2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION
Paper ID #52840

# A Rigorous Evaluation of Agentic Large Language Model Workflows for Multiple-Choice Question Generation in Advanced Engineering Courses

**Eric Robert Tourigny, University of Calgary**
**Armando Miguel Acosta Simancas, University of Calgary**
**Dr. Denis Onen, University of Calgary**

Dr. Onen is an Associate Professor Teaching and a registered professional engineer with a broad industrial background in electrical engineering

**Miriam Nightingale, University of Calgary**
**Ethan MacDonald, University of Calgary**

# A Rigorous Evaluation of Agentic Large Language Model Workflows for Multiple-Choice Question Generation in Advanced Engineering Education

abstract

Large language models (LLMs) present a transformative opportunity for automating multiple-choice question (MCQ) generation in engineering courses, where maintaining large, high-quality question banks aligned with Bloom's Taxonomy remains a labor-intensive process. However, while recent work demonstrates that LLMs can generate pedagogically relevant MCQs, there remains limited understanding of how agentic workflows, those that allow models to reason and act iteratively, affect question quality, validity, and alignment with learning outcomes. This study rigorously evaluates a modular agentic workflow that applies an LLM (Gemini-2.5-Flash) to preprocess course materials, generate topics, create MCQs, and iteratively refine them through structured feedback loops.

An automated evaluation framework was developed to assess question quality across four pedagogically grounded dimensions: (1) item writing flaws (IWFs), using a 19-factor rubric adapted from nursing education research; (2) diversity, measured via Distinct-3 analysis; (3) content alignment, measured by semantic similarity to lecture transcripts; and (4) Bloom's Taxonomy alignment, evaluated using a separate LLM-based classifier achieving 77.4% test accuracy. Experimental data were drawn from lecture transcripts of a third-year FPGA design course, with over 1,500 questions generated across all six Bloom levels.

Results indicate that agentic iteration markedly improves MCQ quality, with most gains achieved in the first two refinement passes. Topic-guided generation enhances quality for small batch sizes but degrades performance when too many topics are processed concurrently, likely due to attention diffusion. Preprocessing transcripts had negligible impact, suggesting modern LLMs are robust to minor transcription noise. Overall, 54% of generated questions met the pedagogical standard of fewer than two IWFs, with strong performance at lower Bloom levels but diminishing reliability for higher cognitive tasks.

These findings demonstrate that agentic LLM workflows can significantly enhance automated MCQ generation when carefully structured and limited to effective iterative stages. The study contributes an open, reproducible framework for systematically evaluating AI-generated

assessment items and provides empirically grounded design principles for integrating LLMs into large-scale engineering education assessment pipelines.

## 1 introduction

Multiple-choice questions (MCQs) are a widely used assessment method in both engineering education and across many other disciplines because they enable scalable, objective grading and rapid feedback for formative assessments [1]. They can effectively assess learning objectives across Bloom's Taxonomy [2], from simple recall to complex evaluation [3]. Targeting different levels of Bloom's Taxonomy allows testing and developing different skills, supporting the need for critical thinking and careful analysis in engineering education.

However, keeping MCQs current, particularly in sufficient quantity to provide parallel exam forms (different versions of a particular exam), is a labor-intensive and time-consuming process. Care must be taken to avoid item writing flaws (IWFs), such as cues that allow test-wise students to guess the correct answer or unnecessarily confusing information, which reduce question validity and discrimination. Due to limited time and training, many questions, including those used in high-stakes assessments, contain IWFs [4].

Recent advancements in large language models (LLMs) have presented a major opportunity to produce MCQs at scale. LLMs can fluently generate natural language, follow instructions, and extract complex concepts from source documents, giving them the potential to quickly generate MCQs from course contents while avoiding common IWFs and aligning with Bloom's Taxonomy. However, instructors are cautious: concerns remain about question quality, diversity, and pedagogical alignment [5].

Agentic methods are an emerging technique in which the LLM is given agency and is allowed to reason, act, and interact with its environment. In particular, multi-step reasoning and LLM-based control flow allow LLMs to follow a structured reasoning and decision-making process [6]. These workflows have already led to significant performance improvements in other domains, such as robotics [7], making them promising techniques to apply to MCQ generation. However, a rigorous evaluation of the impact of these techniques is required to justify the additional LLM usage involved in agentic methods and to ensure the resulting questions are pedagogically useful.

