

RIGOR: Atomic Semantic Normalization and Declarative Constraint Reasoning for Geometry Problem Solving

Yuhang Dong^{1,5}, Weijie Yu^{1,5}, Ziheng Wang^{2,5}, Kunhang Lv^{3,5},
Rui Han^{2,3,5}, Fuqi Jia^{3,5}, Minghao Yin^{4*}, Feifei Ma^{1,5*}

¹Laboratory of Parallel Software and Computational Science, ISCAS, Beijing, China

²Hangzhou Institute for Advanced Study, UCAS, Hangzhou, China

³SKLCS and Key Laboratory of System Software, ISCAS, Beijing, China

⁴Northeast Normal University, Changchun, China

⁵University of Chinese Academy of Sciences, Beijing, China

dongyuhang22@mails.ucas.ac.cn, ymh@nenu.edu.cn, maff@ios.ac.cn

Abstract

Geometry Problem Solving (GPS) is a fundamental task in mathematical reasoning, with applications in intelligent education and automated theorem proving. Existing neuro-symbolic solvers often rely on two restrictive assumptions: textual conditions are expressed as atomic geometric statements, and target derivation can be reduced to eager elimination of intermediate variables. These assumptions break down in realistic problems, where coupled geometric descriptions lead to combinatorial growth in parsing patterns, and nonlinear constraints make explicit elimination inefficient or even intractable. We propose RIGOR, a neuro-symbolic GPS framework based on Atomic Semantic Normalization and Declarative Constraint Reasoning. RIGOR uses a Geometric Semantic Normalizer trained on our newly constructed GSN-15K dataset to normalize geometric texts into atomic statements for reliable formalization. It then adopts a Declarative Symbolic Engine that globally accumulates derived geometric constraints and determines target values through constraint entailment, while also reconstructing proof-grounded reasoning traces. Experiments on multiple standard GPS benchmarks show that RIGOR outperforms state-of-the-art methods while producing interpretable derivations. Our project is available at <https://github.com/dongyuhang22/RIGOR>.

1 Introduction

Geometry Problem Solving (GPS) is a challenging task that requires deriving solutions from multi-modal inputs comprising diagrams and textual descriptions. As a fundamental benchmark for mathematical reasoning, GPS involves spatial and logical

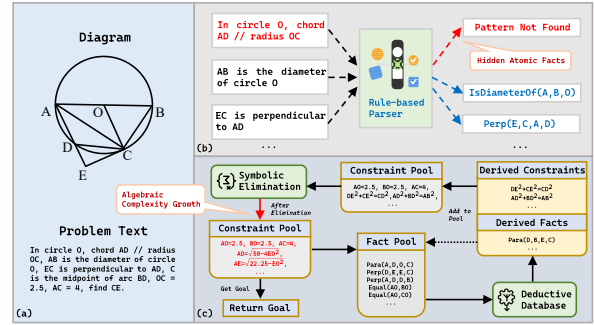


Figure 1: Illustration of two restrictive assumptions in existing DD-based GPS frameworks. (a) A representative geometry problem input. (b) *Text-to-Predicate Mismatch*: coupled geometric descriptions may encode hidden atomic facts but fail to match predicate-level parsing patterns. (c) *Eager Elimination Bottleneck*: newly derived constraints trigger iterative elimination of intermediate variables, which may lead to rapid growth in algebraic complexity.

reasoning and has potential applications in intelligent education. It is also closely related to automated theorem proving (Wen-Tsün, 1986; Chou et al., 2000; Wu and Gao, 2007; Trinh et al., 2024).

Existing GPS methods generally fall into two paradigms. Sequence Generation (SG) approaches formulate GPS as an end-to-end generation task using neural networks, demonstrating adaptability to diverse input variations (Chen et al., 2021; Gao et al., 2025; Xia et al., 2025). However, they rely heavily on large-scale annotations and struggle to guarantee logical rigor. In contrast, Deductive Database (DD)-based methods decompose GPS into parsing and symbolic reasoning stages (Lu et al., 2021; Zhao et al., 2025): a parser first converts multi-modal inputs into formal predicates, and a symbolic engine then performs deductive reasoning to ensure logical rigor. By leveraging neural networks to facilitate formal parsing (Zhang

*Corresponding authors.

et al., 2022) or provide heuristic guidance (Peng et al., 2023; Zhang et al., 2025a), while entrusting rigorous deduction to symbolic engines, this neuro-symbolic paradigm has shown strong potential.

Despite this promise, existing DD-based systems still inherit two restrictive assumptions in their parsing and reasoning stages. Figure 1 uses a representative example to illustrate how these assumptions fail in DD-based GPS systems.

First, in text parsing, existing methods typically rely on rigid regular expressions to map textual descriptions to individual geometric predicates, implicitly assuming that natural geometric text is close to predicate-level forms. While effective for atomic statements, this strategy fails when multiple relations are compressed into a coupled expression through shared primitives. The issue is not merely the absence of specific templates: directly parsing such expressions would require rules covering numerous relation combinations and linguistic variants, making the parser difficult to scale. Consequently, implicit conditions may be missed, breaking the downstream reasoning chain.

Second, in symbolic reasoning, existing engines commonly adopt an eager elimination paradigm for target solving, where intermediate variables are explicitly resolved as new constraints are derived. This strategy assumes that target derivation can be reduced to a tractable sequence of algebraic eliminations. However, geometric constraints are often nonlinear and tightly coupled; eliminating one variable may introduce larger expressions or require branching over multiple cases, leading to rapid growth in algebraic complexity. Practical systems therefore often rely on heuristic pruning or timeout-based termination, which may discard useful constraints and lead to solver failures.

To address these limitations, we propose RIGOR, a neuro-symbolic framework that reformulates text parsing and symbolic reasoning in DD-based GPS. For text parsing, RIGOR introduces *Atomic Semantic Normalization*. It shifts the core challenge from recognizing combinatorially many coupled patterns to producing predicate-level atomic statements: rather than expanding the parser to cover coupled expressions, RIGOR normalizes geometric texts into an atomic semantic space defined by a finite inventory of predicate-level forms. To instantiate this formulation, we introduce a Geometric Semantic Normalizer (GSN), implemented as an instruction-tuned Large Language Model (LLM). We construct GSN-15K, a geometric se-

mantic normalization dataset that pairs geometric descriptions with normalized atomic statements through a decomposition–recombination strategy, and instruction-tune GSN to produce atomic statements for reliable rule-based formalization.

For symbolic reasoning, RIGOR introduces *Declarative Constraint Reasoning*. The key insight is that deriving the final geometric target does not require explicitly resolving all intermediate algebraic quantities. Rather than constructing the target value through sequential symbolic eliminations, RIGOR formulates target solving as an entailment problem over a global constraint system in Nonlinear Real Arithmetic (NRA), which can be queried and verified by a Satisfiability Modulo Theories (SMT) solver. We instantiate this formulation with a Declarative Symbolic Engine (DSE), which first performs deductive fixpoint reasoning to derive and accumulate geometric constraints into a global constraint system. It then simplifies the constraints via value propagation, queries the SMT solver for a candidate target value, and proves its uniqueness by refuting alternative target assignments. The resulting proof-relevant evidence is further grounded in a deductive dependency graph, enabling the reconstruction of fine-grained reasoning traces.

Our main contributions are as follows:

- We formulate *Atomic Semantic Normalization* for geometric text parsing and construct GSN-15K via decomposition–recombination for instruction-tuning GSN for reliable formalization.
- We propose *Declarative Constraint Reasoning* for symbolic GPS, reformulating target solving from eager elimination into entailment over accumulated constraints, and instantiate it with DSE for query-and-refute derivation and tracing.
- Experiments on multiple benchmarks show that RIGOR outperforms state-of-the-art methods, with analyses validating its core components.

2 Background and Problem Formulation

Task Definition. GPS is formally defined as extracting a set of geometric premises Π and a computational goal g from the input diagram $\mathcal{D}$ and text $\mathcal{T}$, followed by a reasoning process to determine the value of g . Existing neuro-symbolic approaches typically map inputs into predicate-based formal descriptions and employ a symbolic engine for reasoning. We define the basic components below:

Definition 1 (Primitive). A *primitive* $\mathcal{E}$ denotes

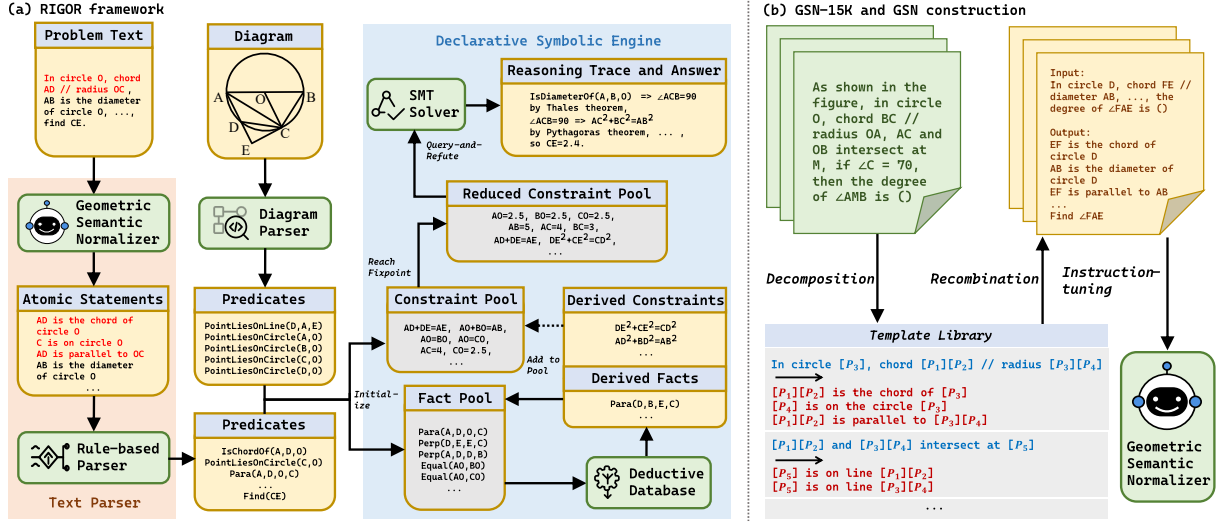


Figure 2: Overview of the proposed RIGOR framework. (a) The overall neuro-symbolic architecture, where the GSN normalizes geometric texts into atomic statements for rule-based formalization, and the DSE derives answers through declarative constraint reasoning. (b) The decomposition–recombination pipeline for constructing GSN-15K and instruction-tuning the GSN.

a basic geometric entity (e.g., points, lines) or a scalar quantity.

