From Natural Language to Certified Geometry Proofs: A Survey of LLM-Augmented Verification and Neuro-Symbolic Theorem Proving

Ioannis Tzachristas [1,2*], **Georgios Tzachristas** [1,3*]
[1]Huawei European Research Institute
[2]Technical University of Munich, Germany
[3]National Technical University of Athens, Greece

Abstract

Large Language Models (LLMs) can produce convincing geometric arguments, yet their outputs are not reliable enough to be treated as proofs without independent verification. In parallel, symbolic geometry tools (e.g. automated theorem provers in dynamic geometry systems) offer strong rigor guarantees, but require formalized inputs and can struggle with problem formalization, auxiliary construction, and proof presentation. This survey synthesizes work at the intersection of these lines: *hybrid LLM–symbolic systems for geometry* that (i) translate natural language and diagrams into formal constraints, (ii) search for solution plans and proof steps using learned or heuristic methods, and (iii) verify the resulting steps using symbolic provers or proof assistants. We propose a taxonomy organized around (a) the role of the LLM in the pipeline (parser, strategist, prover, critic), (b) the target proof artifact (answer-only, informal proof, semi-formal step trace, or kernel-checked formal proof), and (c) the verification backend (numeric testing, algebraic provers, synthetic provers, and proof-assistant kernels). We review representative systems in NLP and AI (e.g. GeoS, Inter-GPS, FormalGeo, AlphaGeometry, AutoGPS, and recent heuristic-only deductive solvers), and connect them to broader neurosymbolic paradigms for *faithful* reasoning (e.g. SatLM, LINC, and autoformalization). Finally, we outline evaluation protocols emphasizing step-level soundness and robustness, and we discuss open problems in multimodal formalization, handling of non-degeneracy conditions, human-readable certified proofs, and reproducibility.

1 Introduction

Geometry sits at the intersection of language, vision, and formal reasoning. Many geometry problems are described in natural language and supported by diagrams, but the desired solution is often a *proof*—a chain of logically valid steps. Classical NLP systems for geometry (e.g. GeoS (Seo et al., 2015)) and later interpretable pipelines (e.g. Inter-GPS (Lu et al., 2021), GeoQA (Chen et al., 2021)) illustrate the core challenge: correct solutions require a consistent interpretation of text, diagram, and domain axioms.

Recent progress in neural theorem proving and LLM-based reasoning has renewed interest in turning informal mathematical reasoning into verifiable proof artifacts (Wu et al., 2022; Li et al., 2024). In Euclidean geometry, this momentum is amplified by strong symbolic engines and new neuro-symbolic solvers that reach Olympiad-level performance (Trinh et al., 2024; Chervonyi et al., 2025; Duan et al., 2025). However, the *verification gap* remains: LLMs can hallucinate steps, omit conditions, and produce plausible-but-invalid proofs, while symbolic provers require careful formalization and may return results that are hard to interpret pedagogically (e.g. algebraic certificates and non-degeneracy conditions).

This survey focuses on **verification-oriented** geometry pipelines that integrate LLMs with symbolic tools. We treat verification broadly, ranging from (i) *symbolic checkers* embedded in dynamic geometry systems such as GeoGebra (Kovács and Sólyom-Gecse, 2016; Botana et al., 2015) to (ii) *formal proof assistants* (Lean/Coq/Isabelle) and their kernels. We argue that modern geometry automation should be evaluated not only by answer accuracy but also by the *soundness and auditability* of its intermediate reasoning steps. We contribute:

1. A **taxonomy** of LLM–symbolic geometry systems grounded in roles, representations, and verification backends.
2. A **comparative review** of symbolic geometry provers (algebraic and synthetic), formal proof assistants, and dynamic geometry environments used for automated verification.

*Equal contribution.
Correspondence: ioannis.tzachristas@tum.de

3. A **survey of neuro-symbolic and multimodal systems** for geometry problem solving, including recent Olympiad-level solvers and formalized geometry frameworks.
4. A set of **evaluation guidelines** emphasizing step-level validity, robustness, and reproducibility.

2 Problem Setting and Terminology

2.1 Inputs, outputs, and levels of rigor

Geometry automation spans multiple input modalities and proof artifacts:

Inputs. (i) *Formal* constructions (e.g. coordinates or a domain-specific language), (ii) *text-only* problem statements, (iii) *text + diagram* (raster or vector).

Outputs. (i) a final *answer* (numeric or multiple-choice), (ii) an *informal proof* in natural language, (iii) a *semi-formal* step trace with explicit theorem applications, (iv) a *formal proof* checked by a proof assistant kernel.

