Assessing the Role of Diversity in LLM Explanations for Enhancing Student Understanding

Kush Patel
Temple University
Philadelphia, PA, US
kushrp@temple.edu

Seth Bernstein
Temple University
Philadelphia, PA, US
sethbern@umich.edu

Rayhona Nasimova
Temple University
Philadelphia, PA, US
rayhona.nasimova@temple.edu

Paul Denny
University of Auckland
Auckland, New Zealand
paul@cs.auckland.ac.nz

Juho Leinonen
Aalto University
Espoo, Finland
juho.2.leinonen@aalto.fi

Stephen MacNeil
Temple University
Philadelphia, PA, US
stephen.macneil@temple.edu

Abstract

Large Language Models **(LLMs)** have shown the potential to generate code explanations that surpass those of peers in quality, offering promising opportunities for computer science education. Inspired by this, we explore whether combining multiple diverse explanations, each emphasizing distinct aspects (e.g., function, concept, goal), can enhance students' understanding of programming exercises compared to generic explanations that do not emphasize distinct conceptual aspects. Insights from other fields, such as computational creativity, suggest that diverse ideas may be more beneficial than relying solely on a single, high-quality option. Variation Theory holds that learners grasp a concept when they see systematic variation that exposes its critical features, helping them distinguish it from related ideas. In creative domains, uniform or homogeneous exemplars can lead to design fixation, whereas varied inputs support more flexible reasoning. In a study with 971 first-year computing students, participants were randomly assigned either diverse or generic LLM-generated explanations for two programming exercises. Students completed multiple-choice **(MCQ)** and open-ended **(OE)** questions for each exercise to assess understanding, followed by Likert-scale questions and OE reflections to understand preferences and perception. Across participants, performance was consistently 7.7% higher when students received diverse explanations, and there was no difference in perceived cognitive load. Performance on the closed-form multiple-choice questions was similar for diverse and generic explanations.

CCS Concepts

• **Social and professional topics** → **CS1**; • **Computing methodologies** → **Artificial intelligence**.

ACM Reference Format:
Kush Patel, Seth Bernstein, Rayhona Nasimova, Paul Denny, Juho Leinonen, and Stephen MacNeil. 2026. Assessing the Role of Diversity in LLM Explanations for Enhancing Student Understanding. In *Proceedings of the 57th ACM Technical Symposium on Computer Science Education V.2 (SIGCSE TS 2026), February 18–21, 2026, St. Louis, MO, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3770761.3777320

SIGCSE TS 2026, February 18–21, 2026, St. Louis, MO, USA
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2255-4/2026/02
https://doi.org/10.1145/3770761.3777320

1 Context and Motivation

Teaching students code comprehension skills has been a long-standing and important pedagogical goal within computing education [2, 7]. Recent advances in LLMs enable the immediate, automatic generation of diverse code explanations [5, 12, 13], and even creative analogies [1] that connect with students on a personal level. Prior work shows that LLMs can generate high-quality explanations [6, 9], but it remains unclear whether presenting multiple, thematically distinct explanations can outperform a single, generic explanation in fostering deeper student understanding.

Insights from other fields, such as computational creativity, suggest that diverse ideas may be more beneficial than relying solely on a single, high-quality option [14]. This aligns with *Variation Theory* [8]. Motivated by this, we investigate whether incorporating multiple LLM-generated diverse code explanations can enhance students' understanding of programming exercises compared to relying on generic explanations. These were compared with generic explanations, which cover the snippet in an all-purpose way, rather than highlighting a specific aspect.

To address this gap, we conducted a large-scale, between-subjects study with **971 first-year computing students**.

2 Methodology and Results

This study was conducted during regular lab sessions for a first-year engineering programming course at a large public research university. A total of 785 students provided sufficiently complete open ended (OE) responses, and all 971 completed the multiple-choice question (MCQ) items. Participants were randomly assigned (between-subjects) to receive three *Generic (G)* or three *Diverse (D)* explanations for two programming snippets.