Definition 2 (Predicate). A *predicate* is a logical function over primitives, denoted by $P(\cdot)$, that asserts a geometric relation or property. For example, $\text{IsMidpointOf}(A, B, C)$ states that A is the midpoint of BC .

Text Parsing Paradigm. Current rule-based text parsers rely on predefined regular expression templates for individual predicates. This paradigm assumes that the input text can be expressed as *Atomic Statements*:

Definition 3 (Atomic Statement). An *atomic statement* is a natural language statement s that describes a single predicate instance, e.g., “point A lies on circle O ”.

However, geometry problems often contain compound descriptions such as “in circle O , chord $AD \parallel$ radius OC ”, where multiple predicate-level facts are coupled around shared primitives. Under the atomic assumption, the parsing mapping can become undefined or incomplete for such cases. While extending predefined patterns is theoretically possible, the combinatorial growth of geometric relations and linguistic variations makes manual rule expansion difficult to scale, motivating Atomic Semantic Normalization as a more scalable text-parsing formulation.

Algebraic Solving Paradigm. During reasoning, the symbolic engine derives both geometric facts

(e.g., $AD \parallel OC$) and algebraic constraints (e.g., $AD^2 + BD^2 = AB^2$). As illustrated in Figure 1(c), existing engines typically operate in an iterative loop: they deduce new facts and constraints using a deductive database and perform explicit symbolic elimination until the target quantity is resolved. We refer to this paradigm as *Eager Elimination*:

Definition 4 (Eager Elimination). Given a set of derived algebraic constraints $\Phi = \{f_1(\mathbf{x}) = 0, \dots, f_m(\mathbf{x}) = 0\}$ over geometric variables $\mathbf{x}$, eager elimination iteratively rewrites constraints into resolved forms such as

$$\Phi' = \{x_{i_1} = h_1(\mathbf{p}), \dots, x_{i_k} = h_k(\mathbf{p})\},$$

where $\mathbf{p}$ denotes constants and remaining free variables. Newly derived constraints are immediately combined with existing substitutions in Φ' for further elimination. The process succeeds when all variables appearing in the expression for the target g are resolved, reducing g to a constant value. An example of eager elimination is provided in Appendix A.

For nonlinear and tightly coupled constraints, eager elimination may increase algebraic complexity or introduce branching, motivating a constraint-entailment formulation for target solving.

3 Methodology

As illustrated in Figure 2(a), RIGOR consists of multi-modal input parsing and symbolic reasoning. For diagrams, RIGOR uses a diagram parser that

combines existing geometric parsing networks with a fallback mechanism to extract visual primitives and relations. For text, RIGOR uses the GSN to map geometric descriptions into normalized atomic statements, which are then formalized by a rule-based parser. The integrated formal descriptions are finally processed by the DSE to perform symbolic reasoning and reconstruct fine-grained reasoning traces. A concrete walkthrough of RIGOR is provided in Appendix A.

3.1 Diagram Parser

Diagrams provide implicit geometric cues, such as point-on-circle relations and label mappings like “ $\angle 1$ ”, that are often unstated in text but critical for reasoning. Following prior works (Zhao et al., 2025; Ping et al., 2026), RIGOR uses PGDP-Net (Zhang et al., 2022) as the primary parser to extract geometric primitives and relations from input images. To mitigate the sensitivity of neural parsers to low-resolution or noisy images, we incorporate a fallback mechanism based on traditional computer vision algorithms, with details provided in Appendix B.

3.2 Text Parser

The text parser bridges natural geometric texts and formal predicates. RIGOR first normalizes text into atomic statements within a finite predicate-level semantic space. These statements are then mapped to formal descriptions by a rule-based parser.

3.2.1 Atomic Semantic Normalization

We instantiate Atomic Semantic Normalization with a Geometric Semantic Normalizer (GSN), implemented as an instruction-tuned LLM. Given a problem text T , GSN generates a sequence of atomic statements $\mathcal{S}$, where each statement describes a single predicate-level geometric relation. To provide supervision for GSN, we construct GSN-15K, a geometric semantic normalization dataset containing 15,000 pairs of geometric texts and normalized atomic statements.

GSN-15K. We build the initial normalization pairs from GeoQA’s official training split by extracting problem texts and using an advanced LLM to generate atomic statement sequences.

To expand the supervision for GSN, we propose a decomposition–recombination strategy based on a key observation: geometric texts exhibit syntactic modularity. Although primitives serve as shared

variables across the whole problem, many text segments have self-contained linguistic structures for expressing predicate-level relations. For example, as shown in Figure 2(b), the segment “in circle O, chord AD // radius OC” can be locally normalized into atomic statements about chord, radius, and parallel relations. This suggests that normalizing text segments primarily depends on their local syntactic patterns, rather than the full semantic context of other segments, motivating template reuse after replacing concrete primitives with placeholders. As illustrated in Figure 2(b), the augmentation pipeline consists of two stages:

- **Template Library Construction (Decomposition):** We use an advanced LLM to decompose raw problem texts into self-contained segments and generate their corresponding normalized atomic statements. Primitives are then replaced with placeholders to build a “geometric segment–atomic statement” template library.
- **Data Synthesis (Recombination):** To mimic the structure of real problem texts, we first sample the number of segments k from the segment-count distribution estimated from the original corpus. We then sample k templates from the template library and instantiate their placeholders with sampled primitives. The instantiated text segments are concatenated into a synthetic problem description, and their corresponding atomic statement sequences are concatenated in the same order to form its normalization target, yielding a new “geometric text–atomic statement” pair.

To reduce annotation noise, we perform targeted manual correction of generated statements, aiming to improve parser compatibility and local semantic correctness. Through this construction pipeline, GSN-15K is expanded to 15,000 samples.

GSN Training. Using GSN-15K, we instruction-tune an open-source LLM as GSN. The model takes the geometric text T as input and predicts the atomic statement sequence $\mathcal{S}$. We optimize the standard negative log-likelihood objective:

$$\mathcal{L} = - \sum_{t=1}^{|\mathcal{S}|} \log P(s_t \mid s_{<t}, T; \theta),$$

where θ denotes model parameters and s_t is the t -th token in the target sequence.

3.2.2 Rule-Based Formalization

Following (Lu et al., 2021), the rule-based parser maps normalized atomic statements to formal de-

scriptions. Since these statements already conform to predicate-level forms, this step preserves formalization precision without enumerating coupled textual patterns.

3.3 Declarative Constraint Reasoning

After parsing, the formal descriptions are processed by the DSE, which instantiates Declarative Constraint Reasoning. Starting from the parsed premises, DSE repeatedly applies deductive rules to derive geometric facts and algebraic constraints until reaching a deductive fixpoint. Instead of eagerly eliminating intermediate variables, DSE accumulates the derived constraints into a global constraint system and establishes the target value through query-and-refute entailment checking. The complete DSE procedure is provided in Algorithm 1 in Appendix C.

Problem Formulation. Let $\Phi = \bigcup_t \{f_t(\mathbf{x}) = 0\}$ denote the accumulated algebraic constraints derived during deductive reasoning. Rather than reducing the target g through sequential symbolic elimination, DSE seeks a value v entailed by the global constraint system:

$$\text{find } v \in \mathbb{R} \quad \text{s.t.} \quad \Phi \models (g = v).$$

Deductive Fixpoint Iteration. The DSE first performs forward chaining to derive a deductive fixpoint. At step t , the reasoning state is denoted as $S_t = (\mathcal{F}_t, \mathcal{C}_t)$, where $\mathcal{F}_t$ stores established geometric facts and $\mathcal{C}_t$ stores accumulated algebraic constraints.

The engine maintains a Knowledge Base (KB) of rules. Each rule $r \in \text{KB}$ maps antecedent predicate patterns to consequent templates:

$$r : \mathcal{A}_r \rightarrow (\mathcal{B}_r^F, \mathcal{B}_r^C),$$

where $\mathcal{A}_r$ denotes antecedent patterns, while $\mathcal{B}_r^F$ and $\mathcal{B}_r^C$ denote templates for geometric facts and algebraic constraints, respectively.

