2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #52010

Knowledge Grapht: AI-Powered Spaced-Repetition Aid and Course Mastery Modeling

Thomas Rife, Loyola Marymount University

Thomas Rife is a senior Computer Science major with a minor in Business Administration at Loyola Marymount University. His work centers on artificial intelligence and quantum computing, exploring how data-driven and generative systems can advance intelligent modeling and decision-making. He has completed two research internships with the Naval Surface Warfare Center, investigating quantum and classical neural network performance for classification tasks. At LMU, he leads projects focused on applied AI development and computational research while serving in multiple leadership and service roles across campus.

Dr. Andrew Forney, Loyola Marymount University

Prof. Andrew Forney is an Associate Professor of Computer Science at Loyola Marymount University with research interests broadly at the intersection of cognitive psychology, artificial intelligence, and experimental design. Although his formal specialty is in the areas of Causal and Counterfactual Inference, his passion for education attempts to blend modern AI tools and analyses for the betterment of the student experience.

Knowledge Grapht: AI-Powered Spaced-Repetition Aid and Course Mastery Modeling

Abstract

Knowledge graphs have traditionally been used to effectively visualize, organize and structure course content by illustrating which concepts support and connect to one another. This work explores a web app extension of knowledge graphs, called Knowledge Grapht, aided by generative AI, and designed with future causal inference in mind to support instructors and students alike. This project enables students to utilize structured class knowledge graphs created by their instructors where each node reflects a concept and each edge encodes the pre-requisite structure. Students can take quizzes for each topic with a mix of instructor curated questions and AI generated questions with each topic's proficiency level dynamically adjusted according to quiz performance and review frequency. Students are encouraged to review topics they struggle with more frequently through a spaced-repetition schedule that emphasizes long-term retention, inspired by validated techniques like the Leitner Box. Instructors also benefit from using Knowledge Grapht as this quiz data is used to calculate and visualize population-level class statistics of concept mastery, as well as individual student proficiencies to help them navigate topics they may be struggling in. This project evaluated the integration of Knowledge Grapht in computer science courses to examine relationships between topic-level mastery and final exam performance. The platform was deployed during regular course activities, and usage data from both students and instructors were analyzed alongside performance on graded assessments. The analysis examined correlations between practice activity, concept-level proficiency, and overall course achievement. Results revealed a positive correlation between students' concept-level mastery scores and final exam performance, with higher engagement in targeted practice and spaced review associated with stronger overall course outcomes. Findings from this implementation provided insight into how AI-assisted knowledge graph systems can support personalized learning and inform future approaches to course design.

Introduction

Conceptually understanding, practicing, and receiving feedback on course material has long posed an instructional challenge for university students. Common study approaches, such as flashcards or isolated assignments, are often fragmented and unguided, making it difficult for students to develop an understanding of how course concepts relate to one another. At the same time, these approaches limit instructors' ability to provide structured, individualized practice. Because most courses are inherently organized by conceptual dependency, where earlier topics support understanding of later ones, visual representations of course structure have become increasingly valuable. Knowledge graphs offer a natural framework for representing such structure, with nodes corresponding to course topics and edges encoding prerequisite relationships. When applied to a classroom setting, these representations can help students organize their studying, make dependencies explicit, and identify gaps in their understanding. Prior work has demonstrated that knowledge graph–based representations can improve learning outcomes; however, many existing systems either conceal the graph from students or lack empirical evaluation in authentic classroom environments. Knowledge Grapht addresses these gaps by providing students with direct access to the course knowledge graph, node-level mastery feedback, and evaluation through deployment in live undergraduate courses.

Related Work

Knowledge graphs have been widely studied as a representation for structuring educational content and modeling prerequisite relationships between concepts [1]. In educational technology, such graphs have been applied to curriculum organization, personalization, and learning analytics across a range of domains [1], [2]. By encoding concepts as nodes and dependencies as directed edges, knowledge graphs provide a structured view of how topics relate and build upon one another, supporting navigation and content organization [3], [4]. However, many existing systems treat knowledge graphs primarily as instructor-facing tools or internal backend representations, rather than as interactive artifacts for students [4], [3]. As a result, learners often do not directly engage with the conceptual structure that underlies course organization.

In parallel, research in learning analytics has emphasized the value of student-facing visualizations for supporting reflection, self-regulated learning, and metacognitive awareness [5]. Dashboards designed for students, rather than exclusively for instructors, have been shown to increase learner agency by enabling learners to interpret feedback and adjust their study strategies accordingly [6]. Despite these benefits, many student-facing analytics tools present only high-level indicators such as grades, completion status, or activity counts, offering limited insight into conceptual understanding or prerequisite structure.

Adaptive learning systems and intelligent tutoring systems further aim to personalize practice by dynamically adjusting content based on student performance [7]. These systems can reduce instructor workload while providing students with immediate feedback and tailored practice opportunities. However, many AI-driven practice systems rely on latent or opaque models of student knowledge that are not visible to learners, limiting transparency and interpretability [8]. Additionally, prior work has noted that large language model–generated educational content often requires instructor oversight, as such models tend to produce generalist responses that may not align cleanly with course-specific learning objectives or conceptual scope [8], [9].

Several studies have examined classroom deployments of AI-enhanced knowledge graph systems, collecting empirical data on student engagement, learning behaviors, and perceived usefulness in authentic instructional settings [10], [11]. While these evaluations demonstrate the feasibility of deploying such systems in practice, they are often domain-specific and rely on coarse outcome measures, such as overall course performance or self-reported learning gains [10], [1]. As a result, there remains limited empirical analysis that directly connects fine-grained, topic-level mastery within a knowledge graph to concrete assessment outcomes such as exam performance.

