参赛队员姓名: 陆知行

中学： 北京市第一0一中学

省份: 北京

国家/地区： 中国

指导教师姓名： 王威 、周宇辰

指导教师单位： 北京通用人工智能研究院

北京市第一0一中学

论文题目： Close the Loop of Neural Intuition and Logical Reasoning for Geometry Theorem Proving

融合语言模型与逻辑推理的几何定理证明

Close the Loop of Neural Intuition and Logical Reasoning for Geometry Theorem Proving

Zhixing Lu
Beijing No.101 High School
amy_lzx@outlook.com

Abstract

Automated geometry theorem proving is a challenging and critical task in the realm of artificial intelligence, demanding both auxiliary constructions and long chains of reasoning to deduce goals from a set of given facts in a geometry diagram. In geometry proofs, auxiliary constructions act as a bridge to fulfill theorem conditions, requiring both an effective geometry intuition and an efficient search strategy. Geometry reasoning aims to plan and search a long path from initial facts to final goals; the intermediate results are deduced by chaining a series of theorems. In this project, we propose a closed loop of neural intuition and logical reasoning to address auxiliary construction and reasoning in geometry theorem proving. Inspired by the human approach to mastering mathematical skills through practice, we derive neural geometry intuition, i.e., the skill to swiftly generate auxiliary candidates simply by viewing the text and diagrams, by learning from an extensive number of geometry proofs. Specifically, we explore the neural geometry intuition capabilities of large pre-trained language models, such as GPT-4, and propose a novel "Mixture-of-Thought" (MoT) prompting and search strategy for auxiliary construction. For *geometry reasoning, we resort to traditional knowledge-based artificial intelligence and exploit a forward chaining algorithm for logical reasoning. Analogous to human trial-and-error problem-solving approaches, we continuously iterate between auxiliary construction and logical reasoning until the final goal is proven. The effectiveness of our method is verified with a 66% accuracy performance on a complex geometry-proof dataset. This demonstrates a strong neural intuition and improved precision and generalization of automated geometric proofs.*

Keywords

Geometry theorem proving, auxiliary construction, neural intuition, logical reasoning, large language model

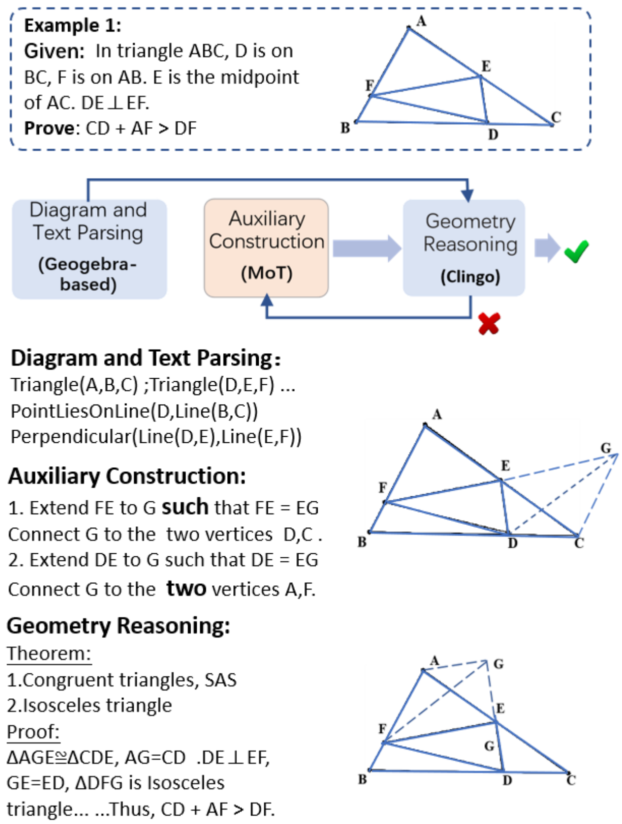


Figure 1. Example demonstrating our closed-loop system. (1) Prompt conversion into formal language (2) Auxiliary Construction with neural intuition (3) Theorem-guided logical reasoning (4) Closing the loop by reconstructing auxiliary lines if the goal is not derived.

1. Introduction

Given a diagram and textual description detailing the relationships and measurements of geometry elements, geometry theorem proving aims to derive reasoning paths towards a final goal using Euclidean axioms. Geometry proof is an essential topic in mathematics education, nurturing students' abilities in abstract and logical thinking. For many students, geometry can be a tough course due to

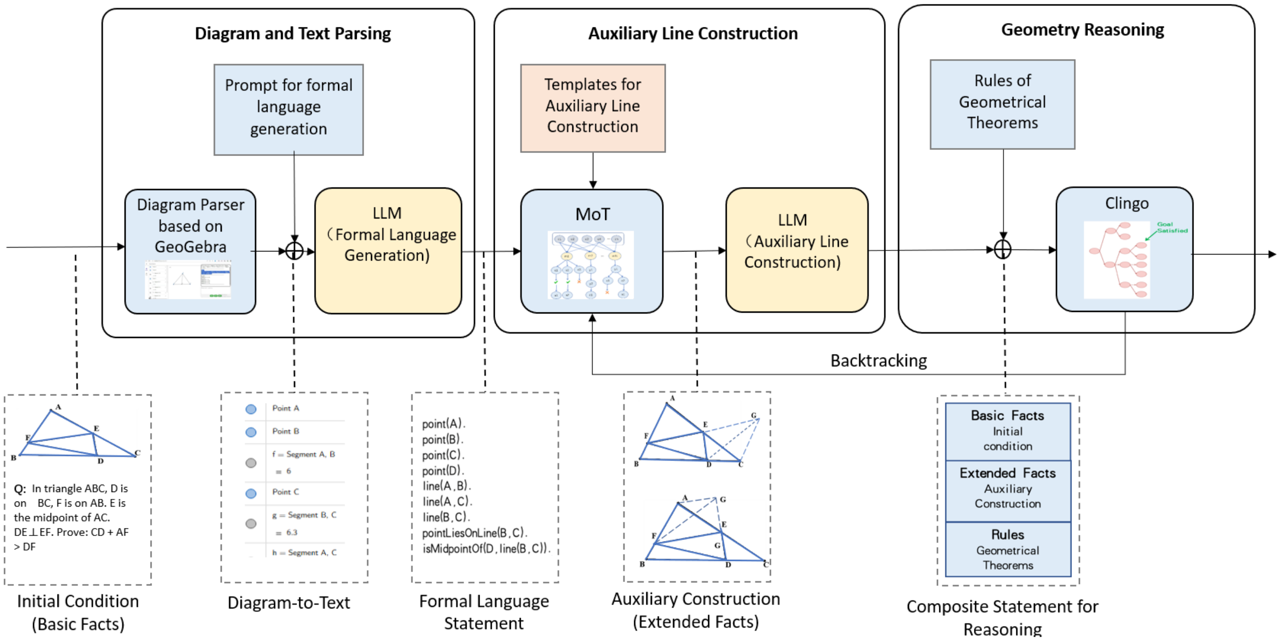


Figure 2: The closed-loop geometry theorem proving process with neural intuition and logical reasoning.

the obstacle of auxiliary construction. It requires the association of the problem with axioms and the search for missing points or lines within a large geometry space. An example of geometry theorem proving is shown in Figure 1.

Since the birth of artificial intelligence, Automated Theorem Proving (ATP) — in particular, geometry theorem proving — has been the focus in the field. Existing methods include the deduction approach [1], algebraic computation [2], and the area method [3]. The deduction approach cannot handle problems that require auxiliaries, and the algebraic computation and area method both demand mathematical expertise, making them difficult for non-experts to master.

Recently, large pre-trained language models (LLMs) have been introduced to reasoning tasks and seem to have achieved some success [4]. However, when conducting comprehensive experiments using LLMs for geometry reasoning, we discover that they often fail to solve geometry problems on their own. For example, in Figure 1, when presented with the proof goal: CD + AF > DF, although LLMs intuitively oriented a proof towards the Triangle Inequality Theorem, their auxiliary constructions and proof are both incorrect. This indicates that LLMs lack the ability of long-term complex reasoning, limiting their performance on formal proof tasks. Yet, their ability to identify relevant theorems (e.g., Triangle Inequality Theorem), suggests that they have the potential in neural intuition, i.e., the skill to swiftly generate auxiliary candidates simply by viewing the text and diagrams.

In this project, we propose a closed loop of neural intuition and logical reasoning to address auxiliary construction and reasoning in geometry theorem proving. Inspired by the human approach to mastering mathematical skills through practice, we derive neural geometry intuition, i.e., the skill to swiftly generate auxiliary candidates simply by viewing the text and diagrams, by learning from an extensive number of geometry proofs. Specifically, we explore the neural geometry intuition capabilities of Large pre-trained Language Models, such as GPT-4, and propose a novel “Mixture-of-Thought” (MoT) prompting and search strategy for auxiliary construction. For geometry reasoning, we resort to traditional knowledge-based artificial intelligence and exploit a forward chaining algorithm for logical reasoning. Analogous to human trial-and-error problem-solving approaches, we continuously iterate between auxiliary construction and logical reasoning until the final goal is proven.

The procedure of our method is explained in Figure 2. Given initial problem statements, our model first parses the diagram to text using GeoGebra and then transforms the text into formal language through LLMs. Subsequently, we employ our proposed MoT prompting strategy in LLMs to search for auxiliaries, also dubbed “neural geometry intuition” — since LLMs are deep neural models trained on large amounts of language data. After acquiring the auxiliary lines, we use a forward chaining tool Clingo to perform reasoning based on basic facts, extended facts (i.e., auxiliary lines), geometry axioms, and basic rules. If the goal is among the deduced results, we have successfully proved the theorem. Otherwise, we go back to search for alternative auxiliaries. The intermediate results of each step are also shown in Figure 1.

The effectiveness of our method is verified with a 66% accuracy performance on a complex geometry-proof dataset. This demonstrates that our model has a strong neural intuition and has widely improved the precision and generalization of automated geometric proofs.

Our **main contributions** are summarized as follows:

- We introduce a closed-loop method for automated geometry theorem proving, joining the neural geometry intuition of modern large language models with typical knowledge-based logical reasoning. This type of neuro-symbolic integration work is rare at the moment.
- We propose a MoT prompting and search strategy that effectively guides LLMs in auxiliary construction. Our MoT provides an example of how to enhance large language models' intuition in scientific domains, such as math and physics.
- Our method has achieved a much higher accuracy in geometry problem-solving compared to other methods.

The rest of the paper is organized as follows. In Section 2, we introduce closely related works to our method. In Section 3, we detail the main components of our method. In Section 4, we present the experimental results. Finally, in Section 5, we conclude the paper.

2. Related Works

In this section, we briefly review the existing literature that closely relates to our method.

Automated Geometry Theorem Proving. There are mainly three types of approaches to automated geometry theorem proving: the deduction method[1], the algebraic computation method (mainly Wu’s method and the Gröbner basis method) [2,5], and the geometric invariant method, like the area method [3]. Geometry theorem proving has been a challenging problem in ATP, which generally needs auxiliary construction to accomplish the proof. Due to the large search space, auxiliary construction is very difficult for the existing reasoning system. Most studies in geometry theorem proving focus on problem parsing and symbolic reasoning [6~8], with less attention devoted to auxiliary constructions.

Auxiliary Construction. There are mainly two categories of approaches that address auxiliary constructions. Earlier approaches focus on the exhaustive search method. Zhou et al.[9] generate all possible auxiliary lines for special points and line segments in the input graphics. They then use binary search to select the useful auxiliary lines. This method optimizes the efficiency to reduce the search space, but still essentially is an exhaustive approach, lacking skill in the actual auxiliary construction and performing extensive deduction with newly given conditions. Matsuda et al. [10] propose a different and more goal-oriented optimization mechanism, constructing auxiliary lines to validate a necessary postulate that is inapplicable to the original set of conditions. This kind of approach, requiring exhaustive searching and deduction, is highly complex and inefficient.