Initially, S_0 is constructed from the parsed formal descriptions. In each iteration, for each rule $r \in \text{KB}$, the DSE matches $\mathcal{A}_r$ against $\mathcal{F}_t$ and applies all triggered instances of r . We denote by $(\Delta\mathcal{F}_r, \Delta\mathcal{C}_r)$ the union of all instantiated facts and constraints produced by rule r in this iteration. Aggregating the outputs of all rules yields the newly derived facts and constraints $(\Delta\mathcal{F}_{\text{new}}, \Delta\mathcal{C}_{\text{new}})$. The state is then updated by

$$S_{t+1} = (\mathcal{F}_t \cup \Delta\mathcal{F}_{\text{new}}, \mathcal{C}_t \cup \Delta\mathcal{C}_{\text{new}}).$$

To ensure that applicable deduction rules can be triggered, the DSE also propagates simple linear relations (e.g., angle equalities) during deduction and adds the resulting facts to $\mathcal{F}_t$. This iteration continues until a **deductive fixpoint** is reached, i.e., $S_{t+1} = S_t$. The resulting state is denoted as $S_{\text{final}} = (\mathcal{F}_{\text{final}}, \mathcal{C}_{\text{final}})$.

We then augment $\mathcal{C}_{\text{final}}$ with type-specific admissibility constraints, $\mathcal{C}_{\text{final}} \leftarrow \mathcal{C}_{\text{final}} \cup \mathcal{D}_{\text{domain}}$, where $\mathcal{D}_{\text{domain}}$ contains constraints induced by geometric quantity types, such as non-negativity constraints for length variables. The updated $\mathcal{C}_{\text{final}}$ is used in subsequent stages.

Value Propagation. Before invoking the constraint solver, the DSE performs value propagation to simplify the accumulated constraints $\mathcal{C}_{\text{final}}$. Let $\mathcal{V}_{\text{known}}$ be the set of variables with determined values. For a constraint $c(\mathbf{x}) \in \mathcal{C}_{\text{final}}$, if substituting the known variables in c reduces it to a univariate equation that uniquely determines an admissible value of x_u , DSE updates $\mathcal{V}_{\text{known}} \leftarrow \mathcal{V}_{\text{known}} \cup \{x_u\}$ and simplifies the constraint set accordingly. This process is repeated until no additional variables can be propagated. The resulting reduced constraint set is denoted as $\mathcal{C}_{\text{red}}$.

Query-and-Refute Target Derivation. After preprocessing, the DSE queries an SMT solver over NRA to obtain a satisfying model of $\mathcal{C}_{\text{red}}$ and reads off a candidate target value v for g . Since a satisfying model alone does not guarantee that g is uniquely determined, the DSE verifies the candidate through refutation-based entailment checking. Specifically, it checks the satisfiability of $\mathcal{C}_{\text{red}} \wedge (g \neq v)$ using the SMT solver. If the result is UNSAT, then $\mathcal{C}_{\text{red}} \models (g = v)$, proving v as the determined target value; otherwise, the current constraints are under-determined.

Proof-Grounded Trace Reconstruction. The DSE uses the unsatisfiable core returned by the SMT solver during verification as proof-relevant evidence, which identifies a subset of derived constraints sufficient to entail the target value. Meanwhile, the DSE maintains a Dependency Graph (DG) throughout deductive fixpoint iteration, recording the provenance of derived facts and constraints. By grounding the proof-relevant constraints in the DG and recursively backtracking their antecedents, the DSE reconstructs a fine-grained reasoning trace. Additional details are provided in Appendix A.

Methods	Angle	Length	All
Generalist Models			
mPLUG-Owl2	5.3 / 19.9	5.0 / 19.6	5.2 / 19.8
LLaVA-v1.5	7.2 / 22.5	5.0 / 19.0	6.2 / 21.0
Qwen-VL	6.5 / 23.3	4.5 / 23.4	5.6 / 23.3
GPT-4o	36.2 / 54.0	43.6 / 62.9	39.5 / 58.0
Specialist Models			
NGS	55.6 / 65.7	34.7 / 47.8	46.3 / 57.7
Geoformer	57.3 / 67.1	33.2 / 47.8	46.6 / 58.5
SCA-GPS	54.9 / 65.5	35.3 / 49.0	46.2 / 58.1
DualGeoSolver	55.2 / 65.7	36.5 / 50.4	46.8 / 58.9
G-LLaVA	55.6 / 70.5	30.9 / <u>55.8</u>	44.6 / 63.9
GeoX	<u>62.8</u> / <u>71.5</u>	44.2 / <u>55.8</u>	<u>54.5</u> / <u>64.5</u>
RIGOR (Ours)	71.0 / 78.7	<u>43.9</u> / 59.3	58.9 / 70.0

Table 1: Performance comparison on the GeoQA dataset. Each cell reports Completion / Choice Accuracy (%). **Bold** and underlined values denote the best and second-best performance among Specialist Models.

4 Experiments

4.1 Settings

Datasets. We conduct experiments on three widely recognized benchmarks: GeoQA (Chen et al., 2021), Geometry3K (Lu et al., 2021), and PGPS9K (Zhang et al., 2023). Specifically, GeoQA comprises 4,998 problems from Chinese secondary school exams, focusing on angle and length derivation. Geometry3K serves as a comprehensive benchmark with 3,002 samples covering diverse geometric primitives, while its extended version, PGPS9K, contains 9,022 problems that cover nearly all plane geometry types across grades 6–12.

Baselines. We comprehensively compare our method against state-of-the-art (SOTA) baselines spanning two GPS paradigms: SG-based methods (including NGS (Chen et al., 2021), Geoformer (Chen et al., 2022), SCA-GPS (Ning et al., 2023), PGPSNet (Zhang et al., 2023), DualGeoSolver (Xiao et al., 2024), G-LLaVA (Gao et al., 2025), GeoX (Xia et al., 2025)) and DD-based methods (including InterGPS (Lu et al., 2021), GeoDRL (Peng et al., 2023), Pi-GPS (Zhao et al., 2025)). To ensure equitable comparison, we adopt the 7B variant of G-LLaVA and exclude image annotation inputs for PGPSNet. Furthermore, we evaluate general-purpose Multi-Modal Large Language Models (MLLMs) under evaluation settings following prior GPS studies (Xia et al., 2025; Zhao et al., 2025), including mPLUG-Owl2 (Ye et al., 2023), LLaVA-v1.5 (Liu et al., 2024), Qwen-VL (Bai et al., 2023), and GPT-4o (OpenAI, 2024).

Metrics. Following prior studies (Zhang et al., 2023; Xia et al., 2025; Zhao et al., 2025), we report two metrics: *Completion*, which measures the accuracy of direct answer derivation; and *Choice*, which evaluates the accuracy of identifying the answer from given options. For *Choice*, we follow prior work and apply random selection when the derived answer matches none of the options. Both metrics are reported using Top-1 accuracy throughout our experiments.

Implementation. For constructing GSN-15K, we use the problem texts from the GeoQA training set as the initial geometric texts, and employ DeepSeek-V3 (DeepSeek-AI et al., 2025) as the advanced LLM in the data generation pipeline. To instantiate GSN, we adopt LLaMA-2-7B (Touvron et al., 2023) as the base model for consistency with specialist baselines, and fine-tune it with LoRA (Hu et al., 2022) on GSN-15K. For text parsing, we apply GSN before rule-based formalization on GeoQA, while applying rule-based formalization directly on Geometry3K and PGPS9K, where existing parsing patterns provide sufficient coverage. We implement the DSE based on Newclid (Sicca et al., 2024) and use Z3 (de Moura and Bjørner, 2008) as the SMT solver for NRA constraints. All experiments are conducted on a compute platform equipped with NVIDIA A100 (40GB) GPUs and AMD EPYC 7763 CPUs. We set the timeout for solving each instance to 1200 seconds. For generalist MLLMs, we use prompts with fixed final-answer formats and a deterministic answer-extraction procedure.

Details of the parameters, extraction rules and prompts are provided in Appendix E.

4.2 Results

We evaluate RIGOR against SOTA methods on three standard benchmarks: GeoQA, Geometry3K, and PGPS9K.

As shown in Table 1, RIGOR achieves the best performance on GeoQA, with completion and choice accuracies of 58.9% and 70.0%, respectively. In completion accuracy, RIGOR surpasses GPT-4o by 19.4 percentage points and the specialist SOTA baseline GeoX by 4.4 percentage points. RIGOR also shows particularly strong performance on angle-related problems.