In addition to knowledge graph–based approaches, spaced repetition and retrieval-based practice are well-established learning principles shown to improve long-term retention compared to unstructured review [12]. Retrieval practice, particularly when paired with immediate feedback, further strengthens learning and supports strong memory formation [13]. Subsequent work has demonstrated the robustness of spaced repetition effects across domains and instructional contexts [14]. Knowledge Grapht incorporates these principles heuristically by applying time-based decay to node-level mastery estimates and prioritizing review based on recent quiz performance.

Computer science courses provide a suitable context for knowledge graph–based learning systems due to the cumulative nature of many core concepts. Prior work in computer science education has shown that novice programmers frequently struggle with foundational ideas, and that misunderstandings at this level can hinder progress on more advanced tasks [15]. Empirical studies further demonstrate that students may succeed on surface-level programming tasks while lacking robust conceptual understanding, a gap that often becomes evident during cumulative assessments [16]. Together, these findings motivate instructional tools that explicitly represent conceptual structure and provide visibility into student understanding at the topic level.

Contributions & Outline

This paper makes three primary contributions:

1) We introduce Knowledge Grapht, a student-facing learning platform that represents course concepts within an explicit knowledge graph, enabling learners to visualize topic dependencies and monitor their progress at the node level.
2) The platform integrates instructor-curated and AI-generated practice questions that are explicitly linked to graph topics, allowing student performance to be contextualized within the course structure.
3) We report findings from an observational classroom deployment in two undergraduate computer science courses, examining associations between student engagement metrics, topic-level mastery estimates, and exam performance.

Background

For a curricular knowledge graph in an instructional context, several key terms must be defined. A node represents a concept or topic within a course, while a directed edge encodes a prerequisite or dependency relationship between nodes. These graphs are directed and acyclic, with nodes appearing lower in the graph depending on a greater number of prerequisite concepts. In Knowledge Grapht, the course knowledge graph is authored by the instructor rather than automatically inferred.

Spaced repetition is a well-established learning principle grounded in cognitive psychology, demonstrating that information reviewed at increasing intervals is retained more effectively than information studied in massed sessions. Techniques such as the Leitner system operationalize spaced repetition by prioritizing review of material that learners find more difficult or have not reviewed recently [17]. Modern educational systems often adapt these principles by dynamically scheduling review based on learner performance and time since last exposure, enabling individualized practice trajectories.

Estimating student knowledge states is a central challenge in adaptive learning systems. Concept-level mastery models attempt to infer a learner's proficiency on topics using observable performance metrics such as quiz responses. When concepts are organized within a graph structure, mastery estimation can be contextualized by prerequisite relationships, allowing performance on downstream topics to reflect understanding of upstream concepts. Such structured representations provide a natural foundation for future causal analysis, where relationships between learning activities, concept mastery, and assessment outcomes can be examined under explicit dependency assumptions.

Figure 1. Example of an instructor created knowledge graph

Method

This study was deployed in an observational classroom setting to examine how student interaction with Knowledge Grapht relates to course performance. Knowledge Grapht consists of an instructor-facing administrative application and a student-facing mobile application. The system was deployed in two undergraduate computer science courses: an introductory programming course and an upper-division artificial intelligence course.

In Knowledge Grapht, instructors define the structure of a course by authoring a curricular knowledge graph in which nodes represent course topics and directed edges encode prerequisite relationships. Instructors can then create lessons associated with specific topics in the graph, each paired with a single quiz.

