symmetry

MDPI

Article

FGeo-ISRL: A MCTS-Enhanced Deep Reinforcement Learning System for Plane Geometry Problem-Solving via Inverse Search

Yang Li [1], Xiaokai Zhang [1], Cheng Qin [2], Zhengyu Hu [1] and Tuo Leng [1,*]

[1] School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China; leeyoung628@shu.edu.cn (Y.L.)
[2] Institute of Artificial Intelligence, Shanghai University, Shanghai 200444, China
* Correspondence: tleng@shu.edu.cn

Abstract

Geometric problem-solving has always been a great challenge in the field of deductive reasoning and artificial intelligence. Symmetry is a defining characteristic of geometric shapes and properties. Consequently, the application of symmetry principles to geometric reasoning arises as a natural choice. To address the efficiency degradation and limited generalization, we propose FGeo-ISRL, a neural-symbolic inverse search framework whose core is the synergistic integration of a task-fine-tuned large language model and Monte Carlo Tree Search. Under the formal framework of FormalGeo, geometric theorems are iteratively applied starting from the given conditions and the target conclusion, in order to infer the necessary supporting premises. The large language model simultaneously serves as a policy network and a value network, guiding theorem application decisions and evaluating intermediate proof states, whereas the Monte Carlo Tree Search performs structured exploration over the state space, both training for policy refinement and inference for online search. The reinforcement learning agent is trained with a hybrid reward scheme, combining immediate feedback from the value difference and a sparse success reward. Experiments demonstrate the effectiveness and correctness of FGeo-ISRL. It not only achieves a Single-Step Theorem Accuracy of 90.2% and a Geometric Problem-Solving Accuracy of 83.14%, but also ensures that every step of the proof process remains readable, verifiable, and traceable.

Keywords: Formal Mathematics; Automatic Reasoning; Neural-Symbolic Networks; Monte Carlo Tree Search; Reinforcement Learning

1. Introduction

In mathematics, symmetry is defined as the property of a mathematical object that remains invariant under certain transformations or operations. This concept is widely studied across various domains such as algebra, combinatorics [1], and especially geometry [2], where geometric symmetry has attracted considerable attention [3]. Geometric problem-solving has always been a great challenge in the field of deductive reasoning and artificial intelligence. Its goal is to start from given geometric conditions and derive the target conclusion through logical reasoning, theorem application, and mathematical operations. The core difficulty of this task lies in enabling computers to process the abstract logical relationships in text and the spatial information in graphics, while achieving high-precision semantic alignment [4] between the two modalities.

Check for updates

Received: 13 March 2026
Revised: 3 April 2026
Accepted: 6 April 2026
Published: 9 April 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Early automated geometric reasoning systems originated from symbolic expert systems [5,6], mostly designed based on manual rules, and performed forward search and reasoning starting from given conditions. The main advantages of such methods lie in their clear logic and verifiable results [7–9]. However, these methods suffer from the performance degradation caused by the huge space complexity of forward search [10–12], as well as the limited coverage dependent on expert experience [13–15]. Therefore, they have poor generalization performance and struggle to adapt to diverse geometric structures and semantic transformations of different types of problems.

With the development of deep learning and large language models (LLMs), researchers have begun to attempt introducing models with strong reasoning capabilities [16–18], especially those based on the Transformer architecture [19], into automated theorem proving and geometric problem-solving tasks [20]. This direction marks a significant advancement of machine intelligence in the field of formal reasoning and has greatly expanded the application prospects of artificial intelligence in mathematics. Leveraging the natural language understanding and generation capabilities of LLMs, the system can provide intuitive explanations and problem-solving strategies for geometric problems [21]. Nevertheless, despite the positive impact of LLMs in problem-solving through reasoning, the suggestions they provide cannot be regarded as rigorous mathematical theorem proofs [22]. Moreover, the inherent black-box nature [23] of neural network architectures makes it challenging to comprehend their internal logic, thereby preventing the derivation of an accurate and verifiable problem-solving path. Additionally, the lack of precise metrics for evaluating executable step sequences hinders the possibility of verifying a problem's complete and correct resolution through the output theorem sequence. Furthermore, using LLMs alone requires a large amount of manually annotated data, which comes at a very high cost. Thus, directly inferring the solvability of geometric problems through natural language models remains a highly challenging task [24].

We use the FormalGeo geometric formalization system as the fundamental tool and theoretical basis for solving geometry problems [25]. This system ensures the effectiveness and consistency of information integration while maintaining invariance in geometric transformations. Furthermore, this system integrates multiple datasets, including Geometry3k [26], GeoQA [27], GeoQA+ [28], and other online resources, thereby constructing a new dataset named FormalGeo7K [29].

Building on the basic inverse solving method of the FormalGeo system, we have constructed FGeo-ISRL, a geometry problem-solving system enhanced by heuristics and driven by reinforcement learning. FGeo-ISRL represents a typical neuro-symbolic architecture [30] that integrates the adaptive learning capabilities of AI agents with the structured precision of formal geometric reasoning. Employing adversarial game mechanisms for performance enhancement has become a common and effective strategy [31]. We introduce an innovative design where a pre-trained natural language model is decomposed into two dedicated components: a value network and a policy network combined with Monte Carlo Tree Search (MCTS). This architecture empowers the system to initiate reasoning from both the provided conditions and the objective goal, incorporating theorem premises iteratively to update the state in a backward fashion. By integrating reinforcement learning and MCTS, the framework effectively mitigates data acquisition costs. The empirical results further demonstrate that this approach achieves effective generalization performance and inference efficiency compared to traditional reasoning.

Our contributions can be categorized as follows:

1. We present FGeo-ISRL, a novel neural-symbolic framework that leverages Inverse Search and Reinforcement Learning for geometric problem-solving. The symbolic component is the formalization system of FormalGeo, while the neural component

is a prediction system that leverages the optimized policy network to directly select theorems, and uses the value network to evaluate state transitions, assisted by the rewards from MCTS.

2. We propose a condition–goal joint inverse reasoning mechanism to address the efficiency degradation and limited generalization observed in traditional forward reasoning, where the reasoning process starts solely from given conditions and becomes less effective as the theorem chain grows longer. By jointly leveraging both the initial conditions and the target conclusion, our approach reverses the reasoning direction and significantly improves inference efficiency and success rates in complex geometric problem-solving.
3. We integrate reinforcement learning and MCTS under a self-play optimization paradigm, enabling the model to iteratively refine theorem selection strategies and reasoning paths without requiring large-scale manual annotation of intermediate reasoning data. This integration effectively reduces supervision cost while enhancing model generalization and search efficiency.
4. We introduce a step-level reasoning evaluation framework that incorporates the Single-Step Theorem Accuracy (SSTA) metric to measure the precision of each inference step alongside the overall Geometric Problem-Solving Accuracy (GPSA). Extensive experiments conducted on the FormalGeo7K dataset demonstrate that FGeo-ISRL achieves a SSTA of 90.2% and GPSA of 83.14%. Furthermore, we perform comprehensive ablation studies by removing the value or policy networks and substituting alternative search strategies to quantify each module's individual contribution and analyze its impact on theorem prediction fidelity and problem-solving success.

2. Related Work

In the domain of automated geometric reasoning, both heuristic algorithms and reinforcement learning (RL) have emerged as key approaches for improving problem-solving efficiency and adaptability.

Heuristic algorithms have historically played a central role in early symbolic systems by guiding geometric reasoning through rule-based strategies, empirical knowledge, and local search mechanisms [32,33]. Systems such as GeoS [34], InterGPS [26], and FGeo-SSS [35] have adopted a heuristic-based search to construct logical proof paths for geometric propositions. AlphaGeometry [36] introduced a heuristic search framework to significantly reduce redundant inference branches, improving both speed and accuracy in solving complex problems. Its integration with classical symbolic approaches such as Wu's method [37] demonstrated the benefits of combining symbolic and neural techniques. The successor system, AlphaGeometry2 [38], further incorporated an experience-sharing mechanism grounded in heuristic learning, showing improved generalization across diverse problem sets. However, traditional heuristic approaches often struggle with large and complex state spaces, leading to unstable performance. To overcome these limitations, recent works have introduced MCTS as a more robust heuristic method [39], using extensive simulation and a reward mechanism based on single-step inference success rates to guide better theorem selection sequences.