On Geometry3K and PGPS9K, as shown in Table 2, RIGOR achieves the highest average completion and choice accuracies of 66.9% and 76.3%,

Methods	Geometry3K	PGPS9K	Average
Generalist Models			
mPLUG-Owl2	2.2 / 26.7	3.0 / 26.4	2.6 / 26.6
LLaVA-v1.5	2.9 / 22.9	1.8 / 21.8	2.4 / 22.4
Qwen-VL	2.5 / 27.5	1.4 / 24.7	2.0 / 26.1
GPT-4o	34.8 / 58.6	33.3 / 51.0	34.1 / 54.8
Specialist Models			
NGS	35.3 / 58.8	34.1 / 46.1	34.7 / 52.5
Geoformer	36.8 / 59.3	35.6 / 47.3	36.3 / 53.3
PGPSNet	48.1 / 70.1	44.4 / 57.6	46.0 / 63.9
GeoX	58.6 / 72.5	52.7 / 63.3	55.7 / 67.9
InterGPS	43.4 / 57.5	-	-
GeoDRL	57.9 / 68.4	55.6 / 66.7	56.8 / 67.6
Pi-GPS	68.2 / 75.4	61.4 / 69.8	64.8 / 72.6
RIGOR (Ours)	65.0 / 74.9	68.8 / 77.6	66.9 / 76.3

Table 2: Performance comparison on Geometry3K and PGPS9K datasets.

Variants	Angle	Length	All
Impact of the GSN			
w/o GSN	53.2	27.9	41.9
w/ GSN (w/o Aug.)	67.4	37.7	54.1
w/ DeepSeek-V3	70.7	37.7	56.0
w/ GSN	71.0	43.9	58.9
Impact of the DSE			
w/ ESE	66.2	31.8	50.8
w/ DSE (w/o Prop.)	71.0	42.1	58.1
w/ DSE	71.0	43.9	58.9

Table 3: Ablation study on the GeoQA dataset.

respectively. Although Pi-GPS obtains slightly higher accuracy on Geometry3K, RIGOR remains competitive on this benchmark and achieves the best performance on the larger and more comprehensive PGPS9K dataset, surpassing Pi-GPS by 7.4 percentage points in completion accuracy.

4.3 Analysis

Ablation. We evaluate the contribution of each core component based on the completion accuracy on the GeoQA dataset, as shown in Table 3.

First, we evaluate the effect of Atomic Semantic Normalization with the DSE fixed. Removing GSN and reverting to purely rule-based text parsing (w/o GSN) leads to a 17.0-percentage-point drop in overall completion accuracy, confirming the importance of normalizing geometric texts into atomic statements. Furthermore, removing decomposition-recombination augmentation (w/o Aug.) decreases accuracy by 4.8 percentage points, showing the value of our data expansion strategy. Directly employing the advanced LLM, DeepSeek-V3 (DeepSeek-AI et al., 2025), as the normalizer (w/ DeepSeek-V3) yields 2.9 percent-

Category	Mutual Runtime		Penalized Runtime	
	Method A	Method B	Method A	Method B
(a) DSE vs. ESE				
Columns:	DSE	ESE	DSE	ESE
Angle	14.72	11.78	31.12	106.81
Length	11.10	80.04	77.89	429.51
All	13.76	30.00	47.12	217.21
(b) DSE vs. DSE (w/o Prop.)				
Columns:	DSE	DSE (w/o Prop.)	DSE	DSE (w/o Prop.)
Angle	15.33	17.44	15.33	17.44
Length	16.59	30.38	17.24	77.79
All	15.74	21.64	15.96	37.56

Table 4: Runtime pairwise comparison on the GeoQA dataset.

age points lower accuracy than our instruction-tuned GSN. This suggests that stronger general-purpose language capabilities do not necessarily yield parser-compatible outputs, as their generations may span a broader range of syntactic forms. In contrast, by constructing a normalization dataset with targeted grammar-level quality control and instruction-tuning a compact GSN, we bias its outputs toward a narrower family of predicate-level formulations, achieving better downstream performance with a substantially smaller model.

We further compare DSE against an eager symbolic reasoning baseline under the same text-parsing setup. Following prior DD-based systems, we use the optimized InterGPS-style engine adopted in (Zhao et al., 2025) as the Eager Symbolic Engine (ESE) baseline. The result (w/ ESE) demonstrates that the DSE outperforms ESE by 8.1 percentage points overall, with a 12.1-percentage-point gain on length problems. Given that nonlinear constraints predominantly reside in length problems rather than angle problems, this improvement supports the advantage of declarative constraint reasoning over eager symbolic elimination. Removing value propagation (w/o Prop.) decreases accuracy by 0.8 percentage points, confirming its contribution.

Additional analyses on direct LLM formalization and failure cases are provided in Appendix D.

Efficiency. We further evaluate solver efficiency using two metrics: Mutual Runtime (MR), the average solving time over instances solved by both solvers, and Penalized Runtime (PR), the average solving time over instances solved by at least one solver. For PR, an unsolved instance for one solver is assigned the timeout penalty of 1200 seconds if the other solver solves it. MR measures raw efficiency, while PR reflects both efficiency and

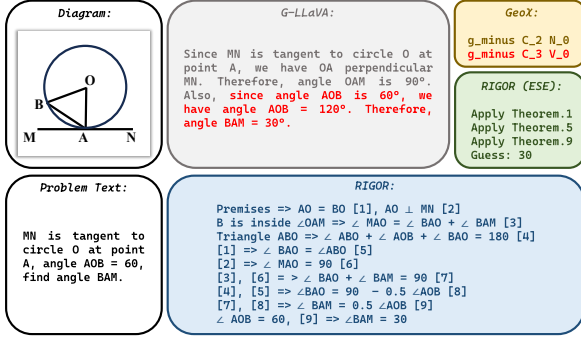


Figure 3: Comparison of reasoning processes across different methods.

robustness under solver failures.

Table 4 details our pairwise comparison on GeoQA and reveals the following:

- **DSE demonstrates significant overall efficiency.** DSE substantially outperforms ESE in PR, requiring only 21.7% of ESE’s time (47.12s vs. 217.21s). The advantage is especially pronounced on length problems with higher constraint complexity, where DSE achieves a $7.2\times$ speedup in MR (11.10s vs. 80.04s), highlighting the effectiveness of declarative constraint reasoning in handling complex constraints.
- **Linear vs. Nonlinear Trade-off.** On angle problems dominated by linear constraints, ESE is slightly faster in MR (11.78s vs. 14.72s), benefiting from the efficiency in linear elimination and early termination upon target derivation. Nevertheless, DSE’s superior performance on non-linear instances yields an over $2\times$ overall MR advantage (13.76s vs. 30.00s).
- **Impact of Preprocessing.** Preprocessing significantly improves DSE efficiency, reducing PR by 57.5% compared to the variant without preprocessing (15.96s vs. 37.56s), confirming its effectiveness in simplifying the constraint space and accelerating search.

Interpretability. We conduct a case study to compare the interpretability of solutions across different methods, as illustrated in Figure 3. While G-LLaVA produces the correct final answer, its reasoning process contains logical inconsistencies or factual errors. For GeoX and RIGOR with ESE, although they provide structured operations or applied theorems, the approaches lack detailed reasoning justifications and explicit premise-conclusion derivations. In contrast, RIGOR re-constructs proof-grounded reasoning traces with

explicit premises, applied rules, and conclusions for each inference step.

5 Related Work

SG-based Methods. SG-based methods use neural networks to generate solution programs or answers from multi-modal inputs. For example, NGS (Chen et al., 2021) encodes inputs using a multi-modal model trained with auxiliary tasks. Existing SG-based methods mainly improve GPS along three dimensions: (1) **Architecture:** Geoformer (Chen et al., 2022) adopts VL-T5 (Cho et al., 2021) as the base model, DPE-NGS (Cao and Xiao, 2022) refines the text encoder, and DualGeo-Solver (Xiao et al., 2024) designs a dual-system model. (2) **Perceptual Enhancement:** SCA-GPS (Ning et al., 2023) strengthens symbol character recognition, PGPSNet (Zhang et al., 2023) and DFE-GPS (Zhang et al., 2025b) enrich textual representations with image features, and LANS (Li et al., 2024) incorporates spatial layout awareness. (3) **MLLM Integration:** G-LLaVA (Gao et al., 2025) enhances LLaVA (Liu et al., 2023) via synthetic data training, GeoX (Xia et al., 2025) builds a geometric MLLM through a three-stage training pipeline, GeoGen (Pan et al., 2025) leverages symbolic systems to generate step-by-step reasoning data for training MLLMs and support symbolic verification, and GNS (Ning et al., 2025) decomposes the GPS process into multiple reasoning stages and provides a dataset with stage-specific annotations for training MLLMs.

Complementary to these model developments, GeoEval (Zhang et al., 2024) provides a comprehensive benchmark for evaluating geometric reasoning in LLMs and MLLMs.

DD-based Methods. DD-based methods decompose GPS into formal parsing and symbolic reasoning. For instance, InterGPS (Lu et al., 2021) constructs a neuro-symbolic geometry solver by combining diagram and text parsing with symbolic deduction. Existing DD-based GPS and geometry theorem proving studies mainly focus on: (1) **Search Optimization:** GeoDRL (Peng et al., 2023) guides the search process via reinforcement learning, E-GPS (Wu et al., 2024) introduces a data synthesizer to reduce solution steps, and FGeo-HyperGNet (Zhang et al., 2025a) uses hypergraphs to encode reasoning states. (2) **Visual Parsing:** PGDPNet (Zhang et al., 2022) designs specialized architectures for diagram parsing, and Pi-

GPS (Zhao et al., 2025) leverages visual information to resolve linguistic ambiguities. (3) **Auxiliary Construction:** AlphaGeometry (Trinh et al., 2024) trains a language model on large-scale synthetic data to propose auxiliary constructions that guide a symbolic deduction engine, substantially advancing Olympiad-level geometry theorem proving. AlphaGeometry2 (Chervonyi et al., 2025) further extends this framework with broader formal-language coverage, improved synthetic-data generation, enhanced symbolic reasoning, and a more effective search procedure.

