VERI-GPS: LEARNING TO COORDINATE GEOMETRIC PROOF AND CALCULATION WITH VERIFICATION AND SOLVER FEEDBACK

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) can solve geometric proof problems with the support of symbolic solvers. However, many geometry problems require more than correct geometric relations, as proof and calculation must also support each other. We introduce Veri-GPS, an interaction learning framework that learns subgoal policies from symbolic solver feedback to coordinate proof and calculation within a given interaction budget. An extended and accelerated Newclid backend enables geometric proof and exact calculation to share verified results. When execution is blocked, goal-related diagnostic feedback identifies missing premises and quantities along candidate rule or formula paths to limit LLM trial and error. To train the LLM to use solver feedback, we construct a dual-source pipeline combining teacher–solver interaction trajectories with Goal Augmentation trajectories. Both sources are split into turn-level training data for supervised fine-tuning (SFT) of Qwen2.5-VL-7B-Instruct, providing a cold start for the interaction policy. We introduce turn-level Dynamic-Difficulty Reinforcement Learning (DDRL), combining adaptive sampling across fixed difficulty buckets with derivation-use rewards for intermediate results used in subsequent reasoning. Across 3,082 problems from Geometry3K, PGPS9K, the planar geometry subset of MathVerse, GeoQA+, and GeoQA, Veri-GPS achieves a problem-weighted pass@1 of 82.12%. Compared with standalone symbolic execution and Qwen3-VL-32B-Instruct direct answering, Veri-GPS improves performance by 10.25 and 17.39 percentage points, respectively. Within Veri-GPS, full feedback yields a 1.27-point gain over reduced feedback at inference with the trained policy fixed, while DDRL improves pass@1 by 3.57 points over the SFT checkpoint.

1 INTRODUCTION

Geometry problem solving (GPS) requires integrating diagrammatic relations, logical proof, and numerical calculation (Wu et al., 2024; Zhang et al., 2025). Neuro-symbolic methods combine the exploratory capabilities of large language models (LLMs) with rigorous symbolic execution, advancing geometric proof (Trinh et al., 2024; Zhao et al., 2025a). Mixed solving requires both a backend that supports the necessary proof and calculation operations and a policy that selects useful subgoals within the interaction budget.

Figure 1 illustrates subgoal switching in a diameter problem with chord $AB = 10$ and inscribed angle $\angle ACB = 45°$. After a direct attempt to compute BD fails, the LLM switches to computing the central angle $\angle AOB = 90°$. It then returns to BD, obtains $BD = 10$, and completes the diameter calculation. AlphaGeometry2 also identifies subproblem decomposition as a promising direction for further progress (Chervonyi et al., 2025). Together, these observations highlight a broader challenge in GPS and raise a natural question: **How can LLMs organize and revise subgoals to coordinate geometric proof and calculation within a given interaction budget, without relying on extensive trial and error?**

To address this question, we introduce Veri-GPS, an interaction learning framework that learns subgoal policies from symbolic solver feedback. We extend the Newclid backend to support geometric proof and exact calculation while reducing redundant matching and deduction costs (Sicca et al., 2024).

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.

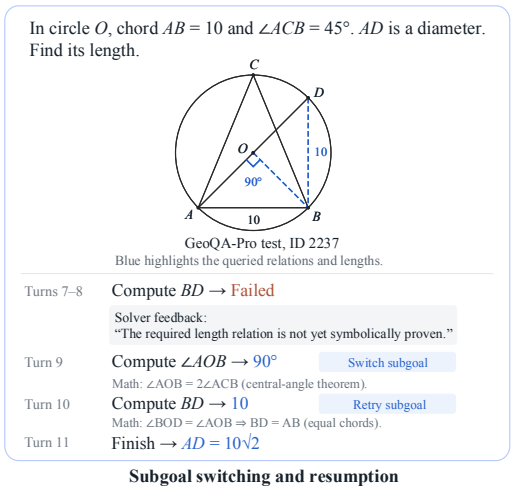


Figure 1: Subgoal switching and resumption. GeoQA+ test, ID 2237, turns 7–11.

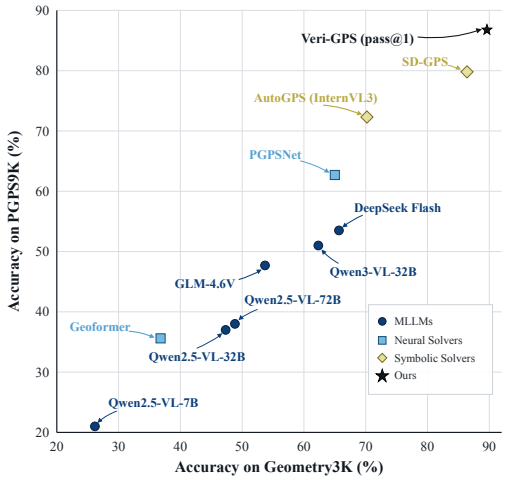


Figure 2: Performance comparison among existing methods.

When execution is blocked, goal-related diagnostics retrieve candidate rules or formulas and bind them to geometric objects. The diagnostic module then uses the verified state to identify missing relational premises and intermediate quantities. Execution status, reasons for unsuccessful execution, newly verified facts, and available diagnostics together form the feedback that guides the LLM's next decision.

To train LLMs to use solver feedback, we use teacher–solver interaction trajectories that record subgoal choices and revisions during solving. To broaden this supervision, Goal Augmentation derives alternative goals and dependency-linked proof–calculation steps from existing configurations, supplementing supervision that is sparse in teacher trajectories. Both trajectory sources are converted into turn-level training data. Each instance takes the problem context, current state, and preceding interaction history as input, with the rationale and executable request of a successful action as the supervision target.

Supervised fine-tuning (SFT) of Qwen2.5-VL-7B-Instruct on these data provides a cold start for the interaction policy. We then introduce Dynamic-Difficulty Reinforcement Learning (DDRL), a training scheme built on Turn-PPO (Li et al., 2026b). DDRL incorporates adaptive sampling over fixed difficulty buckets, using online success rates and problem coverage to match training tasks to the LLM's capability. During autonomous interaction, terminal rewards encourage final goal completion, while derivation-use rewards credit intermediate results used in subsequent reasoning. Reliable admission rejections provide local policy constraints on invalid requests. These signals train the LLM to use symbolic solver feedback when organizing and revising subgoals. After training, the LLM and symbolic solver form the Veri-GPS system, which organizes step-by-step solutions along verified derivation dependencies.

We evaluate Veri-GPS on 3,082 problems across Geometry3K, PGPS9K, the planar geometry subset of MathVerse, GeoQA+, and GeoQA. Veri-GPS achieves a problem-weighted pass@1 of 82.12%, exceeding standalone symbolic execution and Qwen3-VL-32B-Instruct direct answering by 10.25 and 17.39 percentage points, respectively. Figure 2 summarizes the Geometry3K and PGPS9K comparisons. Within Veri-GPS, we further examine feedback and policy training: full feedback outperforms reduced feedback by 1.27 points at inference with the trained policy fixed, while DDRL improves pass@1 by 3.57 points over the SFT checkpoint. For the test problems each model solves, DDRL proposes an average of 5.464 requests per problem, compared with 6.752 for SFT. Alongside DDRL's higher completion rate, this suggests more efficient use of the interaction budget.

2 Related Work

2.1 Symbolic Geometry Solving and Hybrid Execution

AlphaGeometry's DDAR combines deduction with algebraic reasoning (Trinh et al., 2024). AlphaGeometry2 expands the language and accelerates symbolic execution (Chervonyi et al., 2025). Newclid provides a modular foundation with metric support (Sicca et al., 2024), while Euclid-Omni's Euclidea couples deduction with equation solving (Li et al., 2026c). When these solvers are invoked across turns, repeated matching and deduction over an expanding fact set can accumulate execution costs.

2.2 Geometry Solving Strategies

Inter-GPS and FGeo-HyperGNet learn theorem selection (Lu et al., 2021; Zhang et al., 2025), while E-GPS decomposes goals into prerequisites (Wu et al., 2024). GF-Reasoner interleaves natural-language reasoning with formal programs (Yang et al., 2025), and InternGeometry uses execution results to guide proofs and auxiliary constructions (Zhao et al., 2025a). Execution feedback can reveal a blocked path without specifying the next useful subgoal, leaving the policy to decide how to continue within a limited interaction budget.

2.3 Interaction Policy Learning

E-GPS, NeSyGeo, and Euclid-Omni generate geometry goals and trajectories from theorem applications or synthetic configurations (Wu et al., 2024; 2025; Li et al., 2026c). SD-GPS learns autoformalization through solver-guided SFT and RL and uses proof-state diagnostics to propose verified local lemmas (Li et al., 2026a). Its training rewards focus on formalization validity and solvability. Beyond formalization, interactive solving poses a different learning target: adapting subsequent decisions to the state and feedback produced during execution.

ReTool combines supervised initialization with online RL for tool use (Feng et al., 2025). InternGeometry adapts task difficulty to stage-wise performance (Zhao et al., 2025a), while Turn-PPO estimates turn-level advantages (Li et al., 2026b). Successful execution alone does not establish whether an intermediate result helps solve the original problem, making delayed credit assignment important in multi-turn proof–calculation tasks.

3 Method

3.1 Problem Formulation and Framework Overview

Following AutoGPS (Ping et al., 2025), Veri-GPS converts problem text W and diagram I into a geometric DSL D specifying objects and givens through automatic formalization and multimodal LLM alignment. The agent selects and revises proof–calculation subgoals toward the final goal $g^\star$ within an interaction budget.

The strict state S_t stores objects, verified relations, exact quantities, and derivation dependencies. Its visible summary v_t and request–feedback history $H_{<t}$ enter the observation o_t, from which the policy generates a rationale ℓ_t and one request a_t (Figure 3A):

$$\begin{aligned} o_t &= \mathrm{Render}(W, I, D, g^\star, v_t, H_{<t}), \\ y_t &= (\ell_t, a_t) \sim \pi_\theta(\cdot \mid o_t). \end{aligned} \tag{1}$$

The rationale supports deliberation and explanation; only a_t is submitted for execution. Requests cover proof, calculation, supported auxiliary operations, and final-goal verification. Proposal turns and admitted requests have separate budgets; pre-execution rejection consumes only a proposal turn. With q_t comprising S_t, $H_{<t}$, and budget counters, the controller–solver loop (Figure 3B) updates the state and returns feedback:

$$(q_{t+1}, f_t) = \mathrm{Step}(q_t, a_t). \tag{2}$$

Before execution, the controller checks request format, object references, admissibility, and remaining budget. Feedback f_t reports execution status, execution or rejection reasons, and newly verified facts.

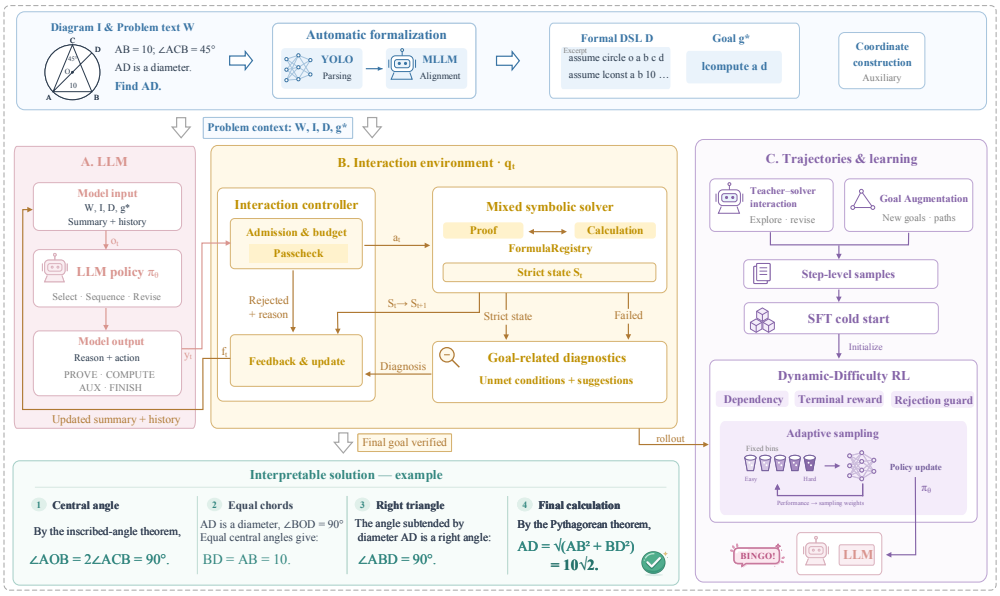


Figure 3: Veri-GPS architecture and training pipeline. (A) Request generation. (B) Request checking, execution, and feedback. (C) Dual-source SFT followed by DDRL. The bottom illustrates a solution based on verified relations for the problem in Figure 1.

For unresolved goals, it also includes available goal-related diagnostics identifying missing relational premises or intermediate quantities along candidate rule or formula paths. The updated state summary and request–feedback history guide the agent in revising subgoals, supplying missing conditions, or retrying previously blocked goals.

Teacher–solver trajectories supervise subgoal selection and revision, while Goal Augmentation adds dependency-linked proof–calculation steps. These data support turn-level SFT followed by Dynamic-Difficulty Reinforcement Learning (DDRL; Figure 3C). At inference, interaction ends when $g^\star$ is strictly verified or a stopping condition is reached. Recorded dependencies support step-by-step explanations for solved problems.

3.2 Hybrid Symbolic Solving and Goal Diagnostics

In multi-turn solving, results from one request must remain available to subsequent proof or calculation. When execution is blocked, the solver should also expose conditions still needed along relevant candidate paths. Building on Newclid's deductive database and algebraic reasoning (Sicca et al., 2024), we extend its support for coordinated proof and calculation and introduce goal-related diagnostics.

