From Symbolic Perception to Logical Deduction: A Framework for Guiding Language Models in Geometric Reasoning

Weichen Dai, Rafael Medeiros Cabral, Ziyi Shou, Yan Cao, Xin Shen, Dongcai Lu*, Yi Zhou*

University of Science and Technology of China

Abstract

Plane geometry remains a significant challenge in AI, requiring the integration of visual perception and mathematical reasoning. While Large Multimodal Models (LMMs) naturally handle visuo-linguistic inputs, they are often computationally intensive and opaque. We demonstrate that a pure Large Language Model (LLM), when equipped with specialized modules, can rival state-of-the-art LMMs on complex geometry problems. Our framework integrates a **Geometric Vision Parser**, which translates diagrams into symbolic form, with a **Symbolic Solver** that performs formal deductions, thereby mitigating hallucinations and promoting interpretable reasoning. To enable rigorous evaluation, we curate a benchmark of challenging problems from the 2025 Chinese Zhongkao examinations, ensuring data novelty and testing deeper deductive skills. Experiments demonstrate that our approach achieves performance comparable to Gemini 2.5 Pro while delivering clearer, human-like solutions.

Introduction

The pursuit of Artificial Intelligence for Mathematics (AI for Math) represents a significant frontier in machine intelligence, with plane geometry standing out as a quintessential grand challenge. Successfully solving geometry problems requires a sophisticated interplay of visual perception and mathematical reasoning—parsing complex diagrams and text, performing logical deductions, and even exhibiting creativity through auxiliary constructions. This process mirrors the progression from perception and cognition to decision-making, marking a critical pathway toward Artificial General Intelligence (AGI). Consequently, geometry problem solving has become a focal point for cutting-edge AI research.

Pioneering works, such as InterGPS (Lu et al. 2021) and AlphaGeometry (Trinh et al. 2024; Chervonyi et al. 2025), demonstrate the power of formal systems in achieving high precision. However, their reliance on formalized languages introduces significant overhead, including the risk of information loss during the translation from naturalistic problems and the inherent brittleness of rule-based systems. Furthermore, their symbolic, non-human-readable outputs pose considerable challenges for user readability. These formal systems are also invariably incomplete, for instance, neither interGPS nor AlphaGeometry can handle inequality or optimization problems in geometry.

The advent of Large Multimodal Models (LMMs), exemplified by systems like Gemini 2.5 Pro, has opened a promising new avenue. By natively processing visuo-linguistic information, LMMs bypass the need for explicit formalization and offer a natural, interactive user experience. Nevertheless, their state-of-the-art performance is not without tradeoffs. These models are computationally expensive, and their reasoning can be opaque. More critically, their performance on specialized domains like geometry is often constrained by distributional shifts between their generalist training data and the specific symbolic logic of geometric diagrams.

This paper challenges the prevailing assumption that end-to-end LMMs are the definitive solution for complex geometric reasoning. We demonstrate that a powerful Large Language Model (LLM), when augmented with specialized geometric perception and deductive reasoning capabilities, can not only match but surpass the performance of leading LMMs. While LLMs like DeepSeek-R1 possess immense abstract reasoning power, their inherent lack of visual processing has previously rendered them unsuitable for such tasks. Our work directly addresses this limitation by introducing a novel framework designed to empower LLMs for advanced geometric problem-solving. At its core are two critical components:

A Geometric Vision Parser, which translates unstructured diagram images into a symbolic, structured representation. By integrating OCR and geometric primitive detection, it provides the LLM with the precise, high-fidelity visual information it naturally lacks.

A Symbolic Solver module, which performs targeted formal deduction on angular relationships—a frequent source of LLM hallucination (Wang et al. 2025a). This not only enhances the model's accuracy but also critically guides it towards more elegant, deductive solutions, steering it away from brittle, brute-force coordinate calculations that plague many current models and degrade readability.

Figure 1 illustrates the overall workflow. The system first converts problem statements and associated diagrams into a formal representation of geometric entities and constraints. Next, the reasoning engine expands the list of geometry re-

*The corresponding authors.

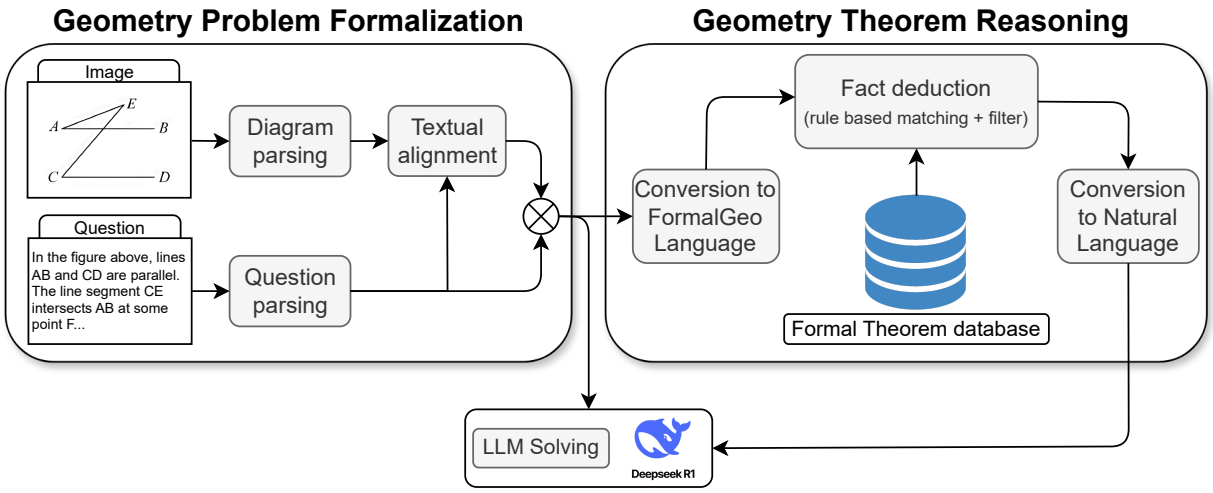


Figure 1: Schematic representation of the pipeline.

lationships already extracted from the problem (image and text), by applying to it theorems available in a theorem database. These intermediate results are translated into natural language and provided to the LLM, which synthesizes them into a complete solution with both deductive rigor and explanatory clarity.

To rigorously evaluate our approach against the state-of-the-art, we identified a critical gap in existing benchmarks. Datasets like GeoQA and Geometry3K, while foundational, are often saturated in performance and may suffer from data contamination, having been publicly available for years. For example, in Appendix B, Table 5 provides a few suspected cases of data contamination in Gemini-2.5-Pro.