In parallel, reinforcement learning has gained traction in the field of geometric theorem proving due to its ability to learn inference policies from feedback rather than static labels [40], more closely resembling human reasoning processes. Compared to manually designed heuristics, RL provides enhanced adaptability and generalization, particularly in complex problem settings. For example, GeoDRL [41] employs a Deep Q-Network to train agents to minimize unnecessary inference paths and identify optimal reasoning sequences. E-GPS [42] integrates bottom-up data generation via RL with a top-down solving

strategy, while rStar-Math [43] proposes a Q-value-based Procedural Preference Model to boost search efficiency. Despite promising progress, the application of deep reinforcement learning in geometric reasoning remains underexplored [44–46]. Recent advancements combine DRL with MCTS [47,48] in a game-theoretic framework, leveraging large-scale simulation and pre-trained policy/value networks to achieve higher stability and accuracy in geometric theorem proving.

3. FGeo-ISRL

3.1. FormalGeo System

As a consistent and extensible formal system for plane geometry, FormalGeo aims to address three core challenges: the inconsistent formalization of geometric knowledge, the unreadability of problem-solving procedures, and the lack of standardized approaches to geometric proof. Grounded in geometric ontology, it abstracts planar geometric knowledge into two dimensions: quantity and form, as well as static properties and dynamic processes. It covers not only topological structures such as points, lines and circles, and metric information including lengths and angles, but also the static attributes of geometric entities and the dynamic processes of theorem derivation, establishing a knowledge base comprising 88 geometric predicates and 196 geometric theorems. The core strengths of the system lie in its uniformity and traceability: it eliminates ambiguities in natural language and ambiguities in diagrammatic representations via formal languages. All supporting datasets are equipped with complete formal annotations and verifiable theorem sequences. Meanwhile, standardized system interfaces are provided to support the construction of neuro-symbolic systems.

FormalGeo incorporates three core types of languages. Geometry Definition Language (GDL), as the metalanguage for knowledge base construction, is divided into predicate definition and theorem definition modules. It decouples solver code from geometric knowledge and allows users to extend novel geometric knowledge; its proposed abstract hierarchical design can reduce the length of reasoning paths. Condition Declaration Language (CDL) serves as the interface for specifying concrete geometric problems. It characterizes the topological structure of diagrams, explicit and implicit given conditions, and solving objectives through construction statements (Construction CDL), condition statements (Text CDL/Image CDL), and goal statements (Goal CDL) respectively, achieving the unified formal conversion of natural language, graphical information, and problem-solving goals. Geometry Predicate Logic (GPL), acting as the core logical layer and the core logical language for the execution engine, defines geometrically contextual logical operations including constrained Cartesian product, set union, and set difference. Through a three-stage pipeline—parsing, normalization, and execution—it integrates symbolic logical reasoning and numerical algebraic computation, guaranteeing the efficiency and traceability of reasoning.

The Formal Geometry Problem Solver (FGPS) is a modular solver built upon the FormalGeo framework, consisting of five components: a parser, a core control module, a core solving engine, a data module, and external interfaces. It supports interactive verification and automated solving, and can generate readable and traceable problem-solving procedures. FGPS implements two fundamental search methods: The forward search starts from the initial given conditions, deriving new conditions by matching and applying theorems until the goal is satisfied. The backward search begins from the problem goal, decomposes it into subgoals, and recursively checks whether these subgoals are satisfied by the given conditions or have been proven. To ensure human readability of the system's problem-solving process, the FormalGeo system integrates a rule-mapping-based grammar mapper for converting problem-solving procedures into natural language.

In prior work, the translation from natural language to formal language was primarily conducted manually. Nevertheless, we have achieved substantial progress in automating formalization using large language models in our subsequent work [46].

3.2. Inference Environment

We model the solution of geometry problems as a Markov Decision Process (MDP) with the help of the FormalGeo environment. In the MDP, we utilize a hypergraph structure [49] to represent the geometric problem-solving process as a state transition system: each node in the graph corresponds to a state (including the initial conditions of the problem or intermediate conditions derived by applying the premises of theorems), while the edges between nodes represent the application of theorems, enabling the transition from one state to another. We further represent the problem-solving process as a process of theorem application, and transform it into relational reasoning, algebraic computation, and logical operations to ensure the correctness of the problem-solving process. In this system, if the conditions accumulated in a certain state node meet the target conditions, the problem is determined to be solved successfully. The sequence of theorem applications from the initial state to the terminal state constitutes an effective proof path.

Regarding search methods, the FormalGeo system implements a basic forward search and backward search. The forward search proceeds from given conditions and achieves problem-solving by matching the premises of theorems, whereas the backward search performs matching against the conclusions of theorems after decomposing the goal. Nevertheless, both approaches yield relatively rigid search paths with insufficient guidance, lead to an excessively large search space, and fail to effectively utilize the intrinsic connections between conditions and goals. To tackle this problem, this study proposes a novel search method named inverse search, which adopts the merged state of given conditions and solving objectives as the initial state. Figure 1 presents a schematic comparison of the inverse search proposed in this work with forward search and backward search.

Unlike these two basic search methods, inverse search is not merely based on pure goal decomposition or state progression. Instead, it initiates from the merged state of conditions and goals, and iteratively incorporates matching theorem premises until the goal state is satisfied. By merging conditions and goals, inverse search leverages human intuition about the correlations between problem conditions and objectives [50] to provide search heuristics and directions, which further shrinks the search space and enhances search efficiency.

In this environment, the starting state S_0 for the inverse search is defined by the set of initial problem conditions and the target goal. By applying theorems, intermediate conditions are incrementally added and integrated with the current state, resulting in updated states that progressively advance the reasoning process. The reasoning is deemed successful once the accumulated condition set of a state satisfies the final goal condition set G. The overall architecture of FGeo-ISRL can be seen in Figure 2. The reasoning system of FGeo-ISRL is a neuro-symbolic framework that enables automated inverse solving of geometric problems through deep reinforcement learning and MCTS. Geometric problems are first processed by the FormalGeo system, then passed to the system agent for reasoning. Within the agent, the MCTS-enhanced policy network specifies the theorem to be executed, and the parameter adapter performs parameter instantiation for the selected theorem. After the reasoning system completes execution and updates the state, the value network is used to score the updated state to determine whether the reasoning path is correct. Once the reasoning process is finished, the reasoning results are transmitted to the FormalGeo system for structured output.

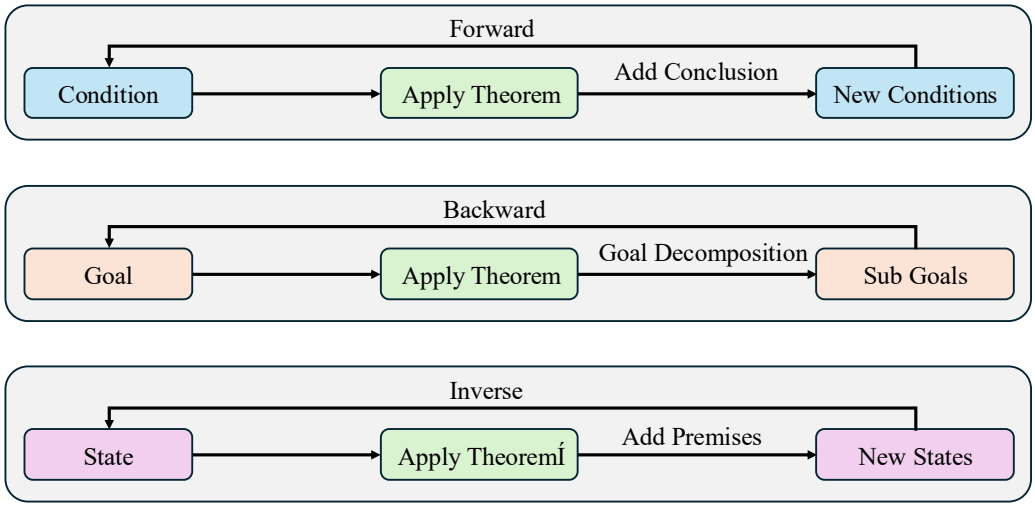

Figure 1. Differences among various search methods.

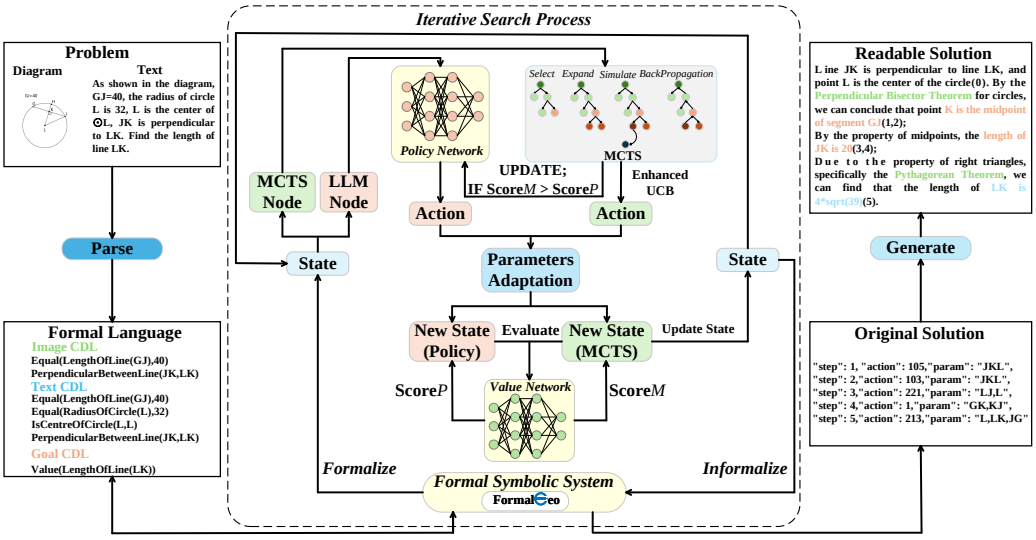

Figure 2. Overall architecture of the FGeo-ISRL system.