Verified proof–calculation coordination. We extend Newclid's metric representations so that objects in geometric relations correspond consistently to variables used in calculation. A formula registry records the objects, geometric premises, and equations required for computation. The solver activates a formula only after its relevant premises have been verified. For example, a base–height area formula requires the altitude foot to lie on the base line and the altitude to be perpendicular to that line. Verified exact quantities and relations are written back to the strict state for later formula solving or relational deduction. Dependency records retain the supporting facts, rules, or equations available from the execution path.

As verified facts accumulate, irrelevant object combinations and repeated scans increase the cost of matching and deduction across requests. Fact indices retrieve candidate relations for rule premises, while specialized variable binding narrows object combinations for high-arity rules such as congruence and similarity. Newly added facts reactivate only the rules they may affect, reducing redundant propagation. Lazy topology supplies topology-related information on demand.

Goal-related diagnostics. Status feedback alone leaves the LLM to infer what to try next, potentially increasing trial-and-error interactions. For an unresolved goal, the diagnostic module retrieves supported proof-diagnostic rules or equation and formula paths whose conclusions or outputs match the target. It binds candidate paths to the current geometric objects and checks their premises against the strict state. Unverified relations suggest possible proof goals, while unknown quantities suggest possible calculation goals. The findings retain the supporting path and potential use of each missing condition, so the LLM can assess how establishing it may advance the blocked goal. The reported conditions are specific to each candidate path rather than required by every solution.

Unlike newly verified facts, diagnostic suggestions identify candidate prerequisites and require verification before entering the strict state. Coordinates can prefilter candidate bindings, but numerical agreement alone does not establish an angle value, an equal-length relation, or an exact quantity. If no usable diagnosis is available, feedback retains the actual execution reason. The LLM remains responsible for selecting and revising subgoals from these clues.

Appendix G.3 records a local-area subgoal adjustment after missing-area-relation feedback.

3.3 Dual-Source Supervision for Subgoal Decisions

Our dual-source supervision uses teacher–solver interaction trajectories as its basis and supplements them through Goal Augmentation. Teacher trajectories record subgoal choices and revisions in response to execution feedback. Goal Augmentation derives additional goals and dependency-linked proof–calculation steps from existing configurations to supplement teacher supervision. Both sources are converted into turn-level training data.

Teacher-interaction trajectories. A teacher LLM (doubao-seed-2-0-pro-260215) proposes requests from the problem context, visible state, and history, then selects subsequent goals in response to solver feedback. The trajectories retain requests, execution outcomes, and state updates, including unsuccessful attempts. Selected successful actions serve as supervision targets for subgoal decisions. Earlier failed and rejected requests and their feedback remain in the input history for later turns. This allows the LLM to learn how to revise its choices after unsuccessful attempts.

Goal Augmentation. Teacher exploration is limited by the original goals and the paths it visits. To broaden supervision, an offline generation process preserves each configuration and its givens while removing the original goal. DDAR first propagates the givens to a fixed point under the available objects and rules. Coordinate-based candidate relations are then checked against established conditions in isolated verification workspaces. Verified relations extend separate proof branches and trigger further deduction, exposing new proof goals and their dependencies. Formula instances available in these states are solved in separate calculation workspaces to derive exact quantities and record their supporting relations and intermediate values. The calculation workspaces do not write their results back into the proof branches. Dependency extraction organizes candidate proof–calculation trajectories and filters duplicate goals, cyclic dependencies, and steps lacking established inputs. Templates provide the corresponding action rationales, and only eligible successful actions enter the turn-level supervision. Algorithm 1 summarizes this generation procedure.

Algorithm 1 Goal Augmentation from Existing Configurations

Require: original problem X, visual hints V, rules R, search budget B

$D \leftarrow \mathrm{RemoveGoal}(X)$
$S_0 \leftarrow \mathrm{SaturateDDAR}(\mathrm{Build}(D, V, R))$
if S_0 is not certified **then return** $\varnothing$
$C \leftarrow \mathrm{RecallNumericalRelations}(S_0)$, $W \leftarrow \{S_0\}$, $G \leftarrow \varnothing$
while $W \neq \varnothing$ and budget remains **do**
  $S \leftarrow \mathrm{Pop}(W)$, $\Delta F \leftarrow \mathrm{NewFormulas}(S)$
  $G \leftarrow G \cup \mathrm{DiscoverCompute}(S, \Delta F)$
  $P \leftarrow \mathrm{VerifyRelations}(S, C)$
  $H \leftarrow \mathrm{NewDDARGoalsOnCopy}(S, P, R)$
  $G \leftarrow G \cup H$, $W \leftarrow W \cup \mathrm{ValidChildren}(S, P, H)$
end while
return $\mathrm{SelectAndExplain}(G)$

Both sources are converted into turn-level training data $\mathcal{D}_{\mathrm{turn}}$. Each instance pairs the pre-action observation o_t, including prior solver feedback, with the rationale and request of an eligible successful action. The current action's outcome is used for sample selection and later observations, not included in its own input. Rationales are supervision annotations, not solver feedback or necessarily verbatim

execution-time reasoning. The resulting dataset contains 32,521 successful-action training instances, with data composition and trajectory examples provided in Appendices A and D.

3.4 Dynamic-Difficulty Reinforcement Learning

We initialize the policy by fine-tuning Qwen2.5-VL-7B-Instruct on $\mathcal{D}_{\text{turn}}$ with the objective

$$\mathcal{L}_{\text{SFT}} = \mathcal{L}_{\text{full}} + \lambda_{\text{act}}\mathcal{L}_{\text{act}}, \tag{3}$$

where $\mathcal{L}_{\text{full}}$ and $\mathcal{L}_{\text{act}}$ are cross-entropy losses over the complete rationale–action response and its action tokens, respectively, and λ_{act} controls the additional emphasis on executable requests. Through SFT, the policy learns to formulate geometric subgoals as executable requests.

Subsequent reinforcement learning optimizes the policy through interaction with the verifier. In this setting, the value of an intermediate result may become apparent only when a later deduction uses it, making final success alone an incomplete signal for crediting individual decisions. We therefore introduce DDRL, which rewards earlier actions through recorded proof dependencies and adapts the training distribution to the policy's evolving competence.

A complete solving attempt produces a trajectory τ consisting of successive agent requests and verifier feedback. Let $\mathcal{U}(\tau)$ contain the successful producer turns whose recorded results support a later successful subgoal or final-goal check through an exact direct or transitive dependency. The turn reward is

$$r_t = r_t^{\text{terminal}} + \eta\,\mathbf{1}[t \in \mathcal{U}(\tau)]. \tag{4}$$

Here r_t^{terminal} reflects the verified task outcome, and the derivation-use term is assigned retrospectively and deduplicated by producer turn. This reward credits intermediate results that verifiably support subsequent deductions.

Alongside this credit for useful intermediate results, we use reliable rejection feedback to penalize invalid or redundant requests during policy optimization.

We use Turn-PPO for turn-level policy optimization (Li et al., 2026b). The complete response $y_t = (\ell_t, a_t)$ is treated as one decision, and generalized advantage estimation propagates later outcomes to earlier turns:

$$\delta_t = r_t + \gamma V(o_{t+1}) - V(o_t), \qquad A_t^{\text{turn}} = \sum_{k=t}^{T} (\gamma\lambda)^{k-t}\delta_k. \tag{5}$$

Here T denotes the final turn of the trajectory. Using the length-normalized turn ratio ρ_t and the resulting policy advantage $\widetilde{A}_t$, we optimize

$$\mathcal{L}_{\text{actor}} = -\mathbb{E}_{\text{valid}}\Big[\min\Big(\rho_t\widetilde{A}_t,\ \text{clip}(\rho_t, 1-\epsilon, 1+\epsilon)\widetilde{A}_t\Big)\Big] + \lambda_{\text{local}}\mathcal{L}_{\text{local}} + \beta_{\text{KL}}\mathcal{L}_{\text{KL}}. \tag{6}$$

$\mathcal{L}_{\text{local}}$ acts on the action span, while $\mathcal{L}_{\text{KL}}$ constrains deviation from the frozen SFT policy.

To align training difficulty with the policy's evolving competence, we adapt the task sampling distribution throughout RL. Before RL, the frozen SFT model scores demonstration prediction loss and partitions the training set into five fixed difficulty buckets. These buckets provide a common difficulty reference throughout training, while recent solving success in each bucket estimates the agent's changing competence. Together, they guide sampling toward challenging tasks suited to the policy's current competence.

The curriculum maintains a continuous difficulty center μ_t and samples across surrounding buckets. Online success estimates guide the progression of this center according to

$$\mu_{t+1} = \min\left\{\mu_{\max},\ \mu_t + \text{clip}\left[v + \alpha\left(\frac{\sum_b \exp[-(b-\mu_t)^2/(2\sigma^2)]\,q_b(t)}{\sum_b \exp[-(b-\mu_t)^2/(2\sigma^2)]} - q^\star\right),\ 0,\ \Delta_{\max}\right]\right\}. \tag{7}$$

Here t indexes curriculum updates, $q_b(t)$ is the calibrated moving estimate of success in bucket b, and the weighted average measures competence around the current difficulty center. Performance relative to the reference success rate $q^\star$ adjusts the base pace v, so stronger performance accelerates progression toward harder tasks and weaker performance slows it. Sampling retains tasks from all difficulty levels and favors less frequently sampled problems. Details of SFT, RL optimization, and curriculum scheduling are provided in Appendix E.

Table 1: Benchmark results and policy-training comparisons (pass@1, %). Overall is problem-weighted over all five test sets. Parentheses show gains over SFT in percentage points. 👍 marks the highest reported result in each column.

Method (completion)	Geometry3K	PGPS9K	MathVerse	GeoQA	GeoQA+	Overall
			MLLMs			
GLM-4.6V (Z.ai, n.d.)	53.70	47.70	—	—	—	—
Qwen2.5-VL-7B-Instruct (Bai et al., 2025b)	26.12	21.00	38.13	42.24	44.70	32.32
Qwen2.5-VL-32B-Instruct (Bai et al., 2025b)	47.25	37.00	58.99	66.94	62.75	51.85
Qwen2.5-VL-72B-Instruct (Bai et al., 2025b)	48.75	38.00	60.79	67.45	64.24	53.02
Qwen3-VL-32B-Instruct (Bai et al., 2025a)	62.23	51.00	70.86	76.96	75.00	64.73
DeepSeek Flash (API)	65.56	53.50	78.42	86.14 👍	82.45 👍	70.12
			Neural Solvers			
Geoformer (Chen et al., 2022)	36.80	35.60	—	—	—	—
PGPSNet (Zhang et al., 2023)	65.00	62.70	—	—	—	—
Euclid-Omni VLM (Li et al., 2026c)	—	—	51.00	—	—	—
			Symbolic and Neuro-symbolic Solvers			
Inter-GPS (Zhao et al., 2025b)	43.40	—	—	—	—	—
GeoCoder-7B (Sharma et al., 2025)	—	—	25.18	—	—	—
AutoGPS (InternVL3) (Ping et al., 2025)	70.20	72.30	—	—	—	—
GF-Reasoner-7B (Yang et al., 2025)	—	68.70	52.20	—	—	—
SD-GPS (Li et al., 2026a)	86.40	79.80	—	—	—	—
			Our backend and agent			
Our symbolic solver	86.19	83.40	65.11	60.10	53.31	71.87
Veri-GPS: SFT	87.69	86.10	74.82	69.95	67.22	78.55
Veri-GPS: RL (Uniform)	84.86	85.50	77.34	73.29	69.21	79.07 (+0.52)
Veri-GPS: DDRL	89.18 👍	87.30 👍	81.29 👍	76.13	72.85	82.12 👍 (+3.57)

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Experimental setup. Veri-GPS uses Qwen2.5-VL-7B-Instruct (Bai et al., 2025b), with SFT on 32,521 successful-action samples followed by DDRL. RL training and all agent evaluations, including ablations, allow at most 13 proposal turns and 10 executed requests per problem. We report pass@1 unless otherwise specified. Training and inference configurations appear in Appendix E.

Datasets and baselines. Our test set comprises 3,082 source problems from Geometry3K (601) (Lu et al., 2021), PGPS9K (1,000) (Zhang et al., 2023), the planar geometry subset of MathVerse (278) (Zhang et al., 2024), GeoQA+ (604) (Cao & Xiao, 2022), and GeoQA (599) (Chen et al., 2021). Table 1 compares direct-answer models, existing geometry solvers, our standalone backend, and successive policy-training settings. Qwen models use local direct-answer evaluation, DeepSeek Flash uses its API, and other results come from the corresponding papers.

Evaluation metrics. We assess overall solving performance, backend accuracy and matching cost, SFT data-source effects, and feedback-guided subgoal revision. The RL analysis examines dynamic sampling, derivation-use rewards, final-goal reuse, and interaction efficiency within the given budget.

4.2 OVERALL RESULTS

Table 1 summarizes overall solving performance. Veri-GPS achieves 82.12% overall success across 3,082 problems and reaches 89.18% on Geometry3K. Relative to the reported SD-GPS results,

its completion rates are 2.78 and 7.50 percentage points higher on Geometry3K and PGPS9K. Against Qwen3-VL-32B-Instruct direct answering, the overall gain is 17.39 points. Direct-answer comparisons evaluate Veri-GPS as a whole, including formalization, symbolic execution, and the learned policy. Table 1 also reports the standalone backend, SFT checkpoint, and uniform-sampling RL, providing intermediate comparisons for the final DDRL-trained Veri-GPS.