Verification targets. A crucial distinction is whether the system produces (or can be converted into) an artifact that a *trusted checker* can validate. We distinguish:

- **Empirical validation:** random instantiation, numeric sampling, or bounded "exact check".
- **Symbolic validation:** algebraic methods (Wu/Gröbner) and semi-algebraic/synthetic methods (area/full-angle/coherent logic).
- **Kernel validation:** proof assistant kernel checking (Lean/Coq/Isabelle/HOL Light/Metamath).

2.2 A canonical hybrid pipeline

Figure 2 sketches the pipeline common to many hybrid systems: (1) *formalization* from text/diagram to constraints, (2) *search* for a proof plan or step sequence, (3) *verification* of each step in a symbolic environment, (4) optionally *natural language rendering* for human consumption.

3 A Taxonomy of LLM–Symbolic Geometry Verification

We propose a three-axis taxonomy:

Axis A: Role of the LLM.

- **LLM as parser:** translate natural language (and/or diagrams) into formal constraints or a DSL (cf. autoformalization (Wu et al., 2022)).
- **LLM as strategist:** propose lemmas, theorem applications, or auxiliary constructions to guide symbolic search (common in neuro-symbolic provers).
- **LLM as prover:** output formal proof scripts (Lean/Coq) directly, checked by a kernel (e.g. LLM provers built on LeanDojo (Yang et al., 2023)).
- **LLM as critic:** score, rerank, or refine candidate steps based on verifier feedback (e.g. solver-in-the-loop refinement (Ye et al., 2023; Olausson et al., 2023)).

Axis B: Proof artifact. Answer-only → informal proof → semi-formal trace → kernel-checked proof.

Axis C: Verification backend. Empirical checking → symbolic algebraic/synthetic provers → proof-assistant kernels.

This taxonomy helps clarify trade-offs: kernel-checked proofs maximize trust but are hard to produce; algebraic provers scale but return conditions and certificates that may be pedagogically opaque; empirical checks are easy but unsound as proof.

4 Symbolic Verifiers for Euclidean Geometry

This section reviews the symbolic tools commonly used to validate geometric claims.

4.1 Algebraic methods

Algebraic geometry theorem proving translates geometric predicates into polynomial equations and uses elimination (e.g. Wu's method or Gröbner bases) to prove that hypotheses imply the conclusion, often generating *non-degeneracy conditions* (NDGs) (Marić et al., 2012). Algebraic methods are powerful and fast in many settings, but their outputs may be less interpretable than classical synthetic proofs.

4.2 Synthetic and semi-algebraic methods

Synthetic approaches aim to produce human-readable proofs. The *area method* is a prominent semi-algebraic technique that produces concise, readable proofs for constructive geometry (Janičić et al., 2012). Tools such as GCLC integrate visualization with theorem proving and can support multiple methods, including the area method (Janičić, 2006).

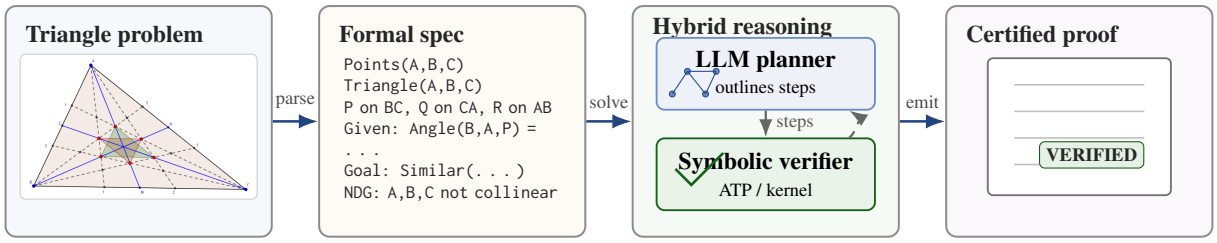


Figure 1: A high-level view of verified geometry reasoning: a natural-language problem with a diagram is translated into a formal specification, solved with LLM-guided symbolic reasoning, and emitted as a certified proof artifact.