In contrast, RIGOR revisits two underexplored interfaces in DD-based GPS: text-to-predicate parsing and algebraic target solving, complementing prior efforts on search optimization, visual parsing, and auxiliary construction.

Intermediate Representations. GSN generates atomic statements before rule-based formalization, thereby introducing an intermediate representation between natural language and formal predicates. Prior work on geometric reasoning has also explored structured intermediate representations that bridge neural and symbolic components, with GOLD (Zhang and Moshfeghi, 2024) being a representative work. GOLD converts formal relations extracted from geometric diagrams into structured natural-language descriptions that can be more readily consumed by LLMs. GSN instead operates in the reverse direction: it transforms natural problem text into atomic statements that facilitate rule-based formalization. Thus, GOLD makes formal information more accessible to neural reasoning, whereas GSN makes natural-language information more accessible to symbolic reasoning.

6 Conclusion

We introduced RIGOR, a neuro-symbolic framework for GPS that reformulates text parsing and symbolic reasoning in DD-based GPS. RIGOR proposes Atomic Semantic Normalization, instantiated by a GSN trained on GSN-15K, to map geometric texts into atomic statements for reliable formalization. It further introduces Declarative Constraint Reasoning through the DSE, which accumulates constraints globally and derives target values via query-and-refute entailment checking rather than eager symbolic elimination. Experiments on mainstream GPS benchmarks show that RIGOR achieves SOTA performance and produces interpretable reasoning traces.

Limitations

Although RIGOR demonstrates strong performance, it still has several limitations. First, RIGOR relies on existing diagram parsing techniques rather than explicitly optimizing visual perception, which may limit its robustness on geometry problems embedded in complex natural scene images. Improving visual perception and representation modeling could further extend RIGOR to more realistic and visually challenging scenarios. Second, RIGOR adopts a modular neuro-symbolic design, where the GSN performs atomic semantic normalization before the DSE conducts symbolic reasoning. While this design improves controllability and maintains a clear separation between linguistic normalization and formal reasoning, it does not yet support iterative feedback from the symbolic engine to the language model. Future work may explore solver-guided refinement or joint optimization, using symbolic failure signals to further improve semantic normalization.

Acknowledgments

We are grateful to the anonymous reviewers for their comments and suggestions. This work has been supported by the National Natural Science Foundation of China (NSFC) under grant No.62572461.

References

- Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. *Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond*. *Preprint*, arXiv:2308.12966.
- Jie Cao and Jing Xiao. 2022. *An augmented benchmark dataset for geometric question answering through dual parallel text encoding*. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 1511–1520. International Committee on Computational Linguistics.
- Jiaqi Chen, Tong Li, Jinghui Qin, Pan Lu, Liang Lin, Chongyu Chen, and Xiaodan Liang. 2022. *Unigeo: Unifying geometry logical reasoning via reformulating mathematical expression*. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 3313–3323. Association for Computational Linguistics.
- Jiaqi Chen, Jianheng Tang, Jinghui Qin, Xiaodan Liang, Lingbo Liu, Eric P. Xing, and Liang Lin. 2021.

- Geoqa: A geometric question answering benchmark towards multimodal numerical reasoning. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, Findings of ACL, pages 513–523. Association for Computational Linguistics.
- Yuri Chervonyi, Trieu H. Trinh, Miroslav Olsák, Xiaomeng Yang, Hoang H. Nguyen, Marcelo Menegali, Junehyuk Jung, Junsu Kim, Vikas Verma, Quoc V. Le, and Thang Luong. 2025. [Gold-medalist performance in solving olympiad geometry with alphageometry2](#). *J. Mach. Learn. Res.*, 26:241:1–241:39.
- Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. [Unifying vision-and-language tasks via text generation](#). In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Proceedings of Machine Learning Research, pages 1931–1942. PMLR.
- Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. 2000. [A deductive database approach to automated geometry theorem proving and discovering](#). *J. Autom. Reason.*, 25(3):219–246.
- Leonardo Mendonça de Moura and Nikolaj S. Bjørner. 2008. [Z3: an efficient SMT solver](#). In *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings*, Lecture Notes in Computer Science, pages 337–340. Springer.
- DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. [Deepseek-v3 technical report](#). *Preprint*, arXiv:2412.19437.
- Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, and Lingpeng Kong. 2025. [G-llava: Solving geometric problem with multi-modal large language model](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Christopher G. Harris and Mike Stephens. 1988. [A combined corner and edge detector](#). In *Proceedings of the Alvey Vision Conference, AVC 1988, Manchester, UK, September, 1988*, pages 1–6. Alvey Vision Club.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [Lora: Low-rank adaptation of large language models](#). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Zhongzhi Li, Ming-Liang Zhang, Fei Yin, and Cheng-Lin Liu. 2024. [LANS: A layout-aware neural solver for plane geometry problem](#). In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Findings of ACL, pages 2596–2608. Association for Computational Linguistics.
- Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. [Improved baselines with visual instruction tuning](#). In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 26286–26296. IEEE.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. [Visual instruction tuning](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. 2021. [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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6774–6786. Association for Computational Linguistics.
- Maizhen Ning, Qiu-Feng Wang, Kaizhu Huang, and Xiaowei Huang. 2023. [A symbolic characters aware model for solving geometry problems](#). In *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023, Ottawa, ON, Canada, 29 October 2023- 3 November 2023*, pages 7767–7775. ACM.
- Maizhen Ning, Zihao Zhou, Qiufeng Wang, Xiaowei Huang, and Kaizhu Huang. 2025. [GNS: solving plane geometry problems by neural-symbolic reasoning with multi-modal llms](#). In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 24957–24965. AAAI Press.
- OpenAI. 2024. Hello GPT-4o. <https://openai.com/index/hello-gpt-4o/>.
- Yicheng Pan, Zhenrong Zhang, Pengfei Hu, Jiefeng Ma, Jun Du, Jianshu Zhang, Quan Liu, Jianqing Gao, and Feng Ma. 2025. [Enhancing the geometric problem-solving ability of multimodal llms via](#)

- symbolic-neural integration. In *Proceedings of the 33rd ACM International Conference on Multimedia, MM 2025, Dublin, Ireland, October 27-31, 2025*, pages 5394–5403. ACM.
- Shuai Peng, Di Fu, Yijun Liang, Liangcai Gao, and Zhi Tang. 2023. *Geodrl: A self-learning framework for geometry problem solving using reinforcement learning in deductive reasoning*. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, Findings of ACL, pages 13468–13480. Association for Computational Linguistics.
- Bowen Ping, Minnan Luo, Zhuohang Dang, Chenxi Wang, and Chengyou Jia. 2026. *Autogps: Automated geometry problem solving via multimodal formalization and deductive reasoning*. Preprint, arXiv:2505.23381.
- Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloé Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross B. Girshick, Piotr Dollár, and Christoph Feichtenhofer. 2025. *SAM 2: Segment anything in images and videos*. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Vladmir Sicca, Tianxiang Xia, Mathis Fédérico, Philip John Gorinski, Simon Frieder, and Shangling Jui. 2024. *Newclid: A user-friendly replacement for alphageometry*. Preprint, arXiv:2411.11938.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. *Llama 2: Open foundation and fine-tuned chat models*. Preprint, arXiv:2307.09288.
- Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, He He, and Thang Luong. 2024. *Solving olympiad geometry without human demonstrations*. *Nat.*, 625(7995):476–482.
- Rafael Grompone von Gioi, Jérémy Jakubowicz, Jean-Michel Morel, and Gregory Randall. 2010. *LSD: A fast line segment detector with a false detection control*. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(4):722–732.
- Wu Wen-Tsün. 1986. *Basic principles of mechanical theorem proving in elementary geometries*. *J. Autom. Reason.*, 2(3):221–252.
- Wenjun Wu and Xiao-Shan Gao. 2007. *Mathematics mechanization and applications after thirty years*. *Frontiers Comput. Sci. China*, 1(1):1–8.
- Wenjun Wu, Lingling Zhang, Jun Liu, Xi Tang, Yaxian Wang, Shaowei Wang, and Qianying Wang. 2024. *E-GPS: explainable geometry problem solving via top-down solver and bottom-up generator*. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 13828–13837. IEEE.
- Renqiu Xia, Mingsheng Li, Hancheng Ye, Wenjie Wu, Hongbin Zhou, Jiakang Yuan, Tianshuo Peng, Xinyu Cai, Xiangchao Yan, Bin Wang, Conghui He, Botian Shi, Tao Chen, Junchi Yan, and Bo Zhang. 2025. *Geox: Geometric problem solving through unified formalized vision-language pre-training*. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Tong Xiao, Jiayu Liu, Zhenya Huang, Jinze Wu, Jing Sha, Shijin Wang, and Enhong Chen. 2024. *Learning to solve geometry problems via simulating human dual-reasoning process*. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pages 6559–6568. ijcai.org.
- Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. 2023. *mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration*. Preprint, arXiv:2311.04257.
- Jiaxin Zhang, Zhongzhi Li, Ming-Liang Zhang, Fei Yin, Cheng-Lin Liu, and Yashar Moshfeghi. 2024. *Geoeval: Benchmark for evaluating llms and multi-modal models on geometry problem-solving*. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, volume ACL 2024 of *Findings of ACL*, pages 1258–1276. Association for Computational Linguistics.
- Jiaxin Zhang and Yashar Moshfeghi. 2024. *GOLD: geometry problem solver with natural language description*. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, volume NAACL 2024 of *Findings of ACL*, pages 263–278. Association for Computational Linguistics.
- Ming-Liang Zhang, Fei Yin, Yi-Han Hao, and Cheng-Lin Liu. 2022. *Plane geometry diagram parsing*. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 1636–1643. ijcai.org.
- Ming-Liang Zhang, Fei Yin, and Cheng-Lin Liu. 2023. *A multi-modal neural geometric solver with textual clauses parsed from diagram*. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pages 3374–3382. ijcai.org.