4.3 Symbolic Solver Analysis

Table 2: Answer accuracy of standalone symbolic backends (%).

Backend	Geometry3K (601)	PGPS9K (1,000)
AutoGPS-DSR	76.04	79.60
Our solver	86.19 (+10.15)	83.40 (+3.80)

Table 2 compares standalone backends under a 600 s limit, using each system's saved formalizations without LLM scheduling. The extended backend combines metric representations with a formula registry for proof and exact calculation, improving accuracy over local AutoGPS-DSR (Ping et al., 2025) by 10.15 and 3.80 percentage points.

On Geometry3K, removing indexed matching or specialized binding raises matching time to 6.83× or 5.89× the full configuration, respectively. All-rule reactivation yields 1.49× (Figure 4), supporting these optimizations for reducing matching overhead.

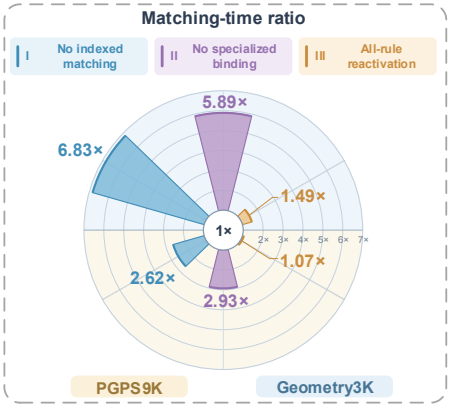


Figure 4: Matching-time ratios for execution ablations, with the full configuration normalized to 1×.

4.4 Feedback and Interaction Behavior

4.4.1 Training-Data Sources

Teacher–solver trajectories supervise subgoal selection and feedback-guided revision. Goal Augmentation supplements them with additional goals and dependency-linked proof–calculation steps, particularly for area calculations and underrepresented formula categories. Combined SFT solves 78.55% of test problems, compared with 78.33% for Teacher-only and 77.29% for Goal-Aug-only, supporting their joint use for policy initialization (Appendices A.2 and C.5).

4.4.2 Feedback in SFT and Inference

Table 3 separates SFT supervision settings from inference-time feedback for Veri-GPS. Completion measures test problems solved; switch & success measures successful action changes after eligible failures. Blocked-trajectory completion covers trajectories containing failed or rejected ordinary subgoals; action rejection is the rejected-request rate. Parentheses give signed percentage-point changes from each block's 01 or Full setting.

Table 3: Supervision settings for SFT models and inference-time feedback for Veri-GPS (%).

Setting	Completion (pass@1)	Switch & success	Blocked-trajectory completion	Action rejection
		SFT models: supervision settings		
01	78.55	11.00	12.70	52.15
02	78.26 (-0.29)	9.47 (-1.53)	11.08 (-1.62)	52.66 (+0.51)
03	77.90 (-0.65)	6.82 (-4.18)	7.98 (-4.72)	64.97 (+12.82)
		Veri-GPS: inference-time feedback		
Full	82.12	13.30	10.56	30.90
Reduced	80.86 (-1.27)	12.89 (-0.41)	7.49 (-3.07)	43.07 (+12.17)

Feedback-conditioned SFT. Setting 01 retains detailed diagnostics and newly derived facts in the input and supervises a rationale explaining each request. Settings 02 and 03 hide both feedback components; 03 also removes rationale supervision. Full supervision yields the highest overall completion and stronger post-blockage revision than 02. Retaining rationale supervision under reduced feedback improves completion by 0.36 points (02 vs. 03; Appendix C.6).

Inference-time feedback. With Veri-GPS parameters fixed, Reduced hides the same two feedback components while retaining request history and execution statuses. Full improves completion by 1.27 percentage points, with higher blocked-trajectory completion and fewer rejections. This supports the joint feedback configuration for guiding subgoal revision, rather than isolating diagnostics alone.

4.5 Analysis of DDRL

Table 1 reports completion on the full 3,082-problem test set. DDRL reaches 82.12%, a gain of 3.57 percentage points over the SFT checkpoint (78.55%) and 3.05 points over uniform-sampling RL (79.07%). For the test problems each model solves, DDRL proposes an average of 5.464 requests per problem, compared with 6.752 for SFT. Alongside DDRL's higher completion rate, this suggests more efficient use of the interaction budget.

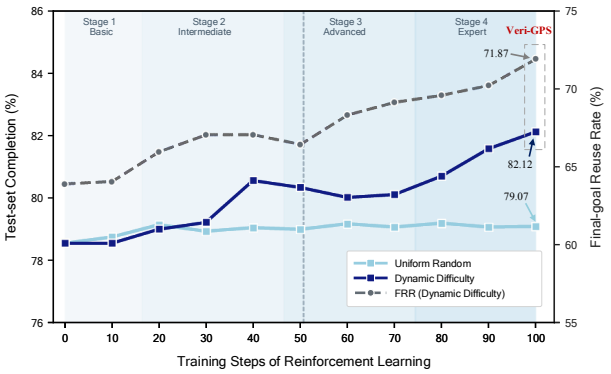


Figure 5: Completion and final-goal reuse during RL on the full 3,082-problem test set.

Dynamic-difficulty sampling. To test whether curriculum sampling improves RL, we compare Dynamic Difficulty with Uniform Random under otherwise matched settings and track completion on the full 3,082-problem test set (Figure 5). Uniform Random leads slightly at steps 10–20, possibly because it encounters harder tasks earlier. Dynamic Difficulty overtakes it at step 30 and improves at every checkpoint from step 60 onward, finishing at 82.12% versus 79.07%. This supports more sustained gains from progressive sampling.

Final-goal reuse. Completion alone does not show whether verified intermediate results contribute to the final solution, as encouraged by the derivation-use reward. We therefore measure FRR: the percentage of successful test-set trajectories whose final-goal derivation depends directly or indirectly on a verified result from an earlier turn. Under Dynamic Difficulty, FRR increases during RL, reaching 71.87% at the final checkpoint (Figure 5), indicating that earlier results contribute to a greater share of successful solutions.

Derivation-use reward ablation. To isolate the contribution of the derivation-use term in Eq. (4), we compare DDRL with an otherwise identical run omitting that term. At step 100 on the full 3,082-problem test set, the ablated run reaches 80.18% completion and 68.27% FRR, respectively 1.95 and 3.60 percentage points below full DDRL. These declines support the reward's contribution to both problem completion and the reuse of earlier verified results.

5 Conclusion

We presented Veri-GPS, an agent that learns subgoal policies from symbolic solver feedback to coordinate geometric proof and calculation. Solver feedback combines verified-state updates with goal-related diagnostics, while dual-source trajectories supervise feedback-conditioned decisions for an SFT cold start. DDRL further optimizes the policy through autonomous interaction with derivation-use rewards and adaptive difficulty sampling. Evaluation across five benchmarks shows gains over stronger direct-answer models and the SFT checkpoint, while behavioral comparisons support feedback-guided subgoal revision. These results highlight the value of learning to organize verified intermediate goals into coherent solutions within a given interaction budget.

AI USE STATEMENT

AI assistants were used to polish the writing, improve clarity, and discuss the presentation of the method and experiments. The authors reviewed and edited the final manuscript. A teacher model was used to construct supervised training data by generating candidate interaction requests, including proposed subgoals and auxiliary constructions, together with natural-language rationales. GPT-5.6 Sol was used as an LLM-based judge to evaluate direct-answer responses to GeoLaux proof problems, as described in Appendix C.3. The authors take responsibility for the final content of this work, including AI-assisted text, claims, data processing choices, evaluation procedures and results, and any remaining errors.

REPRODUCIBILITY STATEMENT

Sections 3 and 4 describe the method and evaluation protocol. Appendix A documents data sources and overlap checks; Appendices B–F provide interface specifications, supplementary experiments, trajectory examples, training configurations, and prompts. Selected implementation and configuration files will accompany the submission as anonymized supplementary material. Upon acceptance, we will release the extended Newclid backend, FormulaRegistry, turn-level data format and synthesis scripts, and evaluation scripts with the corresponding configurations.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a. doi: 10.48550/arXiv.2511.21631. URL https://arxiv.org/abs/2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b. doi: 10.48550/arXiv.2502.13923. URL https://arxiv.org/abs/2502.13923.

Jie Cao and Jing Xiao. An augmented benchmark dataset for geometric question answering through dual parallel text encoding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 1511–1520. International Committee on Computational Linguistics, 2022. URL https://aclanthology.org/2022.coling-1.130/.

Jiaqi Chen, Jianheng Tang, Jinghui Qin, Xiaodan Liang, Lingbo Liu, Eric Xing, and Liang Lin. GeoQA: A geometric question answering benchmark towards multimodal numerical reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 513–523. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.findings-acl.46. URL https://aclanthology.org/2021.findings-acl.46/.

Jiaqi Chen, Tong Li, Jinghui Qin, Pan Lu, Liang Lin, Chongyu Chen, and Xiaodan Liang. UniGeo: Unifying geometry logical reasoning via reformulating mathematical expression. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3313–3323. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.218. URL https://aclanthology.org/2022.emnlp-main.218/.

Yuri Chervonyi, Trieu H. Trinh, Miroslav Olšák, Xiaomeng Yang, Hoang H. Nguyen, Marcelo Menegali, Junehyuk Jung, Junsu Kim, Vikas Verma, Quoc V. Le, and Thang Luong. Gold-medalist

performance in solving olympiad geometry with AlphaGeometry2. *Journal of Machine Learning Research*, 26(241):1–39, 2025. URL https://www.jmlr.org/papers/v26/25-1654.html.

cyborvirtue. GeoAux dataset. Hugging Face dataset repository, n.d. URL https://huggingface.co/datasets/cyborvirtue/Geoaux. Accessed September 26, 2026.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. ReTool: Reinforcement learning for strategic tool use in LLMs. *arXiv preprint arXiv:2504.11536*, 2025. doi: 10.48550/arXiv.2504.11536. URL https://arxiv.org/abs/2504.11536.

Yumeng Fu, Jiayin Zhu, Lingling Zhang, Wenjun Wu, Bo Zhao, Shaoxuan Ma, Yushun Zhang, and Jun Liu. GeoLaux: A benchmark for evaluating MLLMs' geometry performance on long-step problems requiring auxiliary lines. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24784–24809. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1137. URL https://aclanthology.org/2026.acl-long.1137/.

Can Li, Ting Zhang, Junbo Zhao, and Hua Huang. Verifiable geometry problem solving: Solver-driven autoformalization and theorem proposing. *arXiv preprint arXiv:2606.27926*, 2026a. doi: 10.48550/arXiv.2606.27926. URL https://arxiv.org/abs/2606.27926.

Junbo Li, Peng Zhou, Rui Meng, Meet P. Vadera, Lihong Li, and Yang Li. Turn-PPO: Turn-level advantage estimation with PPO for improved multi-turn RL in agentic LLMs. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 6227–6243. Association for Computational Linguistics, 2026b. doi: 10.18653/v1/2026.findings-eacl.328. URL https://aclanthology.org/2026.findings-eacl.328/.

Zhaoyu Li, Hangrui Bi, Youyuan Zhang, Wenjie Ma, Zenan Li, Zhaolei Zhang, Xujie Si, and Kaiyu Yang. Euclid-Omni: A unified neuro-symbolic framework for plane geometry. *arXiv preprint arXiv:2608.14585*, 2026c. doi: 10.48550/arXiv.2608.14585. URL https://arxiv.org/abs/2608.14585.

Zhongzhi Li, Ming-Liang Zhang, Pei-Jie Wang, Jian Xu, Rui-Song Zhang, Yin Fei, Zhi-Long Ji, Jin-Feng Bai, Zhen-Ru Pan, Jiaxin Zhang, and Cheng-Lin Liu. CMMaTH: A chinese multi-modal math skill evaluation benchmark for foundation models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 2690–2726. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.coling-main.184/.

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 and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6774–6786. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.528. URL https://aclanthology.org/2021.acl-long.528/.

Bowen Ping, Minnan Luo, Zhuohang Dang, Chenxi Wang, and Chengyou Jia. AutoGPS: Automated geometry problem solving via multimodal formalization and deductive reasoning. *arXiv preprint arXiv:2505.23381*, 2025. doi: 10.48550/arXiv.2505.23381. URL https://arxiv.org/abs/2505.23381.

Aditya Sharma, Aman Dalmia, Mehran Kazemi, Amal Zouaq, and Christopher Pal. GeoCoder: Solving geometry problems by generating modular code through vision-language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 7355–7371. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.410. URL https://aclanthology.org/2025.findings-naacl.410/.

Vladmir Sicca, Tianxiang Xia, Mathïs Fédérico, Philip John Gorinski, Simon Frieder, and Shangling Jui. Newclid: A user-friendly replacement for AlphaGeometry. *arXiv preprint arXiv:2411.11938*, 2024. doi: 10.48550/arXiv.2411.11938. URL https://arxiv.org/abs/2411.11938.

Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625:476–482, 2024. doi: 10.1038/s41586-023-06747-5. URL https://www.nature.com/articles/s41586-023-06747-5.

Weiming Wu, Jin Ye, Zi-kang Wang, Zhi Zhou, Yu-Feng Li, and Lan-Zhe Guo. NeSyGeo: A neuro-symbolic framework for multimodal geometric reasoning data generation. *arXiv preprint arXiv:2505.17121*, 2025. doi: 10.48550/arXiv.2505.17121. URL https://arxiv.org/abs/2505.17121.