Another type of study uses templates to generate auxiliary lines. K. Wang et al. [11] summarize 6 templates, each adapted to solve a particular class of geometry problems. This provides more guidance for auxiliary construction and is more efficient than an exhaustive search. However, these templates are insufficient to cover all common geometric problems, and selection among these templates is generally complex and inaccurate when figures become more complicated. Additionally, Matsuda et al. [10] do not elaborate on how the designated template is applied to each problem. Other efforts such as Mai C et al. [12], design a structure-based geometry knowledge representation method, and use machine learning to devise auxiliary lines. Up to now, the fully automated process for auxiliary construction has not yet been achieved.

Large pre-trained Language Models (LLMs). Existing efforts do not realize human-like thinking of auxiliary constructions that involve neural intuition. Recent advances in LLMs display human-like thinking, raising the question: can these models have the potential for neural intuition to achieve automated geometric theorem proving?

There are two views on LLMs' capabilities in ATP. Supporters believe that mathematicians can use LLMs to aid them with intuition on complex mathematical objects. S. Polu et al. [13] propose GPT-f, which discovers new, short proofs that are accepted into the main Metamath library. A. Davies et al. [14] describe how DeepMind researchers help mathematicians to prove two new theorems. The opposite view believes that LLMs lack logical reasoning abilities. S. Frieder et al. [15] suggests that, contrary to many of the positive reports in the media, ChatGPT fails to achieve the performance of single-task mathematical training models. OpenAI summarizes GPT-4's mathematical capabilities [16] in that it can abstractly express mathematical problems but makes frequent mistakes in basic computations and reasoning despite appearing to understand mathematical prompts. The latest research indicates that the key to leveraging LLMs' abilities lies in the design of an effective "Chain-of-Thought", guiding the model to think and respond in the desired manner. This is demonstrated by the Chain-of-Thought (CoT) [17] and Tree-of-Thought (ToT) [18,19] prompting methods, which both significantly enhance the reasoning capabilities of the LLMs. Designing the right prompts can yield much better results. In this project, we believe it is possible to take advantage of LLMs with well-designed strategies to achieve neural geometry intuition.

3. Our Method

Our method addresses complex geometry problems that require auxiliary constructions. As shown in Figure 2, we propose a closed-loop method to emulate human cognition to boost the accuracy and generalization of proofs. There are three key functions in this method.

Diagram and Text Parsing. We parse diagrams and convert geometric figures into language descriptions using our parsing tool based on GeoGebra, and then transform these descriptions into formal definitions of points, lines, and edges using GPT-4 [4]. Such formalization of spatial relationships will benefit for later auxiliary construction and logical reasoning.

Auxiliary Construction. We explore the capability of LLMs trained via a huge language dataset to generate neural intuition to construct auxiliaries. We draw inspiration from humans, who solve mathematical problems through repeated practice. We hope that LLMs also exhibit this kind of ability by digesting big data. We propose a novel technology called MoT to generate neural intuition in the LLMs.

Geometry Reasoning. We formulate geometry reasoning as a forward chaining process and utilize Clingo [20] to verify the auxiliary proposals in the previous stage. If the goal (i.e., geometry theorem) is reached, the proof is completed. If the proof does not derive the goal, our method will turn back to searching for alternative auxiliaries until the proof succeeds.

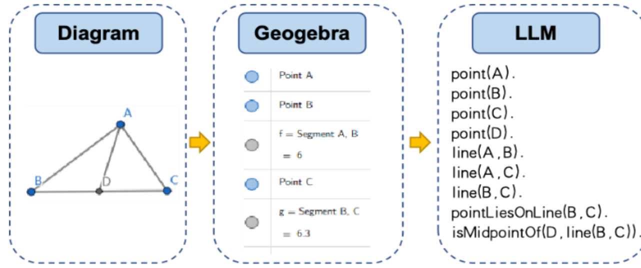


Figure 3: Diagram and Text Parsing. Parsing tool based on GeoGebra first extracts geometry elements from the user's diagram to language descriptions, and then LLMs convert such descriptions into formal language in the Prolog format.

3.1 Diagram and Text Parsing

Understanding the problem statement is the first step to solving problems. In geometry proofs, humans start by sketching the diagram. Such visualization gives them a holistic grasp of the problem and assists them in associating the conditions in the textual description. This depicts the diagram and text parsing process in our method.

It should be noted that directly using LLMs such as GPT-4 cannot achieve such parsing due to its lack of capability of spatial perception. LLMs often make mistakes even in the most basic relationships between vertices, line segments, and angles. A failure case is as follows: when reading a prompt that contains the condition BD=DF, LLMs would perceive that B, D, and F are collinear, and D is the midpoint of BF, which is not necessarily the case.

To address these types of challenges, we develop a Geogebra-based parsing tool to first identify the relative positions of points and lines. This endeavor involves parsing diagrams and facilitates later interaction with LLMs, boosting spatial perception and accuracy in problem comprehension. As depicted in Figure 3, when the user draws the geometry diagram d, the parsing tool based on Geogebra automatically generates textual paradigm description l that characterizes the geometry elements and relationships in d.

$$l \leftarrow Geo(d)$$

Then, we utilize LLMs to transform l into s, which is the prompt diagram in formal language.

$$s \leftarrow LLMs(l)$$

This method incorporates the language generative capability of LLMs, ensuring the accurate representation of key geometric information and providing a reliable basis for auxiliary construction and geometry reasoning.

3.2 Neural Geometry Intuition for Auxiliary Construction

Once the problem statement is parsed into formal language, deriving neural intuition becomes the key to constructing auxiliary lines. Human thinking is characterized by intuition. Practice allows humans to recognize geometry components in new problems that are analogous to those they've encountered in past problems. This helps them establish an intuition for the direction of their proof. The more the practice, the better the intuition. Thus, the two main aspects of the educational value of Euclidean geometry are the ability to form neural intuition and reason logically.

Although LLMs are trained with vast amounts of linguistic data that likely includes a lot of materials on reasoning and theorem proving, they still do not possess accurate intuition in geometry problem-solving, as shown in Figure 4, To address this challenge, we propose a novel prompting and search strategy, i.e., MoT, to guide LLMs in deriving effective auxiliaries.

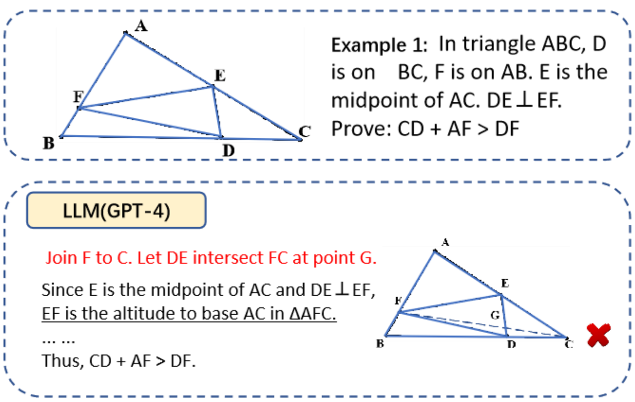


Figure 4: The LLM's performance on Example 1. Its auxiliary construction and proof are both incorrect.

3.2.1 Mixture-of-Thought

MoT comprises three primary components: tactic design, tactic selection, and tactic mapping. Tactic design involves deriving auxiliary patterns/templates from problem samples. Tactic selection involves LLMs' choice of a template proposal by leveraging linguistic intuition. Tactic mapping involves LLMs' decision of how to utilize these abstract templates for specific problems.

Tactic Design. LLMs lack fixed strategies and experience in auxiliary construction. Therefore, we need to induce strategies – a set of stored patterns - from samples to guide LLMs. The design of these strategies has two challenges: proposing auxiliary patterns and expressing these patterns in a clear representation (for LLMs to understand) that captures the relationship between geometry elements for LLMs to understand.

We designed auxiliary patterns by observing 40 geometry problems, identifying the similarities in their auxiliaries, and summarizing them into basic figures. These basic figures are symbolic abstractions that serve as a bridge between the conditions in the prompt and the final goal we ought to prove. To systematically organize these abstractions, we classified them according to their geometry features. Figure 5 describes four basic structures that involve midpoints.

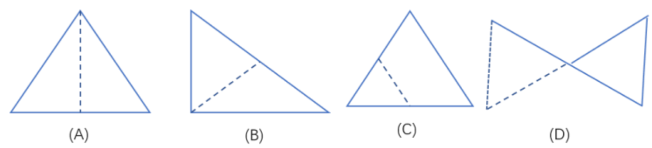


Figure 5: Four templates about midpoints. (A) Median on the base of Isosceles Triangles; (B) Median on the hypotenuse of Right Triangles; (C) Mid-segment of a Triangle; (D) Double the length a Median

We use template (D) as an example to illustrate the effective description and application of these templates. To identify figures in problems that are structurally similar to this template, we first need to clearly represent the template. Through extensive experiments, we propose to use conditions that specify the relationships of key vertices in the template structure. The detailed formal language representation of template (D) is outlined in 4 steps, as follows.

Template (D) Double the length of a Median

Goal: Construct congruent triangles to establish the relations of segments in different triangles

Description:
Point(Q).
Point(X).
Point(Y).
Point(Z).
#Condition 1: point Q lies on line segment XY
pointLiesOnLine(point(Q), line(X,Y)).
#Condition 2: point Q is the midpoint of segment XY
isMidpointOf(point(Q), line(X,Y)).
#Condition 3: point Z is not on line XY
not(pointLiesOnLine(point(Z), line(X,Y))
#Condition 4: there exists line segment ZQ or QZ in the prompt
line(Z,Q).

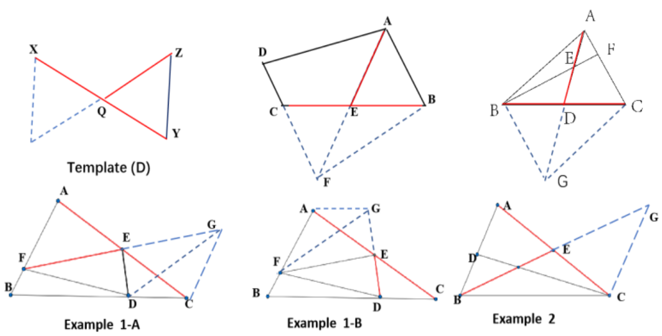


Figure 6. Template (D) and its usage on four problems. There are two possible matches of Template (D) in Example 1

Then, we guide LLMs to search for the structure in Template (D) in problems. Figure 7 shows four examples that can use Template (D). It should be noted that, when searching for template (D) in the same problem, there sometimes exist multiple sets of auxiliary lines, as shown in Figure 6 Example 1-A and 1-B.

After introducing the design of the templates, we will introduce how they are applied to problems by the MoT. The procedural overview of the MoT could be found in Figure 7.

Tactic selection. As shown in Figure 7, the input of MoT can be represented as a tuple (t, s, O), where t is the problem in natural language, s is the formal language representation of the diagram, and O is the option set $O = \{o_{1,} o_{2,} \dots o_{n,}\}$ for auxiliary line features, where each option can be represented by a set of geometry elements $o_i = \{e_{i1,} e_{i2,} \dots e_{in}\}$. The tuple (t, s, O) is then composed as a language sequence and sent to LLMs, where each x[i] is a token, so that

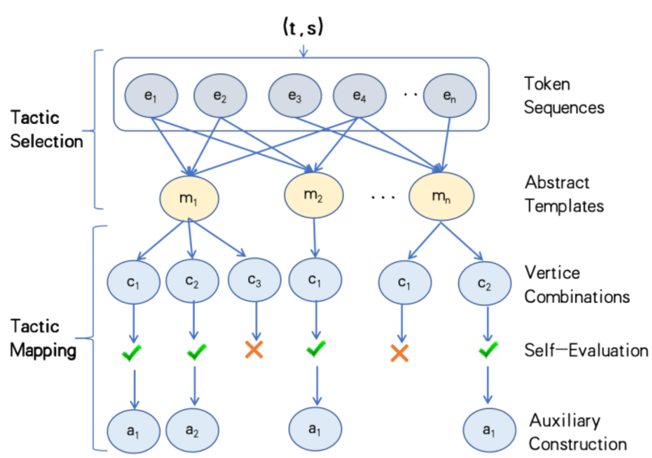


Figure 7. Demonstration of the MoT. Step 1: Tactic Selection. Step 2: Tactic Application

$$p_\theta(x) = \prod_{i=1}^{T} p_\theta(x[i] \mid x[1 \dots i-1])$$

where p_θ denotes a pre-trained LLM with parameters representing the parameters of LLMs[15]. Then LLMs select templates from the template library

$$M = (m_{1,} m_2 \dots m_n)$$

and sort them:

$$rank(m_i) \leftarrow p_\theta(m_i \mid prompt(x))$$

here, $prompt(x)$ wraps input x with task instructions.