To address this gap, our research investigates which steps in an agentic workflow materially improve MCQ quality, based on an automated evaluation framework for engineering education.

We begin by developing an agentic workflow that uses an LLM to preprocess course contents, create topics, generate questions, and iteratively refine the results. We then develop an evaluation framework assessing question diversity, Bloom's level, alignment with course contents, and IWFs. We test each step of the agentic workflow to characterize its overall performance and identify the best combination of steps. We expect that using an agentic workflow will allow higher quality questions to be created.

## 2 background and related work

### 2.1 mcq generation

Even prior to the development of LLMs, many different studies investigated the possibility of generating MCQs, both with rules-based approaches and neural networks [8]. However, the introduction of LLMs has led to a major advancement in the field, with many studies leveraging them to generate MCQs [9 - 17]. Generally, the LLM is instructed, with a carefully engineered prompt, to generate MCQs based on some course contents (like textbook sections). Prior work [16] – [18] uses an agentic workflow to generate question answer-pairs, though not MCQs. However, to our knowledge there is a gap in the research, as iterative, self-feedback-based agentic methods have not been applied to MCQ generation, and the impact of particular steps has not been evaluated.

### 2.2 mcq evaluation

While the gold standard for MCQ evaluation remains manual assessment by a subject matter expert, several frameworks and metrics have been developed to support their evaluations and allow scalable analysis of large datasets. IWF rubrics have been developed to catch common issues that can cue students to the correct answer or cause unnecessary confusion, both of which decrease the ability of questions to discriminate between knowledgeable and uninformed students. Diversity, the range of vocabulary and content present, may be evaluated across a question set to assess whether questions cover a wide range of material. Finally, the difficulty (cognitive complexity) of questions may be characterized with Bloom's Taxonomy to ensure they align with course objectives.

## 3 methodology

### 3.1 agentic workflow

Questions were generated with an agentic workflow, applying an LLM (Gemini-2.5-Flash) to complete several modular steps, each of which could be included or excluded, for experimental verification. The overall process is shown in Figure 1. Response schemas were used to structure the output of each step (except preprocessing) as JSON. This supported the easy parsing of LLM output as questions and allowed for guided chain-of-thought reasoning within a particular step (e.g. having the question evaluator generate general thoughts before its final evaluation). Prompts were manually written then finetuned based on the manual and automatic evaluation of generated questions. Prompts are structured as JSON, as this was found to give the best performance. Gemini-2.5-Flash's large context window was leveraged to provide the entirety of a lecture transcript as context, rather than relying on subsets, as in retrieval-augmented generation (RAG). Each question and topic is instructed to target a particular Bloom's Taxonomy level, by instructing the LLM to target that level based on an explanation and examples.

