scoring pass over the student's existing tokens. The rewrite instruction defines the minimal-edit distribution against which the token-level k_1 estimates are computed.

```
<image>
{figure description}
{question}
--- REFERENCE CORRECT SOLUTION ---
{ref}
```

Table 10: Geoperception pass@1 by predicate for Qwen3-VL-2B students. Per column, the best result is ***bold*** and the second best is underlined.

Method	Angle	Line	OnLine	Equal	OnCirc	Parallel	Perp	Avg.
Base	0.129	0.158	0.093	0.191	0.405	0.128	0.147	0.179
SFT (rejection sampling)	0.165	0.231	0.088	0.254	0.421	0.360	0.188	0.244
SFT (describe-then-solve)	0.346	0.423	0.241	0.277	0.450	0.511	0.284	0.362
SFT (describe-figure)	0.341	0.307	0.207	**0.442**	0.355	0.608	0.253	0.359
SFT (mix)	0.172	0.248	0.168	0.276	0.413	0.527	0.217	0.289
GRPO (plain prompt)	0.262	0.290	0.217	0.243	0.392	0.287	0.262	0.279
GRPO (figure-desc prompt)	0.447	0.461	0.149	0.189	0.259	0.233	0.208	0.278
GRPO (mixed prompts)	0.475	0.478	0.206	0.264	0.107	0.383	0.246	0.308
Naive OPD (8B teacher)	0.207	0.283	0.204	0.249	0.575	0.393	0.259	0.310
Privileged OPD (8B teacher)	0.195	0.296	0.172	0.281	0.435	**0.650**	0.209	0.320
Minimal-edit (8B teacher)	0.511	**0.605**	0.248	0.311	**0.580**	0.557	**0.372**	**0.455**
Minimal-edit (self-teacher)	**0.535**	0.601	**0.252**	0.329	0.504	0.507	0.340	0.438

Table 11: Geoperception pass@1 by predicate for Qwen3-VL-8B students. Per column, the best result is ***bold*** and the second best is underlined.

Method	Angle	Line	OnLine	Equal	OnCirc	Parallel	Perp	Avg.
Base	0.349	0.454	0.004	0.477	0.755	0.615	0.406	0.437
SFT (rejection sampling)	0.345	0.408	0.033	0.448	0.693	0.645	0.398	0.424
SFT (describe-then-solve)	0.468	0.595	0.128	0.459	0.702	0.720	0.428	0.500
SFT (describe-figure)	0.356	0.418	**0.137**	0.547	0.750	0.712	0.309	0.461
SFT (mix)	0.376	0.466	0.110	0.467	0.767	0.761	0.445	0.484
GRPO (plain prompt)	**0.634**	**0.754**	0.012	0.546	0.733	0.590	0.443	0.530
GRPO (figure-desc prompt)	0.537	0.598	0.028	**0.605**	0.739	0.625	0.406	0.506
GRPO (mixed prompts)	0.584	0.616	0.000	0.469	0.760	0.696	0.377	0.500
Naive OPD (32B teacher)	0.418	0.439	0.007	0.388	0.772	0.798	0.438	0.466
Privileged OPD (32B teacher)	0.399	0.467	0.013	0.466	**0.791**	**0.849**	**0.489**	0.496
Minimal-edit (32B teacher)	0.567	0.717	0.016	0.588	0.762	0.765	0.459	**0.553**
Minimal-edit (self-teacher)	0.582	0.715	0.037	0.562	0.759	0.743	0.453	0.550

```
--- END REFERENCE SOLUTION ---

--- STUDENT SOLUTION ---
{student}
--- END STUDENT SOLUTION ---

Rewrite the student solution above into a correct solution by making
the SMALLEST possible change:  copy it VERBATIM and alter only the
specific spans that are actually wrong, keeping every correct
sentence, number, and symbol exactly as written, in the student's
original style and voice.  Write ONLY the corrected solution itself
as a single self-contained solution that reads as if it were correct
from the start.  Do NOT refer to ''the student'', do not state that
anything was wrong, missing, changed, or corrected, and add no
commentary.  End with the final answer in \boxed{}.
```

H Qualitative Examples

Figure 3 illustrates how minimal-edit conditioning concentrates corrective-token selection around a perceptual error. We provide two additional examples spanning a downstream arithmetic mistake and a second diagram-reading error. In each visualization, both teachers score the same fixed student

Table 12: Geoperception pass@1 by predicate on the Qwen2.5-VL-7B backbone. Per column, the best result is ***bold*** and the second best is underlined.

