GeoProblem Factory: A Visual Interaction System for Solvable and Controllable Geometric Problem Generation by Leveraging Symbolic Deduction Engine

Zhuoxuan Jiang[1], Yanpeng Li[1], Tianyang Zhang[2], Jing Chen[3], Yong Li[4], Mo Guang[5], Wen Si[1], Shaohua Zhang[1]

[1]Shanghai Business School, Shanghai, China
[2]Learnable.ai, Boston, MA, USA
[3]Shanghai Jiaotong University, Shanghai, China
[4]Beijing Shangruitong Education Technology Co., Ltd. (TeacherClub.com.cn), Beijing, China
[5]Li Auto, Hangzhou, China
{jzx,siw,zhangsh}@sbs.edu.cn, 23104040113@stu.sbs.edu.cn, tianyang.zhang@learnable.ai, mangguo185@sjtu.edu.cn

Abstract

We propose a novel system, GeoProblem Factory, designed to effectively generate high-quality geometry problems for intelligent education. The system enables to efficiently produce batches of geometry problems for teachers and students, either to save time and manual effort or to support personalized learning. Generating geometry problems is particularly challenging, as it requires ensuring both solvability and controllability from a pedagogical perspective. To address these issues, we adopt a state-of-the-art pipeline method based on a symbolic deduction engine and develop a visual interaction demo. This demo allows users to easily refine the generated problems through visual operations. It provides two modes for inputting controllable information: specifying knowledge points or supplying a reference problem. Moreover, the system can automatically generate a preliminary geometric diagram corresponding to each problem for further refinement. Through human–machine interaction, the system can more efficiently produce high-quality geometry problems than ever.

Introduction

In mathematical education, geometry problems play an important and unique role in cultivating students' logical abilities. Unlike math word problems, which emphasize language comprehension, mathematical modeling, and equation derivation, geometry problems focus on spatial imagination, computational and reasoning skills, as well as the mastery of geometric theorems and properties (Liu et al. 2020). Geometry problems are widely needed for ability evaluation in various scenarios such as quizzes, homework, and examinations. However, creating these problems—especially novel ones tailored for personalized learning (Polozov et al. 2015; Bernacki and Walkington 2018)—often requires considerable time and effort from problem designers (Wang, Lan, and Baraniuk 2021; Cao et al. 2022). Therefore, developing a tool to accelerate the problem-generation process is of great significance.

Designing a system for generating geometry problems is a non-trivial task. Traditionally, ensuring the quality of generated problems has relied heavily on the expertise of problem designers, a process that is both time-consuming and inefficient (Jiao et al. 2023). With the rapid advancement of large language models (LLMs), which excel at content generation, the challenges of cost and efficiency have been largely mitigated. However, problem quality—particularly in terms of solvability and controllability—remains difficult to guarantee (Li and Zhang 2024; Christ, Kropko, and Hartvigsen 2024; Liu et al. 2024). Therefore, it is essential to develop a geometry problem generation system that simultaneously achieves high quality, low cost, and high efficiency.

To develop such a generation system, we first analyze the format and structure of a geometry problem to clarify its essential components. A typical geometry problem usually consists of a textual description containing several clauses and a final question. The clauses provide the given conditions, while the question is typically stated in the last sentence. In some cases, a geometric diagram is included to help illustrate the problem. The solution is a strict, step-by-step reasoning process based on geometric theorems or properties, often expressed in formal language (Chou, Gao, and Zhang 2000; Ida and Fleuriot 2013; Murphy et al. 2024). We also use the term knowledge points to denote the geometric theorems or properties involved. We refer to the prior work (Jiang et al. 2025) and more details can be found there.

More specifically, we propose a demo system called GeoProblem Factory, which integrates front-end visual interaction technologies with a state-of-the-art back-end pipeline method for problem generation. On the front end, we design two modes that allow users to provide controllable variables to initialize the generation process. To enhance visualization, the generated problems are represented in a graph structure, where users can choose base problems, adjust nodes to modify the final question, and change the path length to adjust the problem's difficulty. On the back end, we incorporate the Symbolic Deduction Engine-based Geometry Problem Generation framework (SDE-GPG) (Jiang et al. 2025), which effectively ensures the solvability and controllability of the generated problems, and encapsulate it as an API service. By integrating the front and back ends through API protocols, the demo system supports efficient generation of quality geometry problems via visual human–machine interaction.