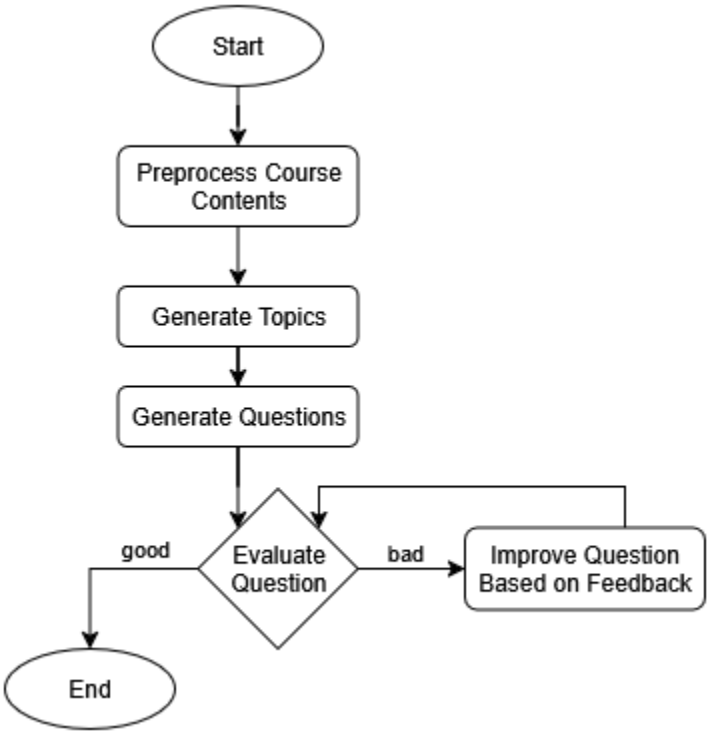


Fig. 1. A workflow diagram, showing the overall agentic workflow if all steps are included, with feedback-driven iteration. For simple iteration, the evaluation is based on whether the LLM makes any changes.

In the preprocessing step, the LLM is provided with the course contents and instructed to address any transcription errors and filler words while maintaining the substance of the material. In our

case, the course contents are lecture transcripts. The output string is then used in place of the original course contents in all later steps. This is done on the basis the LLM should be able to fix inaccuracies in text generated from recordings or extracted from documents, providing later steps with high quality context.

In the topic generation step, the LLM is given the course contents, a target Bloom level, and context on the meaning of that Bloom level, and is instructed to generate as many topics as the number of questions that will be generated (the batch size). Topics are generated on the basis that they should improve the coverage and quality of the questions by guiding the LLM to identify a wide range of topics. This should help avoid very similar questions, particularly when many are generated at once. As an example, some of the topics generated from the transcript of an introductory lecture on FPGAs at the comprehension level were:

- Define what a soft core refers to in FPGA design (A CPU implemented using the FPGA's logic fabric)
- Identify the key characteristic of RISC-V that distinguishes it from proprietary instruction sets (It has an open-source instruction set)
- Identify the relationship between flexibility and performance for ASICs on the continuum of configurable computing (Very high performance but limited flexibility)

Then, the LLM is prompted to generate as many questions as the batch size (all in one go), at the target Bloom level (based on an explanation) while avoiding IWFs. If topics were generated, it is instructed to make each question correspond to a particular topic. For the second topic from above, the LLM generated the question:

What fundamental characteristic distinguishes the RISC-V instruction set architecture from many proprietary instruction sets?

A) Its design is openly available for public use
B) It consistently achieves higher clock frequencies
C) It always results in smaller circuit implementations
D) It includes built-in analog-to-digital converters
E) It consumes less power than other architectures

Finally, the generated questions may be individually iterated on. For this step, we developed and evaluated two different methods: simple iteration and feedback-driven iteration. For simple iteration, the LLM is given the question and instructed to fix any IWFs present, which is repeated until no changes are made. Thus, the LLM decides the question is acceptable by making no changes. Meanwhile, for feedback-driven iteration the LLM first acts as an expert question evaluator (focusing on IWFs). If the question is acceptable, iteration ends, otherwise feedback is generated and used in the next step to fix the question, at which point the process is repeated.

Both methods are capped at a max number of iterations, at which point the question is automatically accepted, to avoid endless repetition.

As an example, the earlier question was modified with feedback-driven iteration. The LLM generated feedback that “Distractor B uses the strong qualifier ‘consistently,’ which makes a broad and unsupported claim,” “Distractor C uses the absolute term ‘always,’ making it an overly general statement that is typically incorrect in multiple-choice questions” and “Distractor D is inconsistent with the question stem, the question asks about a characteristic of the RISC-V *instruction set architecture* (a software/design concept), but the option describes a hardware feature (built-in analog-to-digital converters).” Then, based on this feedback, the LLM modified the question, resulting in:

What fundamental characteristic distinguishes the RISC-V instruction set architecture from many proprietary instruction sets?