3.2.1. Reward

Due to the complexity of inverse reasoning and the need for fine-grained feedback to accelerate the learning process, we design an immediate reward function [51] based on the value network (see Section 4.3).

The value network $V(S_t)$ evaluates the potential value of each state, representing the likelihood of reaching the target state from that state. We define the immediate reward function r_t as follows in Equation (1). This equation first applies a linear transformation to the value vector and compresses the network output into the interval $[0, 1]$ via the sigmoid function; it then compares the normalized values of the subsequent and current states in the sigmoid space to quantify the value increment introduced by a single theorem application, thereby providing continuous and fine-grained positive or negative feedback signals to both the policy network and MCTS.

Since the value network is initialized through pre-training to facilitate a cold start, its state evaluation capability stems largely from the high-quality human-annotated data in FormalGeo7K. Simultaneously, due to the inconsistent number of proof steps and significant fluctuations in theorem sequence lengths, cumulative rewards vary drastically. To mitigate the reward jumping issue, we implement real-time mean normalization, ensuring that the

reward signals consistently remain within a similar order of magnitude. This stabilization allows for smoother and more consistent optimizer step sizes during the learning process.

$$r_t = \sigma\left(\mathbf{w}^{\mathsf{T}}\, V(S_{t+1}) + b\right) \;-\; \sigma\left(\mathbf{w}^{\mathsf{T}}\, V(S_t) + b\right) \tag{1}$$

To facilitate convergence toward the ultimate problem-solving objective, an additional reward of 1 is assigned whenever $S_{t+1} = G$. This hybrid reward scheme, integrating both terminal bonuses and immediate rewards, ensures search flexibility while maintaining a strict focus on proof completion. Crucially, this design prevents the model from disproportionately optimizing for intermediate reward signals rather than the final goal. Such a shaping term effectively steers the model toward the terminal solution state.

3.2.2. Action Space

The initial theorem library consists of 196 fundamental geometric theorems and axioms. However, the FormalGeo system's rigorous requirements for geometric topological structures necessitate that theorems, which are cognitively singular in human reasoning, be partitioned into multiple branches to accommodate diverse topological configurations.

Furthermore, since geometric figures inherently exhibit symmetry, applying the same theorem in different orientations requires distinct parameter sequences. To ensure correct theorem execution, we expand the action space by defining specific branches of a theorem as independent actions. For example, when proving parallel lines via equal corresponding angles, the system must precisely control the counterclockwise definition of angles; different directions of the transversal line lead to varying geometric boundaries. By fixing these distinct branches and defining them as separate new theorems, we systematically expand the action space to 234 branch theorems. Consequently, each state node possesses 234 candidate action choices, providing the model with the necessary granularity for automated reasoning, as illustrated in Figure 3.

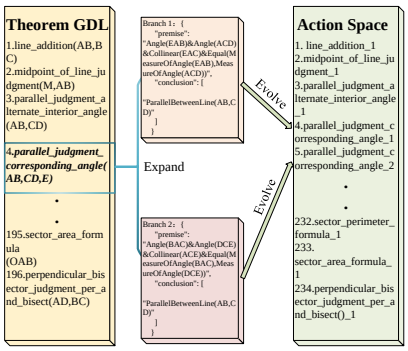

Figure 3. Schematic diagram of theorem-based search space expansion.

3.3. Monte Carlo Tree Search

Monte Carlo Tree Search (MCTS) achieves an effective balance between exploration and exploitation by conducting efficient simulations and evaluations within the state-action space, while integrating heuristic policies with value estimation. It has achieved breakthrough results in strategic games such as Go [52,53], chess, and Shogi [54], establishing itself as a cornerstone method in the integration of reinforcement learning and search algorithms. It accumulates experience through extensive simulations and often achieves good results when facing generalized environments that have not undergone targeted training [55]. This paper leverages the characteristic of MCTS in conducting extensive simulations and the immediate rewards provided by the value network to enhance the policy network. This enhancement improves the performance of the policy network, thereby

enabling the discovery of a more effective sequence of theorems. The comparison between naive MCTS and the improved MCTS described above is shown in Figure 4.

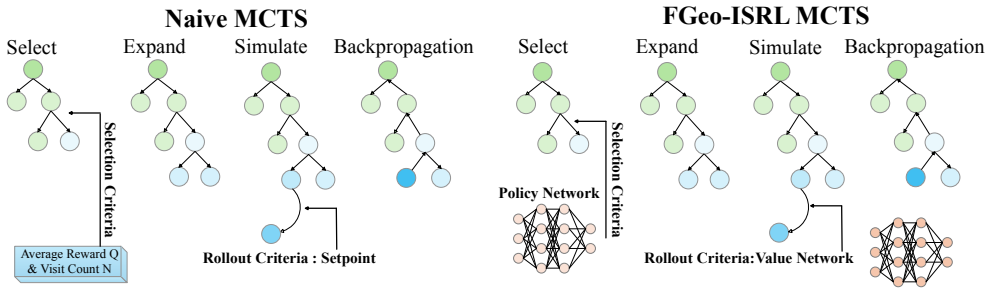

Figure 4. Comparison of the differences between naive MCTS and MCTS employed in our system.

While the traditional application environments of MCTS are often simple, its application to geometric problem-solving is rather complex and challenging. Therefore, we will elaborate on the specific definitions and applications of each stage of MCTS in geometric problem-solving to facilitate readers' understanding. The implementation of MCTS consists of the following four stages:

Select: Starting from the root node, the optimal action a_t is selected. The selection of the optimal action is based on an enhanced Upper Confidence Bound (UCB) equation, which balances both exploration and exploitation. The enhanced UCB equation is as follows in Equation (2).

$$a_t = \arg\max_{a \in A} \left[Q(S_t, a) + c \cdot P(S_t, a) \sqrt{\frac{\log N(S_t)}{N(S_t, a) + \epsilon}} \right] \tag{2}$$

The improved UCB equation incorporates the prior probability $P(S_t, a)$ from policy network, making the search no longer purely driven by random exploration, but guided to some extent by the policy network, thereby reducing ineffective exploration, enhancing both the performance and efficiency of the search process.

Expand: Once an action a_t is selected, the algorithm executes it to create the new node S_{t+1}. When an incompletely expanded node is reached, all admissible legal actions are expanded according to their prior probabilities $P(S_{t+1}, a_t)$. The yet-unexecuted action a_{t+1} is then inserted into the search tree.

Finally, the statistical counters for each newly expanded action are initialized with $Q(S_{t+1}, a) = 0$, $N(S_{t+1}, a) = 0$, and the stored prior $P(S_{t+1}, a)$.

Simulate: The expanded state node is simulated. We set a maximum search step limit and define the number of simulations as the setpoint, with the value network evaluating the state value to guide the simulation process.

For each simulation, starting from the current state S_{t+1}, actions are selected and executed until either the maximum step limit is reached or the target state is achieved. The immediate reward for each state transition is given by r_t. This reward is accumulated to compute the total simulated return G_t. If the target state is reached during the simulation, an additional terminal reward is added.

Backpropagation: For the accumulated rewards and visit counts obtained during the simulation phase, the search tree is updated through backpropagation. The simulation result G_t is fed back to the search tree, updating the average reward $Q(S_t, a_t)$ and visit count $N(S_t, a_t)$ for each state-action pair along the search path. The visit count $N(S_t, a_t)$ is simply incremented by 1, while the update of the action-value $Q(S_t, a_t)$ follows Equation (3).

$$Q(S_t, a_t) \leftarrow \frac{Q(S_t, a_t)\, N(S_t, a_t) + G_t}{N(S_t, a_t) + 1} \tag{3}$$

3.4. Value Network

In the FGeo-ISRL system, the value network (VN) serves as one of the core components, responsible for evaluating the current state and playing a critical role within the MCTS process. By incorporating the value network into MCTS, the system can effectively replace a large number of redundant simulations, thereby improving the quality of the search.