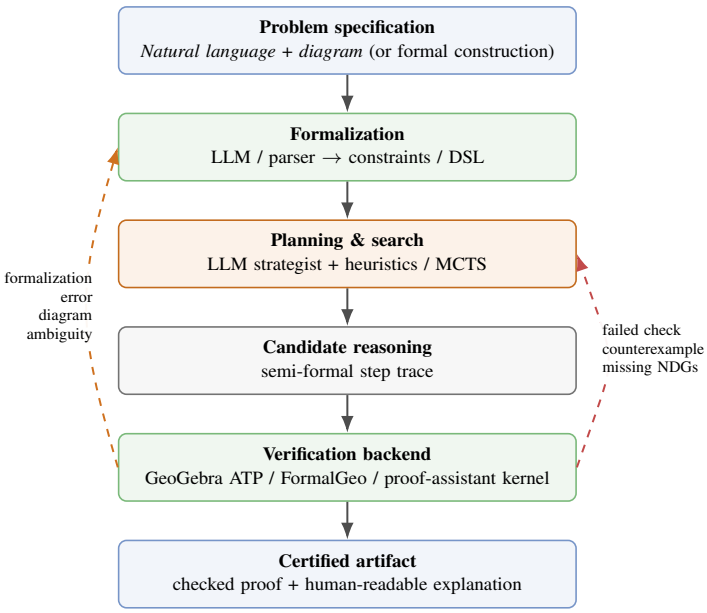


Figure 2: A canonical LLM–symbolic workflow for geometry: propose, check, and repair.

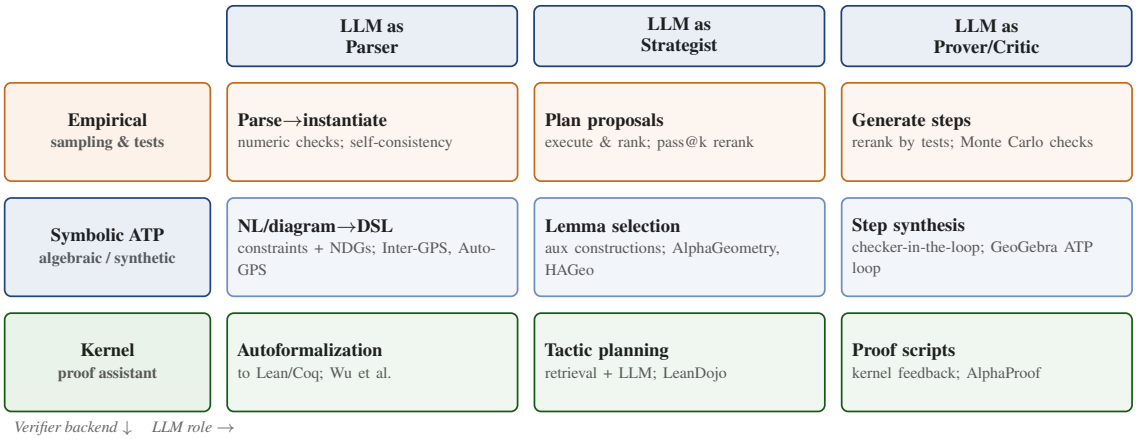


Figure 3: A compact taxonomy of LLM–symbolic geometry systems across LLM roles and verification backends, with representative examples.

4.3 Dynamic geometry environments and embedded provers

Dynamic geometry systems (DGS) provide interactive construction and visualization; some also embed proving functionality. GeoGebra has incorporated automated theorem proving features and a portfolio of provers, including Gröbner-basis-based proving and connections to external provers (e.g. OpenGeoProver for Wu/area) (Botana et al., 2015; Kovács and Sólyom-Gecse, 2016). Recent work

also explores automated *discovery* of geometric properties within GeoGebra constructions (Kovács and Yu, 2022).

4.4 Formal proof assistants

Proof assistants provide the strongest correctness guarantees via small trusted kernels, at the cost of higher formalization effort. For geometry, there is a long line of work on formalizing Euclidean axioms (e.g. Tarski/Hilbert) and connecting automation tactics to kernels. Maric et al. (Marić et al., 2012) describe Isabelle/HOL formalization efforts aimed at bridging algebraic methods and synthetic geometry, enabling verified automation tactics.

5 Representative Geometry Solvers and LLM-Augmented Systems

Table 1 summarizes representative systems along our taxonomy axes. Some entries are not LLM systems by themselves; we include them because they serve as formalizers, verifiers, search environments, or benchmarks inside LLM-augmented geometry pipelines. Figure 4 provides an additional visual summary of these roles.

6 Historical Timeline and Milestones

Figure 5 summarizes a few influential milestones that shaped the modern landscape of verified and neuro-symbolic geometry reasoning, spanning early multimodal QA, embedded automated theorem proving in dynamic geometry software, formalized geometry environments, and Olympiad-level neuro-symbolic solvers.