Model	Angle	Line	OnLine	Equal	OnCirc	Parallel	Perp	Avg.
Qwen2.5-VL-7B	0.382	0.475	0.178	0.258	0.404	0.365	0.247	0.330
Perception-R1-7B	0.526	**0.600**	0.170	0.293	**0.568**	**0.536**	0.248	**0.420**
NoisyRollout-7B	0.490	0.589	0.218	0.273	0.507	0.506	0.287	0.410
PAPO-7B	0.453	0.571	0.162	0.249	0.465	0.334	0.223	0.351
DeepEyes-7B	0.442	0.569	0.155	0.233	0.478	0.402	0.265	0.363
Vision-R1-7B	**0.534**	0.573	0.132	**0.309**	0.488	0.426	0.279	0.392
Minimal-edit (ours)	0.480	0.544	**0.263**	0.278	0.473	0.531	**0.291**	0.409

rollout; only their conditioning differs. Red tokens have $k_t > 0$ and would enter the corrective mask, whereas green tokens have $k_t \leq 0$ and are discarded.

To make the behavior induced by the minimal-edit prompt more concrete, we also show a hypothetical minimal-edit rollout for each example. These edited responses are generated only for exposition. During training, the minimal-edit teacher is never rolled out: it receives the description, reference solution, and fixed student response, then scores the tokens of that existing response in a teacher-forced pass. The textual edits shown below are therefore neither training targets nor inputs to the loss. They illustrate what the teacher could generate under the same minimal-edit prompt, whereas the actual corrective mask is determined solely by the token-level k_1 estimates on the unchanged student rollout.

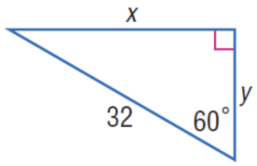
Figure 7 shows the hypothetical edit corresponding to Figure 3. The edit corrects the initial assignment of the side labeled 32 and the hypotenuse, removes the derivation that follows from that misreading, and changes the final answer from $32\sqrt{3}$ to 16. The retained text demonstrates the prompt's preservation objective, while the revision makes explicit the correction suggested by the teacher-disfavored positions.

Figure 8 considers a response whose geometric setup is correct but whose final arithmetic is not. The student identifies AB as the side adjacent to the 42° angle and correctly writes $AB = 56\cos(42^\circ)$, but evaluates the product as 41.2 rather than 41.6. The conventional privileged teacher produces positive k_1 estimates throughout the otherwise usable derivation. By contrast, the minimal-edit teacher assigns at least as much probability as the student to most of those tokens and concentrates its positive estimate near the incorrect multiplication. Thus, this corrective-token selection pattern is not limited to perceptual mistakes: it also concentrates selected positions near a downstream numerical error while preserving the correct setup that precedes it.

The hypothetical edit in Figure 9 preserves the correct geometric setup and cosine relation, removes an unnecessary detour through sine, and corrects the product and rounded answer. This response-level view agrees with the token-level visualization: the substantive correction is numerical rather than geometric.

Figure 10 presents a second perceptual failure. The diagram labels a diameter of 10 cm, but the student treats that value as the radius. Its subsequent arc-length derivation is consistent with this incorrect premise, producing $65\pi/9 \approx 22.69$ instead of the value obtained with $r = 5$. A conventional privileged teacher assigns positive k_1 estimates across much of the response, including the downstream calculation. The minimal-edit teacher instead preserves most of that calculation and focuses its strongest corrective signal on the initial identification of the radius. This is the desired behavior: correcting the misread quantity makes the otherwise coherent derivation reusable rather than treating every consequence of the initial error as an independent mistake.