Wenjun Wu, Lingling Zhang, Jun Liu, Xi Tang, Yaxian Wang, Shaowei Wang, and Qianying Wang. E-GPS: Explainable geometry problem solving via top-down solver and bottom-up generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13828–13837, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wu_E-GPS_Explainable_Geometry_Problem_Solving_via_Top-Down_Solver_and_Bottom-Up_CVPR_2024_paper.html.

Tianyun Yang, Yunwen Li, Ziniu Li, Zhihang Lin, Ruoyu Sun, and Tian Ding. Bridging formal language with chain-of-thought reasoning to geometry problem solving. *arXiv preprint arXiv:2508.09099*, 2025. doi: 10.48550/arXiv.2508.09099. URL https://arxiv.org/abs/2508.09099.

Z.ai. GLM-4.6V. Official release description, n.d. URL https://z.ai/blog/glm-4.6v.

Ming-Liang Zhang, Fei Yin, and Cheng-Lin Liu. A multi-modal neural geometric solver with textual clauses parsed from diagram. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 3374–3382, 2023. doi: 10.24963/ijcai.2023/376. URL https://www.ijcai.org/proceedings/2023/376.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, Peng Gao, and Hongsheng Li. MathVerse: Does your multi-modal LLM truly see the diagrams in visual math problems? In *Computer Vision – ECCV 2024*, pp. 169–186. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-73242-3_10. URL https://link.springer.com/chapter/10.1007/978-3-031-73242-3_10.

Xiaokai Zhang, Yang Li, Na Zhu, Cheng Qin, Zhenbing Zeng, and Tuo Leng. FGeo-HyperGNet: Geometric problem solving integrating FormalGeo symbolic system and hypergraph neural network. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 4733–4741, 2025. doi: 10.24963/ijcai.2025/527. URL https://www.ijcai.org/proceedings/2025/527.

Haiteng Zhao, Junhao Shen, Yiming Zhang, Songyang Gao, Kuikun Liu, Tianyou Ma, Fan Zheng, Dahua Lin, Wenwei Zhang, and Kai Chen. Achieving olympiad-level geometry large language model agent via complexity boosting reinforcement learning. *arXiv preprint arXiv:2512.10534*, 2025a. doi: 10.48550/arXiv.2512.10534. URL https://arxiv.org/abs/2512.10534.

Junbo Zhao, Ting Zhang, Jiayu Sun, Mi Tian, and Hua Huang. Pi-GPS: Enhancing geometry problem solving by unleashing the power of diagrammatic information. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1526–1536, 2025b. URL https://openaccess.thecvf.com/content/ICCV2025/html/Zhao_Pi-GPS_Enhancing_Geometry_Problem_Solving_by_Unleashing_the_Power_of_ICCV_2025_paper.html.

A Data Composition

A.1 Data Sources

We select 4,848 source problems from CMMaTH (Li et al., 2025), GeoAux (cyborvirtue, n.d.), GeoLaux (Fu et al., 2026), GeoQA (Chen et al., 2021), and GeoQA+ (Cao & Xiao, 2022) to construct training data through teacher–solver interactions. Table A1 reports the retained problems, trajectories, and successful-action supervision samples from each source.

Table A1: Sources and sizes of the teacher-interaction data.

Source dataset	Unique source problems	Teacher-interaction trajectories	Successful-action supervision samples
CMMaTH	729	730	2,601
GeoAux	294	294	1,144
GeoLaux	913	913	3,695
GeoQA	1,903	1,903	6,003
GeoQA+	1,009	1,009	3,096
Total	**4,848**	**4,849**	**16,539**

Source problems are deduplicated by dataset name and normalized source-problem identifier. Different trajectories and augmented variants of the same problem count only once. A **trajectory** is an ordered interaction sequence for a particular goal, and one source problem may yield multiple trajectories. A **supervision sample** corresponds to a successful action and pairs its pre-action input state with its target output.

Goal Augmentation preserves the givens and configuration while changing the goal. It generates 3,889 trajectories and 15,982 successful-action samples from 1,143 source problems. The two sources share 1,130 source problems. Together, they cover 4,861 unique source problems, 8,738 trajectories, and 32,521 successful-action samples.

Table A2: Source-problem coverage of the two training-data sources.

Source dataset	Teacher-interaction source problems	Goal Augmentation source problems	Unique problems in the combined data
CMMaTH	729	10	729
GeoAux	294	113	297
GeoLaux	913	245	915
GeoQA	1,903	511	1,909
GeoQA+	1,009	264	1,011
Total	**4,848**	**1,143**	**4,861**

Of the 39,598 original proposals, 32,521 have status solved, 4,326 failed, and 2,751 rejected. Only solved actions serve as supervision targets; failed and rejected turns may remain in the history for later successful actions. The three feedback/rationale supervision variants preserve the same samples, images, actions, and ordering, changing only feedback visibility and output-supervision fields (Appendix E.1). The separate training-data source ablation is reported in Appendix C.5.

A.2 Trajectory and Feedback Coverage

Table A3: Trajectory lengths and successful-action statistics.

Metric	Teacher interaction: median [Q1–Q3]; P90	Goal Augmentation: median [Q1–Q3]; P90
Proposals	4 [3–6]; 9	4 [3–5]; 7
Exported supervision samples	3 [2–4]; 6	4 [3–5]; 7
Successful actions	3 [2–4]; 6	4 [3–5]; 7

All 3,889 augmented trajectories have explicit teaching chains and core-formula labels. Deduplicating this field yields 34 original labels covering mathematical formulas, geometric relations, and solving methods, grouped into 11 families. Table A4 lists the complete mapping. The following distributions describe the composition of the training supervision.

A.2.1 Final-Goal Coverage

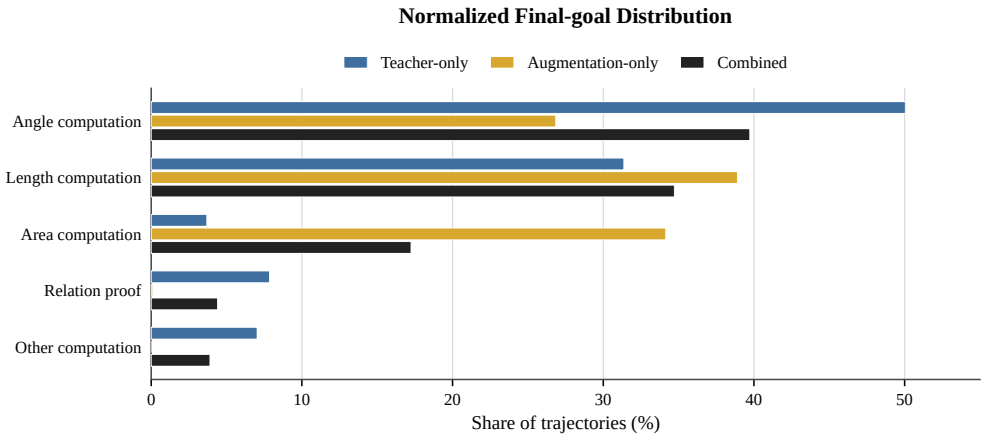


Figure A1: Normalized final-goal distributions for teacher interaction, Goal Augmentation, and their combination.

Teacher-interaction trajectories concentrate on angle calculation (50.05%) and length calculation (31.37%), with area calculation accounting for only 3.69%. Goal Augmentation increases the area-calculation share to 34.15% while retaining angle and length supervision. Area goals account for 17.25% of the combined data. Thus, augmentation does not simply repeat the goals visited by the teacher; it substantially supplements area goals that are sparse in teacher exploration.

A.2.2 Normalized Supporting-Formula Distribution

Table A4: The 34 core-formula labels and their normalized distributions.

Supporting-formula family	Original labels	Teacher interaction	Goal Augmentation	Combined
Angle relations	`angle3_supp`, `angle_system_replay_closure`, `global_circle_central_inscribed_major`, `global_circle_central_inscribed_minor`	37.80%	10.31%	25.57%

Continued on next page

Table A4 (continued)

Supporting-formula family	Original labels	Teacher interaction	Goal Augmentation	Combined
Trigonometric values to angles	`trig_to_angle3`	5.01%	16.53%	10.14%
Pythagorean relations	`global_right_triangle_altitude_geometric_mean`, `pythagorean`	12.06%	11.70%	11.90%
Right-triangle trigonometry	`right_triangle_cos`, `right_triangle_sin`, `right_triangle_tan`	4.04%	4.24%	4.13%
Sine / cosine laws	`law_of_cosines`, `law_of_sines`	2.45%	5.63%	3.87%
Similarity / congruence	`global_contri`, `global_contrir`, `global_parallel_transversal_similarity`, `global_simtri`, `global_simtrir`	8.95%	2.55%	6.10%
Linear lengths / equation systems	`global_coll_between_numeric`, `length_affine`, `minimal_equation_system`	10.58%	13.63%	11.94%
Midpoint relations	`midp`, `midp_length_halving`, `midpoint_area_halving`	4.10%	12.11%	7.67%
Triangle area	`heron_area`, `triangle_area_base_height`, `triangle_area_sine`	7.61%	19.75%	13.01%
Polygon area	`parallelogram_area_height`, `parallelogram_area_sine`, `quadrilateral_area_split`, `rectangle_area`, `regular_polygon_area`, `rhombus_area_sine`, `square_area`	7.36%	3.47%	5.63%
Circle metrics	`global_circle_radius`	0.02%	0.08%	0.05%

Teacher-interaction supervision concentrates on angle relations (37.80%). Goal Augmentation provides more supervision for triangle-area formulas (19.75% versus 7.61%), conversion from trigonometric values to angles (16.53% versus 5.01%), midpoint relations (12.11% versus 4.10%), and the sine and cosine laws (5.63% versus 2.45%). Teacher interactions therefore provide exploratory trajectories and revisions after blockage, while Goal Augmentation supplies verified, dependency-structured supervision for successful decisions in formula categories less frequently covered by teacher exploration.

A.2.3 Auxiliary Actions and Dependency Coverage

Teacher-interaction data contain 554 `connect` and 470 `construct` supervision targets, whereas Goal Augmentation contains neither. Combined retains all 1,024 teacher-sourced auxiliary-action targets. These counts concern actions supervised as outputs, not actions appearing only in the input history.

Goal Augmentation broadens goal and formula coverage through dependency-linked proof–calculation chains. Its calculation workspaces do not write results back into proof branches (Section 3.3), so these statistics do not establish calculation-to-proof dependency coverage. Teacher interactions can retain such transitions through online state updates, but their frequency has not been quantified.

A.3 Test Sources and Separation Checks

Table A5: Evaluation datasets. The first five form the main test set; GeoLaux and GeoAux are supplementary.

Dataset	Source problems
Geometry3K	601
PGPS9K	1,000
MathVerse	278
GeoQA	599
GeoQA+	604
GeoLaux	390
GeoAux	108

The test set comprises 3,082 problems across five datasets. Train–test overlap checks combine source identifiers, normalized text and DSL, and a review of problem content. These checks identified no overlap between our training and test data.

B Symbolic Solver Design

B.1 Predicates and Calculation Interfaces

Relative to official Newclid v2.1.0, the number of registered predicates increases from 29 to 78, adding 49 DSL names. These names cover relations, measurements, structural markers, and calculation entry points.

Table B1: Complete categorization of the added DSL names.

Category	Added names	Count	Function
Ordinary three-point angles	angle3_const, angle3_value, angle3_eq, angle3_supp, angle3_compute	5	Angle values, equality and supplementary-angle relations, and calculation with $[0, \pi]$ semantics
Angle bisectors	angle_bisector	1	Explicitly link angle structure to angle relations
Squared and affine quantities	l2const, r2const, length_affine, angle3_affine, area3_affine	5	Squared lengths, squared ratios, and linear combinations
Area and perimeter	sconst, scompute, scompute_affine, pconst, pcompute	5	Area and perimeter facts and calculation
Circles and arcs	radius_const/compute, circumference_const/compute, arc_measure_const/compute, arc_length_const/compute	8	Radius, circumference, arc measure, and arc length
Triangle-area relations	teqarea, trconst, midpoint_equal_area, midpoint_split_area, triangle_split_area	5	Equal areas, area ratios, and partitions
Trigonometry	cosconst, sinconst, tanconst, cos_compute, sin_compute, tan_compute	6	Exact trigonometric values and calculation
Shapes and special points	shape, incenter, incenter2	3	Structured shapes and incenter representations
Explicit topology	between, same_ray, opp_ray, same_side_line, opp_side_line	5	Betweenness, ray relations, and same-side/opposite-side relations
Circle–line relations and aliases	secant, tangent, constline, eqpoint	4	Secants, tangents, and line and point aliases

Continued on next page

Table B1 (continued)

Category	Added names	Count	Function
Unoriented triangle relations	similar, congruent	2	Similarity and congruence interfaces

B.2 Deduction-Rule Extensions

Table B2: Mathematical functions and implementation identifiers of added rules.