This process of template selection is a heuristic for our search algorithm. We let LLMs to decided which and in what order templates are applied due to their flexibility and linguistic comprehension. LLMs' decisions are made with a holistic understanding of the prompt due to their language characteristics. Among templates that are applicable the prompt, they can prioritize those that matches the best. In Example 3 shown in Figure 8, the problem statement contains the geometry features: triangle, right angle, and midpoint. LLMs were able to prioritize the template "Median on the hypotenuse" (involving all three features) over "Double the length crossing a Median" (involving triangle and midpoint).

We observe that LLMs judgments are highly accurate, most of the time claiming the correct template as its first choice and occasionally ranking it as its second and third choice.

Tactic Mapping. After selecting templates, the challenge becomes to apply these abstract templates to specific problems. For simple geometry figures, we can quickly

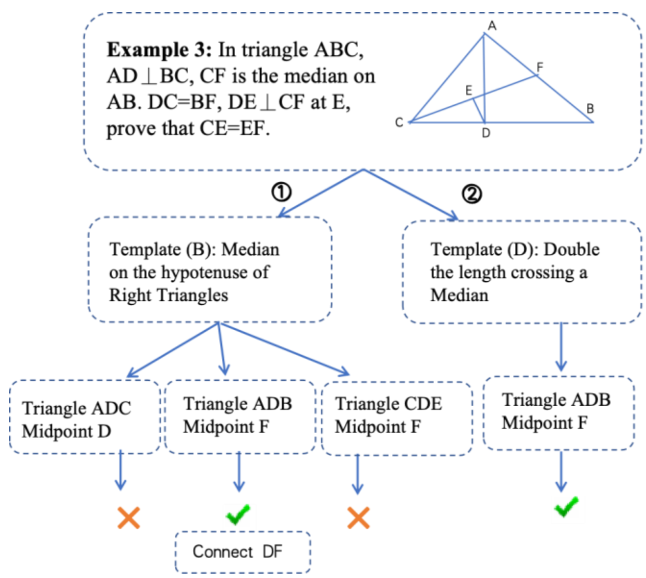


Figure 8. Demonstration of MoT on Example 3.

identify the key geometry elements for the basis of auxiliary constructions. However, for complex geometric figures containing multiple basic shapes, effectively mapping template to the problem is more challenging. These two types of problems vary by difficulty and therefore have different levels of requirements. For simple figures, we can use Chain-of-Thought (CoT) [18] to construct a solution path. While for complex figures, we need to search the problem's solution space using the Tree-of-Thought (ToT) strategy [19]. MoT combines the advantages of both CoT and ToT by generating a flexible searching structure, either chain or tree.

The selected templates are traversed from highest to lowest rank. As shown in Figure 7, the application of each template can be accomplished in three steps: vertices mapping, self-evaluation, and auxiliary construction. For each template m_i, the new input to LLMs is the language sequence x, composed by the tuple (t, s, m_i).

Step 1: Vertices Mapping. Given the input x, LLMs search for a set of vertices combinations $C = \{c_1, c_2, \dots, c_j\}$ in the problem (described by t and s) that map with the template.

$$\{c_1, c_2, \dots, c_j\} \leftarrow p_\theta(\{c_1, c_2, \dots, c_j\} \mid prompt(x))$$

each c_i is a vertices combination, and j is the number of combinations generated from the given template.

As shown in Figure 8, LLMs detected several combinations for Template (B). c_1: Triangle ADC and Midpoint D, and c_2: Triangle ADB and Midpoint F. However, it is evident that some of the combinations are incorrect.

Step 2: Self-Evaluation. To boost efficiency and rule out incorrect combinations, we included the self-evaluation step for LLMs to verify their results.

$$C' \leftarrow p_\theta(C' \mid C)$$

the revised set C' contains only the correct combinations. This step is another heuristic for our MoT search algorithm, keeping only the combinations that satisfy all conditions in the template. Heuristics are typically either programmed (e.g. DeepBlue) or learned (e.g. AlphaGo) [21]. However, LLMs can evaluate states and are especially adept at giving feedback, possessing higher flexibility than the typical means [21]. As shown in Figure 8, the mistakes in the vertices mapping are detected via self-evaluation.

Step 3: Auxiliary Construction. After generation and evaluation, only the correct vertices combinations C' remain. For each combination, an operation on the relationship of the vertices is conducted to construct auxiliary lines $A = \{a_1, a_2, \dots, a_n\}$

$$\{a_1, a_2, \dots, a_n\} \leftarrow p_\theta(\{a_1, a_2, \dots, a_n\} \mid prompt(C'))$$

Each a_i is the auxiliary constructed from combination c'_i, and n is the number of auxiliaries. This step is a strict deduction from the combinations and will not lead to any errors. As shown in Figure 8, for template (B), the auxiliary is directly constructed by connecting two vertices in the combination.

The whole procedure of MoT is described above, and it can be seen that unlike typical inductive methods that start from problem conditions and branch out to all possible outcomes, our MoT prompting is goal oriented. The way MoT characterizes and applies templates engenders LLMs' neural intuition.

3.3 Forward Chaining for Geometry Reasoning

After constructing auxiliaries, since LLMs fail to perform long-term complex logical reasoning, we adopt a forward chaining algorithm for geometry reasoning. Forward chaining starts from known geometric conditions (axioms, theorems, constructions, etc.) and derives new intermediate results and theorems. This process is repeated until the goal is reached. However, it tends to produce an extensive number of redundant intermediate results and steps, requiring effective control over the unfolding process and mechanisms to judge results to obtain the simplest proof. It should be pointed out that, although backward chaining - starting from the ultimate goal and backtracking to the initial conditions - can also be a choice for logical reasoning, we find that this is very time-consuming for existing backward-chaining tools, e.g., Prolog. We hypothesize that geometry theorem proving may generate a vast set of subgoals that fail Prolog to backtrack. Therefore, we adopt Clingo for forward reasoning.

Clingo is an Answer Set Programming (ASP) system. Answer Set Programming (ASP) is a form of declarative programming that performs well in knowledge representation and reasoning, especially with combinatorial search problems. Combinatorial search problems involve searching through extensive possible solutions to find those that satisfy specific criteria.

Our case of geometry theorem proving is a combinatorial search problem. Clingo generates new results from the basic facts, the extended facts from auxiliary lines, geometry axioms, and some basic rules. If our goal is among these generated results, then we have successfully proved the problem. The reasoning process is demonstrated in Figure 9.

Basic Facts and Extended Facts. All the facts for a geometry proof are separated into basic and extended facts. The basic facts are the initial problem statements in formal language, as we have introduced in Section 3.1. The

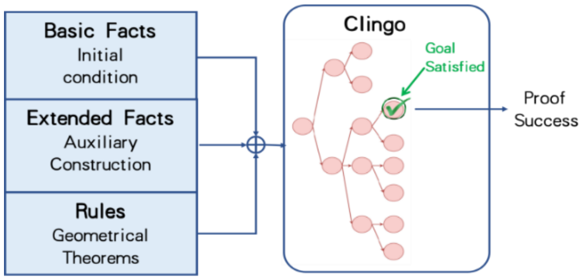


Figure 9. The End-to-End Flow of ASP System Clingo. It puts facts, rules, and the quest into grounding and conducts forward-chaining based on the ground program.

extended facts are from the constructed auxiliaries, also represented in formal language, generated by LLMs introduced in Section 3.2. The facts contain all the geometry elements – points, line segments, and angles – and geometry relationships – points on lines, midpoints, identical lines – in the prompt. However, as mentioned before, not all problems are solvable solely with basic facts, some require auxiliary constructions), which are generated by LLMs. These facts are presented as extended facts. We illustrate Example 1's basic facts and extended facts of Example 1 in Table 2 and Table 3, respectively.

Table 2. Excerpt of Basic Facts for Example 1.

Basic Facts for Example 1
point(a). point(b). line(a,b). line(b,c). ... pointLiesOnLine(d,line(b,c)). pointLiesOnLine(e,line(a,c)). isMidpointOf(point(e),line(a,c)). pointLiesOnLine(f,line(a,b)). measureOf(angle(f,e,d)) == 90. ...

Table 3. Excerpt of Extended Facts for Example 1.

Extended Facts for Example 1
extendLine(line(d,e),line(d,g)). equals(lengthOf(line(d,e)),lengthOf(line(e,g))). line(g,a). line(g,f).

Proof Goal. Here we give an example of the proof goal for Example 1 in the Prolog format in Table 4.

Geometry Axioms. We collect the majority of geometry axioms about triangles, parallel lines, and other related geometry elements in the Prolog format. Several axiom examples are shown in Table 5, e.g., Side Angle Side Theorem (SAS) for congruent triangles.

Table 4. Proof Goal for Example 1.

Proof Goal for Example 1
greater(sumOf(lengthOf(line(c,d)),lengthOf(line(a,f))),lengthOf(line(d,f)).

Table 5. Examples of Geometry Axioms.

Example of Geometry Axioms
%Congruent Triangles %Side Angle Side Theorem (SAS): Two triangles are congruent if they have two sides and their included angle equal. congruent(triangle(A,B,C),triangle(D,E,F)):- triangle(A,B,C),triangle(D,E,F),equals(lengthOf(line(A,B)),length Of(line(D,E))),equals(lengthOf(line(B,C)),lengthOf(line(E,F))),eq uals(measureOf(angle(A,B,C)),measureOf(angle(D,E,F))), not same_triangle(triangle(A,B,C),triangle(D,E,F)). ....

Basic Rules. There exists symmetry, permutation invariant properties in the majority of geometry elements. For example, line(a,b) and line(b,a) are equivalent, the triangles with any order of point a, point b and point c are the same triangle, point a lying on line(b,c) is equivalent to point a lying on line(c,b), etc. We construct a set of basic rules to bridge the facts and geometry axioms. Several examples are shown in Table 6.

Table 6. Examples of Basic rules to Bridge Facts and Geometry Axioms.

Example of Rules
line(A, B) :- line(B, A). pointLiesOnLine(A, line(B,C)) :- pointLiesOnLine(A, line(C,B)). ... isMidpointOf(point(D),line(B,C)) :- isMidpointOf(point(D),line(C,B)). equals(lengthOf(line(B,D)),lengthOf(line(D,C))) :- isMidpointOf(point(D),line(B,C)). ...

Clingo for Geometry Reasoning. Given these facts, geometry axioms, and basic rules, Clingo starts its reasoning which consists of two main phases: grounding and solving.

In the grounding phase, the input is transformed into a ground (variable-free) representation. All variables are replaced by the constants in the facts. In our example, the variables in the rules and axioms are replaced with the vertices and line segments in the facts. The solver then reasons based on the propositional statements and searches for answer sets (solutions) in the ground program. Modern ASP solvers rely upon advanced conflict-driven search procedures, pioneered in the area of satisfiability testing (SAT). In our example, if the proof goal appears in the ground program, we have proved the theorem and finished the task.