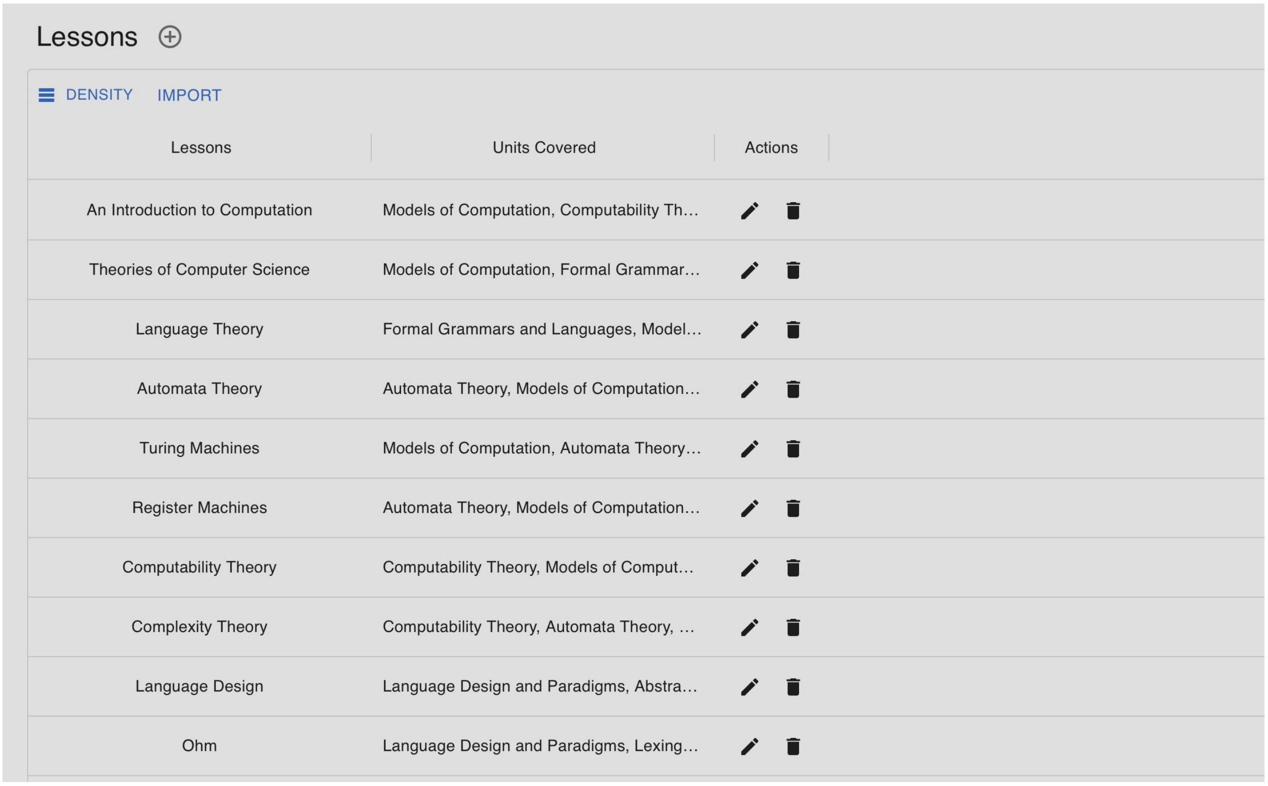

Figure 2. Instructor-facing lesson dashboard showing lesson structure and associated topics

Quizzes consist of multiple-choice questions designed to assess understanding of the lesson's associated topics. To reduce instructor workload, the system incorporates AI-assisted question generation. When creating a quiz, instructors may generate draft questions using AI, which leverages existing human-authored questions as examples to align generated content with the intended nodes and difficulty level. All AI-generated questions are reviewed and optionally edited by instructors prior to being added to quizzes, and are treated equivalently to instructor-authored questions throughout the study.

Students interact with Knowledge Grapht through a mobile application that presents lessons and quizzes defined by the instructor. Students can view which nodes are covered by each lesson and complete associated quizzes as part of their regular coursework.

Upon quiz submission, students receive immediate feedback on their responses. Quiz interaction data are logged at the node level, allowing student practice activity to be associated with specific concepts in the knowledge graph. This data is later used to estimate node-level mastery and to support spaced review scheduling.