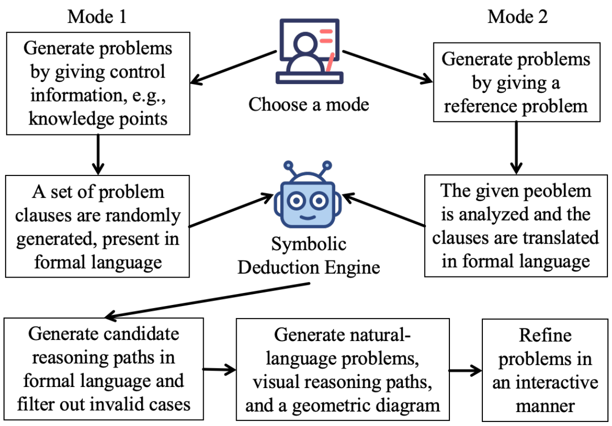

Figure 1: Process of using GeoProblem Factory.

System Overview

GeoProblem Factory integrates a front end for interactive control with a back end for generating batches of geometric problems, where Figure 1 shows how the users can use it.

Front-End Visual Interaction

The front end provides an intuitive interface that supports two controllable modes. In Mode 1 (as shown in Figure 2 (a)), users select from a three-tier structured tree of 43 geometry knowledge points[1], optionally with multi-selection, and specify both difficulty level (from Easy to Expert) and the number of problems to generate. In Mode 2, users input an existing problem for reference, and the system automatically extracts its underlying knowledge points to guide the creation of new variations. These two entry points provide flexible initialization of problem generation for different educational scenarios.

After configuration, users trigger generation by clicking the Generate button. The system displays a progress bar and then presents results through candidate problems, a reasoning graph and accompanying geometric diagrams (as shown in Figure 2 (b) and (c)). In the reasoning graph, nodes denote conditions, intermediate facts, or conclusions, and edges indicate deduction steps. When users click a base problem among candidates or choose a reasoning path, the right part correspondingly switch to the detailed information of the selected problem. The graph-based view makes the logical structure of each problem explicit, while the automatically drawn diagram provides a visual complement.

The interface also supports interactive refinement (as shown in Figure 2 (d)). Users can change the focus by selecting a different endpoint in the reasoning graph, thereby altering the problem's conclusion and difficulty. They also can drag given condition nodes directly for more clear presentation, such as modifying a length or angle value. The system then re-computes a consistent reasoning path in real time, updating the graph, text, and diagram simultaneously.

[1]This volume is what we can support now (limited in Euclidean plane geometry), and the range can be expanded by adding knowledge points in the form of formal language (Trinh et al. 2024).

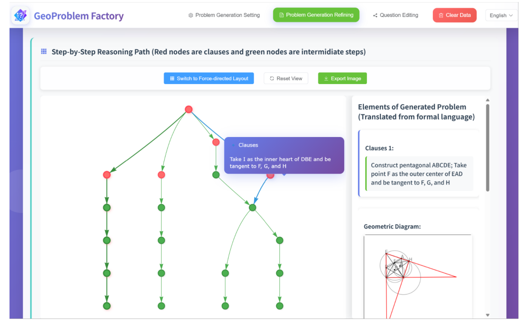

Figure 2: The snapshot of generated problems with reasoning paths and diagrams.

Back-End Problem Generation Pipeline

The back-end pipeline connects to broader work on formal theorem proving and proof assistants (De Moura et al. 2015; Moura and Ullrich 2021). The back end is built on the SDE-GPG framework (Jiang et al. 2025) and AlphaGeometry (Trinh et al. 2024), which are used to produce problems that are both solvable and controllable.

The system first translates user inputs into formal initial conditions: Mode 1 maps knowledge-point selections via a curated table, while Mode 2 uses reference problems as premises. Next, the symbolic deduction engine applies geometric theorems to derive new facts, with reasoning depth controlled by user-specified difficulty. A checker module then validates candidate problems by ensuring all knowledge points are used, conditions are non-redundant, and solution length matches the target range. Finally, retained problems are converted into natural-language statements and diagrams, producing a complete package including the problem, reasoning graph, and diagram. The details of the method can be referred to the aforementioned works.