Function	Core premises and conclusions	Implementation identifiers
Angle propagation through parallel lines	Derive corresponding directed-angle relations from parallelism	r02a
Ordinary angles in isosceles triangles	Equal sides and non-collinearity imply equal ordinary base angles	r13a
Tangent definition and properties	A point on a circle and perpendicularity to its radius establish tangency; tangency implies perpendicularity	r16a, r16b
Secant definition	A and B lie on circle O, P, A, and B are collinear, and $OA = OB$; these conditions establish a secant relation	r16c
Secant–secant power of a point	Two secants of the same circle imply $PA \cdot PB = PC \cdot PD$	r16d
Tangent–secant power of a point	A tangent and a secant of the same circle imply $PT^2 = PA \cdot PB$	r16e
Converse of the diameter theorem	If O is the center, A/B/C lie on the circle, and $AB \perp BC$, then A, O, and C are collinear	r20b
Base angles of isosceles trapezoids	Equal legs imply two pairs of equal ordinary base angles	r21a, r21b
Parallelogram recognition	Two pairs of parallel opposite sides and non-degeneracy establish a parallelogram	r24a
Parallelogram diagonals	The diagonal intersection is the midpoint of both diagonals	r26a
Kite symmetry axis	The diagonal on the symmetry axis bisects the other diagonal	r26b
Ordinary angles in similar triangles	Same-orientation or opposite-orientation similarity implies equal corresponding ordinary angles	r52a, r53a
Angle-bisector ratio	If BX bisects $\angle ABC$ and X lies on line AC, then $AX/XC = AB/BC$	r57a
Perpendicular-bisector theorem for a chord	If A and B lie on circle O, M lies on AB, and $OM \perp AB$, then M is the chord midpoint	r57b
HL congruence and direct angle conclusions	Right triangles with equal corresponding hypotenuses and one equal corresponding leg are congruent; includes shared-point specializations and corresponding acute-angle conclusions	r70, r71, r70s, r71s, r70a, r71a
Expansion of congruence properties	Same-orientation or mirror congruence implies corresponding side, ordinary-angle, and directed-angle relations	r72, r73, r73a, r74, r75, r75a
Radius perpendicular to a chord	`circle O A B C` and $OC \perp AB$ imply $CA = CB$	r23a

B.3 Formula Registry

The FormulaRegistry records object templates, applicability conditions, and equation-generation procedures, allowing premises established through relational deduction to support subsequent calculation.

Table B3: The 18 registered formula templates.

Functional group	Templates
Triangle sides, angles, and area	CosineLaw, SineLaw, TriangleAreaSine, HeronArea, TriangleBaseHeightArea
Polygon perimeter and decomposition	PolygonPerimeter, PolygonAreaDecomposition
Measurements of special polygons	RectangleArea, SquareArea, RhombusArea, KiteArea, RegularPolygonArea, ParallelogramArea, TrapezoidArea
Circles and arcs	CircleCircumference, ArcLength, CircleArea, SectorArea

B.4 Interaction Controller

The interaction controller first checks whether a request satisfies interface and policy constraints through admission checking (passcheck), then dispatches admitted requests to the corresponding executor.

Table B4: Model actions and their effects.

Action form	Execution semantics	State effects
Ordinary predicate or calculation request	Prove a relation or compute a target quantity	Execute in the online state; verified results remain available to later requests
`connect A B`	Connect two existing points	Add a usable segment without directly asserting a new geometric relation
`construct ...`	Perform a supported auxiliary construction	On success, add new objects and state information produced by the valid construction
`finish`	Check the fixed final goal	Verify in a cloned state without writing temporary deductions back to the online state

Table B5: Execution statuses and feedback semantics.

Status	Meaning	State updates and visible feedback
Rejected	Rejected before the corresponding execution stage	Do not symbolically execute the request; record the rejection reason
Success	The requested goal or auxiliary operation succeeds	Retain verified ordinary-goal results and valid auxiliary changes; final-goal checking uses an isolated state
Failed: ordinary goal	Executed, but the requested goal is not completed under the current conditions or budget	Intermediate facts verified during execution may remain; record the actual reason and available diagnostics
Failed: final-goal check	The fixed final goal does not pass verification	Do not write temporary deductions from the cloned state back to the online state; record the failure reason

Pre-execution checks reject requests whose normalized identifiers match those of existing facts or completed goals. However, an admitted calculation request may still return a known numerical value and be marked Success without adding new facts. Thus, Success indicates request completion, not necessarily progress in the proof state. After a failed final-goal check, retrying `finish` requires a change in the proof state since that check; an intervening Success is not sufficient. The interaction prompt is provided in Appendix F.2.

B.5 Mechanism Comparison with SD-GPS

Table B6 distinguishes the learning objectives and interaction mechanisms, rather than treating verified states or diagnostics alone as unique to Veri-GPS. The SD-GPS description follows its reported method (Li et al., 2026a).

Table B6: Mechanism-level comparison with SD-GPS.

Aspect	SD-GPS	Veri-GPS
Model-selected outputs	Formal predicates and bounded sets of local-lemma proposals	One proof, calculation, auxiliary-operation, or final-check request per turn
Available feedback	Known facts, unresolved subgoals, failed theorem matches, and diagnostic messages	Execution statuses and reasons, newly verified facts, and goal-related diagnostics with object-bound rule/formula paths and missing premises or quantities
Learned objective	SFT for formalization; GSPO rewards for format validity, executability, and answer solvability	SFT for feedback-conditioned subgoal decisions; turn-level DDRL with terminal and derivation-use rewards
Role of dependencies	Local proof certificates support verification within the current problem; reported RL rewards concern formalization	Verified dependencies additionally assign policy credit to intermediate results used in later reasoning
Budget scope	Theorem-proposal rounds, candidates per round, and symbolic search	Model-proposed turns and admitted executions, with solver and episode time limits

Both methods use solver-verified information. Veri-GPS links goal-specific proof and calculation diagnostics to a learned request policy and uses recorded derivation dependencies for reward attribution.

C Supplementary Experimental Details

C.1 Execution Ablations of the Symbolic Solver

C.1.1 Ablation Factors

To examine the execution mechanisms of the hybrid solver, we report the full configuration and nine ablations on Geometry3K and PGPS9K. The ablations cover rule matching, variable binding, state updates, numerical premises, and formula entry points. Table C1 summarizes the main mechanisms and their ablated counterparts.

Table C1: Main execution mechanisms and corresponding controls.

Mechanism	Work reduced	Execution when disabled
Indexed rule matching	Bind variables through existing premise facts and indices, avoiding enumeration of all point combinations	Use the corresponding non-indexed matching path
Specialized rule-variable binding	Constrain high-arity rule bindings by triangle roles, shared sides, and orientation	Use generic variable matching
Incremental rule reactivation	Reactivate only rules affected by new facts or structural-index changes	Rescan all rules in each round

Indexed matching narrows candidates using existing premises. For example, after A and B are bound, it first retrieves point pairs satisfying `cong A B P Q` from the equal-length index. Specialized binding uses triangle roles, shared sides, and orientation to constrain high-arity rules. Incremental reactivation retriggers only rules affected by new facts or structural-index changes. Candidates must still satisfy all premises required by the rule. The goal-driven formula switch controls only the specified entry point; it does not remove all formula or algebraic capabilities.

C.1.2 Scoring, Budgets, and Statistical Definitions

The historical execution-ablation batch contains 614 execution tasks for Geometry3K, aggregated over 601 source problems, while PGPS9K is scored over 1,000 source problems. A result is correct only if the solver actually produces a parseable output that agrees with the reference answer after unit and variable transformations and the specified display-precision or equivalence rules. Unsolved, unparseable, and mismatched outputs are incorrect.

Table C2: Ablation execution and statistical settings.

Item	Setting
Answer evaluation	A solver time limit of 600 s
Answer scoring	Use each task's final valid solving result within the 600 s solver time limit; all other outputs are counted as incorrect
Matching-stage timing	Measure DDAR matching time under the same 600 s solver time limit
Concurrency	Usually 32 workers; PGPS9K with all-rule reactivation uses 24 workers
Time ratio	Median of per-task ablated/full matching-stage time ratios
Paired scope	Full-configuration matching time must be positive; zero ablated times are retained; 126/262 pairs for G/P
Interval estimation	2,000 paired-bootstrap resamples for 95% confidence intervals

Strict execution and fallback maintain independent states and record result provenance. Disabling fallback topology also changes the available premises, so this ablation reflects both execution efficiency and premise coverage. Matching-time ratios measure only the matching stage; batch wall-clock time also depends on concurrency and system load.

C.1.3 Complete Ablation Results

Table C3: Accuracy and original matching-stage time ratios for the nine ablations. G/P denote Geometry3K/PGPS9K, with scoring denominators of 601/1,000.

Configuration	Geometry3K accuracy	PGPS9K accuracy	Matching-time ratio G / P
Full configuration (displayed baseline)	86.19%	83.40%	1.000 / 1.000
No indexed matching	84.86%	80.60%	6.831 / 2.623
No specialized binding	85.02%	81.80%	5.892 / 2.934
All-rule reactivation	85.69%	83.00%	1.493 / 1.065
Fixed premise order	86.02%	82.70%	1.084 / 1.003
No deferred numerical premises	85.02%	82.90%	1.028 / 0.974
No goal-driven formula entry point	85.69%	81.90%	1.156 / 1.253
Global matching-cache updates	85.69%	82.70%	1.551 / 1.253
Global algebraic updates	86.19%	83.10%	1.135 / 1.191
No fallback topology	84.86%	81.80%	1.082 / 1.066

Removing indexed matching lowers accuracy by 1.33/2.80 percentage points on the two datasets, with matching-time ratios of 6.831/2.623. Removing specialized binding lowers accuracy by 1.16/1.60 points, with ratios of 5.892/2.934. Under the current configuration, both mechanisms support higher

accuracy and lower matching overhead. All-rule reactivation has a smaller effect; its paired interval is reported in Table C4.

Table C4: Paired matching-time ratios and 95% confidence intervals for the main matching optimizations.

Ablation	**Geometry3K paired median ratio [95% CI]**	**PGPS9K paired median ratio [95% CI]**
No indexed rule matching	6.83 [2.84, 10.22]	2.62 [2.09, 3.69]
No specialized variable binding	5.89 [3.76, 7.87]	2.93 [2.37, 4.23]
All-rule reactivation	1.49 [1.35, 1.56]	1.07 [1.03, 1.11]

The non-primary controls show heterogeneous effects. Removing the goal-driven formula entry point lowers accuracy by 0.50/1.50 points, indicating a contribution to solving coverage. Removing deferred numerical premises lowers accuracy by 1.16/0.50 points, indicating that delayed numerical-premise handling contributes to solving coverage. Replacing the full configuration's local matcher-cache updates with global matching-cache updates lowers accuracy by 0.50/0.70 points, suggesting that the local update policy is more favorable under this protocol.

These three sets of numbers come from different statistical records. Differences in Table C3 use the displayed baseline, while Table C4 reports only paired intervals for matching-time ratios.

C.2 Solving-Time Analysis

To examine the efficiency of the full configuration, we measure correct completion under time budgets up to 600 s and analyze actual solving times for correctly completed problems. On Geometry3K, the extended solver and AutoGPS-DSR correctly complete 86.19% and 76.04%, respectively (Figure C1). Their corresponding rates on PGPS9K are 83.40% and 79.60%. Most correctly completed problems require much less time than the maximum budget. On Geometry3K, the median solving time is 5.59 s and the 95th percentile is 58.67 s (Figure C3); the corresponding values on PGPS9K are 5.54 s and 56.89 s (Figure C4). Thus, approximately 95% of correctly completed problems in both datasets finish within one minute, and the longer budget mainly accommodates a small number of slower problems. Together with the execution ablations in Appendix C.1, these results show that indexed rule matching and specialized variable binding reduce matching overhead and support efficient solver execution.

C.3 Supplementary Evaluation on GeoQA+ and GeoLaux

We additionally evaluate GeoQA+ and GeoLaux to examine solving performance on another geometry benchmark and the ability to use auxiliary lines, respectively. GeoQA+ contains 604 problems. GeoLaux contains 390 problems: 228 short-answer problems and 162 proof problems. Selecting and using auxiliary lines requires the model to propose new connections or constructed points when reasoning is blocked and to continue proving with the resulting geometric relations.

Each dataset uses its corresponding frozen problem list. The data-separation details for GeoQA+ are given in Appendix A.3.

Direct-answer MLLMs receive only the original diagram and problem statement, without a geometric DSL, symbolic solver, or execution feedback, and generate one answer per problem. Short-answer problems are judged by the final answer; direct-answer responses to GeoLaux proof problems are evaluated by GPT-5.6 Sol.

Veri-GPS achieves an aggregate single-attempt success rate of 42.82%, compared with 19.74% for direct answering by Qwen2.5-VL-7B-Instruct. Combining four samples gives an agent pass@4 of 51.54%. This difference indicates that the proposal–symbolic execution–feedback–revision loop can solve some auxiliary-line problems that direct image-and-text generation does not solve reliably. Veri-GPS reaches a pass@4 of 69.14% on proof problems and 39.04% on short-answer problems.

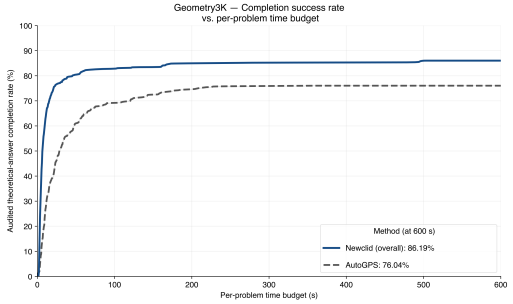


Figure C1: Time budget and correct completion rate on Geometry3K.

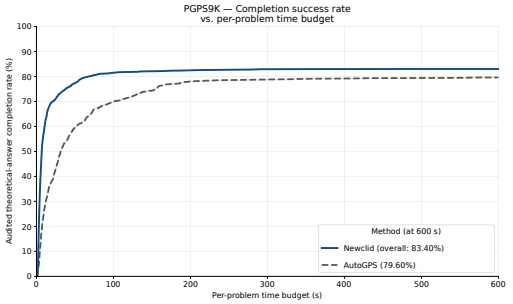


Figure C2: Time budget and correct completion rate on PGPS9K.