A) Its design is openly available for public use
B) It is optimized for achieving the highest possible clock speeds
C) It generally results in very compact hardware designs
D) It follows a Complex Instruction Set Computing (CISC) design philosophy
E) It is specifically optimized for ultra-low power embedded devices

### 3.2 evaluation framework

An automated evaluation framework was developed that combines IWFs, diversity, alignment with course contents, and alignment with the target Bloom’s Taxonomy level. IWFs are automatically evaluated with the rules and LLM call-based method from [5] which implements a proven 19 IWF rubric developed for manual evaluations [4]. The evaluation was adapted to use Gemini-2.5-Flash-Lite, with a temperature of 0. Each IWF is given a binary score, with a 1 indicating the item is flawed. Question diversity is evaluated based on the Distinct-3 score from [5], which measures the number of unique 3-grams per MCQ. However, for consistency with the other metrics (where a higher score indicates a flaw), the question uniformity (uniformity = 1 - distinct3) was used instead, with a higher score indicating questions are less diverse and thus likely test fewer concepts or use repetitive phrases. The alignment of questions with the course contents is examined with the maximum cosine similarity to subsections of the lecture transcripts. A lower misalignment (misalignment = 1 - max cosine similarity) is better, as it indicates the questions are more related to the course contents. Finally, the actual Bloom’s level is evaluated, by providing an LLM with the question stem and context on each Bloom’s Taxonomy level. This is compared to the target Bloom level, with a score of 1 indicating the levels are different, while 0 indicates a match. Finally, a composite score is defined by summing each score, giving the total number of flaws.

### 3.3 experiments

First, the effectiveness of the Bloom's Taxonomy evaluation was investigated based on the set of 2337 free response questions (not MCQs) with expertly rated Bloom's levels from [15]. This was split into 200 training questions used to manually finetune the prompt and 2137 test questions used to evaluate the accuracy of the system's ratings.

Next, the impact of each step in the workflow was evaluated by measuring its effect on final question quality using the proposed evaluation framework. Questions were generated across Bloom's Taxonomy levels and lecture transcripts from a third-year Digital Systems Design course.

## 4 results

### 4.1 evaluation of bloom's taxonomy ratings

A training accuracy of 83.5% and a testing accuracy of 77.4% were achieved, indicating that the Bloom's Taxonomy evaluation is reasonably reliable for determining the true Bloom's Taxonomy level of a particular question. Additionally, most misses were to relatively similar levels of Bloom's Taxonomy, as shown in Figure 2.

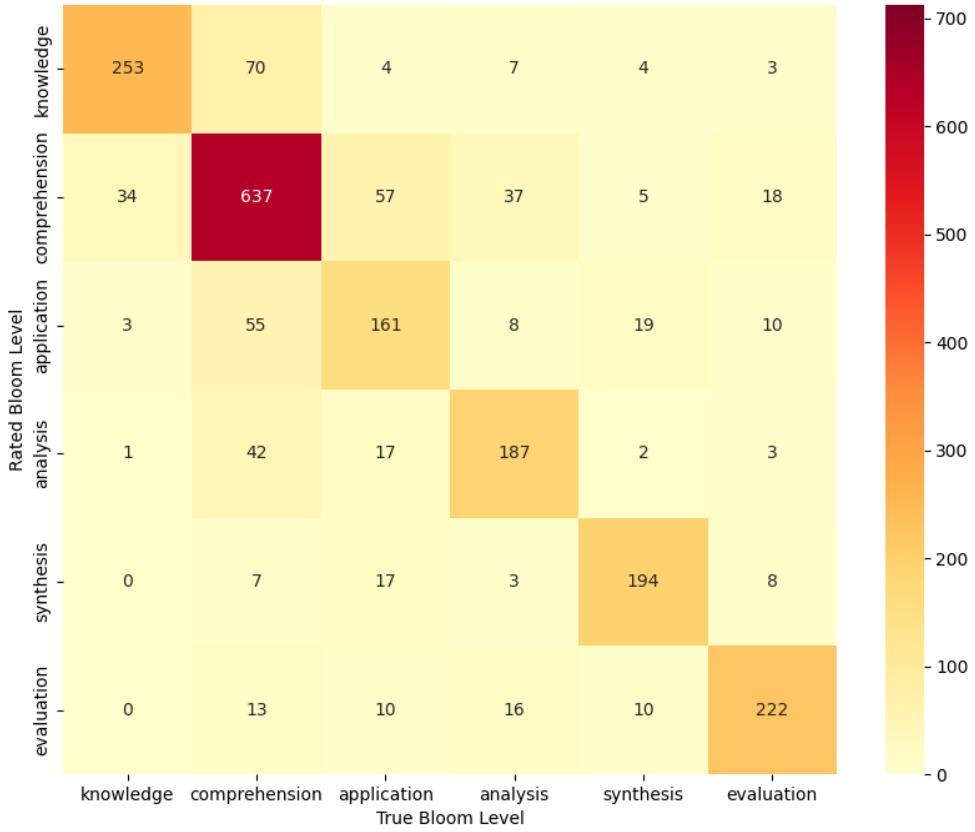


Fig. 2. Rated vs Actual Bloom's Taxonomy levels, on reference set of 2137 questions.