Xiaokai Zhang, Yang Li, Na Zhu, Cheng Qin, Zhen-bing Zeng, and Tuo Leng. 2025a. [Fgeo-hypernet: Geometric problem solving integrating formalgeo symbolic system and hypergraph neural network](#). In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pages 4733–4741. ijcai.org.

Zeren Zhang, Jo-Ku Cheng, Jingyang Deng, Lu Tian, Jinwen Ma, Ziran Qin, Xiaokai Zhang, Na Zhu, and Tuo Leng. 2025b. [Diagram formalization enhanced multi-modal geometry problem solver](#). In *2025 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2025, Hyderabad, India, April 6-11, 2025*, pages 1–5. IEEE.

Junbo Zhao, Ting Zhang, Jiayu Sun, Mi Tian, and Hua Huang. 2025. [Pi-gps: Enhancing geometry problem solving by unleashing the power of diagrammatic information](#). In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pages 1526–1536. IEEE.

A Examples

A.1 An Example of Eager Elimination

Example 1. Consider a triangle ABC with altitude $AD \perp BC$. Starting from the Pythagorean constraints $\Phi = \{AD^2 + BD^2 = AB^2, AD^2 + (BC - BD)^2 = AC^2\}$, the eager elimination yields the resolved set Φ' :

$$\Phi' = \left\{ \begin{array}{l} BD = \frac{AB^2 + BC^2 - AC^2}{2BC}, \\ AD = \sqrt{AB^2 - \left(\frac{AB^2 + BC^2 - AC^2}{2BC} \right)^2} \end{array} \right\}.$$

Here, the side lengths $\{AB, AC, BC\}$ serve as the basis variables, and $\{AD, BD\}$ are explicitly resolved.

A.2 A Concrete Walkthrough of RIGOR

We use the problem in Figure 1 to illustrate the workflow of RIGOR.

Text Normalization, Parsing & Initialization: The GSN maps complex text such as “In circle O , chord $AD \parallel$ radius OC ” to normalized atomic statements: $\{“AD$ is a chord of circle $O”$, “ AD is parallel to OC ”, “ C is on circle O ” $\}$. The rule-based parser then maps these statements to formal predicates, e.g., $\text{IsChordOf}(A, D, O)$ and $\text{Para}(A, D, O, C)$. These predicates are merged with parsed diagram predicates to instantiate the initial state: $S_0.\mathcal{F} = \{\text{IsChordOf}(A, D, O), \dots\}$, $S_0.\mathcal{C} = \{CO = 2.5, AC = 4\}$, with target $g = CE$.

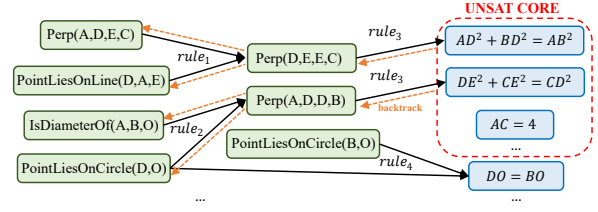


Figure 4: Trace reconstruction. The SMT solver returns an UNSAT core during refutation-based verification, identifying proof-relevant constraints from $\mathcal{C}_{red}$ as blue nodes; green nodes denote derived geometric facts in $S_{final}.\mathcal{F}$. RIGOR maps these constraints to $\mathcal{G}$ and recursively backtracks their antecedents to extract a proof-relevant subgraph, which is then topologically sorted into a human-readable reasoning trace (e.g., $\text{Collinear}(D, A, E) \wedge AD \perp EC \Rightarrow DE \perp EC, \dots, AD^2 + BD^2 = AB^2, \dots \Rightarrow CE = 2.4$).

Deductive Fixpoint Iteration: During iteration t , the DSE matches rule premises against $S_t.\mathcal{F}$. For example, a rule $r \in \text{KB}$, $\{\text{Perp}(x, m, m, y)\} \rightarrow (\emptyset, \{xm^2 + my^2 = xy^2\})$, matches the derived fact $\text{Perp}(D, E, E, C) \in S_t.\mathcal{F}$. The instantiated rule, $\{\text{Perp}(D, E, E, C)\} \rightarrow (\emptyset, \{DE^2 + CE^2 = CD^2\})$, is included in the matched rule set $\mathcal{M}$. APPLYRULES then produces $\Delta\mathcal{F} = \emptyset$ and $\Delta\mathcal{C} = \{DE^2 + CE^2 = CD^2\}$. This constraint is added to $S_t.\mathcal{C}$, and the corresponding dependency is recorded in the graph $\mathcal{G}$.

Value Propagation: At the fixpoint, the accumulated constraint set, e.g., $S_{final}.\mathcal{C} = \{CO = 2.5, AC = 4, CO = BO, \dots\}$, is simplified by propagating known values. For instance, from $CO = 2.5$ and $CO = BO$, the DSE derives $BO = 2.5$, yielding a reduced constraint set $\mathcal{C}_{red}$.

Query-and-Refute Target Derivation: The DSE queries an SMT solver over $\mathcal{C}_{red}$ to obtain a candidate value $CE = 2.4$. The DSE then verifies the candidate by checking $\mathcal{C}_{red} \wedge (CE \neq 2.4)$ with the SMT solver. The UNSAT result proves $\mathcal{C}_{red} \models (CE = 2.4)$.

Proof-Grounded Trace Reconstruction: After refutation, the SMT solver returns an unsatisfiable core that identifies proof-relevant constraints. RIGOR maps these constraints to their corresponding nodes in the dependency graph and recursively backtracks their antecedents to reconstruct the reasoning trace, as shown in Figure 4.

B Diagram Parsing Fallback Mechanism

Following the general scheme of prior work (Lu et al., 2021), we implement a fallback mechanism to process the low-resolution images. The method

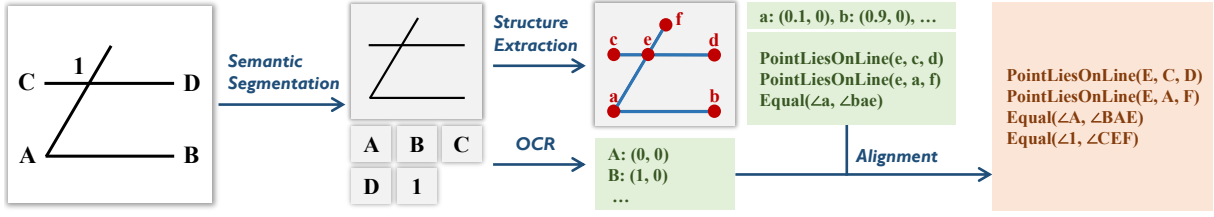


Figure 5: Pipeline of the diagram parsing fallback mechanism.

Algorithm 1 Declarative Symbolic Engine

Require: Initial state $S_0 = (\mathcal{F}_0, \mathcal{C}_0)$, target g , rule base KB

Ensure: Target value v and reasoning trace $\mathcal{T}$, or UnderDetermined

```

1:  $S \leftarrow S_0; \quad \mathcal{G} \leftarrow \emptyset$ 
2: // Deductive Fixpoint Iteration
3: repeat
4:    $S_{\text{old}} \leftarrow S$ 
5:    $\mathcal{M} \leftarrow \text{MATCH}(S, \mathcal{F}, \text{KB})$ 
6:    $(\Delta\mathcal{F}, \Delta\mathcal{C}) \leftarrow \text{APPLYRULES}(\mathcal{M})$ 
7:    $\text{UPDATEGRAPH}(\mathcal{G}, \mathcal{M}, \Delta\mathcal{F}, \Delta\mathcal{C})$ 
8:    $S \leftarrow (S.\mathcal{F} \cup \Delta\mathcal{F}, S.\mathcal{C} \cup \Delta\mathcal{C})$ 
9: until  $S = S_{\text{old}}$ 
10:  $\mathcal{C}_{\text{final}} \leftarrow S.\mathcal{C} \cup \mathcal{D}_{\text{domain}}$ 
11: // Value Propagation
12:  $\mathcal{C}_{\text{red}} \leftarrow \text{VALUEPROPAGATION}(\mathcal{C}_{\text{final}})$ 
13: // Target Derivation
14:  $v \leftarrow \text{SOLVER.GETMODEL}(\mathcal{C}_{\text{red}}, g)$ 
15:  $\rho \leftarrow \text{SOLVER.CHECK}(\mathcal{C}_{\text{red}} \wedge (g \neq v))$ 
16: if  $\rho = \text{UNSAT}$  then
17:   // Trace Reconstruction
18:    $\mathcal{K} \leftarrow \text{SOLVER.GETUNSATCORE}()$ 
19:    $\mathcal{T} \leftarrow \text{BACKTRACK}(\mathcal{G}, \mathcal{K})$ 
20:   return  $(v, \mathcal{T})$ 
21: else
22:   return UnderDetermined
23: end if

```

comprises three key steps: first, extracting semantic elements (e.g., textual labels and symbols) from the image; second, identifying geometric structural elements (e.g., points, lines) and their relationships; and finally, aligning semantic and structural elements to obtain formal descriptions. The process is illustrated in Figure 5.