Specifically, the value network is designed not to directly exploit BERT's robust classification capabilities, but rather to leverage its bidirectional attention mechanism. BERT captures the intricate logical correlations between geometric entities, shifting its role toward deep semantic understanding of states, which enables the system to determine whether the state evolution aligns with the expected reasoning path. On this basis, we model state evaluation as a regression task, requiring the network to output a continuous score $V(S_t) \in [0, 1]$. We adopt Mean Squared Error (MSE) as the primary loss function, as it directly measures the numerical distance between predicted scores and actual cumulative rewards, thereby compelling the model to learn a precise value distribution. The schematic diagram of the value network process is illustrated in Figure 5.

To address the challenges of a vast state space and the scarcity of high-difficulty samples, we incorporate L2 regularization to constrain model parameters and mitigate overfitting. This ensures that the trained value network maintains robust generalization across geometric problems ranging from L1 to L6 difficulty levels.

The training of the value network is based on full proof trajectories from the FormalGeo7K dataset. The detailed expression of the loss function is shown in Equation (4).

$$\mathcal{L}(\theta) = \mathbb{E}_{(S_t, r_t) \sim \mathcal{D}} \left[\left(V_\theta(S_t) - r_t \right)^2 \right] + \lambda \left\| \theta \right\|_2^2 \tag{4}$$

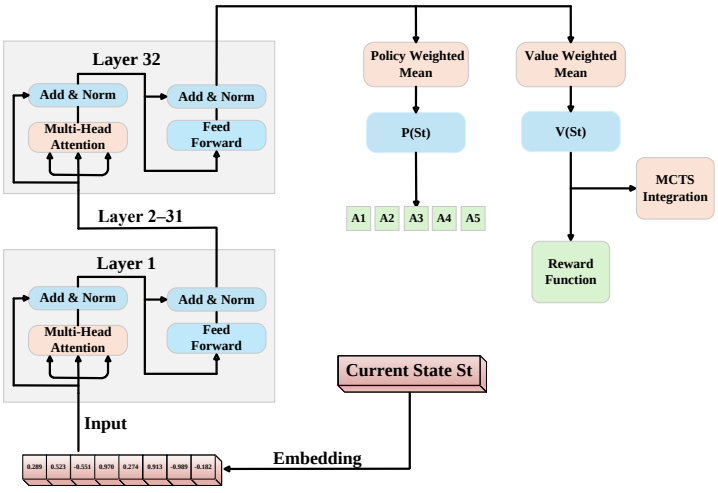

Figure 5. Flowchart of the value network and the policy network fine-tuned from the BERT-Base model.

The cumulative reward is computed from simulation trajectories: if a trajectory successfully reaches the goal state, it receives a terminal reward of 1; otherwise, the reward is accumulated based on changes in estimated state value. Through this training process, the value network gradually learns the distribution of potential value across different states and accurately captures the contribution of each theorem application to achieving the

final goal. Once trained, the value network is directly integrated into the reward function computation, ensuring the stability and reliability of the reward signal.

3.5. Policy Network

As a core component of the FGeo-ISRL system, the policy network (PN) takes the current state S_t as input and outputs a probability distribution over 234 candidate theorem branches, denoted as $\pi(a_t \mid S_t)$. We adopt a BERT-base model as the backbone architecture to achieve an optimal balance between prediction performance and computational efficiency. Since each state in the annotated data corresponds to a unique ground-truth theorem action, we formulate theorem selection as a classic multi-class, single-label classification task. The policy network is trained using a two-stage optimization strategy, beginning with supervised learning on the FormalGeo7K dataset. We optimize the network using the multi-class cross-entropy loss, as defined in Equation (5). This objective is inherently aligned with the Single-Step Theorem Accuracy (SSTA) metric (see Section 4.1), ensuring that the minimization of loss directly translates into the enhancement of individual reasoning precision. The schematic diagram of the policy network process is illustrated in Figure 5.

The multi-class cross-entropy loss function, leveraging the standardized annotations from FormalGeo7K, facilitates the model's transition from a general-purpose pre-trained language model to a specialized geometric reasoning policy network. This fine-tuning process empowers the network to rapidly assimilate human expert logic, effectively circumventing the cold-start problem prevalent in pure reinforcement learning. Moreover, the supervised learning stage ensures the efficient utilization of high-quality state-action pairs, establishing a robust foundation for subsequent long-sequence optimization. This strategic methodology significantly enhances sample efficiency and addresses the challenges of high labeling costs and convergence difficulties typically associated with training complex reasoning agents from scratch. The description of the fine-tuning algorithm for the value network and policy network is detailed in Algorithm 1.

Algorithm 1 Supervised Pretraining for Policy and Value Networks

Require: Human expert proof dataset $\mathcal{D} = \{(S_t\, a_t, v_t)\}_{t=0}^{K}$
1: Initialize Policy Network π_θ and Value Network V_ϕ with random weights
2: **while not** converged **do**
3: Sample a mini-batch of transitions (S, a, v) from $\mathcal{D}$
4: Compute value loss $\mathcal{L}(\theta_1)$
5: Compute policy loss $\mathcal{L}_{SL}(\theta_2)$
6: Update $\theta_1 \leftarrow \theta_1 - \eta \nabla_\theta \mathcal{L}(\theta_1)$
7: Update $\theta_2 \leftarrow \theta_2 - \eta \nabla_\theta \mathcal{L}_{SL}(\theta_2)$
8: **end while**
9: **return** Pretrained networks V_{θ_1} and π_{θ_2}

$$\mathcal{L}_{\mathrm{SL}}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{a \in \mathcal{A}} \mathbb{I}\{a = a_t^{(i)}\} \log \pi_\theta(a \mid S_t^{(i)}) \tag{5}$$

Subsequently, in the reinforcement learning stage designed to enhance the policy network via MCTS, the objective of geometric problem-solving shifts toward achieving a complete end-to-end proof, rather than merely ensuring the correctness of individual theorem applications. Although the supervised learning phase optimizes single-step prediction accuracy through cross-entropy loss, it lacks sufficient handling of the dependencies inherent in long-sequence reasoning paths; a locally correct action does not necessarily lead to a successful final solution. The description of the algorithm is shown in Algorithm 2.

To maximize the cumulative reward, we employ Policy Gradient methods to optimize the network. The loss function, as defined in Equation (6), adopts the form of negative cumulative-reward-weighted log-probability. By utilizing the cumulative reward of the entire reasoning trajectory as a weighting factor, this design directly optimizes the final solving success rate, thereby aligning the loss function perfectly with the ultimate task goal of geometric reasoning.

The core design of this MCTS-enhanced policy network establishes a virtuous cycle of exploration–optimization–exploration. MCTS explores superior theorem paths that surpass the current policy through extensive simulations, generating high-quality trajectories with substantial cumulative rewards. Subsequently, the policy gradient loss leverages these trajectories to drive iterative updates of the policy network. This paradigm significantly reduces the reliance on large-scale human-annotated intermediate reasoning data, effectively lowering supervision costs.

From the perspective of technical compatibility, the action space in geometric reasoning consists of 234 discrete theorem branches. Policy gradient methods optimize the policy directly, thereby avoiding the biases typically introduced by traditional Value Function Approximation. Furthermore, the log-probability formulation ensures the stability of parameter updates. Consequently, this approach possesses a natural suitability for long-sequence decision-making tasks within discrete action spaces, providing robust mathematical support for handling complex geometric proofs.

Algorithm 2 Value-Guided & MCTS-Enhanced Joint Optimization

Require: Pretrained networks π_θ and V_ϕ, Training problem set $\mathcal{P}$
for each problem in $\mathcal{P}$ **do**
  $S_0 \leftarrow \mathit{Init}$ (Text_CDL, Goal_CDL)
  $S' \leftarrow$ Init(Goal_State)
  while not $\mathit{StateReached}(S')$ **do**
    $a_M \leftarrow \mathit{MCTS\_Search}(S_t, V_{\theta_1}, \pi_{\theta_2})$ {Explore via MCTS}
    $S_M \leftarrow \mathit{Apply}(S_t, a_M)$, $v_M \leftarrow V_{\theta_1}(S_M)$
    $a_P \leftarrow \arg\max_a \pi_{\theta_2}(a|S_t)$ {Predict via Policy Net}
    $S_P \leftarrow \mathit{Apply}(S_t, a_P)$, $v_P \leftarrow V_{\theta_1}(S_P)$
    if $v_M > v_P$ **then**
      $\mathcal{L} \leftarrow -\log \pi_\theta(a_M|S_t)$ {Adapt to better MCTS action}
      Update $\theta_2 \leftarrow \theta_2 - \eta \nabla_\theta \mathcal{L}_{\mathrm{RL}}(\theta_2)$
      $S \leftarrow S_M$
    else
      $S \leftarrow S_P$
    end if
  end while
end for
return Optimized networks π_θ

$$\mathcal{L}_{\mathrm{RL}}(\theta) = -\mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} r_t \log \pi_\theta(a_t \mid S_t)\right] \tag{6}$$

As MCTS continues to explore and accumulate experience, if a more effective theorem selection strategy is discovered along a new search path compared to the current policy network, the policy network is updated in a delayed manner. Through collaborative optimization involving MCTS feedback and policy gradient updates, the policy network continuously improves the accuracy and robustness of theorem selection.