## 4.2 evaluation of agentic methods

Figure 3 shows the results for each evaluation criterion of the final generated questions, with and without preprocessing of the lecture contents. As shown, the questions perform slightly better on some criteria and slightly worse on others. Overall, it appears that the preprocessing step had little impact on the final question quality. In fact, counter to the hypothesis, the questions were of a slightly lower quality on average. Thus, preprocessing was not included in the final workflow.

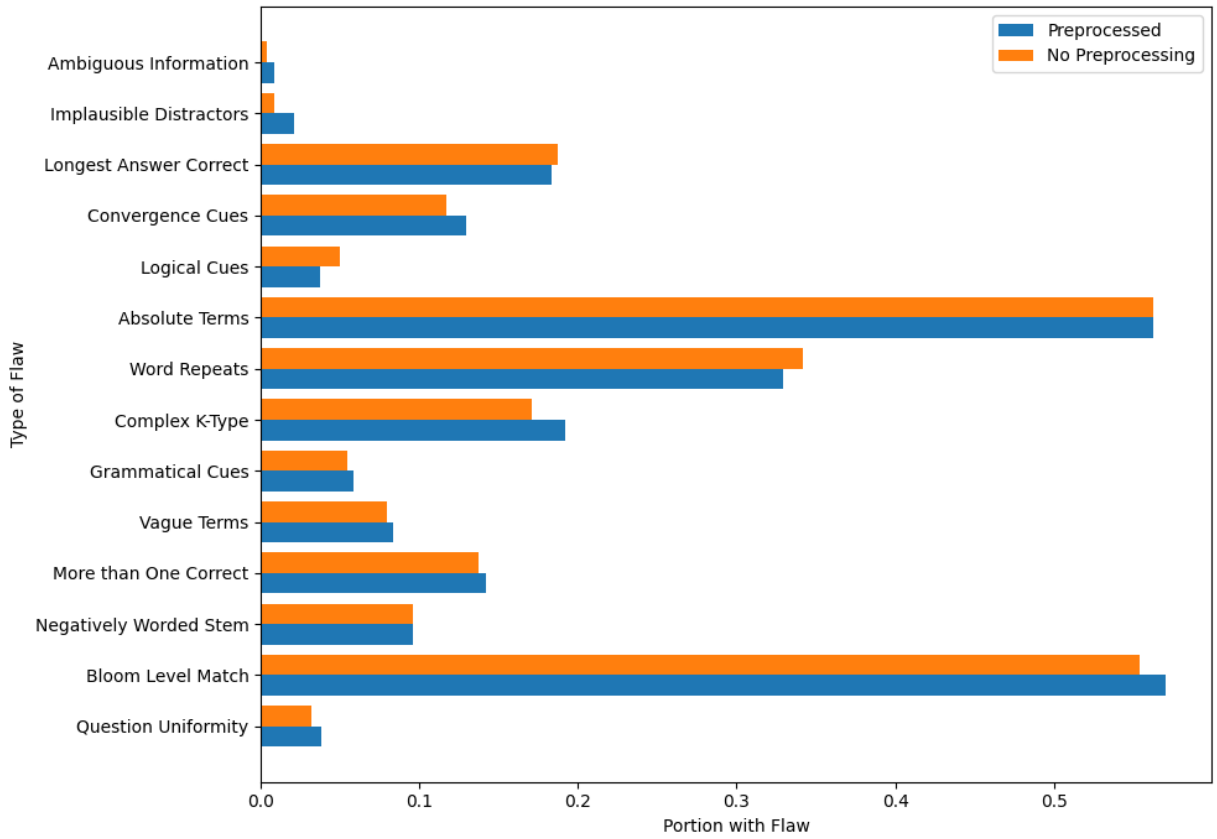


Fig. 3. Comparison of each evaluation criterion for 480 questions generated in batches of five, with and without preprocessing of the course contents, evenly spread across all Bloom levels and 8 lectures. IWFs which did not occur were excluded.