Figure 11 shows the corresponding hypothetical edit. It replaces "radius" with "diameter," introduces the conversion $r = 10/2 = 5$, and updates the subsequent calculation and final answer to $65\pi/18 \approx 11.34$. The edit makes visible how a localized perceptual correction propagates through an otherwise valid solution structure.



Question. Find y.

Internal Monologue:

I see a right triangle here, with a 90-degree angle indicated by the pink square. The hypotenuse is labeled as ~~'x',~~'32', and the side adjacent to the 60-degree angle is ~~32.~~the vertical side 'y'. The angle opposite the side labeled 'y' is 60 degrees. I need to find the length of side 'y'.

Since it's a right triangle with a 60-degree angle, the other acute angle must be 30 degrees because the angles in a triangle add up to 180 degrees. I can use the properties of a 30-60-90 triangle here. In a 30-60-90 triangle, the sides are in a specific ratio: the side opposite the 30-degree angle is half the hypotenuse, and the side opposite the 60-degree angle is half the hypotenuse times the square root of 3.

But wait, in this triangle, I don't know if the side opposite the 60-degree angle is the one labeled 'y'. Let me look again at the diagram. The angle at the bottom is 60 degrees. So the side opposite the 60-degree angle is the one labeled 'y'. The hypotenuse is ~~'x',~~'32', and the side adjacent to the 60-degree angle is ~~32.~~'y'. Since it's a right triangle, I can use trigonometric ratios.

I can use the cosine function because the side adjacent to the angle ~~(32)~~('y') and the hypotenuse ~~(x)~~('32') are known. So, cos(60°) = adjacent/hypotenuse = ~~32/x.~~y/32. I know that cos(60°) is equal to 0.5. So, 0.5 = ~~32/x.~~y/32. Solving for ~~x, I get x = 64. Now that I have the hypotenuse, I can use the sine function to find 'y'. Sin(60°) = opposite/hypotenuse = y/x. So, y = x * sin(60°) = 64 * sin(60°). Since sin(60°) is equal to √3/2,~~y, I get y = 6432 * ~~(√3/2)~~0.5 = ~~32√3.~~16.

Alternatively, ~~So~~I can identify the ~~value~~triangle ~~of~~as ~~'y'~~a 30-60-90 triangle. The angle at the bottom is ~~32√3.~~60°, so the angle at the top is 30°, and the side opposite that 30° angle is 'y'. In a 30-60-90 triangle, the side opposite the 30° angle is half the hypotenuse. The hypotenuse is 32, so y = (1/2) * 32 = 16.

So, y = 16.

Final Answer:

~~\boxed{32\sqrt{3}}~~\boxed{16}

Figure 7: Hypothetical minimal-edit rollout for the leg–hypotenuse example in Figure 3. Red strikethrough marks text removed from the student response, and green marks inserted text. The edit identifies 32 as the hypotenuse, revises the dependent reasoning, and changes the answer from $32\sqrt{3}$ to 16 while preserving text that remains applicable. This rollout is generated only for visualization; training does not decode an edited response and instead scores the original student tokens under the minimal-edit prompt.

Privileged Teacher

Internal Monologue:I am given a rectangle ABCD with diagonals each measuring 56 feet. The angle at point A, ∠BAC, is 42 degrees. I need to find the length of AB. First, I know that in a rectangle, the diagonals are equal in length and bisect each other. So, diagonal AC is 56 feet, and since the diagonals bisect each other, each half of the diagonal is 28 feet. But I don't need that immediately. I'm focusing on triangle ABC. Since ABCD is a rectangle, angle at B (i.e., ∠ABC) is a right angle. That means triangle ABC is a right triangle with a right angle at B, and the hypotenuse is the diagonal AC, which is 56 feet.