6.1 Multimodal problem understanding: text and diagram

NLP-oriented geometry systems often start with the *formalization bottleneck*: extracting entities, relations, and constraints from language and diagrams. GeoS (Seo et al., 2015) pioneered combining text and diagram interpretation for SAT geometry. Inter-GPS (Lu et al., 2021) later emphasized *interpretable* symbolic reasoning with a formal language representation, while GeoQA (Chen et al., 2021) provided a benchmark with annotated programs for multimodal numerical reasoning. Recent systems such as AutoGPS aim to jointly learn formalization and deductive reasoning with tight feedback loops between the modules (Ping et al., 2025).

6.2 Formalized geometry environments for verifiable traces

FormalGeo proposes a consistent formal plane geometry system and datasets that support traceable, verifiable solutions (Zhang et al., 2023). Within such environments, learning can focus on theorem selection and search policy while the environment enforces logical validity. FGeo-DRL builds an RL+MCTS agent that operates in the FormalGeo environment and yields readable, verifiable deductive solutions (Zou et al., 2024).

6.3 Olympiad-level provers and auxiliary construction

AlphaGeometry introduced a neuro-symbolic solver trained on synthetic data that outputs proof-like derivations verified by a symbolic engine (Trinh et al., 2024). AlphaGeometry2 reports expanded language coverage and improved search on IMO geometry sets, and it was part of a system that reached silver-medal standard on IMO-2024 (Chervonyi et al., 2025; Google DeepMind, 2024). Complementary to learned components, purely heuristic deductive approaches have also shown strong performance: HAGeo proposes efficient auxiliary constructions and reports high solving rates on Olympiad benchmarks without neural inference (Duan et al., 2025).

6.4 General neurosymbolic patterns for faithful reasoning

Although not geometry-specific, several ACL-relevant paradigms inform verified geometry pipelines. Program-aided LMs (PAL) offload execution to interpreters (Gao et al., 2022); SatLM translates problems into declarative constraints and uses SAT solving (Ye et al., 2023); LINC translates premises and conclusions into first-order logic and delegates deduction to logic provers (Olausson et al., 2023). Autoformalization systems translate informal mathematics into formal statements for proof assistants (e.g. Isabelle/HOL) (Wu et al., 2022). These frameworks highlight a recurring motif: use LLMs for *semantic parsing and proposal*, but use symbolic engines for *sound inference*.

7 Datasets, Benchmarks, and Evaluation

7.1 Geometry datasets

Table 2 lists major datasets spanning text, diagrams, and formal proof traces.

System	Input	Output	Verifier	Notes / LLM Role
GeoS (Seo et al., 2015)	Text + raster diagram	Answer (SAT-style)	Geometric solver (symbolic)	Early end-to-end pipeline; optimization-based parsing and diagram interpretation.
Inter-GPS (Lu et al., 2021)	Text + diagram	Answer + interpretable steps	Symbolic rule-based reasoning	Neural perception + formal language + theorem-driven symbolic reasoning.
GeoQA (Chen et al., 2021)	Text + diagram	Answer + program	Program executor	Dataset + neural solvers; emphasizes multimodal numerical reasoning.
FormalGeo (Zhang et al., 2023)	Formal language (often derived from text)	Stepwise proof trace	Formal system checker	Formalized predicate/theorem library enabling traceable, verifiable solutions.
FGeo-DRL (Zou et al., 2024)	FormalGeo environment	Stepwise proof trace	FormalGeo checker	RL + MCTS for theorem selection and search in a formal environment.
AutoGPS (Ping et al., 2025)	Text + diagram	Minimal stepwise proof	Deductive symbolic reasoner	Multimodal formalizer + deductive reasoner; emphasizes stepwise coherence.
AlphaGeometry (Trinh et al., 2024)	Domain-specific language	Proof (synthetic-style)	Symbolic engine	Neural + symbolic; trained on synthetic theorems; no human demonstrations.
AlphaGeometry2 (Chervonyi et al., 2025)	Extended DSL / partial NL	Proof	Symbolic engine	Expanded language coverage and improved search; used in IMO-2024 silver system.
HAGeo (Duan et al., 2025)	Formal geometry benchmark	Proof / derivation	Deductive engine (no NN)	Heuristic auxiliary constructions; strong performance without neural inference.
GeoGebra ATP (Botana et al., 2015; Kovács and Sólyom-Gecse, 2016)	Interactive construction	True/False + NDGs	Portfolio provers	DGS interface for algebraic/synthetic proving; useful as step checker in hybrid workflows.

Table 1: Representative geometry systems and their verification backends.