3.6. Rule-Based Parameters Adaptation

In the process of inverse reasoning for geometric problems, the application of a theorem often requires adding its premises to the current state and updating the reasoning state accordingly. Improper parameter instantiation can lead to inconsistent states or deviated reasoning paths, ultimately affecting the stability and effectiveness of the search. To address this, we design a rule-based parameter adaptation mechanism that dynamically filters and matches the required parameter combinations for a theorem under a given geometric state.

This mechanism leverages the formal structure provided by the Geometric Description Language (GDL), where each theorem and predicate is represented in a structured form. Each branch of a theorem is parsed into constructive constraints, logical constraints, and variable groupings. During reasoning, the system examines the current state, extracts all geometric entities and their relationships, and verifies whether the theorem's premises are satisfied. All valid parameter combinations are then extracted and passed forward to support theorem instantiation and state transition. The detailed parameter adaptation process can be found in Figure 6.

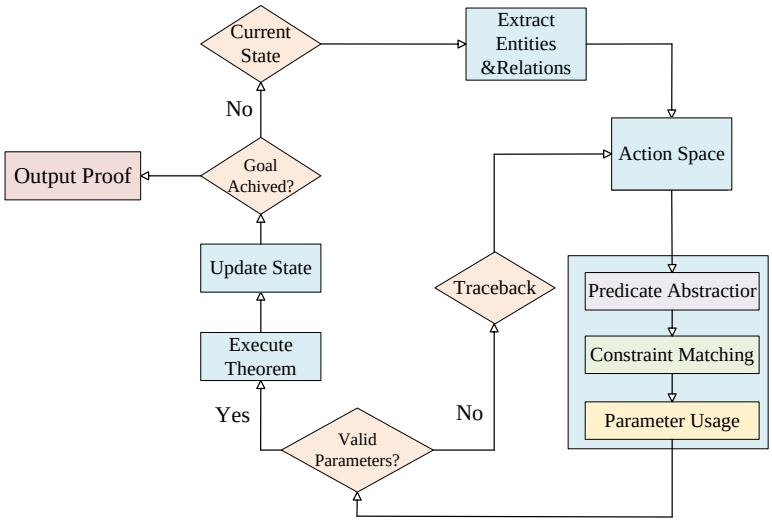

Figure 6. Overall structure of the parameter-adaptation mechanism.

More specifically, the parameter adaptation mechanism consists of three core stages.

First, predicate abstraction and expansion. In the semantic structure of a theorem, both constructive and relational predicates are abstracted and unified. Based on the current problem state, these predicates are expanded into a complete set of geometric entities.

Second, logical constraint matching. The FormalGeo system uses the semantic constraints defined in the theorem structure to perform logical matching against the existing state. It automatically verifies which parameter combinations satisfy the conditions for theorem application.

Third, parameter instantiation. All variable bindings that meet the logical constraints are formatted into standard input structures and passed as valid parameters for the theorem. This ensures stable and consistent state transitions.

This mechanism guarantees both structural legality and semantic consistency in theorem application. It effectively reduces the risk of generating invalid states during the inverse search and significantly improves the robustness and generalization capability of the reasoning process. Compared to traditional exhaustive parameter enumeration, this approach is more efficient at logic matching and highly scalable, making it a critical component in our neuro-symbolic framework for automated geometric theorem proving.

4. Experiments

4.1. Evaluation Metrics

We use Single-Step Theorem Accuracy (SSTA) and Geometry Problem-Solving Accuracy (GPSA) as the evaluation metrics to assess different methods. Their definitions are as follows:

SSTA: Given the current problem state S_t, the model predicts the theorem t_i required to transition to the next state S_{i+1}. This metric evaluates the model's single-step theorem prediction accuracy during the problem-solving process.

GPSA: For a given problem, the model must generate an ordered list of theorems L_T, together with the parameter assignments determined by the parameter-adaptation mechanism. The parameterized theorem list is fed into the symbolic module for verification, determining whether the problem has been solved or not. This metric evaluates the model's overall capability to solve geometric problems.

This section presents the performance analysis of our FGeo-ISRL framework on the FormalGeo7K dataset, evaluating two key metrics: Geometry Problem-Solving Accuracy and Single-Step Theorem Accuracy. Additionally, we conduct ablation studies on different modules of FGeo-ISRL.

4.2. Datasets and Data Processing

By utilizing FormalGeo formal language, we collected widely used datasets and online resources to construct a newly verified dataset, FormalGeo7K, which is validated through the reasoning environment. The statistics collected from FormalGeo7K cover various types of problems, including geometric problems related to angle, length, perimeter, area, and other aspects. We categorize the difficulty levels of the the problems based on the lengths of theorem sequences, measured L_1 to L_6.

Building upon the FormalGeo7K dataset, we utilize FormalGeo to construct state-action pairs (S_t, a_t) from geometric problems, which serve as training corpora for both the value network and policy network. Subsequently, we partition the dataset into training, testing, and validation sets with a 0.7:0.15:0.15 ratio according to problem types and difficulty levels. Specifically, the training set contains 26,031 state-action pairs, while both the testing and validation sets consist of 5578 pairs each. Finally, we train and deploy the preprocessed corpus locally using a BERT-base backbone to obtain a supervised-learning-based value network and initial policy network.

4.3. Benchmark Methods

We evaluate several benchmark methods on the FormalGeo7K dataset, including purely symbolic methods (forward search and backward search); neural language models integrated with the FormalGeo symbolic system, such as T5-small with FGeo and BART-base with FGeo; pure neural methods including DeepSeek-v3; neuro-symbolic systems specifically designed for GPS, such as Inter-GPS, NGS, and DualGeoSolver; and our proposed neuro-symbolic system, FGeo-ISRL. The detailed results are shown in Table 1. The table shows that the performance decreases as the problem difficulty level increases from L_1 to L_6. FGeo-ISRL performs the best in both total performance and all difficulty levels, and its decreasing trend is also better than that of other methods.

The search-based baselines include forward search using a Random Search strategy and backward search employing Breadth-First Search, both constrained to a maximum search tree size of 15 to assess basic heuristic efficiency. For the LLM-based baselines, we utilize T5-small (60 M), BART-base (140 M), and DeepSeek-V3 via API calls. These models are evaluated in a zero-shot setting using prompt engineering without further fine-tuning to establish a performance floor for general-purpose models.

For the specialized solvers, Inter-GPS employs a Transformer-based theorem predictor (TP) with 6 layers, 12 attention heads, and a 768-dimensional hidden state. It follows a Predict + Low-first search strategy with a maximum of 100 steps. To generate forward proof trajectories, the system performs 100 attempts for each problem with a maximum sequence length of 20, utilizing the Adam optimizer with a learning rate of 0.01 for a maximum of 30 epochs. NGS incorporates a multimodal approach with a differential learning rate: a base learning rate of 1×10^{-3} and 1×10^{-5} for the image encoder pre-training. The model is trained using the Adam optimizer with a batch size of 32 for approximately 100 epochs. During inference, it employs a beam search width of 10 by default, which can be increased to 100 for accuracy. DualGeoSolver implements a sophisticated segmented learning rate strategy, specifically 2×10^{-5} for the RoBERTa text encoder, 1×10^{-5} for the multimodal fusion module and the Goal Generation Module (GGM), and 1×10^{-3} for all other modules to balance knowledge retrieval and reasoning. The training process uses the Adam optimizer with a batch size of 32 for a total of 100 epochs, and a beam search width of $B = 10$ is utilized during testing.

Table 1. The performance of different benchmark methods on the FormalGeo7K dataset.