More importantly, their structural simplicity fails to challenge models with complex multi-step deductions or the necessity for auxiliary line constructions. To address this, we introduce a new, challenging benchmark derived from 2025 Chinese Zhongkao (national middle school examination) math problems—a source of difficult problems that demand deep reasoning and creative insight, while also guaranteeing data novelty. Since these questions are from recent public examinations, they are highly unlikely to have been included in the pre-training data of existing large models, thus enabling a fair evaluation of geometric reasoning capabilities without the risk of data contamination.

Our contributions are threefold:

- We propose a novel framework that, for the first time, enables a LLM (R1) to achieves comparable performance to a state-of-the-art LMM (Gemini 2.5 Pro) on complex, unseen plane geometry problems.
- We introduce a new, high-difficulty benchmark curated from 2025 Chinese Zhongkao examination questions, providing a more challenging and reliable testbed for future research in geometric reasoning.
- New Geometric Vision Parser and Symbolic Solver modules that guide LLMs towards human-like, interpretable reasoning by substantially reducing the model's reliance on non-intuitive, "brute-force" coordinate geometry, thereby enhancing the readability and user-friendliness of the solutions.

Related Work

Solving plane geometry problems with AI has been a long-standing challenge, requiring both sophisticated multimodal understanding of text and diagrams, and rigorous logical deduction. Early efforts relied on symbolic systems, which have gradually given way to data-driven neural networks and powerful hybrid architectures that integrate the strengths of both paradigms.

Symbolic-Based Methods

Early approaches to automated geometry problem solving were dominated by symbolic and rule-based systems. These methods prioritize logical soundness and interpretability. They typically employ meticulously crafted rule-based parsers to convert the natural language text into a formal, symbolic language. The diagrammatic primitives are also converted into formal descriptions through manual translation or a trained diagram parser. PGDP-Net (Zhang et al. 2022) models geometric element recognition as an entity segmentation task and uses a Graph Neural Network (GNN) to identify element classes and their relationships, outputting a formal language description. The problem is then solved using symbolic reasoners that operate within this formal system. Pioneering works like Inter-GPS and Formal-Geo (Zhang et al. 2024) are exemplars of this approach. While powerful in logical consistency, their reliance on hand-crafted rules can limit their scalability.

Neural Network-Based Methods

With the rise of deep learning, researchers began exploring end-to-end neural models that learn to solve geometry problems directly from data. These methods encode the problem into dense vector representations, utilizing neural net-

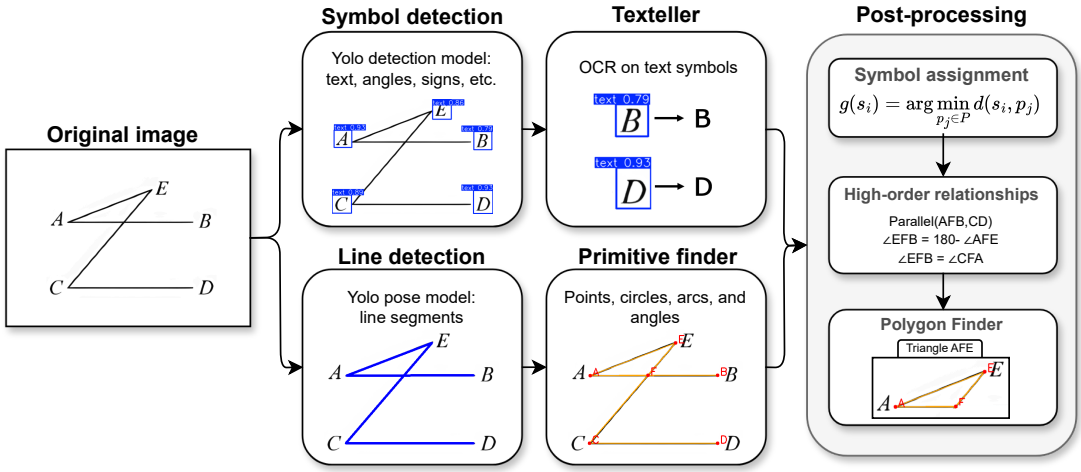


Figure 2: Diagram parser pipeline. From this pipeline we extract relevant primitives and relationships from the diagram image.

works to predict solutions based on the learned features. For instance, NGS (Chen et al. 2021) employed a text encoder to learn textual features of the problem, and introduced auxiliary tasks like a jigsaw puzzle game to learn robust visual representations of geometric diagrams. Building on this, models like PGPSNet (Zhang, Yin, and Liu 2023) utilized dual-stream encoders with bidirectional GRUs to fuse textual and visual features. LANS (Li et al. 2023) further refined this architecture by introducing a layout-aware pre-training module and employing contrastive learning to enhance the alignment between visual points and textual tokens, leading to more precise multimodal understanding before feeding the fused representation to a sequence-to-sequence decoder.

Hybrid Methods

To combine the logical rigor of symbolic systems with the rapid decision-making strengths of neural networks, a significant body of work has focused on hybrid, or neuro-symbolic, methods. A dominant paradigm in this area involves using neural networks as theorem prediction modules to accelerate inference—addressing the fact that traditional symbolic approaches often rely on time- and computationally expensive search processes. For example, GeoDRL (Peng et al. 2023) represents the problem as a geometric logic graph and trains a GNN via reinforcement learning to act as a policy network, which selects the most promising theorem to apply within a formal reasoning system at each step of the reasoning process.

Recent Trends with Large Models

Recently, LLMs have been adopted as powerful backend reasoners, with methods like GeoX (Xia et al. 2024), DFE-GPS (Zhang et al. 2025), and Pi-GPS (Zhao et al. 2025) leveraging them for the final solving stage. The focus has now shifted towards augmenting these large foundational models with specialized strategies to enhance their geometric reasoning capabilities. These strategies include refining the formal problem representation (Ping et al. 2025), generating large-scale annotated datasets (Wu et al. 2025), improving inference-time reasoning with advanced search algorithms (Wang et al. 2025c), and using reinforcement learning to tackle notoriously difficult sub-problems like auxiliary line construction (Wang et al. 2025d).