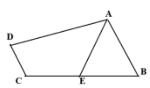
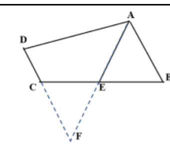
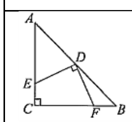
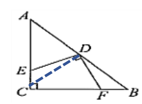
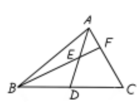
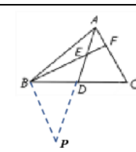
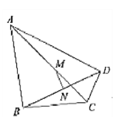
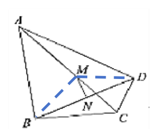
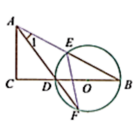
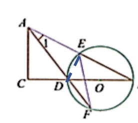
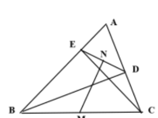
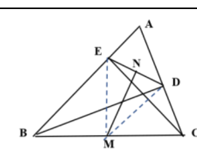
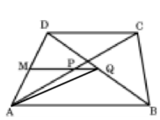
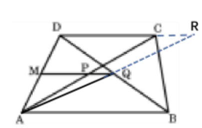
4. Experiments

First, we analyze existing datasets and introduce our task-specific dataset. Then, we evaluate the effectiveness of both our core MoT prompting and our whole method with quantitative and qualitative experiments. The results have proven that our design enlightens the potential of LLMs, simulates the human cognitive process, and improves the effectiveness of automated geometry theorem proving.

4.1 Datasets

Most of the existing research on geometry problem solving involves relatively few proof problems, and even fewer problems requiring auxiliary constructions. For example, the GEOS dataset [22] contains 186 SAT problems, all of which are simple problems without auxiliary lines. The GEOQA dataset [23] contains 4,998 geometric problems but are all multiple-choice and mostly simple computational problems.

Table 7. Examples from Question Dataset

Diagram	Problem	Type	level	Auxiliary line
	In quadrilateral ABCD, E is the midpoint of BC , AB is parallel to CD, If AE bisects angle BAD, Prove: AD = AB + DC.	Quadrilateral	2	
	Triangle ABC, C is a right angle,AC = BC, D is the midpoint of AB. DE⊥DF, points E, F are on AC, BC. Prove: DE = DF	Triangle	2	
	In triangle ABC, AD is the median on BC. E is a point on AD such that BE=AC.	Triangle	3	
	In quadrilateral ABCD, ∠ABC = ∠ADC = 90. M and N are the midpoints of AC and BD. Prove: BM = DM.	Quadrilateral	4	
	In △ABC, ∠C = 90°, D is a point on side BC, circle O with diameter DB passes through the midpoint E of AB, intersects the extension of AD at F, connect EF. Prove:∠1 = ∠F.	Circle	4	
	In triangle ABC, D is on AC such that BD⊥AC at D, E is on AB such that CE⊥AB at E. M and N are the midpoints of BC and DE. Prove: MN⊥DE	Triangle	4	
	In trapezoid ABCD, CD//AB, DQ=BQ, AP=CP, extend QP to intersect AD at M. Prove AM=DM.	Quadrilateral	5	

Matsuda et al. [10] collected 23 geometry-proof problems requiring auxiliary construction, and Wang et al. [11] collected 77 geometry-proof problems, but these datasets and proof procedures are not publicly released. There are other unpublished datasets as well [6,24].

Therefore, we carefully built a new geometry-proof problem test set containing 40 problems, of which 35 require auxiliary lines. These data are collected from various sources, with Chinese geometry exercise books [25] being the main source. Others are from related studies such as [10] [12]. We manually collected the prompt, diagram, and correct proof for each problem and classified them according to difficulty and geometry figures. The difficulty criterion comes primarily from the annotations in the exercise books. Some examples from our dataset are shown in Table 7.

On one hand, we use these problems to test and analyze the logical reasoning abilities and vulnerabilities of LLMs. On the other hand, we analyze patterns of auxiliary constructions through the answers to these problems. For a typical example, we summarized four midpoint-related templates and selected 21 questions with midpoints from the dataset to evaluate our method, using success rate as the evaluation metrics.

4.2 Evaluation Methods

We conducted 2 types of experiments: a comparison experiment focusing on the performance of our MoT-centered mechanism, and a step-by-step analysis holistically analyzing the effectiveness of each step in our approach. Regarding to comparison with other existing methods, some of the methods mentioned in related works either do not reveal their code, such as IGeoTutor[11] and GRAMY[10], or do not address geometry proof problems with auxiliary constructions, such as GeoQA[23]. Therefore, we compare our method with the reproducible prompting methods in the comparison experiments.

The comparison experiment assesses the MoT-centered mechanism by comparing it with existing prompting methods. The capability of LLMs largely depends on the prompt, which requires extensive experiments to optimize. Their performances can be improved by simply providing a few examples (few-shot), describing the task (zero-shot) [26], and in more advanced cases, guiding step-by-step (CoT) [17]. Therefore, in this experiment, we used the GPT-4 to evaluate its performance with our MoT centered mechanism, compared to three other common prompting methods, zero-shot, zero-shot-CoT, few-shot-CoT.

The second experiment involves a step-by-step analysis tracking every part of our complicated method, which is divided into eight tasks. The purpose and intermediate result of each task is assessed.

Table 8. Performance of Our Method Compared to Other Prompting Methods

Method	No. of Successful Case	Success Rate
Zero-Shot	0	0%
Zero-Shot-CoT [26]	2	9%
Few-Shot-CoT [17]	8	38%
Our Method	14	66%

4.3 Experiment Results

Comparison Experiment. We take Example 1 as the typical example to illustrate the difference among results of zero-shot, zero-shot-CoT, few-shot-CoT, and our methods in Figure 10. MoT was able to construct two sets of correct auxiliaries while all other three prompting methods failed.

However, the other methods failed to different extents, and it is worth mentioning how the increasing specificity of the prompting has led to improvement in the auxiliary constructions. The Zero-Shot construction failed to focus on the key information in the prompt: E is the midpoint of AC. However, it shows a hint of intuition; GPT-4 was able to connect FC to form a triangle CDF since CD and DF are in the goal, suggesting that its performance would improve if its intuition is guided on the right track. The Zero-Shot-CoT construction was able to notice the key information and constructed an auxiliary related to midpoint E. Its error is due to a lack of patterns. The Few-shot-CoT provided such a pattern in the form of an example, which led to a partially correct construction: Extending DE to G. However, its lack of precision and generalization caused the error of the two other auxiliary lines. Finally, our MoT, with patterns that acquire precision from the clear-cut conditions and flexibility from the abstraction of vertices, was able to correctly construct two sets of auxiliaries, each containing three lines.

The same procedure shown in Figure 10 is applied to prove 21 selected questions with midpoints using our 4 midpoint-related templates. The experiment result is shown in Table 8 with success rate as the evaluation metric. Our method achieved a significantly higher success rate than all other prompting means, suggesting that MoT-based neural intuition plays a key role in improving the success rate of auxiliary construction. Most of the failure cases of our method can be attributed to two types of errors: template matching and auxiliary construction. Template matching errors occur mainly when there are many misleading conditions. In such case, multiple templates are applicable, and LLMs might have trouble in prioritization. Auxiliary construction errors occur when the right template is selected but there are too many related vertices in the prompt that LLMs are baffled. Example of both errors are shown in Appendix B.

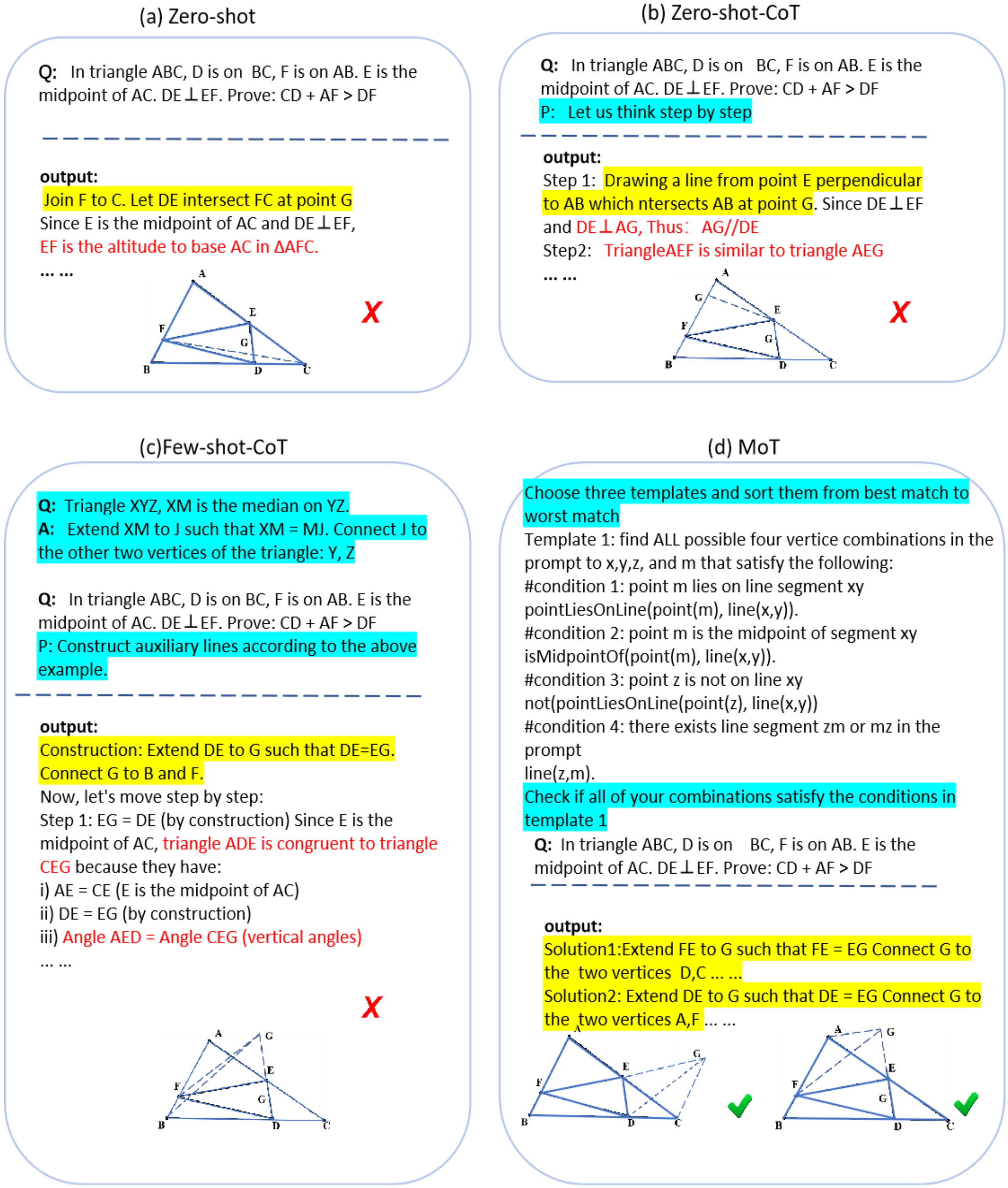


Figure 10. Comparison of MoT's performance with existing prompting methods demonstrated with Example 1. Blue: Prompting. Yellow: LLMs' auxiliary constructions. Red: Errors. (a) Zero-shot. Both the auxiliary construction and proof are incorrect. (b) Zero-shot-CoT. Both the auxiliary construction and proof are incorrect. (c) Few-shot-CoT. The auxiliary construction is partially correct: extending DE to G. The connections from the new point to other vertices are wrong. (d) MoT. Two sets of auxiliary constructions, each set consisting of a correct extension and two correct connections.