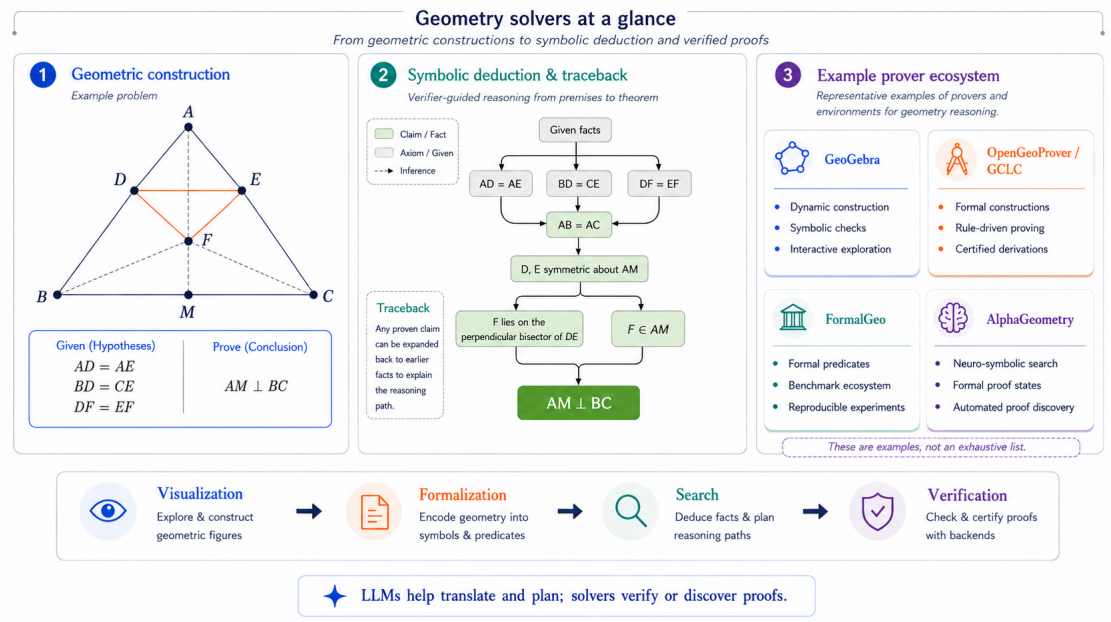


Figure 4: Additional at-a-glance comparison of representative geometry tools and solver families. This visual complements Table 1 by summarizing common stages in LLM-augmented geometry workflows and by showing a representative prover ecosystem; the tools in the right panel are examples rather than an exhaustive list.

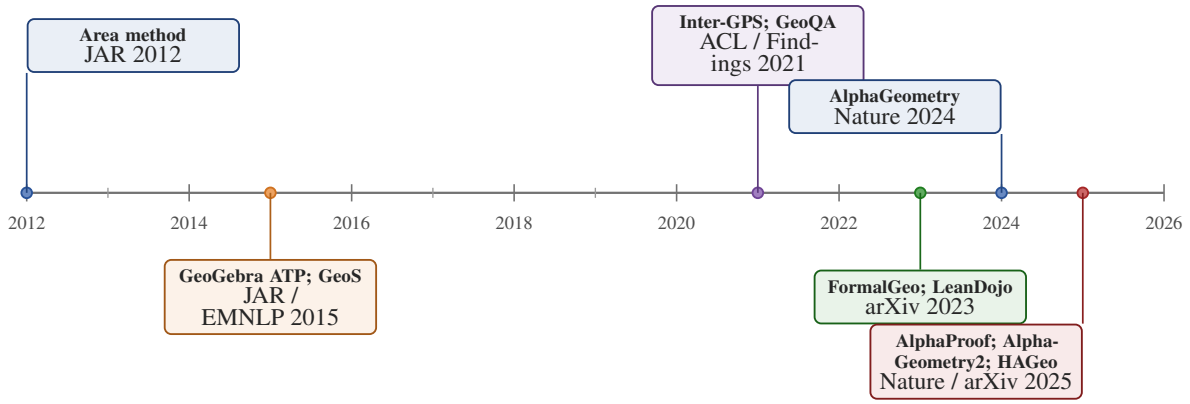


Figure 5: A non-exhaustive timeline of key milestones in multimodal geometry problem solving and verified/neuro-symbolic theorem proving.