Semantic Information Extraction. We employ the segmentation model SAM2 (Ravi et al., 2025) to extract candidate regions from the diagram D , followed by size-ratio filtering to select relevant regions $\{M_i\}_{i=1}^n$. Each region is then fed into an opti-

cal character recognition tool to extract semantic labels, yielding semantic elements $\{s_i\}_{i=1}^n$ with coordinates $\{\text{pos}(s_i)\}_{i=1}^n$, where $\text{pos}(s_i) = \text{pos}(M_i)$.

Structural Information Extraction. We apply computer vision algorithms to extract geometric structures. The segmented semantic regions are removed from the original diagram, leaving $D' = D \setminus \bigcup_i M_i$ to reduce noise in later processing. We apply Hough transforms to D' to extract parametric equations of circles and lines, use a line segment detector (von Gioi et al., 2010) to capture fine-grained segments, and apply Harris corner detection (Harris and Stephens, 1988) to identify intersections and endpoints. This yields geometric elements $\{p_j\}_{j=1}^m$ with positions $\{\text{pos}(p_j)\}_{j=1}^m$. Geometric structural relations such as collinearity and cocircularity are then determined by computing point-to-line and point-to-center distances, respectively.

Information Alignment. After extracting semantic and structural information, we establish correspondences between them by assigning semantic labels to geometric elements. We formulate this task as an optimization problem that finds an assignment matrix X to minimize the spatial distances between semantic and structural elements while satisfying spatial constraints:

$$\begin{aligned}
& \min_{X \in \{0,1\}^{n \times m}} \sum_{i,j} d_{ij} X_{ij} \\
& \text{s.t.} \quad \sum_j X_{ij} = 1, \quad \forall i, \\
& \quad \sum_i X_{ij} \leq 1, \quad \forall j, \\
& \quad X_{ij} = 1 \Rightarrow F(s_i, p_j), \quad \forall i, j,
\end{aligned}$$

where $d_{ij} = \text{dist}(\text{pos}(s_i), \text{pos}(p_j))$, and F defines spatial constraints for valid assignments. For example, assigning the label “ $\angle 1$ ” to an angle requires the semantic element “1” to lie within the angular region, rather than merely close to the vertex.

After determining the assignment matrix, the parser converts the extracted visual information

Setting	Completion Acc.
DeepSeek-V3 Direct Formalization	53.3
DeepSeek-V3 GSN + Rule Parser	56.0
Our GSN + Rule Parser	58.9

Table 5: Effect of rule-based formalization on GeoQA. Direct formalization prompts DeepSeek-V3 to generate solver-compatible formal predicates, while GSN + Rule Parser first normalizes text into atomic statements and then formalizes them by rules.

into formal descriptions. When $n < m$, some geometric elements are unlabeled; the parser assigns new symbolic identifiers to them.

C Pseudo-Code of DSE

Algorithm 1 summarizes the overall procedure of the DSE. Here, $\mathcal{M}$ denotes the set of matched rule instances, $\mathcal{G}$ is the dependency graph maintained for trace reconstruction, and $\mathcal{K}$ is the unsatisfiable core returned by the solver during refutation-based verification. If key conditions are missing due to parsing errors, the resulting constraint system may be under-determined.

D Additional Experiments

D.1 Direct LLM Formalization

Prior work (Lu et al., 2021) suggests that directly relying on language models for end-to-end formalization may suffer from stochastic outputs and insufficient formal precision. We further verify this observation on GeoQA by comparing direct DeepSeek-V3 formalization with atomic semantic normalization followed by rule-based formalization. As shown in Table 5, direct formalization achieves 53.3% completion accuracy, while DeepSeek-V3 GSN + Rule Parser achieves 56.0%, outperforming it by 2.7 percentage points.

This result supports our design choice of separating semantic normalization from rule-based formalization: the LLM handles flexible linguistic normalization, while the rule-based parser preserves the precision of predicate mapping.

D.2 Failure Analysis

We conduct failure analyses for both the GSN and the full RIGOR pipeline. For text normalization, we randomly sample 100 normalized outputs and manually verify whether the generated atomic statements correctly preserve the semantics of the original problem texts. The GSN achieves a sample-level normalization accuracy of 84.0%. We catego-

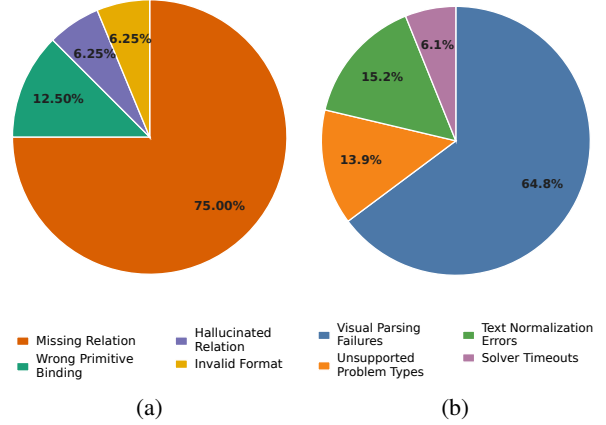


Figure 6: Failure analysis. (a) Error distribution of GSN text normalization among erroneous cases in 100 randomly sampled outputs, where the overall normalization accuracy is 84.0%. (b) Primary failure-source distribution of the full RIGOR pipeline, showing that visual parsing remains the dominant remaining bottleneck.

rize the remaining errors into four types: *Missing Relation*, where the model omits an implicit or explicit geometric relation; *Wrong Primitive Binding*, where the relation type is correct but associated with incorrect points, lines, or circles; *Hallucinated Relation*, where the model generates geometric facts unsupported by the problem text; and *Invalid Format*, where the output does not conform to parser-recognizable forms. The distribution of these error types is shown in Figure 6(a).

We also analyze the failed cases of the full RIGOR system by assigning each failure to its primary source. As shown in Figure 6(b), 64.8% of failures stem from visual parsing failures, especially in complex natural-scene diagrams; 13.9% come from unsupported problem types, such as moving-point problems; 15.2% are caused by text normalization errors, such as missing predicates; and 6.1% result from solver timeouts. Table 6 presents representative cases from the four major failure categories identified above, together with their corresponding failure analyses.

D.3 MR Distribution

Figure 7(a) visualizes the Mutual Runtime (MR) distribution on length problems in GeoQA. Each point denotes an instance solved by both DSE and ESE, and points above the dashed line indicate cases where DSE is faster. Most points lie above the line, with several ESE runtimes reaching the long-tail region, reinforcing the efficiency gains of DSE on algebraically complex length problems.

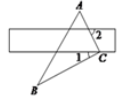
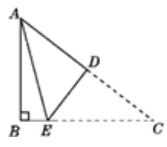
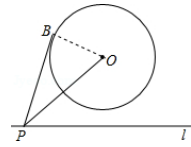
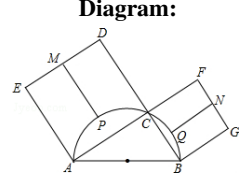
Problem Text: As shown in the figure, $\angle ACB$ is a right angle. Given $\angle 1 = N_0$, find the measure of $\angle 2$. Diagram parsing failure: The diagram parser misses the angle label “2”, so the queried angle cannot be grounded to the corresponding diagram primitives.	Diagram: 
Problem Text: In right $\triangle ABC$, $\angle B = N_0$, $AB = N_1$, and $AC = N_2$. Fold $\triangle ABC$ along ED so that C coincides with A. Find the perimeter of $\triangle ABE$. Text normalization error: GSN misinterprets the folding relation and normalizes it into the incorrect statement that both A and C lie on the crease ED, resulting in an incomplete intermediate representation with spurious relations.	Diagram: 
Problem Text: The radius of $\odot O$ is N_0, the distance from O to line l is N_1, and a moving point P lies on l. If PB is tangent to $\odot O$ at B, find the minimum value of PB. Unsupported problem type: The current formal language does not support moving-point optimization, so this problem falls outside the supported problem class.	Diagram: 
Problem Text: Point C lies on semicircle O with diameter AB; squares $ACDE$ and $BCFG$ are built on AC and BC; M, N are the midpoints of DE, FG; and P, Q are the corresponding arc midpoints. Given $MP + NQ = N_0$ and $AC + BC = N_1$, find AB. Solver timeout: The generated constraints lead to a difficult SMT instance, and the solver does not finish within the time limit.	Diagram: 

Table 6: Representative failure cases of RIGOR.

D.4 Robustness to Problem Scale

We evaluate the robustness of RIGOR to increasing problem scale in comparison with SOTA SG-based methods. We use the human-annotated Reasoning Steps (RS) in GeoQA as a proxy for problem scale, compute completion accuracy across different difficulty levels, and apply linear regression to model performance trends.