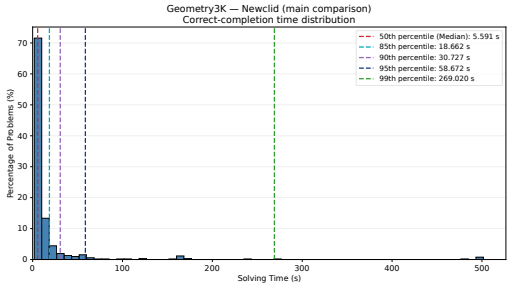


Figure C3: Solving-time distribution for correctly completed Geometry3K problems under the extended solver.

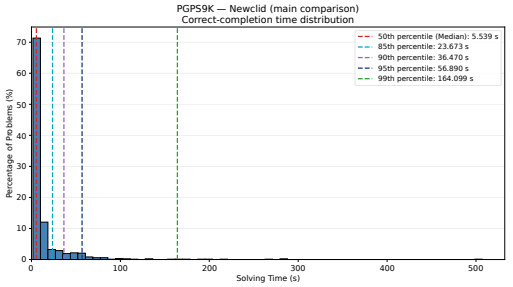


Figure C4: Solving-time distribution for correctly completed PGPS9K problems under the extended solver.

Table C5: Direct-answer and interactive-solving results on GeoLaux.

Model or system	**Short answer**	**Proof**	**Overall**
Qwen2.5-VL-7B-Instruct	22.81%	15.43%	19.74%
Qwen2.5-VL-32B-Instruct	31.14%	26.54%	29.23%
Qwen2.5-VL-72B-Instruct	30.70%	35.80%	32.82%
Qwen3-VL-32B-Instruct	55.26%	59.88%	57.18%
DeepSeek Flash	62.72%	65.43%	63.85%
Veri-GPS (pass@1)	33.33%	56.17%	42.82%
Veri-GPS (pass@4)	39.04%	69.14%	51.54%

Table C6: Direct-answer and interactive-solving results on GeoQA+.

Model or system	**GeoQA+**
Qwen2.5-VL-7B-Instruct	44.70%
Qwen2.5-VL-32B-Instruct	62.75%
Qwen2.5-VL-72B-Instruct	64.24%
Qwen3-VL-32B-Instruct	75.00%
DeepSeek Flash	82.45%
Veri-GPS (pass@1)	72.85%
Veri-GPS (pass@4)	75.00%

On GeoQA+, Veri-GPS achieves 75.00% pass@4 and 72.85% pass@1. For direct answering, the same-backbone Qwen2.5-VL-7B-Instruct achieves 44.70%, while Qwen3-VL-32B-Instruct and DeepSeek Flash achieve 75.00% and 82.45%, respectively. These supplementary results characterize the framework's performance on GeoQA+.

C.4 Supplementary Solver-Only Results

Table C7 reports success rates for direct symbolic-solver execution without LLM interaction on MathVerse, GeoQA, GeoQA+, GeoLaux, and GeoAux. The overall rate is weighted over these 1,979 problems. This aggregate includes GeoLaux and GeoAux but excludes Geometry3K and PGPS9K, so it differs from the 3,082-problem aggregate in Table 1.

Table C7: Solver-only success rates on the supplementary test sets.

Dataset	**Correct / total**	**Success rate**
MathVerse	181/278	65.11%
GeoQA	360/599	60.10%
GeoQA+	322/604	53.31%
GeoLaux	112/390	28.72%
GeoAux	11/108	10.19%
Overall (these five sets)	986/1,979	49.82%

C.5 Training-Data Source Ablation

We compare SFT models trained on teacher-interaction trajectories (Teacher-only; 16,539 successful-action samples), Goal Augmentation trajectories (Goal-Aug-only; 15,982 samples), and their combination (Combined; 32,521 samples). This comparison varies the training-data source, unlike the 01/02/03 feedback/rationale supervision variants in Table 3 and Appendix E.1.

Table C8: Test-set completion by SFT training-data source (pass@1). Cells report solved problems and accuracy; the denominator is shown beside each dataset.

Dataset (problems)	**Teacher-only**	**Goal-Aug-only**	**Combined**
Geometry3K (601)	525 (87.35%)	521 (86.69%)	527 (87.69%)
PGPS9K (1,000)	862 (86.20%)	851 (85.10%)	861 (86.10%)
MathVerse (278)	207 (74.46%)	202 (72.66%)	208 (74.82%)
GeoQA (599)	415 (69.28%)	409 (68.28%)	419 (69.95%)
GeoQA+ (604)	405 (67.05%)	399 (66.06%)	406 (67.22%)
Overall (3,082)	**2,414 (78.33%)**	**2,382 (77.29%)**	**2,421 (78.55%)**

The added Geometry3K problem is counted as unsolved in all three settings. Combined training solves seven more problems than Teacher-only and 39 more than Goal-Aug-only, corresponding to gains of 0.23 and 1.27 percentage points from unrounded proportions.

Table C9: Interaction behavior on the same 763 problems entering LLM interaction in all three settings. Problems solved before LLM interaction are excluded.

Metric	**Teacher-only**	**Goal-Aug-only**	**Combined**
Final completion	101/763 (13.24%)	70/763 (9.17%)	109/763 (14.29%)
Blocked-trajectory completion	82/730 (11.23%)	49/718 (6.82%)	93/732 (12.70%)
Completion among trajectories with a successful ordinary subgoal	90/530 (16.98%)	68/586 (11.60%)	98/504 (19.44%)
Repeated-action termination	149/763 (19.53%)	236/763 (30.93%)	141/763 (18.48%)
Format-error rate	19/8,664 (0.22%)	167/7,797 (2.14%)	26/8,607 (0.30%)
Action rejection	52.96%	28.33%	52.15%

Continued on next page

Table C9 (continued)

Metric	Teacher-only	Goal-Aug-only	Combined
Local action success	15.41%	21.75%	15.27%
Distinct blocked goals, deduplicated within each trajectory	4,222	3,576	4,313
Explicit recovery of a previously blocked goal	5/4,004 (0.12%)	8/3,362 (0.24%)	2/4,083 (0.05%)
Switch & success after an eligible Failed request	10.72%	11.15%	11.00%
Same-goal blockage immediately after Rejected	3.34%	5.78%	3.17%
Proposed auxiliary requests: connect / construct	637	0	597

Blocked-trajectory completion uses each setting's trajectories containing a failed or rejected ordinary subgoal. Local action success is the action-level Success proportion, not admission success or final-goal contribution. Repeated-action termination is trajectory-level. Explicit recovery requires the same normalized goal to fail or be rejected and later succeed; its denominator includes only blocked goals with a subsequent turn. This differs from completing a problem by another path.

Table C10: Auxiliary-action behavior on the common 763-problem interaction subset. Counts are trajectories; each trajectory is counted once per row.

Metric	Teacher-only	Goal-Aug-only	Combined
Proposes an auxiliary action	343 (44.95%)	0	312 (40.89%)
Successfully executes an auxiliary action	135 (17.69%)	0	137 (17.96%)
Successfully constructs a new point	90 (11.80%)	0	94 (12.32%)
Successfully executes an auxiliary action and solves the final goal	14	0	18

No Goal-Aug-only output begins with `connect` or `construct`; its zero count is not solely a parsing artifact. Successful execution and final completion in the last row co-occur but do not establish that the auxiliary action was necessary or used in the final derivation.

Using Teacher-only behavior to define fixed reference subsets, Goal-Aug-only and Combined solve 9/135 (6.67%) and 15/135 (11.11%) problems on which Teacher-only successfully executes an auxiliary action. On the 90-problem subset with successful new-point construction, they solve 5/90 (5.56%) and 10/90 (11.11%). These are behavior-defined subsets, not problems proven to require auxiliary constructions.

Combined has the highest overall completion here, but not the highest local action success. The training coverage and auxiliary-action results support supplementing teacher supervision, not uniform behavioral superiority or a data-volume-controlled benefit. Each setting uses one trained model; Combined has more samples than either source alone.

C.6 SFT Supervision and Feedback Visibility

Table C11 evaluates the three SFT supervision variants in Table 3 and Appendix E.1, using the same 3,082-problem test set. The additional Geometry3K problem is counted as unsolved in every variant.

Table C11: Test-set completion of the SFT supervision variants (pass@1). Cells show solved counts and percentages.

Dataset (problems)	01 Full	02 Reduced feedback + rationale	03 Reduced feedback, action only
Geometry3K (601)	527 (87.69%)	529 (88.02%)	529 (88.02%)
PGPS9K (1,000)	861 (86.10%)	857 (85.70%)	854 (85.40%)
MathVerse (278)	208 (74.82%)	204 (73.38%)	206 (74.10%)
GeoQA (599)	419 (69.95%)	416 (69.45%)	413 (68.95%)
GeoQA+ (604)	406 (67.22%)	406 (67.22%)	399 (66.06%)
Overall (3,082)	2,421 (78.55%)	2,412 (78.26%)	2,401 (77.90%)

Setting 01 solves nine more problems than 02 and 20 more than 03, corresponding to 0.29 and 0.65 percentage points. Retaining rationale supervision under reduced feedback (02 versus 03) adds 11 solved problems, or 0.36 points. These are aggregate comparisons; the ordering is not uniform across datasets.

Feedback visibility. Reduced feedback hides detailed diagnostics and the `Newly derived facts` display, while retaining request history and Success/Failed/Rejected statuses. It changes visible information, not the backend's verified state. The same field distinction applies to the fixed Veri-GPS inference comparison. Because the fields can provide overlapping decision cues, they are removed jointly; the comparison evaluates the feedback configuration, not the isolated contribution of diagnostics. Inference-time removal is also a change from the inputs used to train the full-feedback policy.

D Dual-Source Trajectories and Supervision Samples

D.1 Teacher-Interaction Example

GeoLaux 254_6_3 asks for a circle's radius and illustrates data generation through a switch from blocked calculation to relational proof, followed by a retry.

Table D1: Teacher trajectory.

Turn	Action	Result
1–3	`connect o d`; `connect o c`; `connect b e`	Add usable segments
4	`angle3_const c e d 1pi/4`	Success
5	`angle3_const c o d 1pi/2`	Success
6	`lcompute o c`	Failed: goal timeout
7	`cong o c o d`	Success
8	`lcompute o c`	$OC = \sqrt{2}$
9	`lcompute o b`	$OB = \sqrt{2}$

The saved results establish $OC^2+OD^2 = CD^2$, $OC = OD$, and $CD = 2$, giving $OC = OB = \sqrt{2}$. Turn 6 times out under a 100 s goal budget and produces no new facts; its historical diagnostic status is `deferred_offline`. The teacher proposes turn 7, and subsequent success does not imply that the preceding feedback already contained this suggestion.

D.2 Goal Augmentation Example

GeoQA-1960 gives a right triangle ABC with AB=5, AC=4, and BC=3, and altitude CD to the hypotenuse. Goal Augmentation preserves the givens and replaces the original target CD with the area of triangle BCD.

Table D2: Teaching chain and historical execution.

Step	Teaching-chain request	Historical execution
1	`simtrir a b c c b d`	Rejected: historical prove-prefix parsing failure
2	`eqratio a b c b b c b d`	Same as above
3	`lcompute b d`	Solved
4	`simtrir a b c a c d`	Rejected: historical prove-prefix parsing failure
5	`eqratio a b a c b c c d`	Same as above
6	`lcompute c d`	$CD = 12/5$
7	`scompute triangle b c d`	$S_{BCD} = 54/25$

The success-only conversion retains three successful actions, rendering the final supervision target as `finish`. Diagnostic replay after correcting the prefix obtains $BD = 9/5$, $CD = 12/5$, and $S_{BCD} = 54/25$. Some relational requests are rejected as already known, so not every replayed step can be counted as a newly successful proof. The final `finish` replaces the corresponding successful action rather than adding a fourth sample. Replay does not overwrite historical training records.

D.3 Turn-Level Supervision Format

Each sample takes the pre-action diagram, problem statement, DSL, state, and history as input and uses a successful action as its supervision target. The full variant supervises `reason + action`. The reduced-feedback variant hides detailed diagnostics and the newly derived facts display while retaining prior requests and execution statuses; the action-only variant additionally removes rationale supervision. All three variants preserve the same samples, images, actions, and ordering. Prompt templates are provided in Appendix F.

E Training Configurations

E.1 Supervised Fine-Tuning

We use Qwen2.5-VL-7B-Instruct, freeze its vision tower and multimodal projector, and update the language model through LoRA. All three feedback/rationale supervision variants use the same 32,521 successful-action samples.

Table E1: SFT configuration.

Item	Setting
LoRA	rank=64, alpha=128, target=all
Image / sequence	image_max_pixels=262,144; cutoff_len=2,048; no packing
Optimizer	AdamW; betas=(0.9,0.95); weight decay=0.1
Learning rate	10^{-4}; cosine schedule; warmup=0.05
Batch	8 GPUs $\times$ 2 per GPU $\times$ 4 gradient-accumulation steps; global batch size 64
Training	2 epochs; 1,018 steps; BF16; seed=42
Implementation	SDPA; gradient checkpointing; assistant-only supervision

The loss is $\mathcal{L}_{SFT} = \mathcal{L}_{full} + \mathcal{L}_{action}$. The action term covers only action-content tokens, is normalized within each sample, and is then averaged across samples. `01_full`, `02_without_feedback_with_reason`, and `03_without_feedback_action_only` use full feedback, reduced feedback, and reduced feedback with action-only supervision, respectively. All three complete 1,018 steps. The merged `01_full` weights initialize the RL actor and critic backbone and also serve as the frozen reference model.

Table E2: Input and supervision differences among SFT variants.

Variant	Detailed diagnostics and newly derived facts display	Output supervision
01 Full	Retained	reason + action
02 Reduced feedback	Hidden	reason + action
03 Reduced feedback, action only	Hidden	action

E.2 TURN-PPO AND REWARD ATTRIBUTION

RL starts from the fully supervised SFT checkpoint. The actor interacts with the symbolic solver over multiple turns, the critic estimates turn-state values, and a frozen reference model constrains policy updates through KL regularization.