Dataset	Modality	Notes
GeoS (Seo et al., 2015)	Text+diagram	SAT-style geometry QA; early end-to-end benchmark combining textual and diagrammatic parsing.
GeoQA (Chen et al., 2021)	Text+diagram	Annotated programs for geometric question answering; emphasizes multimodal numerical reasoning.
Geometry3K (Lu et al., 2021)	Text+diagram	Large benchmark used in Inter-GPS and follow-up work; supports interpretable symbolic reasoning.
FormalGeo7K / IMO (Zhang et al., 2023)	Formal	Formalized predicates and theorem libraries with stepwise proof traces enabling verifiable deduction.
IMO-30 (Trinh et al., 2024)	Formal	Standard Olympiad geometry evaluation subset used for neuro-symbolic benchmarking.
HAGeo-409 (Duan et al., 2025)	Formal	Expanded benchmark with human-assessed difficulty levels and emphasis on auxiliary constructions.

Table 2: Selected datasets and benchmarks for geometry reasoning and verification.

7.2 Metrics beyond answer accuracy

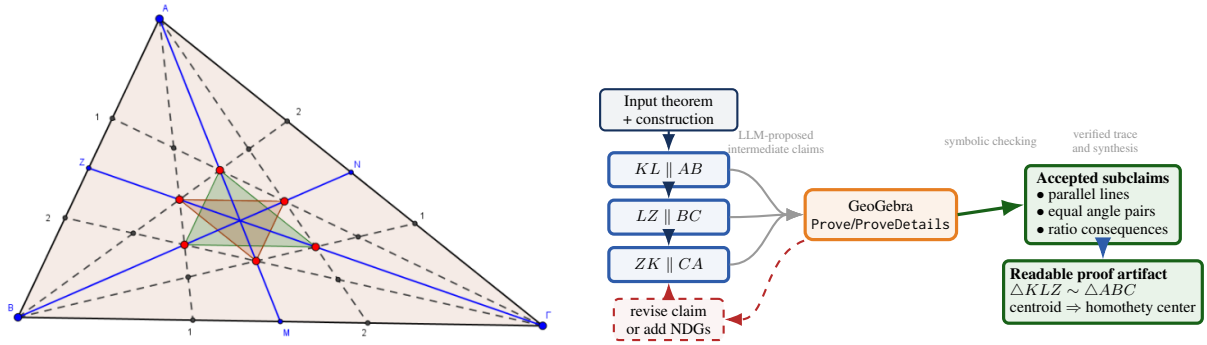
For verification-oriented systems, answer accuracy is insufficient. The checklist below consolidates practices already common in trace-based geometry and verifier-in-the-loop work: executable proof traces, explicit theorem applications, NDG reporting, and reproducibility of verifier settings (Zhang et al., 2023; Zou et al., 2024; Ping et al., 2025; Botana et al., 2015; Trinh et al., 2024). We recommend reporting:

- **Step validity rate:** fraction of generated steps accepted by the verifier.
- **Proof completeness:** whether a full chain from hypotheses to goal is produced.
- **NDG handling:** whether non-degeneracy conditions are made explicit and interpretable.
- **Minimality and readability:** proof length, lemma reuse, and human evaluation.
- **Robustness:** invariance to paraphrases, diagram perturbations, or irrelevant distractors.
- **Reproducibility:** open code/data, deterministic seeds, and clear verifier settings.

8 Case Study: Verifying a Triangle Trisection Generalization with Tool-Augmented LLMs

A representative use-case is to treat an LLM as a *proof planner* that proposes a structured outline, then validate each step with a symbolic geometry prover. For instance, the case study of Tzachristas and Tzachristas (2026) uses a triangle-side trisection configuration to connect natural-language planning, GeoGebra queries, and verified subclaims. The implementation of the case-study workflow was inspired by agentic workflow frameworks such as Hermes Agent and OpenClaw, which organize task execution around persistent agents, tool use, memory, and reusable skills (Nous Research, 2026; OpenClaw Contributors, 2026). Figure 6 adds a visual companion to the textual case-study description.

Concretely, the pipeline input is a theorem statement plus a GeoGebra construction for $\triangle ABC$



(a) Triangle-side trisection configuration from the MICOM case study.

(b) Original schematic of the generate–check–repair loop for verifier-guided proof construction.