Step-by-Step Analysis. We validate the effectiveness of our method via step-to-step tracking with intermediate results checked. In figure 11, we take Example 1 as the typical example to demonstrate the intermediate result of each step in our method. We break down a complex task with a long logical chain into 8 steps. Each step has its own purpose and

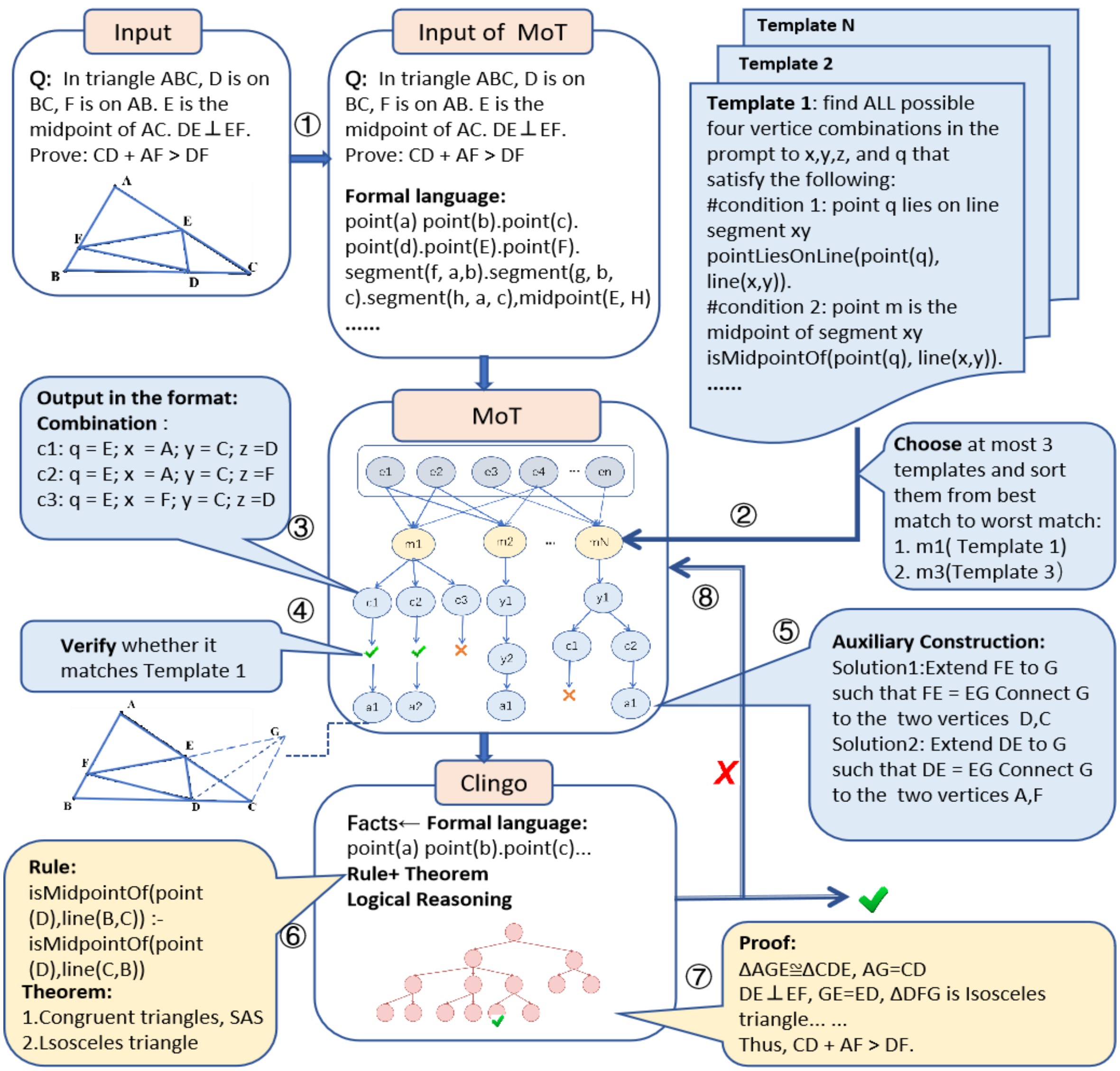


Figure 11. Holistic demonstration of our model in 8 steps on Example 1: (1) Diagram and Text parsing (2) Template selection. (3) Template mapping. (4) LLM Self-evaluation. (5) Auxiliary Construction. (6) Facts and Rules provided to Clingo. (7) Forward Chaining from the conditions to formulate proof. (8) Backtracking to auxiliary construction if proof failed.

is included to solve an issue encountered in the proving process.

① enhances our model's spatial perception – critical in solving geometry problems – by clarifying the relative positions of points, lines, and edges. ② is the heuristic of our search algorithm, determining which templates are related to the problem and which to apply first. ③ is critical in our method, boosting the precision and extending the flexibility of auxiliary constructions by searching for structures in the problem that match the template, which is characterized by vertices and their relationships. ④ is another heuristic, allowing LLMs to evaluate its results in ③, improving both the accuracy and efficiency in searching. ⑤ generates the auxiliary lines and is the output of MoT.⑥and⑦ complement the MoT by finishing the proof with Clingo's strict logical reasoning, which is assisted by rules and theorems. Finally, ⑧ closes the loop of neural intuition and logical reasoning by returning to ② if our goal is not reached; it emulates the human trial-and-error problem-solving thought process.

The generalization ability of our model is shown in Figure 12. We sticked to one example throughout our paper

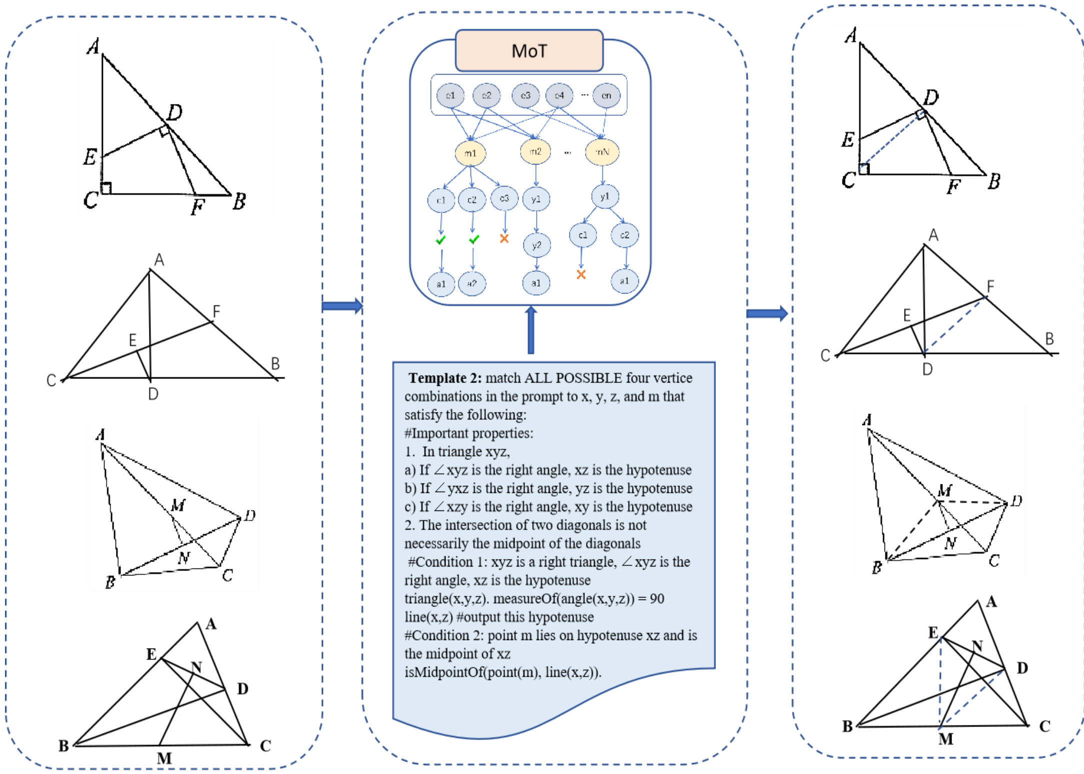


Figure 12. Generalization of Template (B), which can solve a category of problems. Four examples of where Template (B) can be used to construct auxiliary lines.

for clarification. However, each template can solve a large variety of problems. In Figure 12, four examples of problems Template (B) can solve are displayed. Details of the performance of all midpoint-related templates on a variety of problems are illustrated in Appendix B.

5. Conclusions

Automated geometry theorem proving is a critical pursuit in the extension and exploration of the reasoning capabilities of artificial intelligence. In this project, we proposed a closed loop of neural intuition and logical reasoning to address auxiliary construction and reasoning in geometry theorem proving. The neural intuition is derived from our novel design of the MoT prompting method, which engenders the neural intuition capabilities of Large pre-trained Language Models, such as GPT-4, to construct auxiliary construction. The logical reasoning is derived from our resort to traditional knowledge-based artificial intelligence, using a forward chaining algorithm to accomplish the rigorous proofs that LLMs are not able to achieve. These two critical parts of geometry proofs forms a closed-loop when we emulated humans' trial-and-error problem-solving approaches, iterating between auxiliary construction and logical reasoning until the final goal is proven. The effectiveness of our method is verified with a 66% accuracy performance on a complex geometry-proof dataset.

To take advantage of our method's keen relationship to the human thought process, our future efforts would focus on further improving our design and implementing it in real-world tutoring. First, we plan to revise and summarize more templates to cover the vast majority of geometry proof problems. Next, we aim to develop an AI geometry tutor to assist students in geometry studying. Since a big part of learning geometry is back-and-forth guidance and discussion - just like how we guided LLMs - interactive AI would be suitable for this role. It can communicate and provide feedback, addressing the biggest fear most students face in geometry: constructing auxiliary lines.

ACKNOWLEDGMENTS

I would like to express my heartfelt thanks to Dr. Wei Wang from Beijing Insititute of General Artificial Intelligence and Dr. Yu Chen Zhou from Talent Institute Beijing No.101 High School as the supervisors of my project, for the guidance and great help on documentation and experiments.

REFERENCES

[1].Chou S C, Gao X S, Zhang J Z. A deductive database approach to automated geometry theorem proving and discovering[J]. Journal of Automated Reasoning, 2000, 25(3): 219-246.

[2].Elias J M. Automated geometric theorem proving: Wu's method[J]. 2004.

[3].Janičić P, Narboux J, Quaresma P. The area method: a recapitulation[J]. Journal of Automated Reasoning, 2012, 48(4): 489-532.

[4].Bubeck S, Chandrasekaran V, Eldan R, et al. Sparks of artificial general intelligence: Early experiments with gpt-4[J]. arXiv preprint arXiv:2303.12712, 2023.

[5].Chou S C. Automated reasoning in geometries using the characteristic set method and Gröbner basis method[C]//Proceedings of the international symposium on Symbolic and algebraic computation. 1990: 255-260.

[6].Mrinmaya Sachan and Eric Xing. 2017. Learning to solve geometry problems from natural language demonstrations in textbooks. In Proceedings of the 6th Joint Conference on Lexical and Computational Semantics, pages 251–261.

[7].Gan W, Yu X, Wang M. Automatic understanding and formalization of plane geometry proving problems in natural language: A supervised approach[J]. International Journal on Artificial Intelligence Tools, 2019, 28(04): 1940003.

[8].Lu P, Gong R, Jiang S, et al. Inter-GPS: Interpretable geometry problem solving with formal language and symbolic reasoning[J]. arXiv preprint arXiv:2105.04165, 2021.

[9].Zhou M, Yu X. Proving Geometric Problem by Adding Auxiliary Lines-Based on Hypothetical Test[C],International Conference on Artificial Intelligence in Education Technology. Singapore: Springer Nature Singapore, 2021: 151-161.

[10].Matsuda N, Vanlehn K. Gramy: A geometry theorem prover capable of construction[J]. Journal of Automated Reasoning, 2004, 32: 3-33.

[11].Wang K, Su Z. Automated geometry theorem proving for human-readable proofs[C]//Twenty-Fourth International Joint Conference on Artificial Intelligence. 2015.

[12].Mai C, Research on Addition of Auxiliary Quantities toPlane Geometry Problems Based on Graph NeuralNetwork[R]. Master Thesis, 2023.

[13].Polu S, Sutskever I. Generative language modeling for automated theorem proving[J]. arXiv preprint arXiv: 2009.03393, 2020.