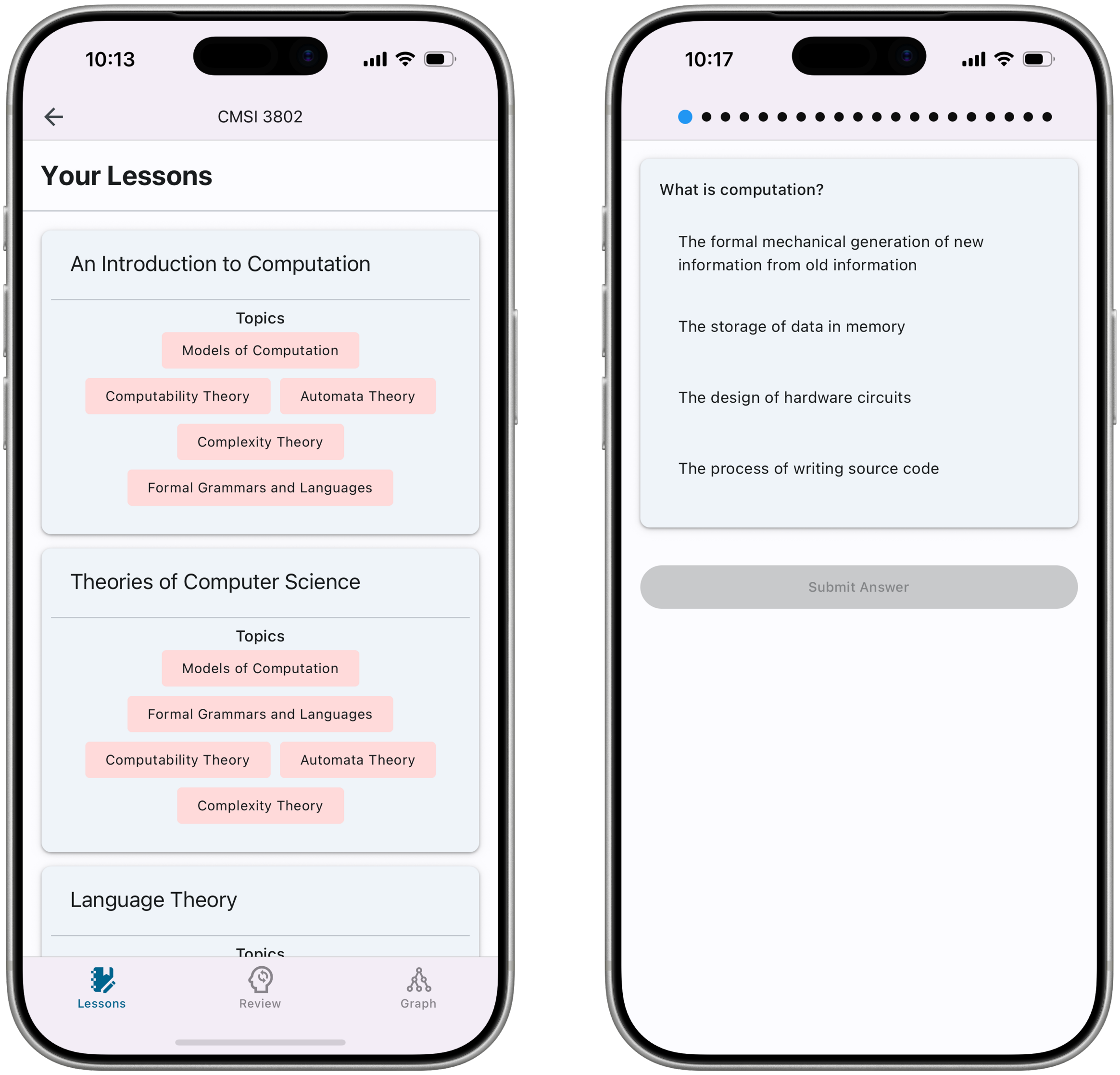

Figure 3. Student-facing lesson selection and quiz interfaces

Student performance on quizzes is used to update node-level mastery estimates. For each student s and topic (node) v, mastery is represented by a score $M_{s,v}(t) \in [0, 1]$ at time t. Let $C_{s,v}$ be the number of correct answers student s has submitted on questions tagged with node v, and let $N_{s,v}$ be the total number of attempted answers tagged with v (with $N_{s,v} \geq 1$). Let $\Delta t_{s,v}$ be the time elapsed since student s last answered any question tagged with v. We define the displayed mastery as:

$$M_{s,v}(t) = \frac{C_{s,v}}{N_{s,v}} \cdot 2^{-\Delta t_{s,v}/h},$$

where h is a fixed half-life (shared across nodes) controlling how quickly mastery decays without review. After a quiz submission at time t, for each node v that appears in that quiz, we update $C_{s,v}$ and $N_{s,v}$ using the new responses and reset $\Delta t_{s,v} = 0$. The system uses a straightforward accuracy-based mastery model rather than more complex approaches such as knowledge tracing. This design prioritizes transparency, allowing both students and instructors to easily understand how mastery is computed from observed quiz performance. Given the short deployment period and limited number of interactions per student, more parameterized models such as Bayesian or neural knowledge tracing were not used, as they typically require larger volumes of data to produce stable estimates. The accuracy-based formulation provides a robust and intuitive way to visualize node-level understanding within the knowledge graph.

In addition to student-facing functionality, Knowledge Grapht provides instructors with aggregate analytics derived from student quiz interactions. The administrative interface displays population-level statistics for each node in the course knowledge graph, including the number of students who attempted questions for a node, average node mastery, and recency of review. These metrics allow instructors to identify concepts where students collectively struggle, observe patterns of engagement across the course, and inform decisions about review sessions or exam preparation. Aggregate metrics were computed from the same question-level logs used in the student-facing analysis and were updated dynamically as students completed quizzes.

The system was deployed in two undergraduate computer science courses: an introductory programming course and an upper-division artificial intelligence course. Knowledge Grapht was integrated into regular course activities during the final two weeks of the academic term as students prepared for their final examinations.

A total of 32 students participated in the study, including five students from the introductory course and 27 students from the upper-division course. Participation was voluntary, and all interaction data were collected passively during normal use of the platform. Informed consent was obtained through the student mobile application, and the study protocol was approved by the institution's Institutional Review Board (IRB).

During deployment, the system logged interaction data for each student. For every quiz question attempt, the selected answer, correctness, and timestamp were recorded. Question-level results were aggregated by node to track review behavior, including total attempts, number of correct responses, and time since last review. This data were used to construct node-level mastery estimates and engagement measures for subsequent analysis.

To examine the relationship between student engagement with Knowledge Grapht and learning outcomes, we constructed a student-level analysis dataset aggregating practice activity and performance metrics. For each student, we computed (1) the number of distinct nodes reviewed, defined as the count of knowledge graph nodes for which the student attempted at least one tagged practice question; (2) total practice volume, measured as the total number of questions attempted; and (3) average node mastery, computed as the mean accuracy across attempted nodes, where node-level mastery was defined as the ratio of correctly answered questions to total attempts. Each quiz question was tagged with one or more nodes from the course knowledge graph, and a student's response to a question contributed to the mastery estimates of all associated nodes.

Student's learning outcomes were measured using their final exam scores and associations between practice metrics and exam performance using both Pearson and Spearman correlation coefficients. Pearson correlation was used to assess linear relationships between engagement metrics and exam scores, while Spearman correlation was used to assess monotonic associations and reduce sensitivity to non-normality and outliers. We first correlated their final exam scores with their individual overall node mastery data across all questions answered, using both Spearman's ρ and Pearson's r.

Each final exam question was associated with a predefined set of topics, and student mastery on those topics was aggregated and correlated with performance on the corresponding exam question using Spearman correlation.

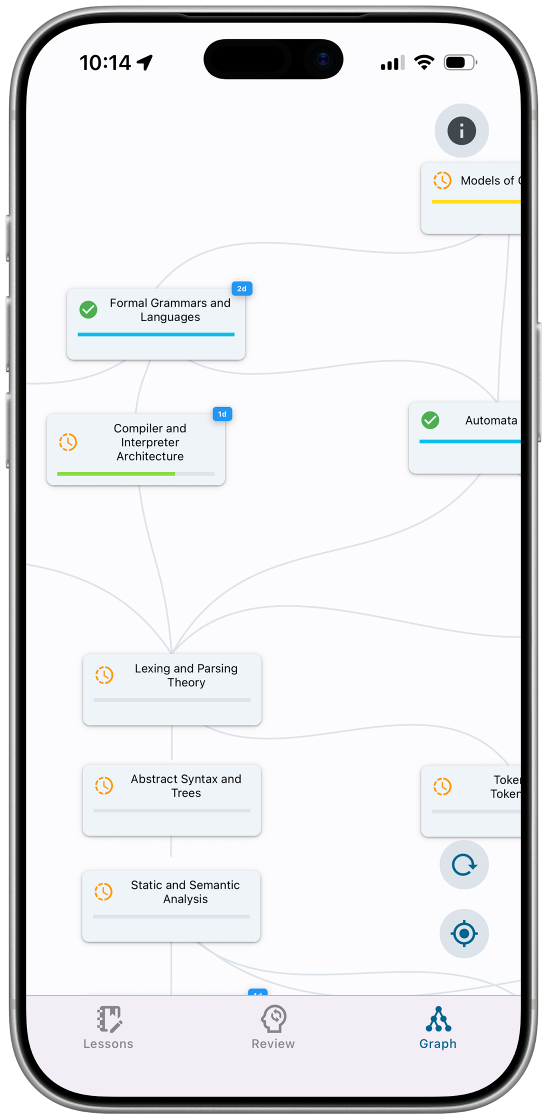

Figure 4. Student-facing knowledge graph showing node dependencies and mastery levels.