We used GPT-4o to generate explanations. For **Generic** explanations, we issued the identical prompt, *"Explain what this code does in plain text,"* three times and showed the three plain-text summaries together. For **Diverse** explanations, we asked three targeted prompts that used different perspectives on the same snippet: an explanation of the function, goal, and concept. After each snippet, students answered one MCQ and one OE question and then rated two Likert items on perceived helpfulness and amount of information provided by the explanation. A rubric was developed to categorize answers as correct or incorrect based on the accuracy and completeness of the conceptual explanations provided by students. Inter-rater reliability was assessed amongst 5 raters using Fleiss' Kappa [4] ($k = 0.75$).

MCQ accuracy was consistently high (>87%) across all conditions, with negligible differences between diverse and generic explanations. In OE (Table 1), we observed consistent improvements for diverse explanations: *sumArray* +7.7%, *randomizeString* +8.1%, *countChar* +7.7%. A logistic regression found the difference not statistically significant (z = 2.15, $p = 0.095$). Helpfulness and perceived information amount did not differ significantly across groups ($H(7) = 4.42$, $p = 0.73$; $H(7) = 3.57$, $p = 0.83$).

Table 1: Performance comparison between Diverse and Generic across three problems (*sumArray, randomizeString, countChar*).

Category	Total	Correct	% Correct
sumArray - Diverse	403	193	47.90
sumArray - Generic	381	153	40.16
Accuracy Increase			**+7.74**
randomizeString - Diverse	187	123	65.80
randomizeString - Generic	201	116	57.70
Accuracy Increase			**+8.10**
countChar - Diverse	211	148	70.10
countChar - Generic	186	116	62.40
Accuracy Increase			**+7.70**

3 Discussion

Our results did not show any significant differences in performance between students who received diverse explanations and those who received generic ones. However, Variation Theory [8] postulates that learning is most effective when learners are exposed to key variations across examples, which allows them to discern critical features and conceptual distinctions. While our intervention aimed to introduce diversity in the explanations provided, it is possible that the sources of variation that we employed were not aligned with the dimensions that students needed to discern in order to develop their understanding.

These findings can also be interpreted through Cognitive Load Theory [3, 15]. Presenting multiple explanations, regardless of type, may have increased extraneous load, and students may have adapted by selectively ignoring some content. In that case, the lack of differences in perceived overload and helpfulness could reflect non-engagement rather than equal cognitive efficiency. It is therefore possible that the task load in both conditions was high enough that many students did not fully engage with the materials, dampening potential benefits of the diverse explanations. While the design aimed to reduce extraneous load via concise, dimension-focused segments [10] consistent with prior work on segmented materials [11], students' self-regulation strategies under load may have limited their uptake.

4 Limitations

While this study suggests that diverse LLM-generated explanations can help students understand recursion, several factors limit what we can conclude. First, all participants came from a single course taught by the same instructor, so strong baseline teaching and shared deadlines may have reduced differences between conditions. Second, students only saw two short code snippets and were tested immediately, which may not capture effects that show up with more practice or over time. Finally, high overall scores point to a possible ceiling effect; when students already perform well, small improvements are harder to detect. Future work should test broader tasks, repeated exposure, and long-term retention to better measure the impact of explanation diversity.

References

[1] Seth Bernstein, Paul Denny, Juho Leinonen, Lauren Kan, Arto Hellas, Matt Littlefield, Sami Sarsa, and Stephen Macneil. 2024. "Like a Nesting Doll": Analyzing Recursion Analogies Generated by CS Students Using Large Language Models. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1.* ACM, Milan Italy, 122–128. https://doi.org/10.1145/3649217.3653533

[2] Malcolm Corney, Sue Fitzgerald, Brian Hanks, Raymond Lister, Renee McCauley, and Laurie Murphy. 2014. 'explain in plain english' questions revisited: data structures problems. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education* (Atlanta, Georgia, USA) *(SIGCSE '14)*. Association for Computing Machinery, 591–596.