[14].Davies A, Veličković P, Buesing L, et al. Advancing mathematics by guiding human intuition with AI[J]. Nature, 2021, 600(7887): 70-74.

[15].Frieder S, Pinchetti L, Griffiths R R, et al. Mathematical capabilities of chatgpt[J]. arXiv preprint arXiv:23 01.13867, 2023.

[16].OpenAI. GPT-4 technical report.arXiv preprint arXiv:2303.08774, 2023.

[17].Wei J, Wang X, Schuurmans D, et al. Chain of thought prompting elicits reasoning in large language models[J]. arXiv preprint arXiv:2201.11903, 2022.

[18].Yao S, Yu D, Zhao J, et al. Tree of thoughts: Deliberate problem solving with large language models[J]. arXiv preprint arXiv:2305.10601, 2023.

[19].Long J. Large Language Model Guided Tree-of-Thought[J]. arXiv preprint arXiv:2305.08291, 2023.

[20].Gebser M, Kaminski R, Kaufmann B, et al. Theory solving made easy with clingo 5[C]//Technical Communications of the 32nd International Conference on Logic Programming (ICLP 2016). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[21].Lee J H, Shin J, Realff M J. Machine learning: Overview of the recent progresses and implications for the process systems engineering field[J]. Computers & Chemical Engineering, 2018, 114: 111-121.

[22].Minjoon Seo, Hannaneh Hajishirzi, Ali Farhadi, Oren Etzioni, and Clint Malcolm. 2015. Solving geometry problems: Combining text and diagram interpretation. In Proceedings of Empirical Methods in Natural Language Processing (EMNLP), pages 1466– 1476.

[23].Chen J, Tang J, Qin J, et al. GeoQA: A geometric question answering benchmark towards multimodal numerical reasoning[J]. arXiv preprint arXiv:2105.14517, 2021

[24].Hao Y, Zhang M, Yin F, et al. PGDP5K: A Diagram Parsing Dataset for Plane Geometry Problems[C]. 2022 26th International Conference on Pattern Recognition (ICPR). IEEE, 2022: 1763-1769.

[25].Xueersi (Chinese),Geometry Auxiliary Line Breakthrough Practice Exercises,Modern Education Publishing House ,2022

[26].Kojima T, Gu S S, Reid M, et al. Large language models are zero-shot reasoners[J]. Advances in neural information processing systems, 2022, 35: 22199-22213.

APPENDIX A.

Source of the selected topic, research background:

The topic of this project comes from the combination of my personal interest and the current research trend in artificial intelligence. Auxiliary constructions have once been one of my biggest headaches when learning geometry. However, my skills largely improved as I communicated with my teachers. I gradually started to gain an instinctual sense of the direction of auxiliary constructions. From there, I realized the beauty and intricacy of geometry intuition. When the newest LLM (GPT-4) emerged and demonstrated striking human-like abilities, I wondered whether it possessed this capability and whether it could communicate with users, assisting them in geometry theorem proving.

The work and contribution of each team member(s):
The project is completed independently under the guidance of the supervisors.

The relationship between the supervisors and the student, the role supervisors played in the process of writing thesis, and whether the tutoring is paid:
Dr. Wei Wang is an off-campus supervisor of the Branch AI Laboratory of Center on Frontiers of Computing Studies Peking University, Talent Institute, Beijing No. 101 High School. Dr. Yu Chen Zhou is the supervisor of the Branch AI Laboratory of Center on Frontiers of Computing Studies Peking University, Talent Institute, Beijing No. 101 High School. I am a project team member of the laboratory.

Research completed with the assistance of others:
The project is completed independently under the guidance of the supervisors.

Team Profile
Zhixing Lu: International Accelerated Class, grade 11, Beijing No. 101 High School. She has a strong interest in AI and mathematics. She has taken several college courses such as Multivariable Calculus, Linear Algebra, and University of California, Berkley's course on Machine Learning and Predictions, Inferential Thinking through Simulations etc. She has completed several AI related projects, including:

- Riverine Debris Detection by Unsupervised Image Segmentation
- Global Surface Temperatures Prediction with Bayesian Structural Time Series
- Analysis of Restaurant Ratings and Prices in Beijing and Los Angeles

She has distinguished accomplishments in science and mathematics competitions，including:

- First place, High School STEM Competition, Industrial Engineering and Operations Management (IEOM) – The 5th European Rome Conference, 2022.07
- Top 1% in American Mathematics Competitions (AMC) 10 (2022.12)
- Top 5% in AMC 12, 2022.12
- Qualification for the final round as one of the seven girls in Beijing, S.-T. Yau High School Girls' Mathematics Contest, 2022.11
- Advance to USAJMO (USA Junior Mathematical Olympiad) , 2022.12

She has also published the following journal paper.

- Zhixing Lu. Analysis of Restaurant Ratings and Prices in Beijing and Los Angeles. Proceedings of the International Conference on Industrial Engineering and Operations Management, 5th Europe Rome Conference, July. 2022.

Supervisor Profile
Wei Wang: received Ph.D. degree in University of Chinese Academy of Sciences in 2011. He has ever been an assistant/associate professor in National Lab of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA) from 2011 to 2021. He is currently a research scientist at Beijing Institute for General Artificial Intelligence (BIGAI). His research interests focus on computer vision, commonsense reasoning and creative problem solving.

Yu Chen Zhou: Ph.D., certificated research fellow, supervisor of the Branch AI Laboratory of Center on Frontiers of Computing Studies Peking University, Talent Institute, Beijing No. 101 High School. He is a senior member of the ACM and IEEE, and former member of Technical Committee, Embedded System Society, China Computer Federation. With 20 years of technical innovation experience in IBM, he served as senior research manager of AI perception in IBM Research China, a member of IBM Academy of Science and Technology, IBM Master Inventor, chair of technical committee and patent review committee of the center, etc. He had won 3 outstanding technical achievement awards, published 1 book, participated, and contributed to 2 international standards, obtained around 50 international patents, and published more than 30 papers.

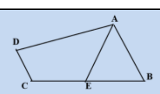
APPENDIX B.

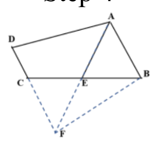
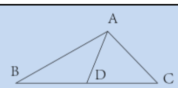
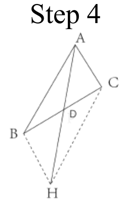
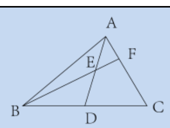
Appendix B contains the experimental results of our method.

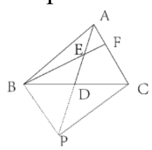
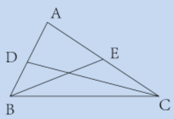
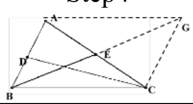
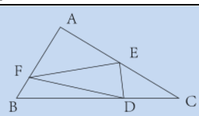
B.1 Examples of MoT's performance on auxiliary constructions.

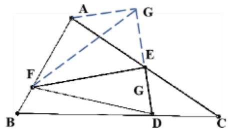
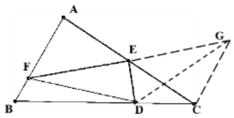
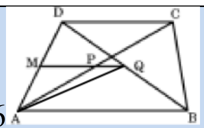
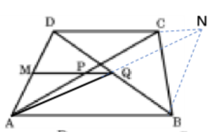
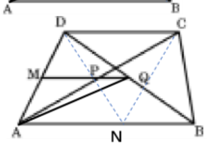
Template 1	

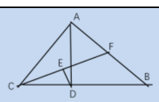
	Step 1 Template Selection
	Here are the keywords of several templates for making auxiliary constructions. According to the matching of the prompt and each of these keywords, choose three templates and sort them from best match to worst match
	Keywords:
	Template 1: Midpoint
	Template 2: Right Triangle and Midpoint
	Template 3: Isosceles triangle
	Template 4: Two or more midpoints
	Template 5: Equilateral triangle
	Template 6: Congruent triangles
	Template 7: Circle
	Template 8: Quadrilateral
	
	Step 2 Template Mapping
	for template 1, find ALL POSSIBLE four vertice combinations in the prompt to x,y,z, and m that satisfy the following:
	#condition 1: point m lies on line segment xy
	pointLiesOnLine(point(m), line(x,y)).
	#condition 2: point m is the midpoint of segment xy
	isMidpointOf(point(m), line(x,y)).
	#condition 3: point z is not on line xy
	not(pointLiesOnLine(point(z), line(x,y))
	#condition 4: there exists line segment zm or mz in the prompt
	line(z,m).
	Output in the format:
	Combination ?: m = ?x = ?y = ?z = ?
	
	Step 3 Self-verification
	Check if all your combinations satisfy conditions in the template.
	
	Step 4 Auxiliary Construction
	For each combination, do the following to construct auxiliary lines:
	Extend zm to n such that zm = mn. Connect n to x and y. By intuition, you may also connect n to other vertices in the prompt that would help solving the problem.
	(x,y,z,m match to the vertices that you claimed in your combination, n is a new point you created)

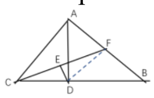
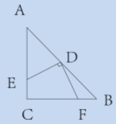
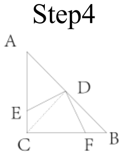
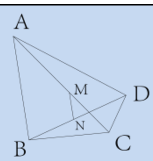
1 	In quadrilateral ABCD, E is the midpoint of BC， AB is parallel to CD, If AE bisects angle BAD, Prove: AD = AB + DC.
Step 1	Template 1 (Midpoint) - The midpoint of BC was used to show BE = EC. Template 8 (Quadrilateral) - Properties of the given quadrilateral ABCD were used. Template 2 (Right triangle and midpoint) - Not used since no right triangles.
Step 2	Combination 1:m = E x = B y = C z = A Combination 2:m = E x = C y = B z = D
Step 3	Combination 1:m = E x = B y = C z = A Does m lie on line segment xy? Yes, E lies on line segment BC. Is m the midpoint of xy? Yes, E is the midpoint of BC.

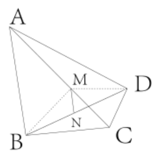
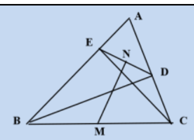
<table>
<tr><td></td><td>Is z not on line xy? Yes, A is not on line BC.
Does line segment zm exist? Yes, AE exists.
This combination satisfies all conditions.
Combination 2: m = E x = C y = B z = D
Does m lie on line segment xy? No, E does not lie on line segment CB.
Is m the midpoint of xy? No, E is not the midpoint of CB.
Is z not on line xy? Yes, D is not on line CB.
Does line segment zm exist? No, DE does not exist.
This combination does NOT satisfy all the conditions.
In conclusion, the only valid combination based on the given conditions is:
m = E x = B y = C z = A</td></tr>
<tr><td>Step 4

</td><td>Construction:
Extend AE to n such that AE = En ,Connect n to B, Connect n to C, Connect n to D .
The constructed auxiliary lines are:
Line En such that AE = En, Line nB, Line nC, Line nD ✔</td></tr>
<tr><td>2 
</td><td>In triangle ABC,P is the midpoint of BC. Prove: AP < 1/2(AB + AC).</td></tr>
<tr><td>Step 1</td><td>Template 1 Midpoint: Since P is already the midpoint of BC, this keyword is directly relevant to the problem.
Template2 Right Triangle and Midpoint: This can be relevant because using a right triangle might give us relationships between sides that can help prove the inequality.</td></tr>
<tr><td>Step 2</td><td>Condition 1: point m lies on line segment xy.
Condition 2: point m is the midpoint of segment xy.
Condition 3: point z is not on line xy.
Condition 4: there exists a line segment zm or mz in the prompt.
Let's examine the possible combinations:
Combination 1:m = P (as P is the midpoint of BC) x = B y = C
z = A (since A is not on line BC, and AP exists in the triangle)</td></tr>
<tr><td>Step 3</td><td>All the conditions are satisfied for Combination 1.</td></tr>
<tr><td>Step 4