For valid trajectories retained for training, completing the final proof within the interaction budget yields a terminal reward of $+1$; failure to complete it yields -1. We also assign intermediate rewards using proof dependencies recorded by the solver. If a conclusion produced at turn t is used directly or transitively in later successful reasoning, that turn receives an additional reward of 0.04, with at most five rewarded turns per trajectory. Even when the trajectory does not solve the problem, dependency-based attribution can provide a learning signal for useful intermediate deductions.

Generalized advantage estimation (GAE) propagates rewards across interaction turns. Let r_t denote the reward at turn t and $V(s_t)$ the pre-action value estimate:

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t), \qquad A_t = \sum_{l \geq 0} (\gamma\lambda)^l \delta_{t+l},$$

where $\gamma = 0.99$ and $\lambda = 0.90$. Advantages are standardized within the training batch. For clearly invalid actions and premature termination, a fixed negative value modifies the actor-side learning signal after standardization, while the critic continues to learn from trajectory returns.

Turn-PPO optimizes complete interaction turns. For the L_t output tokens at turn t, the policy ratio is

$$\rho_t(\theta) = \exp\left[\frac{1}{L_t}\sum_{k=1}^{L_t}\left(\log \pi_\theta(y_{t,k} \mid o_t, y_{t,<k}) - \log \pi_{\text{old}}(y_{t,k} \mid o_t, y_{t,<k})\right)\right].$$

Both reason and action tokens participate in turn-level policy updates. The loss is normalized over valid turns within each trajectory and then averaged across trajectories, preventing longer trajectories from receiving excessive weight. In addition to the Turn-PPO objective, an action-level auxiliary objective has coefficient 0.25. Successful termination actions and intermediate goals used by later successful reasoning receive positive signals; invalid or repeated actions receive negative signals, and premature termination receives a weaker penalty.

Table E3: RL configuration.

Item	Setting
Task pool / schedule	2,394 problems; 101 rollout–update rounds $\times$ 64 planned trajectories per round; one sequence per problem
PPO	2 epochs; minibatch=16 trajectories
Actor	q/k/v/o LoRA, rank=16, alpha=32; learning rate 5×10^{-6}
Critic	Same backbone with a value head; full-parameter training; learning rate decreases linearly from 5×10^{-6} to 10^{-6}
Optimization	AdamW (0.9,0.999); gradient-norm limit 1; policy/value clip=0.2; action-level auxiliary coefficient=0.25; reference KL=0.01
Parallelism	BF16, FSDP2, 8 GPUs; 4 rollout services, each with TP=2

Continued on next page

Table E3 (continued)

Item	Setting
Generation	temperature=0.8; top-p=1; 768 tokens per turn; context length 8,192
Interaction budget	13 proposals, 10 executions; construct$\leq$2, connect$\leq$4
Time budget	solver 15 s; hard goal 100 s; episode 1,800 s

RL comprises 101 rollout–update rounds and yields 6,315 valid training trajectories.

E.3 Dynamic-Difficulty Buckets and Sampling

Dynamic sampling matches training difficulty to the policy's current solving ability. Very easy problems provide limited learning signals, whereas very difficult problems tend to produce sparse rewards. We therefore establish fixed difficulty buckets before RL and dynamically adjust their sampling probabilities using online success and coverage statistics.

Before training, a frozen historical SFT model scores successful demonstrations. Turn difficulty is the length-normalized negative log-likelihood of content tokens in the complete reason/action response, excluding structural tags. Problem difficulty is the nearest-rank 70th percentile of its successful-turn scores. The 2,394 training problems are sorted by this score and divided into five fixed, approximately equal-sized buckets containing 479/479/479/479/478 problems. Bucket membership remains fixed throughout RL; online statistics adjust only sampling probabilities.

For difficulty bucket b, the sampling weight combines distance from the current curriculum center, recent success, and coverage:

$$w_b \propto \exp\left(-\frac{(b-\mu)^2}{2\times 1.1^2}\right)\left[0.25+4q_b(1-q_b)\right]c_b.$$

Here μ is the curriculum center, q_b is the bucket success rate estimated from recent trajectories, and c_b compensates for coverage. The factor $q_b(1-q_b)$ emphasizes buckets with intermediate success rates and room for exploration and improvement. Bucket success rates are updated by an exponential moving average and monotonically calibrated to remain non-increasing with difficulty.

The curriculum center starts at $\mu = 1.9$ and advances according to the success rate at the current sampling frontier, s_{frontier}:

$$\Delta\mu = \text{clip}\left(0.025 + 0.10(s_{\text{frontier}} - 0.30),\ 0,\ 0.05\right),$$

with μ capped at 4.6. If observed success remains below 0.25 for three consecutive batches, curriculum progression pauses for the next three batches. Batch allocation also keeps predicted success within $[0.32, 0.60]$.

Coverage compensation increases the weights of under-sampled buckets. Let $p_k = (k+1)/101$ denote training progress and E_b/N_b the mean effective exposure per problem in bucket b. With target coverage

$$\tau_b \in \{0.8, 1.2, 2.8, 4.0, 4.6\},$$

the coverage term is

$$c_b = \min\left[3,\ 1 + 2\max\left(0, \tau_b p_k - \frac{E_b}{N_b}\right)\right].$$

Each batch reserves at least one slot for every difficulty bucket and allocates the remaining slots by the largest-remainder method. Within each bucket, less frequently exposed problems are prioritized. Every trajectory uses the same interaction limits of 13 proposals and 10 actual executions.

Algorithm E1 summarizes the dynamic-difficulty RL procedure. Starting from the SFT checkpoint, each iteration samples problems according to the current curriculum and collects agent–solver trajectories. Solver-verified outcomes determine terminal rewards, while proof dependencies credit earlier turns whose conclusions support later successful reasoning. Turn-PPO updates the actor and critic using these trajectories, with action-level auxiliary signals, penalties for clearly invalid actions and premature termination, and KL regularization against a frozen SFT reference. Batch success and exposure statistics then update the curriculum for subsequent sampling.

Algorithm E1 Solver-Coupled Turn-PPO with Dynamic Difficulty

Require: training problems D, SFT policy π_{SFT}, critic V_ϕ, solver environment E, curriculum state C, iterations K

$\pi_\theta \leftarrow \pi_{\mathrm{SFT}}, \quad \pi_{\mathrm{ref}} \leftarrow \mathrm{Freeze}(\pi_{\mathrm{SFT}})$
for $k = 1, \ldots, K$ **do**
  $\pi_{\mathrm{old}} \leftarrow \mathrm{Freeze}(\pi_\theta), \quad V_{\mathrm{old}} \leftarrow \mathrm{Freeze}(V_\phi)$
  $X \leftarrow \mathrm{SampleProblems}(D, C)$ ▷ difficulty and coverage
  $B \leftarrow \varnothing$
  for $x \in X$ **do**
    $q \leftarrow E.\mathrm{Reset}(x), \quad \tau \leftarrow \varnothing$
    while $\neg q.\mathrm{done}$ **do**
      $o \leftarrow \mathrm{Observe}(q)$
      $y = (\ell, a) \sim \pi_{\mathrm{old}}(\cdot \mid o)$
      $q', f, \xi \leftarrow E.\mathrm{Step}(q, a)$ ▷ execution and dependency evidence
      $\tau.\mathrm{Append}(o, y, f, \xi), \quad q \leftarrow q'$ ▷ retain rejected turns and feedback
    end while
    $B \leftarrow B \cup \{\tau\}$
  end for
  $B \leftarrow \mathrm{RetainReliableEpisodes}(B)$
  $r \leftarrow \mathrm{TerminalRewards}(B)$
  $r \leftarrow r + \mathrm{DerivationUseCredit}(B)$ ▷ credit earlier supporting turns
  $A, R \leftarrow \mathrm{TurnGAE}(B, r, V_{\mathrm{old}})$
  $\widetilde{A} \leftarrow \mathrm{WhitenNormalThenOverride}(A, B)$ ▷ invalid-action and early-stop penalties
  $u, M \leftarrow \mathrm{BuildLocalTargetsAndMasks}(B)$
  for each PPO epoch and episode minibatch **do**
    $\phi \leftarrow \mathrm{UpdateCritic}(R, V_{\mathrm{old}}, M)$
    $\theta \leftarrow \mathrm{UpdateActor}(\widetilde{A}, u, \pi_{\mathrm{old}}, \pi_{\mathrm{ref}}, M)$ ▷ turn/action objectives and reference KL
  end for
  $C \leftarrow \mathrm{CommitCurriculum}(C, B)$ ▷ success and exposure guide sampling
end for
return π_θ

F Prompt Templates

The experiments use multimodal large language models for two tasks:

- **Direct-answer task.** The MLLM generates concise reasoning and a final answer from the original diagram and problem statement, without receiving DSL input or symbolic solver feedback.
- **Agent task.** The MLLM acts as a closed-loop planner. At each turn, it selects one executable action from the problem statement, formalized conditions, current symbolic state, usable segments, newly derived facts, and action history. Newclid executes the action and returns the updated state.

The prompt templates for both tasks are given below. Image inputs are omitted for brevity; the actual diagrams are supplied through multimodal message fields. Braced variables highlighted in red are placeholders replaced with the corresponding content at runtime.

F.1 DIRECT ANSWERING

Direct Answering: System Prompt

```
You are a mathematics problem-solving assistant. Independently solve a geometry or
mathematics problem from its original image and question. You cannot use external
tools or a symbolic geometry solver.
Use only the question and the supplied original image. Carefully read the point
labels, numerical values, symbols, and geometric markings in the image. Do not
assume that the diagram is drawn to scale, and do not infer exact lengths, angles,
parallelism, perpendicularity, or other geometric relations solely from its visual
appearance. Do not introduce conditions that are not given in the problem.
Give a concise reasoning process and the final result actually requested by the
problem. Include only the key relations or calculations needed to support the answer.
Do not output lengthy reasoning, additional analysis, a confidence score, or any
other text.
Your response must strictly follow the format below and contain only these two tags
in this order:
<reason>concise reasoning process</reason>
<final_answer>answer</final_answer>
Answer-format requirements:
- Put only the final mathematical answer or conclusion inside final_answer.
- Prefer exact numerical expressions, such as 5*sqrt(3), 7/2, or pi/3.
- If a decimal is necessary, retain at least six significant digits when possible,
unless the problem specifies a different precision.
- Do not output physical or geometric units for lengths, areas, volumes, or similar
quantities.
- For an angle expressed in degrees, output only its numerical value; do not output
degrees, degree, or °.
- For an angle expressed in radians, preserve an exact expression containing pi.
- A percent sign cannot be discarded as a unit. Preserve % in a percentage answer
unless the problem explicitly requests decimal form.
- If multiple quantities are requested, label each one, for example x=3; y=4.
- Put only the concise reasoning needed to support the answer inside reason.
- Do not output anything before, between, or after these two tags.
If the answer cannot be determined from the question and original image, output:
<reason>a concise explanation of why the answer cannot be determined</reason>
<final_answer>UNKNOWN</final_answer>
Do not make a random guess.
```

Direct Answering: User Prompt

```
Problem
Solve the following problem using the attached original image together with the
problem statement. Carefully inspect all labels, numerical values, symbols, and
geometric markings visible in the image. Use only the information explicitly
provided by the image and statement, without introducing any unstated assumptions.
Problem statement:
{question}
Provide the key reasoning needed to support your result, followed by the final
answer. Follow the exact response format specified in the system instructions.
```

F.2 Interactive Decision-Making

The system outputs one reason/action request per turn. The following templates specify the system instructions and dynamic input structure.

Interactive Decision-Making: System Prompt

```
Role
You are a Newclid geometry closed-loop planner. Based on the problem statement,
problem DSL, current symbolic state, usable segments, newly derived facts, and
action history, choose the one next executable action that most helps advance the
final goal.
Output Format
Output exactly the following two tags. Do not output JSON, Markdown, extra
explanation, or multiple alternatives:
<reason>a concise geometric rationale for the action</reason>
<action>one executable action</action>
Action Constraints
Return exactly one action. It may be a valid Newclid subgoal, a connection between
two existing points, a valid dynamic auxiliary construction, or the controller
action finish. An action must use points already present in available_points; only a
construction may introduce one new point, whose name must not duplicate an existing
point.
State Semantics
A connection only adds a segment between two existing points to usable_segments; it
does not assert a geometric fact. Do not add an unrelated or duplicate usable
segment, or use a connection as a mathematical proposition. A dynamic construction
must use supported Newclid construction syntax. Do not invent predicates, theorem
names, function names, point names, or construction names.
If the current conditions are sufficient for the symbolic solver to solve the final
goal directly, output <action>finish</action>. Submitting a subgoal exactly
equivalent to the final goal is also accepted as a finish request. Both forms verify
the final goal on an isolated state; a failed check adds no facts. Ordinary subgoals
do not automatically trigger a final check.
Action History lists earlier actions in chronological round order, each marked
Success, Failed, or Rejected. Failed actions may include a short Solver feedback
line. Reuse successful actions through the current state. Use the current state and
earlier feedback to decide whether to revise a request, pursue another goal, or
retry. Reason may use natural language; action must be executable by the runtime
parser.
Solver Outcome Semantics
Success means that the goal was completed or the connection or construction was
committed. Rejected means that the action was refused before its corresponding
execution stage. Failed means that execution did not complete the action; it does
not mean the proposition was disproved. For Rejected actions, correct the DSL or
choose an admissible action as indicated by the feedback. Querying an already-known
relation is Rejected; reuse it to pursue a new target. A past timeout does not
permanently forbid a target.
```

Interactive Decision-Making: Dynamic User Input

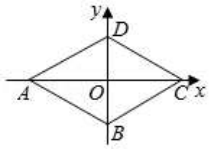
```
Problem
Problem statement:
{problem_text}
Formal DSL:
{initial_dsl_without_coordinates}
Final goal:
{final_goal}
Current Geometry State
Available points:
{available_points}
Usable segments:
{usable_segments}
Newly derived facts:
{newly_derived_facts_or_None}
Numeric display note: rounded values are approximate reading hints, not exact proof
facts.
Action History
{action_history_or_None}
```

G SOLUTION CASES

G.1 SUCCESSFUL AND UNSUCCESSFUL CASES

Success 1 | geoqa+/geoqa_plus_11570: Revision and Resumption after Blockage

In the Cartesian coordinate system xOy, quadrilateral $ABCD$ is a rhombus with $\angle ABC = 120°$ and $B = (0, -2)$. Find the area of $ABCD$.




```
a = free a; b = free b; c = free c; d =
free d; o = midpoint o a c; assume shape
rhombus a b c d; assume angle3_const a b c
2pi/3; assume lconst o b 2; assume coll a
o c; assume coll b o d; assume midp o b d;
assume perp a c b d
? scompute quadrilateral a b c d
```

Table G1. Recorded interaction.