Method

Solving geometry problems, especially those that heavily rely on diagrams, requires a precise understanding of visual geometric relationships. Unlike algebraic or purely text-based tasks, geometric problem solving often depends on recognizing implicit constraints, such as collinearity, angle properties, that are visually encoded in diagrams rather than explicitly stated in text. However, current multimodal models exhibit significant limitations in accurately parsing and reasoning about geometric diagrams (Wang et al. 2025b), which poses a major challenge for automated geometry problem solving. We propose a hybrid framework for automated geometry problem solving that integrates formalized problem parsing, theorem-driven reasoning, and LLM inference. Unlike traditional symbolic geometry solvers, our approach leverages both rule-based theorem application and the generative capabilities of LLMs to produce human-readable solutions. As shown in Figure 1, the framework consists of three core modules: (1) Geometric Problem Formalization, (2) Geometric Theorem Reasoning, and (3) LLM-based Answer Generation.

Geometric Problem Formalization

Solving geometry problems requires a precise formal representation of both diagrammatic and textual information. Inspired by Inter-GPS (Lu et al. 2021), we first design a bottom-up diagram parsing pipeline for extracting and constructing geometric information from diagrams. A visual representation of this pipeline is shown in Figure 2.

The pipeline begins with the extraction of geometric primitives, which serve as the foundational elements of the diagram representation. A primitive is defined as basic geometric elements such as a point, line, circle, or arc segment. We start by using a fine-tuned YOLO pose model (Khanam and Hussain 2024) to identify geometric line segments. This is followed by a "primitive finder" step that merges overlapping or closely aligned detections into consistent line entities. During this phase, we also detect more primitives by using Hough transforms to find circles and angular structures. Points and angles are then located based on the intersections between the detected lines and circles.

Beyond geometric primitives, the parser also identifies diagrammatic symbols (such as angle marks and right-angle indicators) and textual annotations (such as point labels or length values). Symbol and text regions are first localized using a finetuned YOLO detection model. The content in the text regions is then recognized with a specialized recognition model (Texteller[1]). At this stage, we obtain a primitive set $P = \{p_1, p_2, \ldots, p_m\}$ and a symbol set $S = \{s_1, s_2, \ldots, s_n\}$. To create meaningful connections between these components, we assign each symbol with its corresponding primitive using a greedy assignment strategy. This process is formally defined as:

$$g(s_i) = \arg\min_{p_j \in P} d(s_i, p_j), \ \ s.t. \ (s_i, p_j) \in \text{Feasibility Set } F.$$

Here, $d(s_i, p_j)$ represents the Euclidean distance between the detected location of symbol s_i and the candidate primitive p_j. A key constraint is that the symbol s_i and primitive p_j must be compatible, belonging to a defined feasibility set F. For example, the text symbol "10°" can only be assigned to angle or arc measure primitives, and the perpendicular symbol can only be assigned to orthogonal line primitives. This ensures that each symbol is assigned to the closest compatible primitive.

We further introduce post-processing steps to derive higher-order geometric relations and align them with textual descriptions. Such enriched representations are essential for guiding the LLM in subsequent reasoning stages. While the primitive-level elements (e.g., points, lines, circles) provide a structural foundation, additional relational information enables a more expressive and semantically meaningful formalization. For example, besides extracting simple relationships like `PointLiesOnLine(Point(F),Line(A,B))`, which only indicates a collinearity relation, we also encode collinear relationships in a systematic left-to-right, top-to-bottom order to fully capture the spatial relationships between points. In addition, we extend the relationship list to incorporate cyclic relations, polygonal structures, and angle dependencies. A detailed catalog of these formal commands, along with additional examples, is provided in the Appendix B, Table 4.

Diagrams in geometry problems are often just schematic sketches, not precise drawings, so their visual information can be redundant or even misleading. To ensure we only use relevant and correct geometry information, after diagram parsing, we perform a step called "textual alignment" (see Figure 1) to align the visual data with the text. We utilize a rule-based parsing strategy to filter and refine the information extracted from the diagram image. For example, if the problem text states that the figure contains a rectangle $ABCD$, our system automatically adds the defining properties of a rectangle —such as opposite sides being parallel ($AB \parallel DC$) and adjacent sides being perpendicular ($AD \perp AB$) —to its list of known relationships. This approach guarantees that our system's reasoning is based on correct geometric properties derived from the text, not on potentially inaccurate visual relationships extracted from the image.

[1]https://github.com/OleehyO/TexTeller

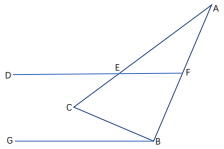


Figure 3: Example problem for applying M-theorem.

Geometric Theorem Reasoning

While LLMs have demonstrated remarkable capabilities in complex reasoning, they frequently produce hallucinations, particularly in problems concerning angles. The core of this issue lies in the LLM's reliance on analytical methods, such as constructing parametric equations, to validate steps. This paradigm fails to capture fundamental, non-analytic geometric knowledge. For example, analytic geometry can confirm the equality of vertical angles ($\angle 1 = \angle 2$) via dot products but cannot intrinsically represent the concept of "being vertical angles". This relationship is defined axiomatically ("opposite angles at an intersection"), a form of knowledge that LLMs struggle to ground.

To bridge this gap, we propose a Theorem Reasoning Module that deduces angle properties based on formal geometric rules. This module acts as an external verifier, providing the LLM with reliable geometric facts for its subsequent reasoning. The process includes three key steps: Literal Expansion, Fact Deduction, and Fact Filtering.

We first process the diagram parser's output. This involves not only translating the elements into the FormalGeo symbolic language but also augmenting the representation by inferring implicit relationships. For instance, we expand collinear relations (e.g., `Collinear(A,B,C) & Collinear(B,C,D)` implies `Collinear(A,B,C,D)`) and composite angles (e.g., `Collinear(A,B,C) & Angle(D,A,C)` implies `Angle(D,A,B)`). This augmentation provides a richer set of primitives and relations for the next step, reducing inferential gaps.