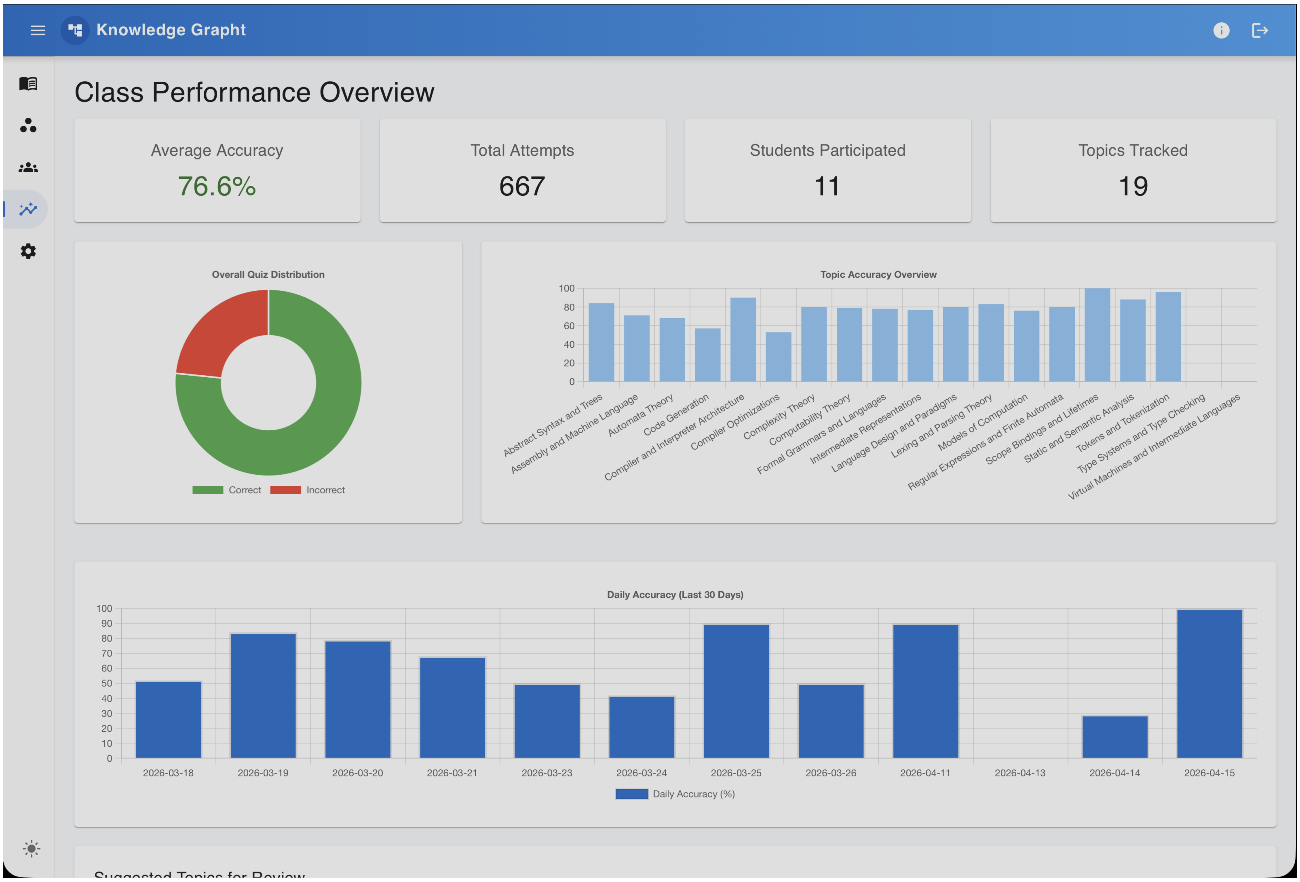

Figure 5. Instructor aggregate metrics from student quiz performance

Results

Table I
DESCRIPTIVE STATISTICS OF STUDENT ENGAGEMENT AND PERFORMANCE

Variable	**Mean**	**SD**	**Min**	**Max**
Upper-division AI course (n = 27)				
Final exam score	55.76%	20.14%	20.5%	98.0%
Nodes reviewed	8.93	3.79	3.0	13.0
Total question attempts	39.48	30.53	3.0	118.0
Average node mastery	67.41%	24.61%	0.0%	100.0%
Lower-division intro course (n = 5)				
Final exam score	71.00%	5.26%	66.25%	80.0%
Nodes reviewed	10.40	6.54	5.00	18.0
Total question attempts	48.40	31.65	17.00	87.0
Average node mastery	84.44%	12.32%	65.76%	97.5%

A total of 27 students in the upper division AI course answered at least one quiz question to practice for their final exam. The mean final exam score for the class was 55.76% with a standard deviation of 20.14%. There was a broad min-max range of final exam scores for the AI class.