Case Studies

We apply our demo system to several cases. Figure 2 illustrates the interface of GeoProblem Factory, where generated problems are displayed with reasoning graphs and diagrams. For more detailed demonstrations of the interactive features, please refer to the supplementary demo video.

Conclusion

In this paper, we present GeoProblem Factory, a demo system designed to efficiently generate batches of high-quality geometry problems through human–machine visual interaction. On the front end, we develop two modes to initial the process and implement visualization of reasoning paths and graph structures to facilitate problem refinement and editing. On the back end, we employ a cutting-edge geometry problem generation method based on a symbolic deduction engine. The integration of both components ensures solvability, controllability, and efficiency simultaneously. GeoProblem Factory streamlines the generation process and enhances the pedagogical value of the generated problems. Future work will focus on extending the system's interactive functions and supporting the editing of generated diagrams.

Acknowledgments

This work is supported by Shanghai Science International Cooperation Project, China (Grant No. 24170790602), Program of Shanghai Committee of Science and Technology, China (Grant No. 24511103204), Special Program on Education Examinations of the China Education Development Strategy Society (Grant No. jyks2024038), 2024 Special Program of Artificial Intelligence Promotion on the Reform of Scientific Research Paradigms of Shanghai Committee of Education, China, and Natural Science Foundation of Shanghai (Grant No. 25ZR1401273). We thank all the anonymous reviewers for their insightful and constructive comments. Zhuoxuan Jiang is the corresponding author.

References

Bernacki, M. L.; and Walkington, C. 2018. The role of situational interest in personalized learning. *Journal of Educational Psychology*, 110(6): 864.

Cao, T.; Zeng, S.; Xu, X.; Mansur, M.; and Chang, B. 2022. DISK: Domain-constrained Instance Sketch for Math Word Problem Generation. *arXiv preprint arXiv:2204.04686*.

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

Christ, B.; Kropko, J.; and Hartvigsen, T. 2024. MATHWELL: Generating Educational Math Word Problems Using Teacher Annotations. In *EMNLP 2024*, 11914–11938.

De Moura, L.; Kong, S.; Avigad, J.; Van Doorn, F.; and von Raumer, J. 2015. The Lean theorem prover (system description). In *CADE*, 378–388.

Ida, T.; and Fleuriot, J. 2013. *Automated Deduction in Geometry*. Springer.

Jiang, Z.; Zhang, T.; Peng, P.; Chen, J.; Xun, Y.; Zhang, H.; Li, L.; Li, Y.; and Zhang, S. 2025. Towards Generating Controllable and Solvable Geometry Problem by Leveraging Symbolic Deduction Engine. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, 1378–1398.

Jiao, Y.; Shridhar, K.; Cui, P.; Zhou, W.; and Sachan, M. 2023. Automatic educational question generation with difficulty level controls. In *AIED*, 476–488.

Li, K.; and Zhang, Y. 2024. Planning First, Question Second: An LLM-Guided Method for Controllable Question Generation. In *ACL*, 4715–4729.

Liu, S.; Feng, J.; Yang, Z.; Luo, Y.; Wan, Q.; Shen, X.; and Sun, J. 2024. COMET:“cone of experience” enhanced large multimodal model for mathematical problem generation. *Science China Information Sciences*, 67(12): 1–2.

Liu, T.; Fang, Q.; Ding, W.; Li, H.; Wu, Z.; and Liu, Z. 2020. Mathematical word problem generation from commonsense knowledge graph and equations. *arXiv preprint arXiv:2010.06196*.

Moura, L. d.; and Ullrich, S. 2021. The Lean 4 theorem prover and programming language. In *CADE*, 625–635.

Murphy, L.; Yang, K.; Sun, J.; Li, Z.; Anandkumar, A.; and Si, X. 2024. Autoformalizing Euclidean Geometry. *arXiv preprint arXiv:2405.17216*.

Polozov, O.; O’Rourke, E.; Smith, A. M.; Zettlemoyer, L.; Gulwani, S.; and Popović, Z. 2015. Personalized mathematical word problem generation. In *IJCAI*.

Trinh, T. H.; Wu, Y.; Le, Q. V.; He, H.; and Luong, T. 2024. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995): 476–482.

Wang, Z.; Lan, A. S.; and Baraniuk, R. G. 2021. Math word problem generation with mathematical consistency and problem context constraints. *arXiv preprint arXiv:2109.04546*.