</td><td>We have extended line segment AP to a new point n, such that AP = Pn.
We have drawn lines from point n to B and C.✔</td></tr>
<tr><td>3 
</td><td>In triangle ABC, AD is the median on BC. E is a point on AD such that BE=AC.
Extending BE intersects AC at F. prove that AF= EF</td></tr>
<tr><td>Step 1</td><td>Template 1: Midpoint
Template 3: Isosceles triangle</td></tr>
<tr><td>Step 2</td><td>Combination 1:m = D   x = B   y = C   z = E
Combination 2: m = E   x = D   y = B   z = A
Combination 3:m = E   x = D   y = C   z = F</td></tr>
<tr><td>Step 3</td><td>Combination 1: Does not satisfy Template 1 fully since D is not constructed as a midpoint, it is given
Combination 2： :m = E x = D   y = B   z = A Satisfies conditions 1-4
Satisfies Template 1: E is constructed as midpoint of DB</td></tr>
</table>

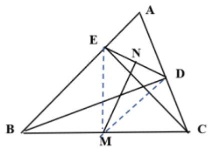
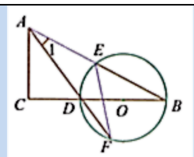
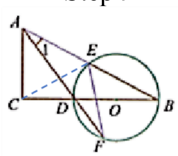
	Combination 3: m = E x = D y = C z = F Satisfies conditions 1-4 Satisfies Template 1: E is constructed as midpoint of DC
Step4 	Combination 2:Extend EA to N such that EA = EN. Connect N to D and B. Also connect N to C.✓ Combination 3:Extend EF to N such that EF = EN. Connect N to D and C. Also connect N to B.✗
 4	In triangle ABC, D is on AB, E is the midpoint of AC. BE and CD intersect at F. BD = DF. Prove: AB = CF.
Step1	Template 1: Midpoint - This is clearly relevant since E is given as the midpoint of AC. Template 3: Isosceles triangle - Given that BD = DF, the idea of an isosceles triangle (a triangle with two sides of equal length) seems to fit with the information provided.
Step2	Combination 1:m = E x = A y = C z = B
Step3	#condition 1: E lies on AC. This is true as E is the midpoint of AC. #condition 2: E is the midpoint of AC. This is also true as stated in the prompt. #condition 3: B is not on AC. True, B is a vertex distinct from the line segment AC. #condition 4: BE exists. Yes, the line segment BE is mentioned in the prompt. It seems that all conditions are satisfied for this combination.
Step4 	Extend BE to a point N such that BE = EN. Connect N to A. Connect N to C. ✓
 5	In triangle ABC, D is on BC, F is on AB. E is the midpoint of AC. DE⊥EF. Prove: CD + AF > DF
Step 1	Template 2: Right Triangle and Midpoint - This template seems the most relevant since it combines two key elements mentioned in your problem: the right triangle and midpoint. Template 1: Midpoint - The concept of a midpoint is directly mentioned in your problem, and this template would undoubtedly be of use.
Step 2	Template 2: Combination 1: m = E x = D y = E z = F✗ Template 1: Combination 1:m = E x = A y = C z = F Combination 2:m = E x = A y = C z =D
Step 3	Template 2: Using the above data, the vertices can be matched as:Combination 1: m = E x = D y = E z = F Template 1: :m = E x = A y = C z = D or F Condition 1: m lies on xy. E lies on AC. This condition is met. Condition 2: m is the midpoint of xy. E is the midpoint of AC. This condition is met. Condition 3: z is not on line segment xy. If we take z = D, then D lies on BC and is not on AC. This condition is met. If we take z = F, then F lies on AB and is not on AC. This condition is met. Condition 4: There exists a line segment zm or mz. For z = D, we have DE. This condition is met. For z = F, we have EF. This condition is met.
Step 4	Template 2: Following the new instruction: Connect ym.✗ However, since y and m are the same point (E), connecting y to m doesn't result in a new line. They are already the same point, so there's no line to be drawn between them in this

  	combination. Template 1: For z = D Constructing Auxiliary Lines: Extend DE to a point N such that DE = EN. This implies DE = EN. Connect N to A and C. So, we have lines NA and NC. Connect GF For z = F Constructing Auxiliary Lines: Extend FE to a point N such that FE = EN. Connect N to A and C. So, we have lines NA and NC., Connect GD
 6	In trapezoid ABCD, CD//AB, Q is the midpoints of BD, P is the midpoints of AC, extend QP to intersect AD at M. Prove AM=DM
Step 1	Template 1: Midpoint - Directly relates to the given conditions. Template 8: Quadrilateral - The whole figure is a trapezoid which is a type of quadrilateral. Template 2: Right Triangle and Midpoint - Midpoints are relevant, and while right triangles aren't explicitly mentioned, they might be involved in the solution.
Step 2	Condition 1: m lies on segment xy. This means m is between points x and y. Condition 2: m is the midpoint of segment xy. This directly indicates two possibilities from the problem - Q is the midpoint of BD and P is the midpoint of AC. Condition 3: z is not on the line xy. Condition 4: There exists a segment zm or mz in the problem statement. Based on the problem, we have segments AQ, QP, PK, and PA. Using these conditions, let's identify the combinations: Using Q as midpoint: m = Q x = B y = D For z, the possibilities based on the segments given are A (since AQ exists). So, A cannot lie on line BD. Combination 1: m = Q x = B y = D z = A Using P as midpoint: m = P x = A y = C For z, the possibilities based on the segments given are D (since PK exists) and A (since PA exists). However, A lies on AC which contradicts Condition 3. So, only D remains as a valid option. Combination 2: m = P x = A y = C z = D
Step 3	Combination 1: m = Q x = B y = D z = A Combination21: m = P x = A y = C z = D On re-evaluation, these remain as the only two combinations that fit the given conditions from the problem
Step 4   	Combination 1: m = Q x = B y = D z = A Extend AQ to a point n such that AQ = Qn. Connect n to B and D. Combination 2: m = P x = A y = C z = D Extend DP to a point n such that DP = Pn. Connect n to A and C.
Template 2	
	Step 1 Template Selection Here are the keywords of several templates for making auxiliary constructions. According to

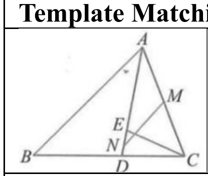
	the matching of the prompt and each of these keywords, choose three templates and sort them from best match to worst match Keywords: Template 1: Midpoint Template 2: Right Triangle and Midpoint Template 3: Isosceles triangle Template 4: Two or more midpoints Template 5: Equilateral triangle Template 6: Congruent triangles Template 7: Circle Template 8: Quadrilateral **Step 2 Template Mapping** for template 2, match ALL POSSIBLE four vertice combinations in the prompt to x, y, z, and m that satisfy the following: #Important properties: 1. In triangle xyz, a)If ∠xyz is the right angle, xz is the hypotenuse b)If ∠yxz is the right angle, yz is the hypotenuse c)If ∠xzy is the right angle, xy is the hypotenuse 2.The intersection of two diagonals is not necessarily the midpoint of the diagonals #Condition 1: xyz is a right triangle, ∠xyz is the right angle, xz is the hypotenuse triangle(x,y,z). measureOf(angle(x,y,z)) = 90 line(x,z) #output this hypotenuse #Condition 2: point m lies on hypotenuse xz and is the midpoint of xz isMidpointOf(point(m), line(x,z)). Output in the format:Combination ?: m = ? x = ? y = ? z = ? **Step 3 Self-verification** Check if all your combinations satisfy conditions in the template. **Step 4 Auxiliary Construction** For each combination, do the following to construct the auxiliary line: Connect ym. (y,m match to the vertices that you claimed in your combination)
7 	In triangle ABC, AD⊥BC, F is the midpoint of AB. DC=BF, DE⊥CF at E, prove that CE=EF.
Step1	1. Template 2 (Right Triangle and Midpoint) - This is the best match because the prompt mentions right angles and a midpoint. 2. Template 1 (Midpoint) - The midpoint is a key element of the prompt so this template is relevant
Step2	Combination 1: m = F x = A y = D z = B Combination 2: m = F x = A y = D z = C
Step3	Looking at the prompt again, the only right angle specified is ∠ADB. So for the vertex combinations to satisfy the conditions: x and z must be A and B in some order, since AB is the hypotenuse y must be D since ∠ADB is the right angle

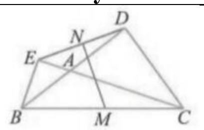
	m must be F since F is the midpoint of AB The vertex combinations satisfying all conditions are: Combination 1:m = F x = A y = D z = B
Step4 	Connect DF
 8	Triangle ABC, C is a right angle,AC = BC, D is the midpoint of AB. DE⊥DF, points E, F are on AC, BC, respectively. Prove: DE = DF
Step1	Template 2: Right Triangle and Midpoint - Relevant because ABC is a right triangle and D is the midpoint of AB. Template 1: Midpoint - Relevant because D is the midpoint of AB. Template 3: Isosceles triangle - Relevant because AC = BC, making ABC an isosceles triangle.
Step2	Combination 1: m = D x = A y = C z = B
Step3	#Condition 1: Considering the right angle at C, the combinations for xyz can be: Combination 1:x = B y = C z = A With BC being the side opposite the right angle, and BA is the hypotenuse. #Condition 2: Point m lies on hypotenuse xz and is the midpoint of xz. For the above combination, xz corresponds to BA. And D is the midpoint of BA. Thus, this combination satisfies both conditions: Combination 1: m = D x = B y = C z = A
Step4 	Connect C to D.
 9	In quadrilateral ABCD, ∠ABC = ∠ADC = 90. M and N are the midpoints of AC and BD, respectively. Prove: BM = DM.
Step1	Template 2：Right Triangle and Midpoint: This seems to be the most relevant template because we have a quadrilateral with two right angles and we are also dealing with midpoints (M and N). This is likely to provide tools or steps relevant to the problem. Template 1 :Midpoint: The problem involves the midpoints of diagonals of a quadrilateral. Therefore, any template focused on midpoints might offer insight or steps towards a solution, but it might be less specific than the "Right Triangle and Midpoint" template.
Step2	Combination 1 (Triangle ABC):m = M x = A y = B z = C Combination 2 (Triangle ADC):m = M x = A y = D z = C
Step3	For triangle ABC: y = B (Because the right angle is at vertex B, which is ∠ABC) x = A z = C So, AC (xz) would be the hypotenuse. For triangle ADC: y = D (Because the right angle is at vertex D, which is ∠ADC) x = A z = C So, AC (xz) would be the hypotenuse.