Secondly, we match the parsed geometric elements against the input parameters of rules defined in our theorem library to derive new facts. However, an exhaustive matching approach, which iterates through every possible combination, is computationally prohibitive. As illustrated in Figure 3, a simple diagram with 7 points may yield 21 distinct line segments. To apply a theorem that requires four

Dataset	# Problems	Block	Counts					
			1/A	2/B	3/C	4/D	5/E	Avg./F
ZhongkaoGeo-L1	89	Sub-questions	69	15	5	0	0	1.281
		Knowledge types	17	38	7	7	12	8
ZhongkaoGeo-L2	83	Sub-questions	46	14	17	5	1	1.807
		Knowledge types	28	20	4	9	17	5

Table 1: A detailed statistical comparison of the ZhongkaoGeo-L1&L2 datasets, highlighting the increased difficulty of our proposed benchmark. In each data block, the top row shows the distribution of problems by the number of sub-questions (‘1‘ to ‘5‘ and their average), while the bottom row shows the distribution by knowledge type (‘A‘ to ‘F‘). The knowledge types are defined as A: Angle, B: Length, C: Area, D: Comprehensive, E: Transformation, and F: Optimization.

specific line segments, such as an ”M-theorem” defined as `m_model_angle_equation(AB,BC,CD,DE)`, a brute-force approach would need to evaluate at least $21 \times 20 \times 19 \times 18 = 143,640$ permutations, resulting in an intractably large search space. To mitigate this combinatorial explosion, we introduce constraints by defining strict ordering rules for points within our theorem definitions. We further leverage geometric priors, such as parallel and perpendicular relationships, to prune invalid argument combinations. Through these heuristics, we successfully reduced the number of candidate permutations.

In the final step, we assess the relevance of the derived conclusions to the original problem statement. Only the pertinent conclusions are selected and provided as supplementary input to the LLM.

Formal Definitions of Reasoning Steps Crucially, the inclusion of the filtered deductions is the primary mechanism for constraining the LLM's reasoning space. We define the search space as a state graph $\mathcal{G} = (S, E)$, where S represents the set of known geometric facts. Within the Theorem Reasoning Module, we model the problem-solving process as a state graph search, effectively pruning the search space to mitigate the combinatorial explosion often seen in theorem application. This is achieved through two strategies: (1) *Rule-based Literal Expansion*, where an expansion operator $\Phi_{exp} : S \rightarrow S^{'}$ applies axiomatic transitivity without costly database searches, e.g., inferring ordered segments from collinear primitives; and (2) *Constraint-Based Pruning*, which employs a geometric feasibility filter $\mathcal{F}$. Here, for example, a theorem $\mathcal{T}(arg_1, arg_2)$ is strictly triggered only if its arguments satisfy specific geometric types and spatial constraints (e.g., $arg_1 \in$ **Lines**, $arg_2 \in$ **Angles**), significantly reducing the branching factor from $O(N^k)$ to $O(N_{valid})$, where $N_{valid} \ll N$.

LLM-based Answer Generation

The final stage of our framework leverages a LLM to produce human-readable solutions. To achieve this, we synthesize the outputs from all preceding modules to construct a comprehensive, task-specific prompt. This prompt is meticulously structured to integrate several key streams of information:

1. the verbatim textual description of the geometry problem,
2. the parsed spatial and logical relationships between geometric entities and the pixel coordinates of key points, which serve as a reference for visual grounding, and
3. the set of high-relevance geometric deductions produced and filtered by our Theorem Reasoning Module.

By assembling the input in this manner, we create an augmented prompt that grounds the LLM in the specific context of the problem. Rather than navigating a vast stochastic search space, the LLM is guided by these verified conclusions, keeping its generation path aligned with a logically sound trajectory and reducing the risk of hallucination. This enriched input is then fed to the LLM to generate the final, step-by-step derivation.

ZhongKao Geometry Benchmark

As the reasoning capabilities of LLMs and LMMs continue to advance, the need for robust and challenging evaluation benchmarks becomes increasingly critical. We posit that a high-quality benchmark must satisfy three core principles: Quality (well-posed problems with clear descriptions), Difficulty (requiring a significant number of reasoning steps or solving for multiple objectives), and Diversity (broad coverage of problem types and concepts).

However, existing plane geometry benchmarks are becoming saturated, failing to adequately challenge the capabilities of state-of-the-art models. For instance, the GeoQA (Chen et al. 2021) training set is heavily skewed towards angle and length calculations, with only 267 of its 3,509 problems involving area or other concepts. Its complexity is limited, with an average of just 1.96 reasoning steps (max 4). While the GeoQA+ (Cao and Xiao 2022) dataset improves on diversity, it remains constrained in difficulty, with an average of 2.23 reasoning steps (max 8). [2]

To address these limitations, we introduce a new benchmark suite, systematically sourced from recent Chinese Zhongkao Examination, which are renowned for their quality, complexity, and diversity. Furthermore, we contend that a fourth crucial principle for modern benchmarks is Contamination Prevention. The risk that a model has been exposed

[2]We report statistics from the training sets as presented in the original papers; their random-split methodology ensures these distributions are representative of the test sets.

to test data during its pre-training phase poses a significant threat to the validity of the evaluation.

Adhering to these four principles, we designed a three-tiered evaluation suite for plane geometry with progressively increasing difficulty and reduced risk of data contamination:

ZhongkaoGeo-L1. Sourced primarily from official and mock examination papers from 2023 and earlier. This dataset covers a wide spectrum of formats (e.g., multiple-choice, calculation, and proof problems) and assesses diverse concepts, including: angle calculation, length (ratio/perimeter/trigonometric-value) calculation, area (ratio) calculation, comprehensive problems (requiring a combination of the above), composite transformations (rotation, translation, symmetry, dynamic points), and optimization (finding extremal values).

ZhongkaoGeo-L2. Sourced from examinations administered between 2024 and early 2025. This dataset increases complexity by featuring a higher proportion of problems with multiple sub-questions and a greater emphasis on difficult problem archetypes (37% vs. L1's 30% on Knowledge type D+F+G). See Table 1 for a detailed comparison.

ZhongkaoGeo-L3. Comprises plane geometry problems from the official 2025 examinations in Chinese provincial capitals and major municipalities. This dataset consists of entirely novel data, providing the most reliable assessment of a model's un-memorized reasoning capabilities.

After a meticulous curation process, where we discarded problems with unclear images, text-diagram mismatches, or missing/ambiguous solutions, our final benchmark suite consists of 89 problems in ZhongkaoGeo-L1, 83 in ZhongkaoGeo-L2, and 105 in ZhongkaoGeo-L3. Notably, unlike traditional plane geometry benchmarks which typically present a single problem with one question and one solution target, the problems in our benchmark are often structured as multi-step tasks, containing several related sub-questions that require achieving multiple solution targets.

Experiments

In this section, we conduct a series of experiments to evaluate the effectiveness of our proposed method on three datasets of increasing difficulty: ZhongkaoGeo-L1, ZhongkaoGeo-L2, and ZhongkaoGeo-L3. Our method is built on Deepseek-R1.