Figure 4 shows the composite score for generating questions in different batch sizes, with and without topics. For each point, 150 questions were generated with a particular batch size and across the different Bloom's Taxonomy levels. Using topics is found to increase the quality of the generated questions for smaller batch sizes (1 or 5) but harmed the quality for a batch size of 25. When using topics, question quality got consistently worse at larger batch sizes. However, without topics, question quality was the worst when 5 questions were generated, being lower for 25 questions. Additionally, the impact of topics was found to depend heavily on the Bloom's Taxonomy, as they provided the greatest benefit for evaluation questions, while generally harming performance on synthesis and application questions. Thus, the benefit of topics depends heavily on the surrounding situation.

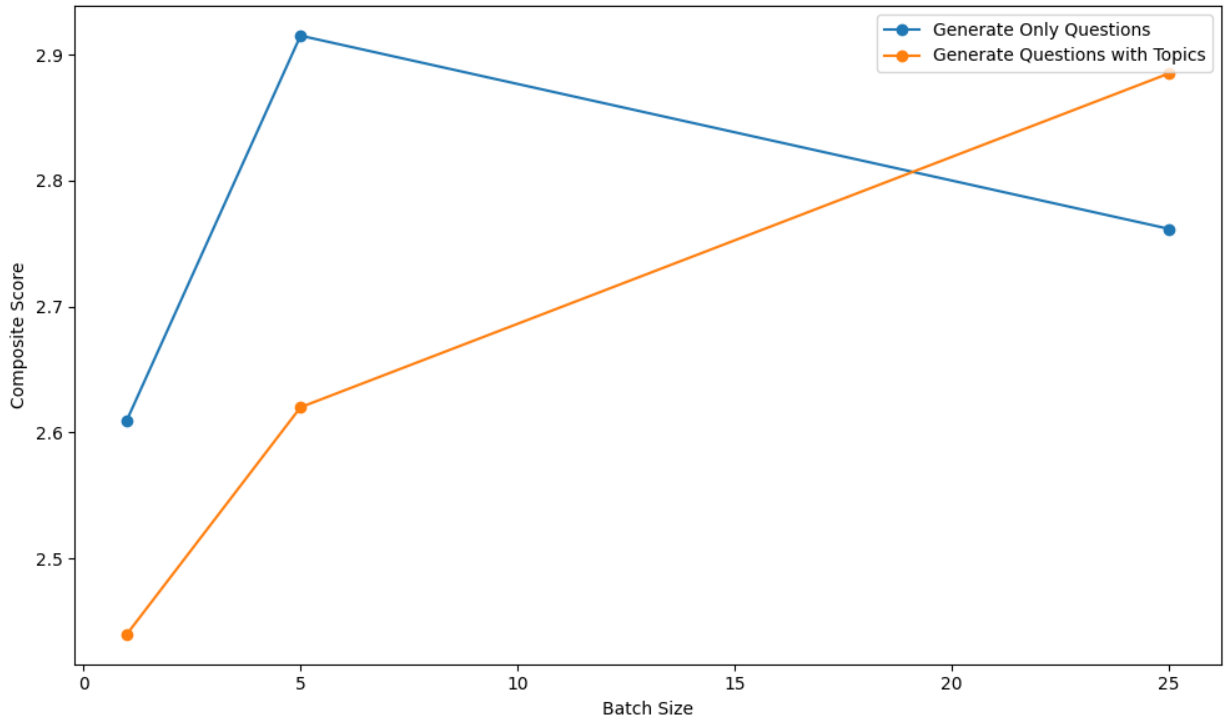


Fig. 4. Comparison of composite evaluation score vs. the batch size for 900 total questions generated with and without topics, evenly spread across all Bloom levels.