In right triangle ABC, ∠ABC = 90°. The angle at A, ∠BAC = 42°. The side opposite to ∠BAC is BC, and the side adjacent to ∠BAC is AB. I need to find the length of AB. The hypotenuse is AC = 56 ft, and ∠BAC = 42°. Since in a right triangle, the side opposite the angle is AC, but that's the hypotenuse. So, I can use trigonometric ratios. Specifically, I can use the sine function. In triangle ABC: - ∠BAC = 42° - Hypotenuse = AC = 56 ft - Opposite side = BC - Adjacent side = AB I need to find AB. The sine of an angle is the ratio of the length of the opposite side to the hypotenuse. So, sin(∠BAC) = opposite / hypotenuse sin(42°) = BC / 56

But I don't need BC. I need AB. AB is the adjacent side. So, I can use cosine. Cosine of ∠BAC = adjacent / hypotenuse cos(42°) = AB / 56 Therefore, AB = 56 × cos(42°) Now I need to calculate this. Using a calculator: cos(42°) ≈ 0.7431

So AB = 56 × 0.7431 ≈ 41.2036 Rounded to the nearest tenth of a foot, AB ≈ 41.2 ft.

Final Answer:\boxed{41.2} feet

Minimal Edit Teacher

Internal Monologue:I am given a rectangle ABCD with diagonals each measuring 56 feet. The angle at point A, ∠BAC, is 42 degrees. I need to find the length of AB. First, I know that in a rectangle, the diagonals are equal in length and bisect each other. So, diagonal AC is 56 feet, and since the diagonals bisect each other, each half of the diagonal is 28 feet. But I don't need that immediately. I'm focusing on triangle ABC. Since ABCD is a rectangle, angle at B (i.e., ∠ABC) is a right angle. That means triangle ABC is a right triangle with a right angle at B, and the hypotenuse is the diagonal AC, which is 56 feet.

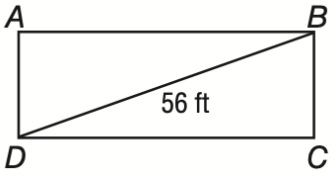
In right triangle ABC, ∠ABC = 90°. The angle at A, ∠BAC = 42°. The side opposite to ∠BAC is BC, and the side adjacent to ∠BAC is AB. I need to find the length of AB. The hypotenuse is AC = 56 ft, and ∠BAC = 42°. Since in a right triangle, the side opposite the angle is AC, but that's the hypotenuse. So, I can use trigonometric ratios. Specifically, I can use the sine function. In triangle ABC: - ∠BAC = 42° - Hypotenuse = AC = 56 ft - Opposite side = BC - Adjacent side = AB
I need to find AB. The sine of an angle is the ratio of the length of the opposite side to the hypotenuse. So, sin(∠BAC) = opposite / hypotenuse
sin(42°) = BC / 56

But I don't need BC. I need AB. AB is the adjacent side. So, I can use cosine. Cosine of ∠BAC = adjacent / hypotenuse
cos(42°) = AB / 56 Therefore, AB = 56 × cos(42°) Now I need to calculate this. Using a calculator: cos(42°) ≈ 0.7431

So AB = 56 × 0.7431 ≈ 41.2036 Rounded to the nearest tenth of a foot, AB ≈ 41.2 ft.

Final Answer:\boxed{41.2} feet

Figure 8: Edit conditioning concentrates corrective-token selection near a numerical error. Token-level k_1 estimates for an incorrect solution that selects the correct relation, $AB = 56\cos(42°)$, but evaluates it incorrectly as 41.2 rather than 41.6. The conventional privileged teacher assigns a positive k_1 estimate to many otherwise acceptable wording choices (top), whereas minimal-edit conditioning assigns high probability to most of the student's derivation and concentrates the corrective signal near the erroneous calculation (bottom). Colors follow Figure 3.



Question. The diagonals of rectangle *ABCD* each have a length of 56 feet. If $m\angle BAC = 42°$, what is the length of $\overline{AB}$ to the nearest tenth of a foot?

Internal Monologue:

I am given a rectangle ABCD with diagonals each measuring 56 feet. The angle at point A, ∠BAC, is 42 degrees. I need to find the length of AB.

First, I know that in a rectangle, the diagonals are equal in length and bisect each other. So, diagonal AC is 56 feet, and since the diagonals bisect each other, each half of the diagonal is 28 feet. But I don't need that immediately.