Baselines

To rigorously evaluate our approach, we compare it against a suite of state-of-the-art large models, renowned for their powerful reasoning and general problem-solving capabilities. Our selection prioritizes models that have demonstrated leading performance on various general-purpose benchmarks, ensuring a high-standard and relevant comparison. The chosen baselines are LLMs: Qwen3-32B & Qwen3-235B, DeepSeek-R1, and LMMs: GPT-o1, Gemini 2.5-Pro.

Evaluation Metrics

For the L1 and L2 datasets, we use a strict accuracy metric. A question is marked as correct (score of 1) only if the generated answer perfectly matches the ground truth. Any deviation, however minor, results in a score of 0. The final accuracy is calculated as the ratio of correctly answered questions to the total number of questions.

Since the L3 dataset is directly sourced from the Zhongkao examinations held this year, we adopt a Scoring Rate metric based on the official grading rubrics of the Zhongkao examinations. Each sub-question within a problem is assigned a specific point value. A model receives the full points for a sub-question if its answer is correct and zero points otherwise. The total score for a problem is the sum of scores from its constituent sub-questions. The final Scoring Rate is the ratio of the total score achieved by the model to the maximum possible score across the entire dataset.

MODEL	ZhongKao-L1	ZhongKao-L2
Qwen3-32B	84.64	64.26
Qwen3-235B	86.52	69.48
DeepSeek-R1	88.39	67.87
GPT-o1	82.02	56.63
Gemini2.5-Pro	**94.38**	73.49
Ours	92.13	<u>74.30</u>
Ours (Major@3)	<u>93.26</u>	**78.31**

Table 2: Performance (Accuracy) on ZhongkaoGeo-L1&L2

Implementation Details

All experiments are conducted using the official APIs or publicly available weights of the baseline models to ensure fair comparison. The input for LLMs was solely the textual description of the problem, while for LMMs, it included both the image and the text. For closed-source models, we set the Temperature to 0.1 to obtain stable outputs, which also helps reduce API call costs. For open-source models, specifically the Qwen series, since the official documentation states that in reasoning mode "greedy decoding should not be used as it may lead to performance degradation and endless repetition," we followed the official recommendations and used the default parameters, reporting the average performance over three runs. Additionally, since DeepSeek-R1's reasoning mode does not support adjusting the Temperature parameter—meaning greedy decoding cannot be enabled—its single-run outputs are less stable compared to other closed-source models. As a result, we also report the average performance of R1 across three runs. To assess performance stability and potential gains from ensembling, we also report results for "Ours (Major@3)". This strategy involves performing three independent inference runs and selecting the most frequent answer as the final output through a majority vote. The prompt for our full pipeline is designed as `Prompt Template with Geometric Information` in Appendix A, while for all other situations, we use the `Normal Problem-Solving Prompt`.

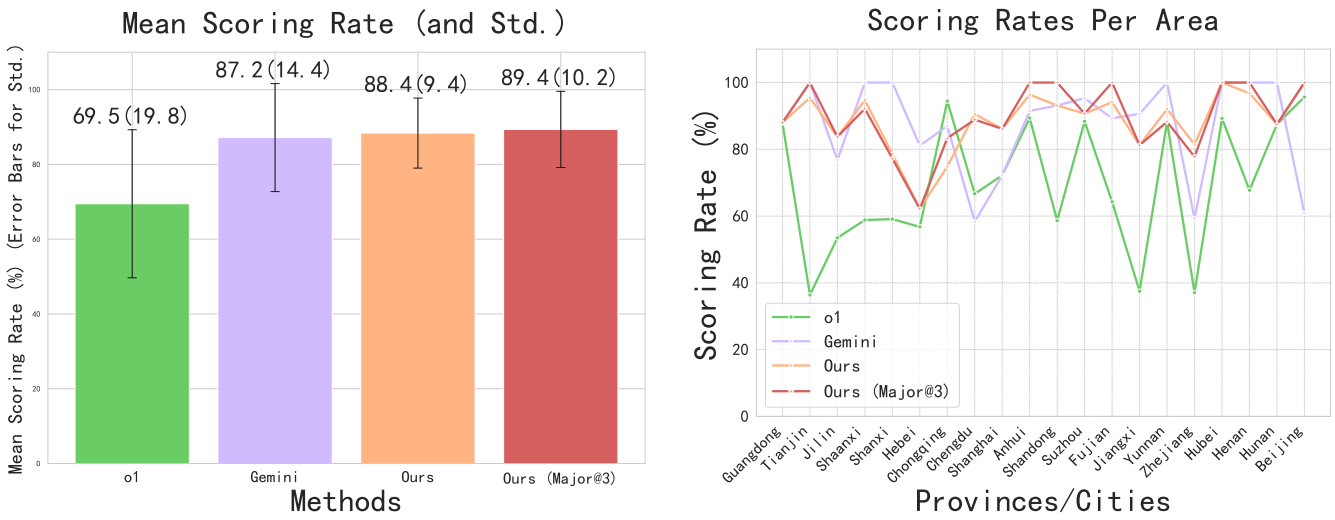


Figure 4: Performance (Scoring Rate) on ZhongkaoGeo-L3. We also report the standard deviation of score rates computed over different regional exam papers.

Relation Type	Qwen-VL-2.5-72B	Ours	Improvement (△)
Collinearity	45	**85**	+40
Concyclicity	74	**95**	+21
Angular Position Relations	46	**86**	+40
All Relations Correct	23	**81**	**+58**

Table 3: Performance comparison on the parsing of implicit geometric relations on the ZhongkaoGeo-L3 dataset. We report the number of problems where each type of relation was correctly identified. "All Relations Correct" is a holistic metric counting problems where all three relation types were successfully parsed. Our method demonstrates a substantial improvement across all categories, highlighting its superior diagram understanding capabilities.

Results and Analysis

Performance on ZhongkaoGeo-L1 & L2. The comparative results on the L1 and L2 datasets are presented in Table 2. Our method demonstrates highly competitive performance. In its standard configuration ("Ours"), it achieves an accuracy of 92.13% on L1 and 74.30% on L2, outperforming most SOTA baselines, including the large-scale Qwen3-235B and DeepSeek-R1 models. The significant performance drop from L1 to L2 across all models underscores the increased difficulty and complexity of the L2 dataset. This further reveals the higher risk of data leakage associated with earlier benchmarks. When employing the majority voting strategy ("Ours (Major@3)"), our model sets a new state-of-the-art on both datasets, reaching 93.26% on L1 and a remarkable 78.31% on L2. This result highlights the robustness and reliability of our approach.