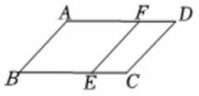
Turn	Action	Outcome
1	`lcompute o d`	Success: `lconst d o 2`
2	`scompute triangle a b o`	Success: `sconst triangle a b o`
3	`finish`	Failed: `goal_hard_timeout`
4	`finish`	Rejected: `finish_without_new_facts`
5	`scompute triangle a c d`	Failed: `missing_triangle_area_relation`
6	`finish`	Failed: `goal_hard_timeout`
7	`angle3_compute a b d`	Success: `angle3_const a b d 1pi/3`
8	`scompute triangle a c d`	Success: `sconst triangle a c d 4*sqrt(3)`
9	`finish`	Success: `sconst quadrilateral a b c d 8*sqrt(3)`

Final answer: $8\sqrt{3}$; solved. After the area request is blocked at turn 5, the agent establishes an angle fact and retries the area subgoal at turn 8. The final check at turn 9 uses this area result.

Success 2 | geolaux/369_2_1: Adding an Auxiliary Segment after Blockage

In parallelogram $ABCD$, points E and F lie on BC and AD, respectively, and $AF = AB = BE$. Join EF and prove that quadrilateral $ABEF$ is a rhombus.

Given $AF = AB = BE$, it suffices to prove $EF = AB$. Then $AF = AB = BE = EF$, establishing that $ABEF$ is a rhombus by equality of its four sides. The final symbolic goal is therefore reduced to `cong a b e f`.




```
a = free a; b = free b; c = free c; d =
free d; e = free e; f = free f; assume
shape parallelogram a b c d; assume coll b
e c; assume coll a f d; assume cong a f a
b; assume cong a b b e
? cong a b e f
```

Table G2. Recorded interaction.

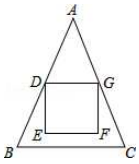
Turn	Action	Outcome
1	`cong b e a f`	Success: `cong a f b e`
2	`finish`	Failed: `compute_check_failed`
3	`connect e a`	Success: `connected a e` (auxiliary segment AE added)
4	`construct g = intersection_ll g a b e f`	Failed: `dynamic_aux_construction_rejected(No intersection)`
5	`angle3_eq b a e e a f`	Success: `angle3_eq b a e e a f`
6	`finish`	Success: `cong a b e f`

Conclusion: quadrilateral $ABEF$ is a rhombus; solved. The conditions are insufficient for direct completion at turn 2, so the agent adds auxiliary segment AE at turn 3. After an additional intersection construction fails, the agent establishes an angle relation and completes the goal. The triangle-congruence derivation in the final support certificate explicitly uses the newly available segment `segment:a:e`. Thus, the auxiliary segment contributes to the final proof rather than merely being drawn.

At turn 4, the executor attempts to construct the intersection of AB and EF. In this configuration, $AB \parallel EF$, so no unique finite intersection exists, and the constructor correctly prevents the infeasible construction. This turn calls the construction executor and consumes execution budget, so it retains its original Failed status rather than the admission-stage status Rejected.

Failure 1 | geoqa/geoqa_7898

In triangle ABC, $AB = AC = 18$ and $BC = 12$. Vertices E and F of square $DEFG$ lie inside the triangle, while D and G lie on AB and AC, respectively. Given $AD = AG$ and $DG = 6$, find the distance from F to BC.




```
a = free a; b = free b; c = free c; d =
free d; e = free e; f = free f; g = free g;
h = foot h f b c; assume lconst a b 18;
assume lconst a c 18; assume lconst b c 12;
assume shape square d e f g; assume coll a
d b; assume coll a g c; assume cong a d a
g; assume lconst d g 6
? lcompute f h
```

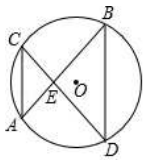
Table G3. Recorded interaction.

Turn	Action	Outcome
1	`lcompute f g`	Success: `lconst f g 6`
2	`finish`	Failed: `length_relation_not_proven`
3	`lcompute d f`	Success: `lconst d f 6*sqrt(2)`
4	`finish`	Failed: `length_relation_not_proven`
5	`scompute triangle a b c`	Success: `sconst triangle a b c 72*sqrt(2)`
6	`cos_compute b e f`	Failed: `coordinate_model_only`
7	`scompute triangle b f g`	Failed: `goal_hard_timeout`
8	`finish`	Failed: `length_relation_not_proven`
9	`lcompute d b`	Success: `lconst b d 9`
10	`lcompute e f`	Success: `lconst e f 6`

Reference answer: $6\sqrt{2} - 6$**; the execution budget is exhausted without completing the final goal.**

Failure 2 | geoqa/geoqa_3306

Chords AB and CD of circle O intersect at E. Given $BE = DE$ and $\angle B = 40^\circ$, find $\angle A$.




```
a = free a; b = free b; c = free c; d =
free d; e = free e; o = free o; assume
circle o a b c d; assume coll a e b;
assume coll c e d; assume cong b e d e;
assume angle3_const d b e 2pi/9
? angle3_compute c a e
```

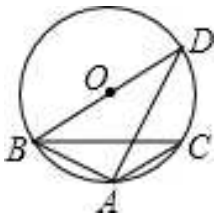
Table G4. Recorded interaction.

Turn	Action	Outcome
1	`angle3_compute b e d`	Success: `angle3_const b e d 5pi/9`
2	`finish`	Failed: `coordinate_model_only`
3	`angle3_compute c e a`	Success: `angle3_const a e c 5pi/9`
4	`finish`	Rejected: `finish_without_new_facts`
5	`angle3_compute a b o`	Failed: `coordinate_model_only`
6	`angle3_compute b o d`	Failed: `coordinate_model_only`
7	`angle3_compute a e d`	Success: `angle3_const a e d 4pi/9` (no new facts)
8	`angle3_compute e c d`	Success: `angle3_const d c e 0pi/1` (known collinear angle; no new facts)
9	`angle3_compute c d b`	Success: `angle3_const b d c 2pi/9` (no new facts)
10	`finish`	Rejected: `finish_without_new_facts`
11	`angle3_compute d b e`	Success: `angle3_const d b e 2pi/9`
12	`angle3_compute a c e`	Failed: `coordinate_model_only`

Reference answer: 40°**; the execution budget is exhausted without completing the final goal.**
Although turns 7–9 return Success, both `new_facts` and `committed_fact_delta` are empty, and `state_changed=false`. The recorded final check at turn 10 is rejected with `finish_without_new_facts`.
At turn 2, the numerical branch of the final check obtains $2\pi/9$, but the target angle has not entered the angle system (`target_in_angle_system=false`). The check therefore returns `coordinate_model_only` rather than accepting the numerical answer.

G.2 Comparison with Open-Source Models: geoqa/geoqa_8501

Triangle ABC is inscribed in circle O, with $\angle BAC = 120°$, $AB = AC$, diameter BD, and $AB = 3$. Find AD. The reference answer is $3\sqrt{3}$.




```
a = free a; b = free b; c = free c; d = free d;
o = midpoint o b d; assume circle o a b c d;
assume angle3_const b a c 2pi/3; assume cong a b a c;
assume lconst a b 3; assume coll b o d
? lcompute a d
```

Veri-GPS: Recorded Execution

Table G5: Veri-GPS: Recorded Execution.

Turn	Action	Solver outcome
1	`angle3_compute a d b`	Success: $\angle ADB = 30°$
2	`lcompute a o`	Failed: the length relation is not proven
3	`lcompute o b`	Failed: the length relation is not proven
4	`scompute triangle a b d`	Success: $S_{ABD} = 9\sqrt{3}/2$
5	`finish`	Success: $AD = 3\sqrt{3}$

Solution explanation. Since $AB = AC$, $\angle ACB = 30°$. Inscribed angles subtending chord AB give $\angle ADB = 30°$. Since BD is a diameter, $\angle BAD = 90°$. Therefore, $AD = AB/\tan 30° = 3\sqrt{3}$.

Qwen2.5-VL-72B-Instruct: Incorrect Response

Response excerpt (translated). "$\angle ABC = \angle ACB = (180° - 120°)/2 = 30°$. Since BD is a diameter, $\angle BAD = 90°$" ... "In right triangle ABD, $AB = 3$ and $\angle ABD = 30°$" ... "$AD = AB\tan 30° = 3(\sqrt{3}/3) = \sqrt{3}$."

Final answer: $\sqrt{3}$ (incorrect).

Error: the response substitutes $\angle ABD$ for the base angle $\angle ABC$, overlooking that BC and BD are different rays. The $30°$ angle is $\angle ADB$, whereas $\angle ABD = 60°$.

Qwen3-VL-32B-Instruct: Incorrect Response

Response excerpt (translated). "Both base angles are $30°$. Since BD is a diameter, $\angle BAD = 90°$" ... "It follows that $\angle ABD = 30°$ (because $\angle ABC = 30°$ and B, A, D are concyclic, $\angle ABD$ and $\angle ABC$ are the same angle)" ... "$AD = AB\tan 30° = 3(1/\sqrt{3}) = \sqrt{3}$."

Final answer: $\sqrt{3}$ (incorrect).

Error: concyclicity does not imply that $\angle ABD$ and $\angle ABC$ are the same angle. They subtend chords AD and AC, respectively, and cannot be equated directly.

Comparison note. These are excerpts from two complete responses saved for the same problem; omissions are marked with ellipses. Incorrect inferences, answers, and error explanations are highlighted in red. Veri-GPS uses DSL input and symbolic execution, whereas the baselines receive only the original diagram and problem statement. Coordinates and non-degeneracy guards are omitted from the displayed DSL.

G.3 Solving after Missing-Relation Feedback: geoqa+/geoqa_plus_12811

This example comes from our trained Veri-GPS model's evaluation trajectory. In rhombus $ABCD$, the diagonals AC and BD intersect at O, with $AC = 4$ and $BD = 3$. Given $BH \perp AD$ at H, the final goal is to compute BH (`lcompute b h`).

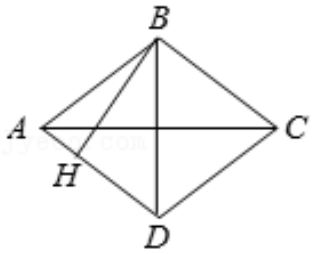


Figure G3: GeoQA+ 12811: finding the height of a rhombus.

Table G6: Recorded requests, feedback, and outcomes. All four turns are retained: three subgoal requests followed by a final-goal check.

Turn	Request	Outcome and feedback
1	`rconst a b b o 2/3`	Failed: "The current facts do not certify this target."
2	`scompute triangle a b c`	Failed: "Missing area relations required for this computation."
3	`scompute triangle a b o`	Success: `sconst triangle a b o 3/2`.
4	`finish`	Success: `lconst b h 12/5`.

Recorded response (summary). The turn-2 feedback is present in the actual input at turn 3. The model's response refers to the failed area request and insufficient area relations, then changes the target from triangle ABC to the smaller triangle ABO. The latter request succeeds, and the subsequent final check returns $BH = 12/5$. This paragraph summarizes the response rather than reproducing the model's original rationale verbatim.

Mathematical explanation (post hoc). The rhombus diagonals bisect each other at right angles, giving $AO = 2$, $BO = 3/2$, and $S_{ABO} = 3/2$. Hence $S_{ABCD} = 6$ and $AD = \sqrt{2^2 + (3/2)^2} = 5/2$, so $BH = S_{ABCD}/AD = 12/5$. This explanation is separate from the recorded model rationale and is not a verbatim solver trace.

Final answer: $BH = 12/5 = 2.4$. The case illustrates successful subgoal adjustment after missing-relation feedback. The feedback identifies a missing relation category, not the exact next action; this single trajectory does not independently establish a causal feedback effect.

H LIMITATIONS

Solver coverage. Veri-GPS is limited by its geometry DSL, rule library, and formula registry. General inequality goals lie outside the supported representation and execution interface. A failed request indicates that the target was not established under the current state and budget, not that the statement is false.

Auxiliary construction and generalization. Veri-GPS can propose auxiliary constructions, but this capability remains limited. It underperforms the stronger DeepSeek Flash direct-answer baseline on GeoQA+ and GeoLaux, including the auxiliary-construction evaluation (Appendix C.3). These results leave room for improving construction strategies and their use in subsequent reasoning.