I'm focusing on triangle ABC. Since ABCD is a rectangle, angle at B (i.e., ∠ABC) is a right angle. That means triangle ABC is a right triangle with a right angle at B, and the hypotenuse is the diagonal AC, which is 56 feet. In right triangle ABC, ∠ABC = 90°. The angle at A, ∠BAC = 42°. The side opposite to ∠BAC is BC, and the side adjacent to ∠BAC is AB.

I need to find the length of AB. The hypotenuse is AC = 56 ft, and ∠BAC = 42°. Since in a right triangle, the side opposite the angle is AC, but that's the hypotenuse. So, I can use trigonometric ratios. Specifically, I can use the sine function.

In triangle ABC: - ∠BAC = 42° - Hypotenuse = AC = 56 ft - Opposite side = BC - Adjacent side = AB

I need to find AB. ~~The sine of an angle is the ratio of the length of the opposite side to the hypotenuse.~~

~~So, sin(∠BAC) = opposite / hypotenuse~~
~~sin(42°) = BC / 56~~

~~But I don't need BC. I need AB.~~ AB is the adjacent side. So, I can use cosine.

Cosine of ∠BAC = adjacent / hypotenuse
cos(42°) = AB / 56

Therefore, AB = 56 × cos(42°) Now I need to calculate this. Using a calculator: cos(42°) ≈ 0.7431

So AB = 56 × 0.7431 ≈ ~~41.2036~~41.6136

Rounded to the nearest tenth of a foot, AB ≈ ~~41.2~~41.6 ft. **Final Answer:** ~~\boxed{41.2} feet~~\boxed{41.6}

Figure 9: Hypothetical minimal-edit rollout for the arithmetic-error example in Figure 8. Red strikethrough marks removed text, and green marks inserted text. The edit preserves $AB = 56\cos(42°)$, corrects the product from 41.2036 to 41.6136, and changes the rounded answer from 41.2 to 41.6 feet. This rollout is generated only for visualization; the training procedure scores the fixed student response without decoding this revision.

Privileged Teacher

Internal Monologue:I need to find the length of the arc $\widehat{ZY}$ on a circle. The given information includes the radius of the circle, which is 10 cm, and the central angle, which is 130 degrees. I recall that the formula to find the length of an arc is: $$\text{Arc Length} = \frac{\theta}{360^\circ} \times 2\pi r$$ where θ is the central angle in degrees, and r is the radius of the circle. Since I have all the necessary values, I can proceed to plug them into the formula. First, I plug in the values:

$$\text{Arc Length} = \frac{130^\circ}{360^\circ} \times 2\pi \times 10$$ I simplify the fraction: $$\frac{130}{360} = \frac{13}{36}$$

Now, I calculate:$$\text{Arc Length} = \frac{13}{36} \times 2\pi \times 10$$
$$= \frac{13}{36} \times 20\pi$$
$$= \frac{260\pi}{36}$$

I can simplify this fraction:$$\frac{260\pi}{36} = \frac{65\pi}{9}$$

Now, I calculate the numerical value: $$\frac{65\pi}{9} \approx \frac{65 \times 3.1416}{9} \approx \frac{204.204}{9} \approx 22.6893$$ Rounding to the nearest hundredth:$$\approx 22.69$$ So, the length of $\widehat{ZY}$ is approximately 22.69 cm.

Final Answer:\boxed{22.69}

Minimal Edit Teacher

Internal Monologue:I need to find the length of the arc $\widehat{ZY}$ on a circle. The given information includes the radius of the circle, which is 10 cm, and the central angle, which is 130 degrees. I recall that the formula to find the length of an arc is: $$\text{Arc Length} = \frac{\theta}{360^\circ} \times 2\pi r$$ where θ is the central angle in degrees, and r is the radius of the circle. Since I have all the necessary values, I can proceed to plug them into the formula. First, I plug in the values:

$$\text{Arc Length} = \frac{130^\circ}{360^\circ} \times 2\pi \times 10$$ I simplify the fraction: $$\frac{130}{360} = \frac{13}{36}$$