Performance on ZhongkaoGeo-L3. As shown in Figure 4, we further compare our method with LMMs on the most challenging ZhongkaoGeo-L3 dataset, which requires greater reliance on integrating image and text information. The bar chart on the left illustrates the mean scoring rate. Our method ("Ours") achieves a mean scoring rate of 88.4%, which is already superior to Gemini 2.5-Pro's 87.2%. Furthermore, the error bars, representing standard deviation, indicate that our method's performance is stable, with a variance comparable to that of Gemini 2.5-Pro. The line plot on the right provides a more granular, per-province/city view of the scoring rates. This plot reveals that while all models exhibit performance fluctuations depending on the specific exam questions, our method (orange and red lines) consistently tracks or exceeds the performance of the best baseline (Gemini 2.5-Pro, purple line). The performance of our method under the major@3 setting is largely consistent with that under the standard configuration ("Ours"), which further demonstrates that our approach can enhance the stability of the model in solving problems. This fine-grained analysis confirms that our model's superiority is not due to overfitting on specific question types but rather a consistent, high-level reasoning capability across a diverse range of problems.

In summary, the experimental results across all three datasets demonstrate that our proposed method achieves state-of-the-art performance, outperforming even the most advanced proprietary LMMs.

More Experiments

Analysis of Geometric Parsing Performance To solve geometry problems, a model must parse implicit visual relations from diagrams. We evaluated our method's ability to do this on the ZhongkaoGeo-L3 dataset, comparing it against the strong Qwen-VL-2.5-72B baseline. The results, shown in Table 3, unequivocally demonstrate our method's superiority. Component-wise, our model nearly doubles the baseline's performance in identifying individual relations like collinearity and angular positions. More critically, for holistic understanding (parsing all relations in a problem correctly), our model achieves a score of 81 compared to the baseline's 23—a greater than 3.5-fold improvement. This substantial leap shows that our method consistently constructs a comprehensive and accurate symbolic representation of the geometric figure. While baseline models often miss necessary conditions, our approach provides a much stronger foundation for successful, multi-step reasoning.

Effect of Geometric Theorem Reasoning With the help of Geometric Theorem Reasoning, our method is available to guide the model to a readable solution instead of the brute-force coordinate method, as shown in Appendix B, Figure 5 and 6.

Conclusion

This work challenges the prevailing paradigm that ever-larger multimodal models are the predominant solution to advancing geometric reasoning. We have demonstrated that a pure LLM, when augmented with a specialized Geometric Vision Parser and a Symbolic Solver, can achieve and even surpass the performance of a state-of-the-art LMM like Gemini 2.5 Pro. Our introduction of the challenging, contamination-free ZhongkaoGeo benchmark provides a rigorous new standard for evaluating true deductive capabilities. More importantly, by promoting human-like deductive reasoning over opaque, brute-force methods, our framework advances the development of transparent and interpretable AI for mathematical problem-solving.

Acknowledgments

This work was supported by grants from the Anhui Province Science and Technology Tackle Plan Project (NO.202423k09020008).

References

Cao, J.; and Xiao, J. 2022. An augmented benchmark dataset for geometric question answering through dual parallel text encoding. In *Proceedings of the 29th international conference on computational linguistics*, 1511–1520.

Chen, J.; Tang, J.; Qin, J.; Liang, X.; Liu, L.; Xing, E. P.; and Lin, L. 2021. Geoqa: A geometric question answering benchmark towards multimodal numerical reasoning. *arXiv preprint arXiv:2105.14517*.

Chervonyi, Y.; Trinh, T. H.; Olšák, M.; Yang, X.; Nguyen, H.; Menegali, M.; Jung, J.; Verma, V.; Le, Q. V.; and Luong, T. 2025. Gold-medalist performance in solving olympiad geometry with alphageometry2. *arXiv preprint arXiv:2502.03544*.

Khanam, R.; and Hussain, M. 2024. Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*.

Li, Z.-Z.; Zhang, M.-L.; Yin, F.; and Liu, C.-L. 2023. LANS: A layout-aware neural solver for plane geometry problem. *arXiv preprint arXiv:2311.16476*.

Lu, P.; Gong, R.; Jiang, S.; Qiu, L.; Huang, S.; Liang, X.; and Zhu, S.-C. 2021. Inter-GPS: Interpretable Geometry Problem Solving with Formal Language and Symbolic Reasoning. In *The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*.

Peng, S.; Fu, D.; Liang, Y.; Gao, L.; and Tang, Z. 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*, 13468–13480.

Ping, B.; Luo, M.; Dang, Z.; Wang, C.; and Jia, C. 2025. Autogps: Automated geometry problem solving via multimodal formalization and deductive reasoning. *arXiv preprint arXiv:2505.23381*.

Trinh, T.; Wu, Y.; Le, Q.; He, H.; and Luong, T. 2024. Solving Olympiad Geometry without Human Demonstrations. *Nature*.

Wang, X.; Wang, Y.; Zhu, W.; and Wang, R. 2025a. Do Large Language Models Truly Understand Geometric Structures?

Wang, X.; Wang, Y.; Zhu, W.; and Wang, R. 2025b. Do Large Language Models Truly Understand Geometric Structures? In *The Thirteenth International Conference on Learning Representations*.

Wang, Y.; Wang, S.; Cheng, Q.; Fei, Z.; Ding, L.; Guo, Q.; Tao, D.; and Qiu, X. 2025c. Visuothink: Empowering lvlm reasoning with multimodal tree search. *arXiv preprint arXiv:2504.09130*.

Wang, Y.; Wang, Y.; Wang, D.; Peng, Z.; Guo, Q.; Tao, D.; and Wang, J. 2025d. Geometryzero: Improving geometry solving for llm with group contrastive policy optimization. *arXiv preprint arXiv:2506.07160*.

Wu, W.; Wang, Z.-k.; Ye, J.; Zhou, Z.; Li, Y.-F.; and Guo, L.-Z. 2025. NeSyGeo: A Neuro-Symbolic Framework for Multimodal Geometric Reasoning Data Generation. *arXiv preprint arXiv:2505.17121*.

Xia, R.; Li, M.; Ye, H.; Wu, W.; Zhou, H.; Yuan, J.; Peng, T.; Cai, X.; Yan, X.; Wang, B.; et al. 2024. Geox: Geometric problem solving through unified formalized vision-language pre-training. *arXiv preprint arXiv:2412.11863*.

Zhang, M.-L.; Yin, F.; Hao, Y.-H.; and Liu, C.-L. 2022. Plane geometry diagram parsing. *arXiv preprint arXiv:2205.09363*.