Method	Total	L_1	L_2	L_3	L_4	L_5	L_6
Forward Search [35]	39.71	58.47	41.01	34.16	16.4	5.45	4.79
Backward Search [35]	35.44	66.43	34.98	11.78	6.56	6.09	1.03
T5-small [56] with FGeo	36.14	49.90	34.84	34.59	23.57	8.06	3.33
BART-base [57] with FGeo	54.00	73.90	56.12	50.38	26.75	16.13	8.33
DeepSeek-v3 [58]	60.79	75.99	56.38	63.91	43.31	32.26	28.33
Inter-GPS [26]	60.50	76.20	63.30	60.90	39.49	17.74	15.00
NGS [27]	62.60	62.22	64.97	72.79	57.47	56.41	36.59
DualGeoSolver [59]	62.11	62.96	67.80	65.44	60.92	53.85	34.15
FGeo-ISRL	83.14	98.20	96.35	88.31	78.21	60.22	29.21

4.4. Experiments Results

The core architecture of the FGeo-ISRL system is co-driven by policy and value networks, both of which utilize a BERT-base model as the primary feature extractor to achieve balance between inference performance and computational efficiency. All fine-tuning tasks are implemented via Low-Rank Adaptation (LoRA) and processed on an NVIDIA RTX 4090 24 GB GPU. In terms of training configuration, both networks share a unified learning rate of 1×10^{-5} and a batch size of 16, with ReLU activation functions introduced in the hidden layers to incorporate non-linearity. The entire training process is executed by the AdamW optimizer for a maximum of 100 epochs, complemented by an early stopping mechanism and a learning rate scheduling strategy to ensure stable convergence via step-size reduction when the validation loss plateaus.

During the reasoning phase, each MCTS decision cycle performs a variable number of simulations determined dynamically by the value network. The exploration constant c is set to 1.5 to effectively weight the exploitation of known high-value paths against the exploration of unvisited nodes. To maintain controllability and prevent infinite loops, the system strictly limits the maximum inference depth to 30 steps per problem. Within the main search loop, the system expands the Top-10 candidate actions with the highest predicted probabilities to ensure a broad yet focused search space. Furthermore, a built-in exception termination mechanism is triggered if the policy network predicts the highest-probability action to be 0 for 3 consecutive times, allowing the system to preemptively halt unproductive search trajectories.

We evaluate our approach on the test set using MCTS, the value network, and the enhanced policy network. The testing procedure is outlined in Algorithm 3, the results are summarized in Table 2, and we also present the results in the form of a heatmap in Figure 7. Among the methodologies, FGeo-ISRL demonstrates superior performance, yielding a GPSA of 83.14%. This result surpasses the majority of existing approaches, including FGeo-TP and FGeo-DRL. While most approaches exhibit significant performance degradation when handling higher difficulty levels (particularly L_5 and L_6), FGeo-ISRL maintains robust solving capabilities, attaining GPSA scores of 60.22% and 29.21% for L5 and L6 difficulty levels, respectively. These results demonstrate the method's exceptional competence in addressing complex geometric problems.

Table 2. Experimental results for different methods in GPSA and SSTA. L_1 to L_6 represent performance metrics across levels. Black bold marking denotes the best result. Meanwhile, this experiment also refers to other FormalGeo series solving methods and conducts technical analysis on them.

Method	GPSA	SSTA	L_1	L_2	L_3	L_4	L_5	L_6
MCTS	39.11	65.22	67.51	53.22	39.31	28.12	17.53	11.82
PN	53.64	59.48	76.51	78.21	48.92	45.13	34.22	9.04
PN-RS	57.91	65.49	78.51	79.52	50.21	46.53	35.82	10.51
PN-DFS	60.03	67.63	82.11	82.23	53.42	49.71	39.02	13.51
PN-BFS	56.52	69.63	85.22	88.81	58.81	52.12	38.41	13.55
PN-BS	62.51	74.17	80.31	84.61	56.62	52.31	41.61	16.02
PN-MCTS	64.49	72.45	89.12	87.51	65.51	51.22	40.51	14.81
PN-VN	73.02	80.15	91.51	87.21	68.11	59.02	42.31	14.51
FGeo-TP-RS [45]	80.86	-	96.43	85.44	76.12	62.26	48.88	29.55
FGeo-DRL-BS [44]	80.85	-	97.61	91.88	70.82	57.55	36.17	27.59
FGeo-HGNet [49]	**88.36**	-	96.24	91.76	87.59	**82.17**	56.45	**56.67**
FGeo-ISRL	**83.14**	**90.20**	**98.20**	**96.35**	**88.31**	78.21	**60.22**	**29.21**

Algorithm 3 Inference via MCTS-Enhanced Policy

Require: Optimized networks π_θ and V_ϕ, Unseen test problem
$S \leftarrow Init(\text{Text\_CDL}, \text{Goal\_CDL})$
while not $GoalReached(s)$ **and** within step limit **do**
  for iteration $i = 1$ **to** N **do**
    Select: Traverse tree from s to leaf node S_{leaf} using UCB and π_θ
    Expand: Expand legal actions under S_{leaf} guided by prior $\pi_\theta(a|S_{leaf})$
    Evaluate: Compute node value via $V_\phi(S_{leaf})$ (and/or rollout)
    Backup: Update visit counts and accumulated rewards ω to root S
  end for
  $a^* \leftarrow$ Action with maximum visit count from S
  $S \leftarrow Apply(S, a^*)$
end while
return *Proof Solution* **or** *Failure*

The improvement from the naive MCTS approach is limited. The first notable performance gain is achieved when incorporating an LLM-based policy network directly into the solving process. However, due to the LLM's lack of geometric intuition and insufficient handling of geometric constraints, the enhancement from simply combining traditional search methods with a single policy network remains constrained. This suggests that relying solely on a single search strategy or online update mechanism has inherent limitations in boosting model performance.

Further improvement is observed when integrating both a policy network and value network. The combined PN + VN approach significantly outperforms the standalone PN, highlighting the critical role of the VN in providing more effective guidance by evaluating

the value of each decision step. The complete FGeo-ISRL model (PN + VN + MCTS) achieves the best overall performance, demonstrating the complementary nature of PN's local action precision and VN's global value assessment, while MCTS further amplifies their synergistic effect.

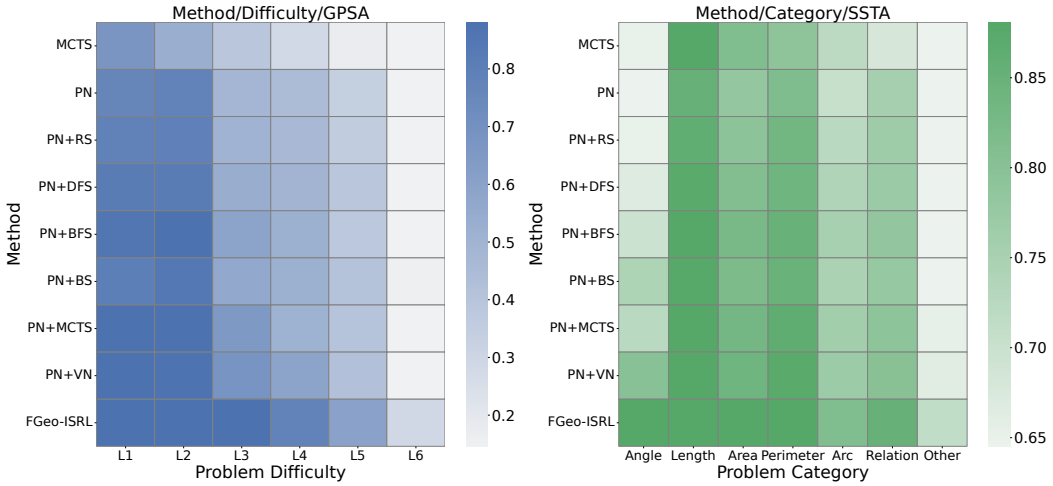

Figure 7. GPSA performance of different methods across various question difficulty levels and the corresponding SSTA results across diverse question types.

Compared with other FormalGeo-series solving methods, such as FGeo-DRL which also adopts deep reinforcement learning for problem-solving, this method performs inferiorly to FGeo-ISRL due to the lack of an explicit supervision mechanism similar to a value network. In contrast, FGeo-HyperGNet, which fully embeds the state transition process through a hypergraph neural network, outperforms our method in overall solving efficiency and long-sequence solving. This also points out a direction for future improvements: how to integrate reinforcement learning to better describe the state transition process, thereby enhancing solving efficiency and capability.

4.5. Ablation Study and Discussion

The ablation study consists of two parts. The first part specifically evaluates the contribution of each core module in the FGeo-ISRL framework to the geometric problem-solving performance. As shown in Table 3, this study designs three sets of comparative experiments:

Table 3. Performance comparison of FGeo-ISRL with different ablations.