Figure 6: Additional case-study visualization. Left: the triangle-side trisection construction discussed by Tzachristas and Tzachristas (2026). Right: an original, redrawn schematic showing how LLM-proposed subclaims are checked with GeoGebra and assembled into a readable proof artifact; this preserves the general deduction-and-traceback idea without reusing the source-uncertain image.

with side/intersection points K, L, Z and the target that the constructed inner triangle is similar and homothetic to ABC (with the centroid as homothecy center). The intermediate structures are verifier-addressable Boolean claims, such as $KL \parallel AB$, $LZ \parallel BC$, and $ZK \parallel CA$, issued through commands of the form `Prove(AreParallel(...))` and optionally expanded with `ProveDetails` / NDG output. The output is a checked trace of accepted subgoals plus a human-readable proof deriving similarity and the homothecy claim. In such a workflow:

1. The LLM proposes a sequence of intermediate claims (collinearity, parallelism, ratios, or Ceva/Menelaus-style equalities).
2. Each claim is translated into a form accepted by a verifier (e.g. GeoGebra's `Prove` /`ProveDetails` commands).
3. Failed steps trigger refinement: the LLM revises the claim or introduces missing NDG conditions.
4. A final verified step trace is rendered into human-readable proof text.

This "generate–check–repair" loop aligns naturally with neurosymbolic paradigms in NLP, and can be evaluated via step validity, completeness, and interpretability.

9 Open Challenges and Future Directions

Multimodal autoformalization. Moving from text+diagram to a formal specification remains brittle; robust datasets with aligned language, diagram, and formal constraints are scarce despite progress in GeoS, Inter-GPS, GeoQA, and AutoGPS-style pipelines (Seo et al., 2015; Lu et al., 2021; Chen et al., 2021; Ping et al., 2025).

Auxiliary construction and search control. Olympiad geometry often hinges on creative constructions. Balancing learned heuristics, symbolic search, and interpretability remains an open problem in AlphaGeometry-style and heuristic-only systems (Trinh et al., 2024; Chervonyi et al., 2025; Duan et al., 2025).

Non-degeneracy conditions (NDGs). Algebraic provers generate NDGs; mapping them to human-friendly geometric conditions and ensuring they are tracked across proof steps is essential for trustworthy proofs (Marić et al., 2012; Botana et al., 2015).

Certified yet readable proofs. Producing *kernel-checked* proofs that are also pedagogically meaningful is an ongoing challenge. Bridging proof-assistant scripts, prover certificates, and classical Euclidean style is a key opportunity (Wu et al., 2022; Yang et al., 2023; Kovács and Sólyom-Gecse, 2016).

Evaluation culture. We encourage the community to report verifier settings, failure modes, and ablations that isolate formalization errors from reasoning errors, following the traceability emphasis of FormalGeo/FGeo-DRL and the reproducible benchmark style of recent solvers (Zhang et al., 2023; Zou et al., 2024; Duan et al., 2025).

10 Conclusion

Hybrid LLM–symbolic systems offer a promising path from natural language and diagrams to verified geometry proofs. The strongest systems tightly couple learned proposal mechanisms with symbolic or kernel-level verification, enabling traceable derivations and reducing hallucinations. This survey provided a taxonomy of such systems, reviewed symbolic verifiers and geometry datasets, and argued for evaluation protocols that prioritize step-level correctness and reproducibility.

Limitations

This survey emphasizes verification-oriented pipelines and may omit purely neural answer-only systems when they do not expose verifiable intermediate artifacts. The field is moving rapidly; despite including recent work up to early 2026 in our bibliography, some contemporaneous results may be missing.

Ethics Statement

We do not anticipate direct harmful applications from surveyed methods. However, educational deployments should clearly communicate the difference between plausible explanations and verified proofs, and should avoid over-reliance on unverified LLM output. Releasing datasets should respect copyright constraints for sourced problem statements and diagrams.

Acknowledgements

The authors thank the open research community for making datasets, code, and preprints publicly available. We acknowledge the use of automated writing tools for language editing and clarity improvements during the preparation of this manuscript. All technical content, interpretations, and final decisions remain the responsibility of the authors.

References

Francisco Botana, Markus Hohenwarter, Predrag Janičić, Zoltán Kovács, Ivan Petrović, Tomás Recio, and Simon Weitzhofer. 2015. Automated theorem proving in GeoGebra: Current achievements. *Journal of Automated Reasoning*, 55:39–59.

Jiaqi Chen, Jianheng Tang, Jinghui Qin, Xiaodan Liang, Lingbo Liu, Eric Xing, and Liang Lin. 2021. Geoqa: A geometric question answering benchmark towards multimodal numerical reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 513–523.

Yuri Chervonyi, Trieu H. Trinh, Miroslav Olšák, Xiaomeng Yang, Hoang Nguyen, Marcelo Menegali, Junehyuk Jung, Junsu Kim, Vikas Verma, Quoc V. Le, and Thang Luong. 2025. Gold-medalist performance in solving olympiad geometry with AlphaGeometry2. arXiv:2502.03544.