	Condition 2: point m lies on hypotenuse xz and is the midpoint of xz. Given, M is the midpoint of AC. For triangle ABC: m = M (Since M is on AC) For triangle ADC: m = M (Since M is on AC) Now, combining the above results: Both the combinations indeed satisfy the conditions in the template:
Step4  	Auxiliary Line 1: BM Auxiliary Line 2: DM
 10	In triangle ABC, D is on AC such that BD⊥AC at D, E is on AB such that CE⊥AB at E. M and N are the midpoints of BC and DE, respectively. Prove: MN⊥DE.
Step1	Template 2: Right Triangle and Midpoint This seems like the best match since the prompt mentions right angles, midpoints, and perpendicular lines which suggests working with right triangles and midpoints. Template 1: Midpoint Midpoints are mentioned directly in the prompt so this is likely useful. Template 3: Isosceles triangle While not directly mentioned, it's possible isosceles triangles could be constructed as part of a proof.
Step2	Let's break it down step by step. For Condition 1: "xyz is a right triangle, ∠xyz is the right angle, xz is the hypotenuse". Given the prompt: We have ∠BDC = 90 and ∠CEB = 90. Considering the right angle at BDC, BD or DC could be the legs of the triangle, making BC the hypotenuse. Similarly, for the right angle at CEB, CE or EB could be the legs, making CB the hypotenuse again. For Condition 2: "point m lies on hypotenuse xz and is the midpoint of xz". From the prompt: M is the midpoint of BC. Now, considering the above, let's try to match the conditions to the given triangle: Combination 1: Given the triangle BDC with the right angle at BDC: m = M (Midpoint of BC) x = B y = D z = C Combination 2: Given the triangle CEB with the right angle at CEB: m = M (Midpoint of BC) x = C y = E z = B
Step3	Condition 1:xyz is a right triangle, ∠xyz is the right angle, xz is the hypotenuse. Condition 2:point m lies on hypotenuse xz and is the midpoint of xz. Considering the triangle ABC: Combination 1: For the triangle BDC with ∠BDC as the right angle: x = B y = D z = C Thus, BC (or equivalently z-x) would be the hypotenuse. For the midpoint condition, M is the midpoint of BC. So m = M satisfies Condition 2. Combination 2:

	For the triangle CEB with ∠CEB as the right angle: x = C y = E z = B Thus, CB (or equivalently z-x) would be the hypotenuse. For the midpoint condition, M is the midpoint of BC. So m = M satisfies Condition 2. Upon re-evaluation, both combinations do indeed satisfy the conditions provided in the template.
Step4  	Thus, for the two combinations, we have: In the first combination, the auxiliary line is DM. In the second combination, the auxiliary line is EM.
 11	In △ABC, ∠C = 90°, D is a point on side BC, circle O with diameter DB passes through the midpoint E of AB, intersects the extension of AD at F, connect EF. Prove: ∠1 = ∠F.
Step1	Template 2: Right Triangle and Midpoint - this involves a right triangle and midpoint, both of which are given Template 7: Circle - a circle is given Template 1: Midpoint - the midpoint of AB is used
Step2	Combination 1:m = E x = A y = B z = C
Step3	Combination 1 fully satisfies the conditions laid out for Template 2
Step4  	To construct the auxiliary line: Connect ym y = A m = E Therefore, the auxiliary line construction is: Connect AE

B.2 Examples for two types of typical failure cases

Template Matching Error	
 	In triangle ABC, D is on BC , E is on AD such that ∠ACE = ∠B, CE=CD, M is the midpoints of AC, MN//AB ,N is on DE. Prove: EN=DN.
Step1	Based on the keywords given in the prompt, here is how I would rank the template matches from best to worst: Template 1 (Midpoint) - The prompt directly mentions constructing the midpoint M of AC, so this template is the best match. Template 4 (Two or more midpoints) - The prompt only mentions one midpoint, but this template could still be useful as we can construct more midpoints as needed. Template 8 (Quadrilateral) - The prompt mentions quadrilateral ABCD which we can use properties of. The other templates don't seem applicable since the prompt does not mention right triangles, isosceles triangles, equilateral triangles, congruent triangles, or circles. So in summary, the best to worst template matches are: 1 Midpoint

	4 Two or more midpoints ✗ 8 Quadrilateral ✗
Auxiliary Construction Error	
	In right triangles BDC and BEC, ∠BDC = 90°, ∠BEC = 90°. M and N are midpoints of line segments BC and DE respectively. Prove: MN⊥DE.
Step1	Here is how I would rank the template keywords from best match to worst match for this geometry proof prompt: 1. Template 4: Two or more midpoints This template matches directly with the given information that M and N are midpoints. Using properties of midpoints would likely be useful. 2. Template 2: Right triangle and midpoint ✓ This template matches the right triangles given in the prompt. Midpoint properties could still be useful with right triangles. 3. Template 1: Midpoint This template broadly matches that midpoints are mentioned, though does not account for multiple midpoints or right triangles. The other template keywords do not seem directly relevant to this specific proof prompt involving midpoints and perpendicularity. So I would rank templates 3-8 as worse matches than the top 3 templates.
Step2	Here are the possible matches for the vertices in the prompt to the variables x, y, z, and m in the template, satisfying the given conditions: Combination 1: m = M x = B y = D z = C Combination 2: m = M x = C y = B z = D Combination 3: m = N x = D y = B z = E Combination 4: m = N x = E y = D z = B The explanations are: - BDC and BEC are specified as right triangles, so B, D, C and B, E, C can match x, y, z in a right triangle combination. - M is specified as the midpoint of BC and N is the midpoint of DE. So M and N can match the midpoint m in the template. - The hypotenuses identified using the right angle properties are BC for BDC, and EB for BEC. - This covers the 4 possible combinations of mapping the givens to the template variables.
Step3	You're right, let me re-check my combinations against the conditions: Combination 1: m = M x = B y = D z = C This satisfies the conditions. BDC is a right triangle with right angle at D. BC is the hypotenuse with M as its midpoint. Combination 2: m = M x = C y = B z = D This does NOT satisfy the conditions. BDC has right angle at D, not at B. Combination 3: m = N x = D y = B z = E This does NOT satisfy the conditions. BEC has right angle at B, not at E. Combination 4: m = N x = E y = D z = B This satisfies the conditions. BEC is a right triangle with right angle at B. EB is the hypotenuse with N as its midpoint.So you're correct, only Combinations 1 and 4 fully match the template conditions. ✗

B.3. LLMs' ability to solve geometry problems.

Our early extensive experiments on the two LLMs GPT-4 and Claude 2 showed that they have poor spatial ability. After simple prompting, the LLMs' ability was significantly improved, indicating the prompting is the key to improving LLM's mathematical abilities. However, this is only for fundamental spatial abilities and LLMs are tested on simple computational problems. Solving complicated geometry theorem proving problems that require auxiliary lines demands intricate prompting methods such as our MoT. Following are some of our experiments:

Zero-shot	
Example 1	Q: Triangle ABC， ∠B=1/3∠C, ∠ACB=90, solve ∠BAC
GPT-4	Given triangle ABC where: Let's use the sum of the angles in a triangle, which is always 180°. Let x be the measure of ∠C in degrees. According to the problem, ∠B = (1/3)x. Thus: ∠A + ∠B + ∠C = 180° ∠A + (1/3)x + x = 180° (since ∠ACB = ∠A = 90°) 90° + (1/3)x + x = 180°Now, combine like terms: (1/3)x + x = 90°Multiplying every term by 3 to clear the fraction: x + 3x = 270 4x = 270 x = 67.5°So, ∠C = 67.5° and ∠B = (1/3)(67.5°) = 22.5° ∠BAC = 180° - 90° - 22.5° = 67.5°
Claude2	Okay, let's solve this step-by-step: ∠B = 1/3 ∠C， ∠ACB = 90 degrees (right angle) * Angles in a triangle sum to 180 degrees So ∠B + ∠C + 90 = 180 * ∠B + ∠C = 90 (rearranging) We know: ∠B = 1/3 ∠C ∠C = 60 degrees Since ∠B = 1/3 ∠C, ∠B = 1/3 * 60 = 20 degrees Therefore, ∠BAC = 180 - 90 - 20 = 70 degrees
IFlytek Spark	∠B = 1/3 ∠C，∠B + ∠C = 90， ∠C=67.5◦,∠BAC = 180° - 90° - 22.5° = 67.5°
Error	∠ACB=∠A
Few-shot	
Example 1	Pay attention to the equivalence relations between the angles: ∠XYZ=∠ZYX=∠Y, solve：Triangle ABC， ∠B=1/3∠C, ∠ACB=90, solve ∠BAC
GPT-4	Given: 1. Triangle ABC with ∠ACB = 90° (Right triangle) 2. ∠B = 1/3∠C 3. The equivalence relation between the angles: ∠XYZ = ∠ZYX = ∠Y. We know the sum of angles in a triangle is 180°.∠A + ∠B + ∠C = 180° ....(i) Given ∠ACB (or ∠C) = 90°Substitute this value in equation (i): ∠A + ∠B + 90° = 180° ∠A + ∠B = 90° ....(ii) Also, given that ∠B = 1/3∠C: ∠B = 1/3 * 90°,∠B = 30° Now, substitute the value of ∠B in equation (ii): ∠A + 30° = 90°; ∠A = 60° Therefore, ∠BAC (or ∠A) = 60°
Claude2	Okay, let's break this down step-by-step: Given: ∠XYZ = ∠ZYX = ∠Y (Angle relationships) ΔABC ∠B = 1/3 ∠C ∠ACB = 90° (Right angle)， We need to solve for ∠BAC Using the fact that ∠B = 1/3 ∠C, we can write: ∠B = 1/3 ∠C And since ∠ACB is a right angle, ∠C = 90° Substituting: ∠B = 1/3 (90°) ∠B = 30° Now using the fact that the sum of angles in a triangle is 180°, we can write: ∠B + ∠BAC + 90° = 180° 30° + ∠BAC + 90° = 180° ∠BAC = 180° - (30° + 90°) ∠BAC = 60°

	Therefore, the measure of ∠BAC is 60°.
IFlytek Spark	∠AYX=∠BXZ ∠AZY=∠CXY； ∠AYX+∠X+∠AXY=180°...

Zero-shot	
Example 2	Q:AB is 5 units long. BC is 3 units long. Triangle ABC is a right triangle. ∠C is the right angle. What is tan(∠B) ?
GPT-4	A: From the Pythagorean theorem for right triangles: $AC^2 = BC^2 + AB^2$ $AC^2 = 3^2 + 5^2 = 34$ (tan(∠B) =frac{opposite side}{adjacent side} For angle B, the opposite side is AB and the adjacent side is BC. tan(∠B) = AB/BC So, (tan(∠B) =5/3.
Claude2	A: AB is 5 units long, BC is 3 units long, Triangle ABC is a right triangle,Angle C is the right angle,We want to find tan(angle B) In a right triangle, tan(angle) = opposite side / adjacent side The opposite side of angle B is BC, which is 3 units The adjacent side of angle B is AB, which is 5 units So, tan(angle B) = BC/AB = 3/5 Therefore, tan(∠B) = 3/5.
IFlytek Spark	A: Since ∠C is the right angle, we know that tan(∠ B) =BC/AB， Since AB = 3 units and BC = 3 units, we have: tan(∠B) = 3/5
Analyze	The problem gives a right angle, but the LLM cannot distinguish between the hypotenuse and the adjacent side to ∠B.
Few-shot	
Example 2	Pay attention to the relations in a right triangle: If Triangle XYZ is a right triangle, Note the relationship between the side and the right angle: 1、If ∠X is a right angel, so YZ is is the hypotenuse. 2、The opposite side of ∠Y is XZ 3、 The adjacent side of ∠Y is XY Solve: AB is 5 units long. BC is 3 units long. Triangle ABC is a right triangle. ∠C is the right angle. What is tan(∠B) ?
GPT-4	AB is the side opposite to ∠C, and BC is the side adjacent to ∠C. Using the relationships given: 1. If ∠C is the right angle, then AB is the hypotenuse. 2. For ∠B, BC is the side opposite ∠B and AC (not mentioned, but by deduction) is the side adjacent to ∠B. However, you've provided the lengths for AB and BC, and not for AC. - Side adjacent to ∠B = BC = 3 units - Side opposite to ∠B = AC. We can find AC using the Pythagorean theorem: Given, AB (hypotenuse) = 5 units BC = 3 units \(AC^2 \) + \(BC^2 \) = \(AB^2 \) AC = 4 unitsThus,

	\(\tan(∠B) \) = 4/3
Claude2	Okay, let's go through this step-by-step:- Triangle ABC is a right triangle - AB is the hypotenuse and is 5 units long - AC is the side opposite to ∠B - BC is the side adjacent to ∠B and is 3 units long - tan(∠B) = opposite side / adjacent side - tan(∠B) = BC / AC - Since we don't know the length of AC, we can use the Pythagorean theorem: - AC^2 + BC^2 = AB^2 - AC^2 + 3^2 = 5^2 - AC^2 = 25 - 9 = 16 - AC = 4 units Therefore, tan(∠B) = BC / AC = 4 /3
IFlytek Spark	$AC^2 = BC^2 + AB^2$, $AC^2 = 3^2 + 5^2 = 34$, tan(∠ B) =AC/BC