Students from the AI course reviewed on average nine nodes while studying using Knowledge Grapht. This includes repeat questions in addition to new attempts. They achieved 67.41% accuracy on average for all nodes studied, with mastery being calculated by averaging scores across attempted nodes, and nodes with no attempts being excluded from aggregate mastery results.

As for the introductory course, a total of five students answered at least one question, with a mean score of 71% on their exam and an average node mastery of 84.44%. Students in this section attempted a max of 87 total questions, and a minimum of 17.

Table II
CUMULATIVE CORRELATIONS BETWEEN ENGAGEMENT METRICS AND FINAL EXAM PERFORMANCE

Course	**Metric**	**Pearson** r	p**-value**	**Spearman** ρ	p**-value**
AI	Nodes reviewed	0.26	0.193	0.26	0.195
AI	Total question attempts	0.22	0.273	0.15	0.455
AI	Average node mastery	0.34	0.083	0.37	0.059
Intro	Nodes reviewed	0.32	0.598	0.21	0.741
Intro	Total question attempts	0.50	0.393	-0.13	0.833
Intro	Average node mastery	-0.70	0.187	-0.36	0.553

Table 2 reports Pearson and Spearman correlation coefficients and their respective p-values between Knowledge Grapht engagement metrics and final exam performance for both the upper-division AI course and the introductory class. For each course, correlations were computed between final exam scores and three engagement measures: number of nodes reviewed, total number of practice question attempts, and average node mastery.

Table III
PER-QUESTION SPEARMAN CORRELATIONS BETWEEN NODE MASTERY AND EXAM PERFORMANCE FOR UPPER-DIVISION AI COURSE

Exam Question and Associated Topics	**Spearman** ρ	p**-value**
Question 1 *(n = 26)* (Sampling / Approximate Inference)	0.06	0.770
Question 2 *(n = 23)* (HMMs and Filtering)	-0.25	0.241
Question 3 *(n = 24)* (Naive Bayes / Parameter Estimation)	-0.11	0.598
Question 4 *(n = 13)* (Linear Perceptrons)	0.05	0.882
Question 5 *(n = 11)* (Logistic Regression)	0.40	0.218
Question 6 *(n = 19)* (Supervised Learning)	-0.08	0.744

In addition to the cumulative correlations reported in Table 2, each of the six questions on the upper-division AI final exam were tagged with the nodes assessed. Spearman correlation coefficients and p-values were computed between each student's score on an individual exam question and their corresponding node mastery on Knowledge Grapht for the nodes associated with that question. Because student practice varied across nodes, not all participating students had prior practice data for every exam question, resulting in different sample sizes across questions.

Per question correlation analyses was not conducted for the intro class because only five students participated and many exam questions did not have sufficient corresponding practice data.

Discussion

The two courses examined differed substantially in both engagement patterns and outcome variance. The upper-division AI course exhibited substantially greater variance in exam performance and a larger sample size, making it more informative for examining relationships between engagement and assessment outcomes.

Across the three aggregate metrics in Table 2, moderate positive correlations were observed between student exam performance and (i) the number of nodes reviewed, (ii) total question attempts, and (iii) average node mastery, with average node mastery exhibiting the strongest association. While none of the observed correlations reached conventional thresholds for statistical significance, the p-value for average node mastery approached significance in the upper-division course, suggesting a potentially meaningful trend. It remains possible that engagement with Knowledge Grapht contributed to improved exam performance, but alternative explanations such as prior ability or motivation cannot be ruled out.

Importantly, differences between engagement quantity and engagement quality emerged. The correlation between total question attempts and final exam score was weaker overall than the correlation between higher average node mastery and final exam score. This distinction suggests that targeted, accurate practice, as captured by node-level mastery, may be more informative than raw practice volume alone, highlighting the value of explicit mastery modeling in knowledge graph–based systems over simple activity counts.

However, aggregate correlations mask substantial variation in how practice relates to performance across individual exam questions. In Table 3, the question-level analysis shows diverse relationships between node mastery and exam performance, with some questions exhibiting near-zero or negative correlations and others showing stronger positive associations. Although these per-question correlations were not statistically significant, their variability suggests that the effectiveness of practice depends on alignment between practiced concepts and assessed material rather than overall engagement.

Several factors may explain weaker or negative per-question associations. First, measurement noise is introduced by multiple-choice practice questions, where correct responses may occasionally result from guessing rather than understanding. Second, node breadth plays a role: broad nodes such as “Supervised Learning” or “Sampling” encompass multiple subskills, causing node-level mastery to average over student competencies. In contrast, more narrowly defined nodes such as “Logistic Regression” might align more directly with specific exam questions, yielding stronger correlations. Additionally, instructor-authored practice questions may have differed in structure or emphasis from corresponding exam questions, limiting the extent to which practiced skills transferred to assessed problems. Together, these findings suggest that fine-grained node modeling and explicit prerequisite structure may be important for accurately capturing the relationship between practice and assessment performance.

Extension: Causal Review

While the present study evaluates mastery-based review using accuracy-derived node-level estimates, the underlying knowledge graph structure also enables more advanced review strategies based on prerequisite relationships. Since the initial deployment in Fall 2025, Knowledge Grapht has been extended to incorporate a causal review prioritization mechanism that estimates the downstream impact of improving mastery on individual nodes. Rather than selecting nodes solely based on lowest observed mastery, this approach prioritizes nodes whose improvement is expected to produce the largest gains in dependent concepts. By leveraging these dependencies, causal review prioritization aims to better align student practice with the conceptual structure of the course. While this system is currently deployed in the platform, it was not evaluated within the scope of the present study.