Boyan Duan, Xiao Liang, Shuai Lu, Yaoxiang Wang, Yelong Shen, Kai-Wei Chang, Ying Nian Wu, Mao Yang, Weizhu Chen, and Yeyun Gong. 2025. Gold-medal-level olympiad geometry solving with efficient heuristic auxiliary constructions. arXiv:2512.00097.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2022. PAL: Program-aided language models. arXiv:2211.10435.

Google DeepMind. 2024. Ai achieves silver-medal standard solving international mathematical olympiad problems. Blog post.

Predrag Janičić. 2006. GCLC—a tool for constructive euclidean geometry and more than that. In *International Congress on Mathematical Software (ICMS)*, pages 58–73.

Predrag Janičić, Julien Narboux, and Pedro Quaresma. 2012. The area method. *Journal of Automated Reasoning*, 48(4):489–532.

Zoltán Kovács and Csilla Sólyom-Gecse. 2016. Geogebra tools with proof capabilities. *arXiv preprint arXiv:1603.01228*.

Zoltán Kovács and Jonathan H. Yu. 2022. Automated discovery of geometrical theorems in GeoGebra. In *Proceedings 10th International Workshop on Theorem Proving Components for Educational Software (THedu'21)*, volume 354 of *Electronic Proceedings in Theoretical Computer Science*, pages 1–12.

Zhaoyu Li, Jialiang Sun, Logan Murphy, Qidong Su, Zenan Li, Xian Zhang, Kaiyu Yang, and Xujie Si. 2024. A survey on deep learning for theorem proving. arXiv:2404.09939.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. 2021. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6774–6786.

Filip Marić, Ivan Petrović, Danijela Petrović, and Predrag Janičić. 2012. Formalization and implementation of algebraic methods in geometry. In *Theorem Proving Components for Educational Software (THedu'11), EPTCS 79*, pages 63–81.

Nous Research. 2026. Hermes agent: The self-improving ai agent built by nous research. https://github.com/NousResearch/hermes-agent. Open-source software repository. Accessed 2026-05-14.

Theo X. Olausson, Alex Gu, Benjamin Lipkin, Cedegao E. Zhang, Armando Solar-Lezama, Joshua B. Tenenbaum, and Roger Levy. 2023. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. arXiv:2310.15164.

OpenClaw Contributors. 2026. Openclaw: Personal ai assistant. https://github.com/openclaw/openclaw. Open-source software repository. Accessed 2026-05-14.

Bowen Ping, Minnan Luo, Zhuohang Dang, Chenxi Wang, and Chengyou Jia. 2025. Autogps: Automated geometry problem solving via multimodal formalization and deductive reasoning. arXiv:2505.23381.

Minjoon Seo, Hannaneh Hajishirzi, Ali Farhadi, Oren Etzioni, and Clint Malcolm. 2015. Solving geometry problems: Combining text and diagram interpretation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1466–1476.

Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, He He, Thang Luong, et al. 2024. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482.

Ioannis Tzachristas and Georgios Tzachristas. 2026. Proofing techniques in geometry using LLMs and GeoGebra: A case study of the triangle-side trisection theorem. In *7th International Congress on Mathematics (MICOM 2026)*, Thessaloniki, Greece. Mathematical Society of South-Eastern Europe (MASSEE). Presentation slides.

Yuhuai Wu, Albert Q. Jiang, Wenda Li, Markus N. Rabe, Charles Staats, Mateja Jamnik, and Christian Szegedy. 2022. Autoformalization with large language models. In *Advances in Neural Information Processing Systems*.

Kaiyu Yang, Aidan M. Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar. 2023. Leandojo: Theorem proving with retrieval-augmented language models. arXiv:2306.15626.

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. 2023. SatLM: Satisfiability-aided language models using declarative prompting. In *Advances in Neural Information Processing Systems*.

Xiaokai Zhang, Na Zhu, Yiming He, Jia Zou, Qike Huang, Xiaoxiao Jin, Yanjun Guo, Chenyang Mao, Yang Li, Zhe Zhu, Dengfeng Yue, Fangzhen Zhu, Yifan Wang, Yiwen Huang, Runan Wang, Cheng Qin, Zhenbing Zeng, Shaorong Xie, Xiangfeng Luo, and Tuo Leng. 2023. Formalgeo: An extensible formalized framework for olympiad geometric problem solving. arXiv:2310.18021.

Jia Zou, Xiaokai Zhang, Yiming He, Na Zhu, and Tuo Leng. 2024. Fgeo-drl: Deductive reasoning for geometric problems through deep reinforcement learning. arXiv:2402.09051.