Method	Top K	GPSA	SSTA
FGeo-ISRL	3	74.86	80.62
	5	78.32	87.93
	10	80.69	88.28
-w/o MCTS	3	70.11	80.11
	5	72.02	80.55
	10	72.88	82.15
-w/o VN	3	49.20	60.27
	5	55.12	63.45
	10	60.35	70.18

Our observations reveal that as Top K increases, both GPSA and SSTA metrics improve across all methods. To assess MCTS's enhancement effect on the policy network, comparing

Experiment 1 and Experiment 2 demonstrates that under identical Top K conditions, methods without MCTS augmentation exhibit lower GPSA and SSTA than FGeo-ISRL. This confirms MCTS's capability to explore more effective theorem selections. To evaluate the value network's role, the comparison between Experiment 1 and Experiment 3 shows that without the value network's guidance, the system experiences significant performance degradation in problem-solving capability. This verifies that the value network's guidance substantially improves solving success rates, and the synergistic optimization mechanism combining local policy, global evaluation, and search strategy yields significant effectiveness.

The second part demonstrates the extent to which the number of training epochs and the single-step reasoning time of the FGeo-ISRL model influence the problem-solving results. As shown in Table 4, with the increase in training epochs and single-step reasoning duration, the performance improvement of SSTA hits a bottleneck, and the growth rate of GPSA also gradually slows down. This indicates that in the early stages of model training, enhancing computational instances helps to improve problem-solving capabilities. However, as parameters increase, the mere scaling of computational resources may have limited effectiveness in improving problem-solving outcomes.

Table 4. The problem-solving performance of different versions of FGeo-ISRL under various parameters.

Model Version	Epochs	Step Time	GPSA	SSTA
FGeo-ISRL-V1	20	300	74.49	88.30
FGeo-ISRL-V2	50	300	78.31	90.20
FGeo-ISRL-V3	50	600	83.14	90.20

4.6. Case Analysis

For the final results, we selected representative cases in Figure 8 for in-depth analysis. GT represents Ground Truth, and PT represents Predicted Theorem. Taking Case 1 as an example, we can see that FGeo-ISRL is highly accurate in solving problems with long theorem sequences, with a high hit rate in selecting and applying theorems, and the order of theorems is accurate. However, the model also has certain limitations. Taking Case 2 as an example, the selection of the theorem is correct but the parameters cannot be matched, leading to the failure of solving. This indicates that the subsequent work should improve the adaptation of theorem parameters. Taking Case 3 as an example, in solving extremely difficult problems, due to insufficient training samples, the solving ability is limited and the theorem sequence does not converge. Therefore, future work can refine effective data augmentation.

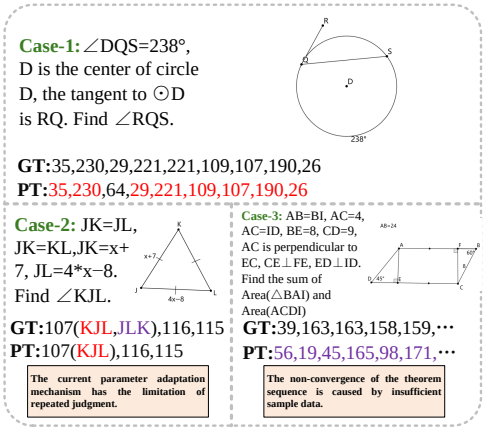

Figure 8. Typical case analysis. Case 1 is a positive case; Case 2 and Case 3 are negative cases.

Notably, the best performance in certain categories does not always come from the overall strongest model. For instance, FGeo-ISRL-V1 achieves the highest accuracy in L2, while FGeo-ISRL-V2 performs best in L4. This indicates some degree of training instability, possibly due to the stochastic nature of MCTS or parameter update fluctuations. However, these variations are relatively minor and do not significantly impact the model's overall reliability. The consistent alignment between the metric trends of SSTA and GPSA demonstrates that SSTA can serve as a reliable leading indicator for GPSA performance, and accurate step-by-step theorem application constitutes the fundamental prerequisite for successful end-to-end geometric reasoning.

5. Conclusions

This study proposes FGeo-ISRL, a theorem sequence prediction model that integrates MCTS, heuristic algorithms, and a reinforcement learning framework. In this system, we incorporate a pre-trained natural language model fine-tuned for geometric reasoning tasks as an initial guide, which is subsequently combined with MCTS and a symbolic reasoning-based reinforcement learning environment. Through iterative learning and experimentation on the FormalGeo7K dataset, we validate the system's deductive reasoning capabilities in geometric problem-solving.

Author Contributions: Conceptualization, Y.L. and C.Q.; methodology, Y.L.; software, Y.L.; validation, Y.L.; formal analysis, Y.L., C.Q., Z.H. and X.Z.; data curation, Y.L., Z.H. and C.Q.; writing—original draft preparation, Y.L.; writing—review and editing, Y.L. and T.L.; supervision, T.L.; and funding acquisition, T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (NSFC) grant 12071282.

Data Availability Statement: The project is available at https://github.com/leeyoung628/FGeo-ISRL (accessed on 12 March 2026).

Acknowledgments: We sincerely thank everyone who has contributed to this work.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hilton, P.; Pedersen, J. Symmetry in mathematics. *Comput. Math. Appl.* **1986**, *12*, 315–328. [CrossRef]
2. Shegeva, S.; Goel, A. The Role of symmetry in geometric intelligence. *Balt. J. Mod. Comput.* **2021**, *9*, 260–275. [CrossRef]
3. Čulina, B. An elementary system of axioms for Euclidean geometry based on symmetry principles. *Axiomathes* **2018**, *28*, 155–180. [CrossRef]
4. Rocco, I.; Arandjelović, R.; Sivic, J. End-to-end weakly-supervised semantic alignment. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6917–6925.
5. Tan, H. A brief history and technical review of the expert system research. In *Proceedings of the IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2017; Volume 242, p. 012111.
6. O'Keefe, R.M.; O'Leary, D.E. Expert system verification and validation: A survey and tutorial. *Artif. Intell. Rev.* **1993**, *7*, 3–42. [CrossRef]
7. Zhang, Y.; Zhang, H.; Li, L.; Xing, E. Evaluating step-by-step reasoning through symbolic verification. In Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, 16–21 June 2024; pp. 2984–3002.
8. Yang, S.; Li, X.; Cui, L.; Bing, L.; Lam, W. Neuro-symbolic integration brings causal and reliable reasoning proofs. In Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, NM, USA, 29 April–4 May 2025; pp. 5732–5744.
9. He, P.; Huang, Y.; Sachan, M.; Jin, Z. Uncovering Hidden Correctness in LLM Causal Reasoning via Symbolic Verification. In Proceedings of the NeurIPS 2025 Workshop on CauScien: Uncovering Causality in Science, San Diego, CA, USA, 6 December 2025.
10. Cohen, E.; Beck, C. Empirical analysis of beam search performance degradation in neural sequence models. In *Proceedings of the International Conference on Machine Learning*; PMLR: New York, NY, USA, 2019; pp. 1290–1299.