In practice, causal review prioritization is implemented as a dynamic ranking procedure triggered after each student interaction. When a student completes a lesson or quiz, their updated node-level mastery estimates are sent to a backend service that evaluates the expected downstream impact of improving each node in the knowledge graph. This process produces a ranked set of candidate review nodes, from which the top recommendations are presented to the student as review priorities. When a student initiates a review quiz session, the system generates a targeted quiz by aggregating questions tagged with the highest-ranked node. Upon completion of the review quiz, mastery estimates are updated and the ranking process is repeated, enabling iterative, feedback-driven adjustment of review priorities.

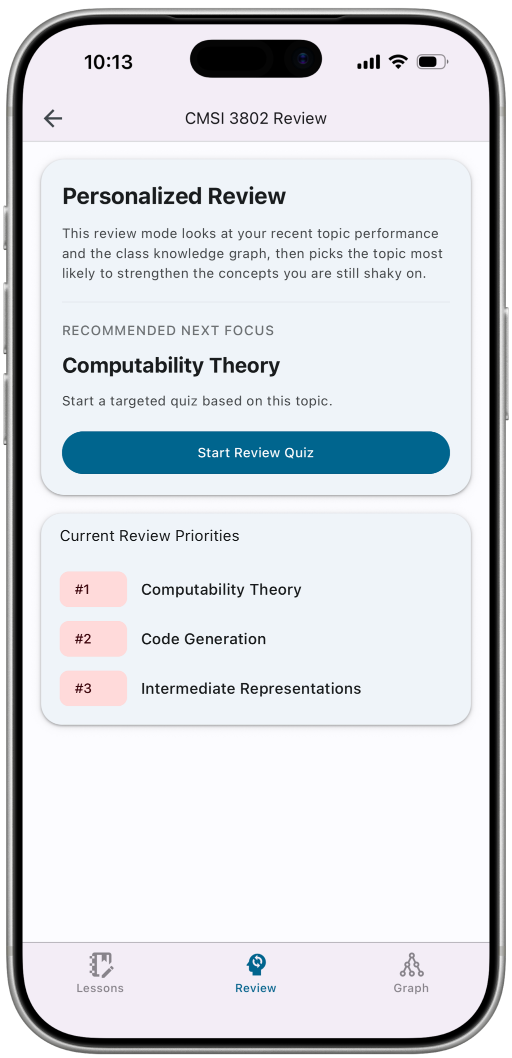

Figure 6. Student-facing causal review interface showing recommended next focus and prioritized review nodes

This formulation is illustrated in the following example, which demonstrates how causal reasoning can alter review priorities compared to mastery-based heuristics alone.

Table IV
EXAMPLE PRIOR MASTERY DISTRIBUTIONS

Node	P(Mastered = 1)
Conditionals (C)	0.30
Variables (V)	0.50

Table V
EXAMPLE CONDITIONAL PROBABILITY TABLE FOR MASTERING DEPENDENT NODE (F)

C	V	F	$P(F = 1 \mid C, V)$
0	0	1	0.20
0	1	1	0.70
1	0	1	0.40
1	1	1	0.80

Although a lesser proportion of students have mastered the node *C* overall, increasing mastery of *V* produces a larger marginal improvement in the probability of mastering *F*. For example, holding conditions fixed at $C = 0$, increasing variable mastery from 0 to 1 increases $P(F = 1)$ from 0.20 to 0.70, a gain of 0.50. In contrast, holding variables fixed at $V = 0$, increasing condition mastery raises $P(F = 1)$ from 0.20 to 0.40, a gain of only 0.20.

In this example, selecting review nodes solely based on lowest observed mastery would prioritize conditions over variables, despite variables having a larger causal influence on the downstream concept. This illustrates how causal reasoning over a knowledge graph could enable more effective and targeted review recommendations than mastery-based heuristics alone. Such causal estimates could be incorporated into instructor-facing aggregate metrics, guiding in-class review toward prerequisite nodes with the greatest downstream impact on student understanding.

Limitations

This study has several limitations that should be considered when interpreting the results. The first is that deployment period was relatively short. Knowledge Grapht was available to students for approximately two weeks prior to final examinations, limiting both the amount of practice students could complete and the extent to which spaced repetition effects could emerge. A longer deployment would allow for more stable mastery estimates and richer engagement patterns.

Sample sizes were also uneven and limited, particularly for the introductory course. While the upper-division AI course provided sufficient variation to support correlational analysis, the small number of introductory students restricts the strength and generalizability of findings for that course. Additionally, per-question analyses are limited by the small number of observations available for individual exam questions, reducing the reliability of these estimates.

Mastery estimates were derived from multiple-choice question performance, which introduces inherent noise due to guessing and question difficulty variation. Node mastery values therefore approximate, but do not fully capture, underlying conceptual understanding. Mastery was also averaged across nodes, potentially masking differences in subnode competence within broader conceptual areas.

Finally, the analysis focused on correlational relationships between practice metrics and exam performance. While Knowledge Grapht's knowledge graph structure enables reasoning about prerequisite relationships, causal inference methods were not implemented in this study. Future work is required to evaluate whether causal or dependency-aware models can better inform review prioritization and learning outcomes.

Future Directions

Future work will focus on empirically evaluating causal review prioritization in classroom settings, including its impact on student learning outcomes and alignment with assessment performance. A longer-term deployment across multiple courses would enable more stable mastery estimates and allow investigation of how review strategies influence retention over time. Additionally, expanding the system beyond computer science courses would test the generalizability of knowledge graph–based and causal review in domains with different conceptual structures.