[3] Rodrigo Duran, Albina Zavgorodniaia, and Juha Sorva. 2022. Cognitive Load Theory in Computing Education Research: A Review. *ACM Trans. Comput. Educ.* 22, 4, Article 40 (Sept. 2022), 27 pages. https://doi.org/10.1145/3483843

[4] Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin* 76, 5 (1971), 378.

[5] Breanna Jury, Angela Lorusso, Juho Leinonen, Paul Denny, and Andrew Luxton-Reilly. 2024. Evaluating LLM-generated Worked Examples in an Introductory Programming Course. In *Proceedings of the 26th Australasian Computing Education Conference* (Sydney, NSW, Australia) *(ACE '24)*. Association for Computing Machinery, 77–86. https://doi.org/10.1145/3636243.3636252

[6] Juho Leinonen, Paul Denny, Stephen MacNeil, Sami Sarsa, Seth Bernstein, Joanne Kim, Andrew Tran, and Arto Hellas. 2023. Comparing Code Explanations Created by Students and Large Language Models. (2023), 124–130. https://doi.org/10.1145/3587102.3588785

[7] Raymond Lister, Elizabeth S Adams, Sue Fitzgerald, William Fone, John Hamer, Morten Lindholm, Robert McCartney, Jan Erik Moström, Kate Sanders, Otto Seppälä, et al. 2004. A multi-national study of reading and tracing skills in novice programmers. *ACM SIGCSE Bulletin* 36, 4 (2004), 119–150.

[8] Mun Ling Lo and Ference Marton. 2012. Towards a science of the art of teaching. *International Journal for Lesson and Learning Studies* 1, 1 (2012), 7–22. https://doi.org/10.1108/20468251211179678 Publisher: Emerald Group Publishing Limited.

[9] Stephen MacNeil, Andrew Tran, Arto Hellas, Joanne Kim, Sami Sarsa, Paul Denny, Seth Bernstein, and Juho Leinonen. 2023. Experiences from Using Code Explanations Generated by Large Language Models in a Web Software Development E-Book. In *Proc. SIGCSE'23*. ACM, 6 pages.

[10] Lauren E Margulieux, Mark Guzdial, and Richard Catrambone. 2012. Subgoal-labeled instructional material improves performance and transfer in learning to develop mobile applications. In *Proceedings of the ninth annual international conference on International computing education research.* 71–78.

[11] Richard E. Mayer, , and Roxana Moreno. 2003. Nine Ways to Reduce Cognitive Load in Multimedia Learning. *Educational Psychologist* 38, 1 (Jan. 2003), 43–52. https://doi.org/10.1207/S15326985EP3801_6 Publisher: Routledge _eprint: https://doi.org/10.1207/S15326985EP3801_6.

[12] Nishat Raihan, Mohammed Latif Siddiq, Joanna CS Santos, and Marcos Zampieri. 2025. Large language models in computer science education: A systematic literature review. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1.* 938–944.

[13] Sami Sarsa, Paul Denny, Arto Hellas, and Juho Leinonen. 2022. Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models. In *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1* (Lugano and Virtual Event, Switzerland) *(ICER '22)*. Association for Computing Machinery, 27–43. https://doi.org/10.1145/3501385.3543957

[14] Pao Siangliulue, Kenneth C. Arnold, Krzysztof Z. Gajos, and Steven P. Dow. 2015. Toward Collaborative Ideation at Scale: Leveraging Ideas from Others to Generate More Creative and Diverse Ideas. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15)*. Association for Computing Machinery, 937–945. https://doi.org/10.1145/2675133.2675239

[15] John Sweller. 2011. Cognitive Load Theory. In *Psychology of Learning and Motivation.* Vol. 55. Elsevier, 37–76.