As shown in Figure 7(b), RIGOR and its variant exhibit shallower performance degradation slopes than SG-based methods, indicating better robustness to increasing problem scale.

D.5 Generalization of GSN

To evaluate whether GSN generalizes beyond GeoQA wording and the template structures in GSN-15K, we use the same 100 GeoQA examples from the normalization analysis in Appendix D.2 as the source set, denoted by *Original*. We construct two controlled linguistic-shift sets, *Paraphrase* and *TemplateShift*, by rewriting each source example while preserving its geometric semantics. *Paraphrase* rewrites each example with different surface wording while preserving the original problem structure. *TemplateShift* changes sentence structure and condition organization through operations such

as reordering, splitting, or merging conditions and replacing geometric templates with alternative formulations. We use GPT-4o to generate one variant of each type for every source example under the constraint that the semantic content of the original text remains unchanged. Each shifted set therefore contains 100 manually verified examples.

We quantify the resulting linguistic shifts with three complementary statistics. Let $\tau(\cdot)$ be a template abstraction function that replaces instance-specific geometric symbols with placeholders while preserving the surrounding sentence pattern. In the following, S denotes a shifted set, and R denotes a reference set (*Original* or GSN-15K).

Text template overlap. We first collect the templates of all problem texts in R :

$$\mathcal{T}_{\text{full}}(R) = \{\tau(x) \mid x \in R\}.$$

The text template overlap is the proportion of problem texts in S whose templates occur in this reference template set:

$$\frac{1}{|S|} \sum_{\tilde{x} \in S} \mathbf{1}[\tau(\tilde{x}) \in \mathcal{T}_{\text{full}}(R)].$$

This metric captures template overlap at the full-text level.

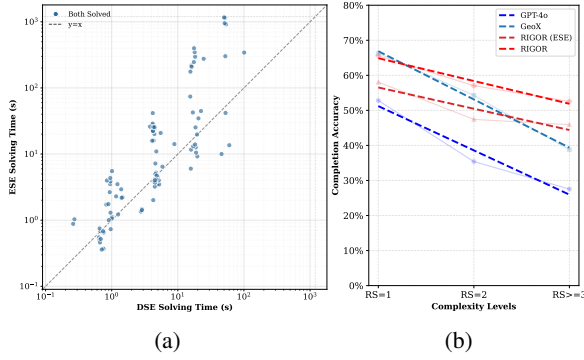


Figure 7: (a) MR distribution on length problems in GeoQA. (b) Completion accuracy across different reasoning-step levels in GeoQA.

Metric	Reference	Paraphrase	TemplateShift
Full-text overlap	Original	0.0%	0.0%
Full-text overlap	GSN-15K	0.0%	0.0%
Segment overlap	Original	6.3%	3.7%
Segment overlap	GSN-15K	5.8%	4.2%
3-gram Jaccard	Original	0.120	0.086

Table 7: Template and lexical shift statistics for the controlled evaluation sets.

Segment template overlap. Let $\text{Seg}(x)$ denote all local condition segments in problem x . We first collect the templates of all segments in R :

$$\mathcal{T}_{\text{seg}}(R) = \{\tau(s) \mid x \in R, s \in \text{Seg}(x)\}.$$

The segment template overlap is the proportion of all segment occurrences in S whose templates occur in this reference segment-template set:

$$\frac{1}{\sum_{\tilde{x} \in S} |\text{Seg}(\tilde{x})|} \sum_{\tilde{x} \in S} \sum_{\tilde{s} \in \text{Seg}(\tilde{x})} \mathbf{1}[\tau(\tilde{s}) \in \mathcal{T}_{\text{seg}}(R)].$$

This metric captures template overlap at the local segment level.

3-gram Jaccard similarity. To measure surface wording similarity, let $\{(x_i, \tilde{x}_i)\}_{i=1}^N$ be the paired original and rewritten examples, where $N = 100$. We tokenize x_i and $\tilde{x}_i$, construct their 3-gram sets $G_3(x_i)$ and $G_3(\tilde{x}_i)$, and report the average Jaccard similarity:

$$\frac{1}{N} \sum_{i=1}^N \frac{|G_3(x_i) \cap G_3(\tilde{x}_i)|}{|G_3(x_i) \cup G_3(\tilde{x}_i)|}.$$

Table 7 shows no full-text template overlap with either *Original* or GSN-15K. Segment-level overlap is at most 6.3%, and 3-gram similarity remains

Set	GSN	DeepSeek-V3
<i>Original</i>	84.0%	68.0%
<i>Paraphrase</i>	78.0%	64.0%
<i>TemplateShift</i>	75.0%	70.0%

Table 8: Normalization accuracy under controlled linguistic shifts.

low. These statistics indicate that both evaluation sets introduce substantial changes in wording and template structure.

We then manually evaluate normalization accuracy on the three sets. As shown in Table 8, GSN reaches 78.0% on *Paraphrase* and 75.0% on *TemplateShift*, corresponding to decreases of 6.0 and 9.0 percentage points from *Original*. GSN nevertheless outperforms the DeepSeek-V3 normalizer by 14.0 and 5.0 percentage points on the two shifted sets, respectively. These results suggest that GSN learns semantic normalization patterns that transfer beyond exact GeoQA-style wording.

E Experimental Details

E.1 Parameter Settings

We fine-tune GSN with LoRA (Hu et al., 2022), using a rank of $r = 128$ and a scaling factor of $\alpha = 256$. The model is trained for 3 epochs with a batch size of 2, using the AdamW (Loshchilov and Hutter, 2019) optimizer and a learning rate of 1×10^{-4} . All experiments are conducted with a random seed of 42.

E.2 Answer Extraction

We use a deterministic answer-extraction parser for general-purpose MLLMs. It first searches for final-answer markers such as “Answer:” and “The answer is”. For *Completion*, it parses the numeric answer from this span; for *Choice*, it extracts one of A–D. If no explicit marker is found, *Completion* falls back to the last numeric value in the response, while *Choice* falls back to the option whose text or numeric value is mentioned last in the response.

E.3 Prompts

We summarize the prompts used in our experiments. Figure 8 presents the prompts used to evaluate general-purpose MLLM baselines under different metrics, following the evaluation protocol of (Xia et al., 2025). Figure 9 presents the prompt templates used with the advanced LLM to generate normalized atomic statements (top) and construct the segment-level template library (bottom).

Completion Evaluation Prompt

System Prompt: First perform reasoning, then finally give the answer in the following format: e.g., Answer: [12.1].

Diagram: The Diagram is image_id.png

Question: In rhombus LMPQ, angle QLM = $2x^2 - 10$, angle QPM = $8x$, and MP = 10.0. Find the perimeter of LMPQ.

Choice Evaluation Prompt

System Prompt: First perform reasoning, then finally select the answer from the choices in the following format: The answer is (A), (B), (C) or (D).

Diagram: The Diagram is image_id.png

Question: As shown in the figure, BD bisects angle ABC, CD parallel AB, if angle BCD = 70.0 , then the degree of angle CDB is ().

Choices: (A): 55 (B): 50 (C): 45 (D): 30

Figure 8: Prompts used to evaluate general-purpose MLLM baselines. Different system prompts are used for completion and choice tasks.

Task Description.

Given a natural language description of a geometry problem, please decompose it into atomic geometric statements using the patterns below:

[Atomic statement patterns omitted for brevity.]

Examples.

Input: As shown in the figure, in circle O, chord BC $\parallel$ radius OA, AC and OB intersect at M, if the degree of $\angle C$ is N_0 , then the degree of $\angle AMB$ is ().

Output:

BC is the chord of O
A is on the circle O
BC is parallel to OA
M is on line AC
M is on line OB
 $\angle C = N_0$
Find $\angle AMB$

[Additional few-shot examples omitted.]

Input: [New problem description]

Output:

Task Description.

Given a natural language description of a geometry problem, please identify independent syntactic structures in the text and construct template mappings by translating each structure into atomic geometric statements using the patterns below:

[Atomic statement patterns omitted for brevity.]

Examples.

Input: As shown in the figure, in circle O, chord BC $\parallel$ radius OA, AC and OB intersect at M, if the degree of $\angle C$ is N_0 , then the degree of $\angle AMB$ is ().

Output:

in circle $[p_3]$, chord $[p_1][p_2] \parallel$ radius $[p_3][p_4] \rightarrow [p_1][p_2]$ is the chord of $[p_3]$, $[p_4]$ is on the circle $[p_3]$, $[p_1][p_2]$ is parallel to $[p_3][p_4]$
 $[p_1][p_2]$ and $[p_3][p_4]$ intersect at $[p_5] \rightarrow [p_5]$ is on line $[p_1][p_2]$, $[p_5]$ is on line $[p_3][p_4]$
if the degree of $\angle [p_1]$ is $N_0 \rightarrow \angle [p_1] = N_0$
then the degree of $\angle [p_1][p_2][p_3]$ is () $\rightarrow$ Find $\angle [p_1][p_2][p_3]$

[Additional few-shot examples omitted.]

Input: [New problem description]

Output:

Figure 9: Prompt templates used for generating normalized atomic statements (top) and constructing the segment-level template library (bottom).