Conclusion

The lack of tools that effectively visualize structured conceptual dependencies and support targeted review has long posed a challenge for student preparation. Knowledge Grapht addresses this gap by providing a student-facing knowledge graph that displays node-level mastery derived from tagged quiz questions, alongside instructor-facing aggregate analytics to support in-class review. The system was deployed in two undergraduate courses, where observational correlational analyses were conducted between student practice behavior and exam performance. Across engagement metrics, average node mastery exhibited the strongest association with exam outcomes, suggesting that targeted, accurate practice may be more informative than raw practice volume alone. By making conceptual structure and mastery visible to both students and instructors, knowledge graphs offer a promising foundation for supporting more focused review. Future work incorporating causal reasoning over prerequisite relationships may further improve the effectiveness of both student study strategies and instructor-led intervention.

Acknowledgements

I would like to thank the instructors and students who participated in this study for their time and engagement, Diego Acosta and Aiden Dionisio for creating the initial versions of the Knowledge Grapht platform and Dr. Andrew Forney for guidance and feedback throughout the development of this project.

References

[1] B. Abu-Salih and S. Alotaibi, "A systematic literature review of knowledge graph construction and application in education," *Heliyon*, 2024. [Online]. Available: https://doi.org/10.1016/j.heliyon.2024.e25383

[2] Y. Hou, B. Liu, Q. Fan, and J. Zhou, "Research on the application mode of knowledge graph in education," *Proceedings of the 6th International Conference on Educational Technology Management*, 2024. [Online]. Available: https://doi.org/10.1145/3637907.3637976

[3] P. Sun and L. Gu, "Fuzzy knowledge graph system for artificial intelligence-based smart education," *Journal of Intelligent & Fuzzy Systems*, 2021. [Online]. Available: https://doi.org/10.3233/JIFS-201770

[4] L. Lin and F. Wang, "Adaptive learning system based on knowledge graph," *Proceedings of the 9th International Conference on Education and Training Technologies (ICETT)*, 2023. [Online]. Available: https://doi.org/10.1145/3599640.3599647

[5] K. Verbert, E. Duval, J. Klerkx, S. Govaerts, and J. L. Santos, "Learning analytics dashboard applications," *American Behavioral Scientist*, 2013. [Online]. Available: https://doi.org/10.1177/0002764213479363

[6] R. Bodily and K. Verbert, "Review of research on student-facing learning analytics dashboards and educational recommender systems," *IEEE Transactions on Learning Technologies*, 2017.

[7] C. Dong, Y. Yuan, K. Chen, S. Cheng, and C. Wen, "How to build an adaptive ai tutor for any course using knowledge graph-enhanced retrieval-augmented generation (kg-rag)," *arXiv preprint arXiv:2311.17696*, 2025. [Online]. Available: https://arxiv.org/abs/2311.17696

[8] S. Elkins, E. Kochmar, J. C. K. Cheung, and I. Serban, "How teachers can use large language models and bloom's taxonomy to create educational quizzes," *arXiv preprint arXiv:2401.05914*, 2025. [Online]. Available: https://arxiv.org/abs/2401.05914

[9] E. Kasneci, K. Sessler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier, S. Krusche, G. Kutyniok, T. Michaeli, C. Nerdel, J. Pfeffer, O. Poquet, M. Sailer, A. Schmidt, T. Seidel, M. Stadler, J. Weller, J. Kuhn, and G. Kasneci, "ChatGPT for good? On opportunities and challenges of large language models for education," *Learning and Individual Differences*, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1041608023000195

[10] T. Chen, L. Gong, N. Wang, and X. Ling, "Integrating ai teaching assistants and knowledge graphs in college english listening and speaking education: A study of evaluation and practice," *Proceedings of the 2025 International Conference on Digital Education and Information Technology*, 2025. [Online]. Available: https://doi.org/10.1145/3732299.3732316

[11] Q. Liu, Z. Wang, and Q. Yang, "From barriers to bridges: A tpack-xl framework for knowledge graph integration in higher education," *Proceedings of the 2nd Guangdong-Hong Kong-Macao Greater Bay Area Education Digitalization and Computer Science International Conference*, 2025. [Online]. Available: https://doi.org/10.1145/3746469.3746588

[12] N. J. Cepeda, E. Vul, D. Rohrer, J. T. Wixted, and H. Pashler, "Spacing effects in learning: A temporal ridgeline of optimal retention," *Psychological Science*, 2008. [Online]. Available: https://doi.org/10.1111/j.1467-9280.2008.02209.x

[13] H. L. RoedigerIII and J. D. Karpicke, "Test-enhanced learning: Taking memory tests improves long-term retention," *Psychological Science*, 2006. [Online]. Available: https://doi.org/10.1111/j.1467-9280.2006.01693.x

[14] S. H. K. Kang, "Spaced repetition promotes efficient and effective learning: Policy implications for instruction," *Policy Insights from the Behavioral and Brain Sciences*, 2016. [Online]. Available: https://doi.org/10.1177/2372732215624708

[15] A. Robins, J. Rountree, and N. Rountree, "Learning and teaching programming: A review and discussion," *Computer Science Education*, 2003. [Online]. Available: https://doi.org/10.1076/csed.13.2.137.14200

[16] R. Lister, E. S. Adams, S. Fitzgerald, W. Fone, J. Hamer, M. Lindholm, R. McCartney, J. E. Moström, K. Sanders, O. Seppälä, B. Simon, and L. Thomas, "A multi-national study of reading and tracing skills in novice programmers," *SIGCSE Bull.*, 2004. [Online]. Available: https://doi.org/10.1145/1041624.1041673

[17] S. Leitner, *So lernt man lernen*. Herder, 1974.