Zhang, M.-L.; Yin, F.; and Liu, C.-L. 2023. A multi-modal neural geometric solver with textual clauses parsed from diagram. *arXiv preprint arXiv:2302.11097*.

Zhang, X.; Zhu, N.; He, Y.; Zou, J.; Huang, Q.; Jin, X.; Guo, Y.; Mao, C.; Zhu, Z.; Yue, D.; Zhu, F.; Li, Y.; Wang, Y.; Huang, Y.; Wang, R.; Qin, C.; Zeng, Z.; Xie, S.; Luo, X.; and Leng, T. 2024. FormalGeo: An Extensible Formalized Framework for Olympiad Geometric Problem Solving. arXiv:2310.18021.

Zhang, Z.; Cheng, J.-K.; Deng, J.; Tian, L.; Ma, J.; Qin, Z.; Zhang, X.; Zhu, N.; and Leng, T. 2025. Diagram formalization enhanced multi-modal geometry problem solver. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Zhao, J.; Zhang, T.; Sun, J.; Tian, M.; and Huang, H. 2025. Pi-gps: Enhancing geometry problem solving by unleashing the power of diagrammatic information. *arXiv preprint arXiv:2503.05543*.

Appendix

A. Prompt Design

Problem-Solving Prompt Template with Geometric Information in English

Please strictly follow the requirements below to accurately solve the geometry problem:
* Task Requirements
** Step 1: Attempt to Reconstruct the Diagram
1. If the diagram can be directly reconstructed from the "problem text" alone, proceed directly to Step 2.
2. If the "problem text" alone is insufficient to fully reconstruct the diagram, use the "spatial logical relationships" and "pixel coordinates of the points" from the "diagram information" to determine the relative spatial relationships. Note that "pixel coordinates" should only be used as a reference for reconstructing the diagram and must **not** be used for solving or calculating.
3. If there is a conflict between the diagram information and the problem description, the "problem text" takes precedence.
** Step 2: Analyze and Solve Each Sub-question
For each sub-question in the problem:
1. Clearly identify the common conditions shared by the problem statement and those specific to this sub-question.
2. Determine whether conclusions from previous sub-questions can be used, but only if the conditions match.
3. If the diagram structure changes based on the problem's conditions, re-construct the diagram based on both the "diagram information" and the problem description.
4. Use the "diagram information" to perform geometric analysis, reasoning, and detailed calculations, and provide the final answer for the sub-question.
* Problem Text
[Problem Text]
* Diagram Information
** Diagram Analysis Result:
[Geometry Problem Formalization Module Result]
** Diagram Analysis Inferences
[Geometric Theorem Reasoning Result]
* Output Requirements
For each question, provide a clear and structured problem-solving process:
1. Clearly list the conditions used. If conclusions from earlier parts are referenced, explain why.
2. If necessary, explain how key information is extracted from the diagram.
3. Show key steps and necessary calculations, providing a clear answer.
4. If there are multiple sub-questions, answer them individually and summarize all answers at the end.

Diagram Information Template in English

**** Diagram Analysis Result:**
- Concyclic relationship (points strictly in counterclockwise order):
[Example: Points A, B, C are concyclic, lying on a circle with O as its center.]
- Collinear relationship (points strictly in left-to-right and top-to-bottom order):
[Example: Points G, F, D are collinear.]
- Perpendicular relationship:
[Example: Line AD is perpendicular to line DC.]
- Parallel relationship:
[Example: Line GH is parallel to line ED.]
- Angle relationship:
[Example: ∠C and ∠BCA are the same angle.]
- Pixel Coordinates of Points (for positional judgment only, not for numerical calculations)
[Example: A: x=32, y=29; B: x=147, y=30]
**** Diagram Analysis Inference:**
[Example:
- The inscribed angle theorem: ∠ACB = ∠AOB / 2.
- Alternate interior angles are equal: ∠CED = ∠ADE, ∠EDG = ∠DGH.]

Normal Problem-Solving Prompt in English

Please solve the following math problem. Provide a step-by-step explanation, including the problem analysis, followed by the solution steps, and finally the answer. Use the following output format: **Problem Analysis**, **Solution Steps**, **Answer**

Diagram Parsing Prompt in English

Please describe the geometric information in the following image, including spatial logical relationships: collinearity, concyclicity, and angle relationships. The format is as follows:
Spatial Logical Relationships:
- Concyclic relationship (points strictly in counterclockwise order):
[Example: Points A, B, C are concyclic, lying on a circle with O as its center.]
- Collinear relationship (points strictly in left-to-right and top-to-bottom order):
[Example: Points G, F, D are collinear; Points A, G, B are collinear; Points B, E, C are collinear.]
- Angle relationship:
[Example: ∠C and ∠BCA are the same angle.]

B. Supplementary Tables, Figures, and Case Study

We provide examples of our higher-order predicates, data contamination found in Gemini, and case study of our method as below.

Predicate	Description and Example
`PointsLieOnCircle(`P_1`, `P_2`, ..., `P_k`, Circle(O, r))`	All listed points lie on the same circle, given in clockwise order. Example: `PointsLieOnCircle(C, A, B, D, Circle(O, radius_0))`
`Collinear(`P_1`, `P_2`, ..., `P_k`)`	All listed points lie on the same straight line, arranged in left-to-right and top-to-bottom order according to the diagram. Example: `Collinear(A, O, B, F)`
`Shape(`P_1`, `P_2`, ..., `P_k`)`	Defines a polygonal figure (triangle, quadrilateral, etc.), with vertices specified in clockwise order. Example: `Shape(A, F, D)`
`Angle(`P_i`, `P_j`, `P_k`) = Angle(...)`	Encodes equalities or additive relations between angles. Example: `Angle(B,A,C) = Angle(B,A,D) + Angle(C,A,D)`

Table 4: Examples of higher-order predicates.