Now, I calculate:

$$\text{Arc Length} = \frac{13}{36} \times 2\pi \times 10$$

$$= \frac{13}{36} \times 20\pi$$

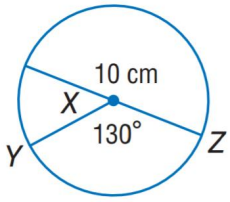
$$= \frac{260\pi}{36}$$

I can simplify this fraction: $$\frac{260\pi}{36} = \frac{65\pi}{9}$$

Now, I calculate the numerical value: $$\frac{65\pi}{9} \approx \frac{65 \times 3.1416}{9} \approx \frac{204.204}{9} \approx 22.6893$$ Rounding to the nearest hundredth: $$\approx 22.69$$ So, the length of $\widehat{ZY}$ is approximately 22.69 cm.

Final Answer:\boxed{22.69}

Figure 10: Edit conditioning concentrates corrective-token selection near a diameter–radius error. The student misreads the labeled diameter of 10 cm as the radius and consequently computes the arc length using $r = 10$. The conventional privileged teacher assigns positive k_1 estimates throughout the response (top), whereas the minimal-edit teacher concentrates the corrective signal on the tokens that identify 10 cm as the radius (bottom). Colors follow Figure 3.



Question. Find the length of ZY. Round to the nearest hundredth.

```
**Internal Monologue:**

I need to find the length of the arc $\widehat{ZY}$ on a circle. The given information includes the ~~radius~~diameter of the circle, which
is 10 cm, and the central angle, which is 130 degrees. I recall that the formula to find the length of an arc is:

$$\text{Arc Length} = \frac{\theta}{360^\circ} \times 2\pi r$$

Where $\theta$ is the central angle in degrees, and $r$ is the radius of the circle. Since I have the diameter, I must divide it by 2 to
get the radius: $r = 10 / 2 = 5$ cm. I have all the necessary values, so I can proceed to plug them into the formula.

First, I plug in the values: $$\text{Arc Length} = \frac{130^\circ}{360^\circ} \times 2\pi \times ~~10~~5$$ I simplify the fraction:
$$\frac{130}{360} = \frac{13}{36}$$ Now, I calculate: $$\text{Arc Length} = \frac{13}{36} \times 2\pi \times ~~10~~5$$

$$= \frac{13}{36} \times ~~20\pi~~10\pi$$

$$= ~~\frac{260\pi}{36}~~\frac{130\pi}{36}$$

I can simplify this fraction: $$~~\frac{260\pi}{36}~~\frac{130\pi}{36} = ~~\frac{65\pi}{9}~~\frac{65\pi}{18}$$

Now, I calculate the numerical value:$$~~\frac{65\pi}{9}~~\frac{65\pi}{18} \approx \frac{65 \times ~~3.1416}{9}~~3.1416}{18} \approx
~~\frac{204.204}{9}~~\frac{204.204}{18} \approx ~~22.689~~11.3446$$

Rounding to the nearest hundredth:$$\approx ~~22.69~~11.34$$

So, the length of $\widehat{ZY}$ is approximately ~~22.69~~11.34 cm.

**Final Answer:**

~~\boxed{22.69}~~\boxed{11.34}
```

Figure 11: Hypothetical minimal-edit rollout for the diameter–radius example in Figure 10. Red strikethrough marks removed text, and green marks inserted text. The edit recognizes 10 cm as the diameter, sets $r = 5$, and updates the arc length from $65\pi/9 \approx 22.69$ to $65\pi/18 \approx 11.34$. This rollout is generated only for visualization; during training, the teacher generates no revised response and only scores the original student tokens.

Together with Figure 3, these examples cover three regimes: a localized misidentification of a triangle's leg and hypotenuse, a localized arithmetic error after a correct setup, and a diameter–radius reading error whose consequences propagate through an otherwise valid derivation. Across all three, conventional privileged conditioning responds to broad distributional differences, whereas minimal-edit conditioning better preserves tokens that do not need to change. The corrective mask then converts that response-specific preference into a sparse training signal.