11. Barker, J.; Korf, R. Limitations of front-to-end bidirectional heuristic search. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 29.
12. Vendrell, J.; Bent, R.; Kia, S. FORWARD: Feasibility Oriented Random-Walk Inspired Algorithm for Radial Reconfiguration in Distribution Networks. In *Proceedings of the 2025 American Control Conference (ACC)*; IEEE: New York, NY, USA, 2025; pp. 4689–4694.
13. Hong, S.; Lee, J.; Park, B.; Alwusaibie, A.A.; Alfadhel, A.H.; Park, S.; Hyung, W.J.; Choi, M.K. Rethinking generalization performance of surgical phase recognition with expert-generated annotations. *arXiv* **2021**, arXiv:2110.11626. [CrossRef]
14. Al Makdah, A.A.; Krishnan, V.; Pasqualetti, F. Learning Lipschitz Feedback Policies From Expert Demonstrations: Closed-Loop Guarantees, Robustness and Generalization. *IEEE Open J. Control Syst.* **2022**, *1*, 85–99. [CrossRef]
15. Saad, E.M.; Blanchard, G. Fast rates for prediction with limited expert advice. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 23582–23591.
16. Lee, S.; Sim, W.; Shin, D.; Seo, W.; Park, J.; Lee, S.; Hwang, S.; Kim, S.; Kim, S. Reasoning abilities of large language models: In-depth analysis on the abstraction and reasoning corpus. *ACM Trans. Intell. Syst. Technol.* **2024**, *16*, 137. [CrossRef]
17. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24824–24837.
18. Webb, T.; Holyoak, K.J.; Lu, H. Emergent analogical reasoning in large language models. *Nat. Hum. Behav.* **2023**, *7*, 1526–1541. [CrossRef]
19. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6000–6010.
20. Orrù, G.; Piarulli, A.; Conversano, C.; Gemignani, A. Human-like problem-solving abilities in large language models using ChatGPT. *Front. Artif. Intell.* **2023**, *6*, 1199350. [CrossRef]
21. Xu, S.; Luo, Y.; Shi, W. Geo-LLaVA: A Large Multi-Modal Model for Solving Geometry Math Problems with Meta In-Context Learning. In Proceedings of the 2nd Workshop on Large Generative Models Meet Multimodal Applications, Melbourne, Australia, 28 October–1 November 2024; pp. 11–15.
22. Gao, J.; Pi, R.; Zhang, J.; Ye, J.; Zhong, W.; Wang, Y.; Hong, L.; Han, J.; Xu, H.; Li, Z.; et al. G-llava: Solving geometric problem with multi-modal large language model. *arXiv* **2023**, arXiv:2312.11370.
23. ŞAHiN, E.; Arslan, N.N.; Özdemir, D. Unlocking the black box: An in-depth review on interpretability, explainability, and reliability in deep learning. *Neural Comput. Appl.* **2025**, *37*, 859–965. [CrossRef]
24. Zhang, J.; Li, Z.; Zhang, M.; Yin, F.; Liu, C.; Moshfeghi, Y. Geoeval: Benchmark for evaluating llms and multi-modal models on geometry problem-solving. *arXiv* **2024**, arXiv:2402.10104.
25. Zhang, X.; Zhu, N.; He, Y.; Zou, J.; Huang, Q.; Jin, X.; Guo, Y.; Mao, C.; Li, Y.; Zhu, Z.; et al. FormalGeo: An Extensible Formalized Framework for Olympiad Geometric Problem Solving. *arXiv* **2023**, arXiv:2310.18021.
26. Lu, P.; Gong, R.; Jiang, S.; Qiu, L.; Huang, S.; Liang, X.; Zhu, S.C. Inter-GPS: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv* **2021**, arXiv:2105.04165. [CrossRef]
27. Chen, J.; Tang, J.; Qin, J.; Liang, X.; Liu, L.; Xing, E.P.; Lin, L. GeoQA: A geometric question answering benchmark towards multimodal numerical reasoning. *arXiv* **2021**, arXiv:2105.14517.
28. Cao, J.; Xiao, J. An augmented benchmark dataset for geometric question answering through dual parallel text encoding. In Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, 12–17 October 2022; pp. 1511–1520.
29. Zhang, X.; Zhu, N.; Qin, C.; Yang, L.; Zeng, Z.; Leng, T. Formal Representation and Solution of Plane Geometric Problems. In Proceedings of the 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24, Vancouver, BC, Canada, 14 December 2024.
30. Xie, X.; Kersting, K.; Neider, D. Neuro-symbolic verification of deep neural networks. *arXiv* **2022**, arXiv:2203.00938. [CrossRef]
31. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
32. Soutchanski, M.; Young, R. Planning as theorem proving with heuristics. *arXiv* **2023**, arXiv:2303.13638. [CrossRef]
33. Kwon, K. A Heuristic Proof Procedure for Propositional Logic. *arXiv* **2022**, arXiv:2202.10639. [CrossRef]
34. Seo, M.; Hajishirzi, H.; Farhadi, A.; Etzioni, O.; Malcolm, C. Solving geometry problems: Combining text and diagram interpretation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 1466–1476.
35. Zhang, X.; Zhu, N.; He, Y.; Zou, J.; Qin, C.; Li, Y.; Leng, T. FGeo-SSS: A Search-Based Symbolic Solver for Human-like Automated Geometric Reasoning. *Symmetry* **2024**, *16*, 404. [CrossRef]
36. Trinh, T.H.; Wu, Y.; Le, Q.V.; He, H.; Luong, T. Solving olympiad geometry without human demonstrations. *Nature* **2024**, *625*, 476–482. [CrossRef] [PubMed]
37. Sinha, S.; Prabhu, A.; Kumaraguru, P.; Bhat, S.; Bethge, M. Wu's Method can Boost Symbolic AI to Rival Silver Medalists and AlphaGeometry to Outperform Gold Medalists at IMO Geometry. *arXiv* **2024**, arXiv:2404.06405.

38. Chervonyi, Y.; Trinh, T.H.; Olšák, M.; Yang, X.; Nguyen, H.; Menegali, M.; Jung, J.; Verma, V.; Le, Q.V.; Luong, T. Gold-medalist Performance in Solving Olympiad Geometry with AlphaGeometry2. *arXiv* **2025**, arXiv:2502.03544. [CrossRef]
39. Xin, H.; Ren, Z.; Song, J.; Shao, Z.; Zhao, W.; Wang, H.; Liu, B.; Zhang, L.; Lu, X.; Du, Q.; et al. Deepseek-prover-v1. 5: Harnessing proof assistant feedback for reinforcement learning and monte-carlo tree search. *arXiv* **2024**, arXiv:2408.08152.
40. Kaliszyk, C.; Urban, J.; Michalewski, H.; Olšák, M. Reinforcement learning of theorem proving. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 8836–8847.
41. Peng, S.; Fu, D.; Liang, Y.; Gao, L.; Tang, Z. Geodrl: A self-learning framework for geometry problem solving using reinforcement learning in deductive reasoning. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023, Toronto, ON, Canada, 9–14 July 2023; pp. 13468–13480.
42. Wu, W.; Zhang, L.; Liu, J.; Tang, X.; Wang, Y.; Wang, S.; Wang, Q. E-gps: Explainable geometry problem solving via top-down solver and bottom-up generator. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 13828–13837.
43. Guan, X.; Zhang, L.L.; Liu, Y.; Shang, N.; Sun, Y.; Zhu, Y.; Yang, F.; Yang, M. rStar-Math: Small LLMs Can Master Math Reasoning with Self-Evolved Deep Thinking. *arXiv* **2025**, arXiv:2501.04519.
44. Zou, J.; Zhang, X.; He, Y.; Zhu, N.; Leng, T. Fgeo-drl: Deductive reasoning for geometric problems through deep reinforcement learning. *Symmetry* **2024**, *16*, 437. [CrossRef]
45. He, Y.; Zou, J.; Zhang, X.; Zhu, N.; Leng, T. FGeo-TP: A Language Model-Enhanced Solver for Geometry Problems. *arXiv* **2024**, arXiv:2402.09047. [CrossRef]
46. Zhu, N.; Zhang, X.; Huang, Q.; Zhu, F.; Zeng, Z.; Leng, T. FGeo-Parser: Autoformalization and Solution of Plane Geometric Problems. *Symmetry* **2024**, *17*, 8. [CrossRef]
47. Yang, X.W.; Zhou, Z.; Wang, H.; Li, A.; Wei, W.D.; Jin, H.; Li, Z.; Li, Y.F. CARTS: Advancing Neural Theorem Proving with Diversified Tactic Calibration and Bias-Resistant Tree Search. In Proceedings of the Thirteenth International Conference on Learning Representations, Singapore, 24–28 April 2025.
48. LaBelle, E. Monte Carlo Tree Search Applications to Neural Theorem Proving. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2024.
49. Zhang, X.; Zhu, N.; Qin, C.; Li, Y.; Zeng, Z.; Leng, T. FGeo-HyperGNet: Geometric Problem Solving Integrating Formal Symbolic System and Hypergraph Neural Network. *arXiv* **2024**, arXiv:2402.11461.
50. Liu, Y.; Ayzenberg, V.; Lourenco, S.F. Object geometry serves humans' intuitive physics of stability. *Sci. Rep.* **2024**, *14*, 1701. [CrossRef] [PubMed]
51. Pandey, P.; Pandey, D.; Kumar, S. Reinforcement learning by comparing immediate reward. *arXiv* **2010**, arXiv:1009.2566. [CrossRef]
52. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489. [CrossRef]
53. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. Mastering the game of go without human knowledge. *Nature* **2017**, *550*, 354–359. [CrossRef]
54. Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv* **2017**, arXiv:1712.01815. [CrossRef]
55. Xiao, C.; Mei, J.; Müller, M. Memory-augmented monte carlo tree search. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
56. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 5485–5551.
57. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7871–7880.
58. Zhao, C.; Deng, C.; Ruan, C.; Dai, D.; Gao, H.; Li, J.; Zhang, L.; Huang, P.; Zhou, S.; Ma, S.; et al. Insights into deepseek-v3: Scaling challenges and reflections on hardware for ai architectures. In Proceedings of the 52nd Annual International Symposium on Computer Architecture, Tokyo, Japan, 21–25 June 2025; pp. 1731–1745.
59. Xiao, T.; Liu, J.; Huang, Z.; Wu, J.; Sha, J.; Wang, S.; Chen, E. Learning to solve geometry problems via simulating human dual-reasoning process. *arXiv* **2024**, arXiv:2405.06232. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.