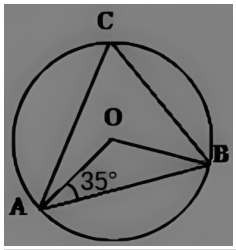
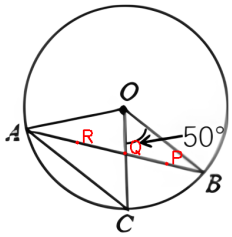
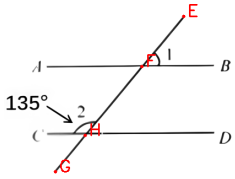
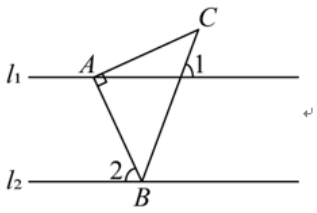
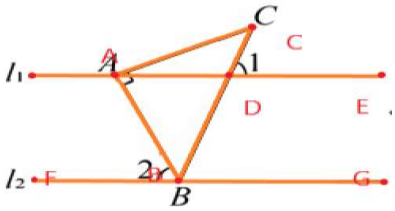
Image	Question	DP Information	Gemini-2.5-Pro Answer
	As shown in the figure, triangle ABC is the inscribed triangle of circle O, then the degree of angle ACB is...	..."Formulas": ["MeasureOf(Angle(B,O,A), 35°)", "Angle(B,A,C) = Angle(B,A,O) + Angle(C,A,O)"...	From the figure, we are given that the measure of angle OAB is 35°...
	As shown in the figure, chord AC is parallel to radius OB, then angle OAB is...	..."Formulas": ["MeasureOf(Angle(O,B,P), 50%)", "Angle(C,A,O) = Angle(O,A,P) + An...	From the figure, the central angle ∠BOC is given as 50°...
	As shown in the figure, AB is parallel to CD, then the degree of angle 1 is...	[No information about angles given or 135°]	The figure shows an angle of 135°...

Table 5: Suspected cases of data contamination on the Mathvista test dataset. Gemini-2.5-Pro corrects erroneous Diagram Parsing (DP) information, appearing to infer details from the original image despite **not being provided with it**. The model's corrections are highlighted in red.

13. 如图，已知$l_1 // l_2$，$\triangle ABC$是等腰直角三角形，$\angle BAC = 90°$，顶点A, B分别在l_1, l_2上，当$\angle 1 = 70°$时，$\angle 2 =$_____.





Geometry Problem: As shown in the figure, given $l_1 \parallel l_2$, $\triangle ABC$ is an isosceles right-angled triangle with $\angle BAC = 90°$. Vertices A and B are on lines l_1 and l_2 respectively. When $\angle 1 = 70°$, find the measure of $\angle 2$.

Image Parsing Result: ... `(Correct structured data)` ...

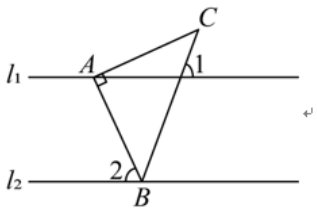
[... 22k+ tokens of reasoning omitted ...]

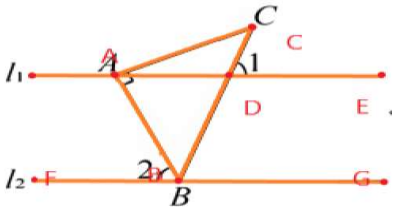
1. **Coordinate System Setup:** Let l_1 be $y = 0$ and l_2 be $y = -h$. Let $A = (0, 0)$ and $B = (b, -h)$. Then $C = (h, b)$.
2. **Equation of Line BC:** The slope of line BC is $\frac{b+h}{h-b}$. The intersection D of line BC with l_1 is $(\frac{h^2+b^2}{h+b}, 0)$.
3. **Analysis of $\angle 1 = 70°$:** The cosine of the angle between vector $\overrightarrow{DC}$ and the positive x-axis is calculated, yielding $\cos 70° = \frac{h-b}{\sqrt{2(h^2+b^2)}}$.
4. **Parameter Solving:** Let $t = h/b$. Solving the equation gives $t \approx 2.146$, so $h \approx 2.146b$.
5. **Calculation of $\angle 2$:** $\angle 2$ is the angle between AB and the horizontal line l_2. Its tangent is $\tan\theta = \frac{h}{b} \approx 2.146$, which corresponds to an angle of approx. $65°$.
6. **Geometric Relation Analysis:** The model reconsidered potential geometric shortcuts, distrusted its calculation, and guessed a different relation, concluding that $\angle 2 = 20°$.

Final Answer: $\boxed{20°}$ **(Incorrect)**

Figure 5: Case Study: Deepseek R1 without Our Method. Despite receiving perfectly parsed visual information, the model defaults to a complex coordinate geometry approach. It makes calculation errors or loses confidence in its complex derivation, ultimately resorting to a guess and providing an incorrect answer.

13. 如图，已知$l_1 // l_2$，$\triangle ABC$是等腰直角三角形，$\angle BAC = 90°$，顶点A, B分别在l_1, l_2上，当$\angle 1 = 70°$时，$\angle 2 =$_____.





Geometry Problem: As shown in the figure, given $l_1 \parallel l_2$, $\triangle ABC$ is an isosceles right-angled triangle with $\angle BAC = 90°$. Vertices A and B are on lines l_1 and l_2 respectively. When $\angle 1 = 70°$, find the measure of $\angle 2$.

Image Parsing Result: ... `(Correct structured data)` ...
Fact Deduction Result: $\angle 2 + \angle ABC + \angle CBG = 180°$...

Key Steps:

1. **Isosceles Right Triangle Properties:** Since $\triangle ABC$ is an isosceles right triangle with $\angle BAC = 90°$, we have $AB = AC$ and $\angle ABC = \angle ACB = 45°$.
2. **Parallel Lines Properties:** Let the intersection of the extension of BC with l_1 be D. The given $\angle 1 = 70°$ is the angle between line CD and line l_1. Since $l_1 \parallel l_2$, the corresponding angle at line l_2 is also $70°$.
3. **Angle Relationships at Vertex B:** Let's denote the angle formed by the line segment BC and the rightward direction of line l_2 as $\angle CBG$. Due to the parallel lines, $\angle CBG = \angle 1 = 70°$.
4. **Calculation:** The angles on the straight line l_2 around point B must sum to $180°$. We have the angle formed by segment AB and the leftward direction of l_2, which is $\angle 2$. The angles are adjacent on a straight line: $\angle 2 + \angle ABC + \angle CBG = 180°$ Substituting the known values: $\angle 2 + 45° + 70° = 180°$ Solving for $\angle 2$: $\angle 2 = 180° - 45° - 70° = 65°$

Final Answer: $\boxed{65°}$ **(Correct)**

Figure 6: Case Study: Our Method. Our method provides supplementary guidance, prompting the model to leverage deductive geometric reasoning. This guides the model to a simple and correct solution based on angle relationships, avoiding the pitfalls of the less-interpretable coordinate method.