Figure 5 shows the impact of each step of iteration on question quality for each of the two different methods of iteration (simple iteration and feedback-driven iteration). The question quality after zero iterations represents the baseline quality, before any changes are made. Then in each iteration the iteration method is applied (either telling the LLM to improve the question for simple iteration or having the LLM generate feedback then improve the question based on that feedback for feedback-driven iteration) once. For each method, the iteration method is repeatedly applied to the results of the last iteration, the iteration number represents how many iterations have been applied. Both of the methods resulted in an overall improvement to question quality, although in feedback-driven iteration, the question quality seems to cyclically improve on one iteration and regress on the next. Simple iteration resulted in a larger improvement, thus these questions were used for the final evaluation of the workflow. However, most of the benefit came from the first two iterations, suggesting that using only two rounds of iteration would likely be sufficient in practice.

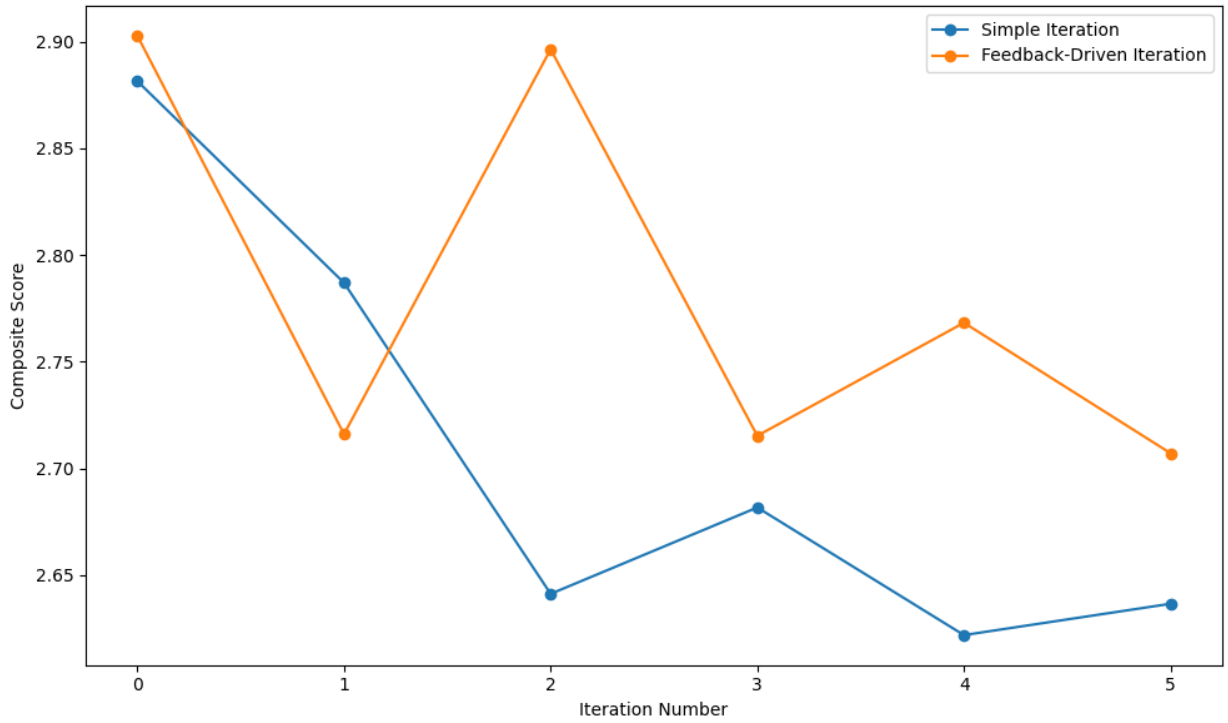


Fig. 5. Comparison of change in question quality for each iteration step for simple and feedback-driven iteration.

## 4.3 evaluation of overall method

Figure 6 shows the quality of 150 of these final questions on each evaluation criterion, based on the target Bloom level. Generally, questions generated at higher Bloom's Taxonomy levels have more IWFs, with the exception that questions at the Evaluation level performed relatively well. Prior research [4] suggests that questions are acceptable if they have less than 2 IWFs, we find that on average across the Bloom's levels, 54% of the generated questions are acceptable, while at the knowledge level, all of the 25 generated questions were acceptable.

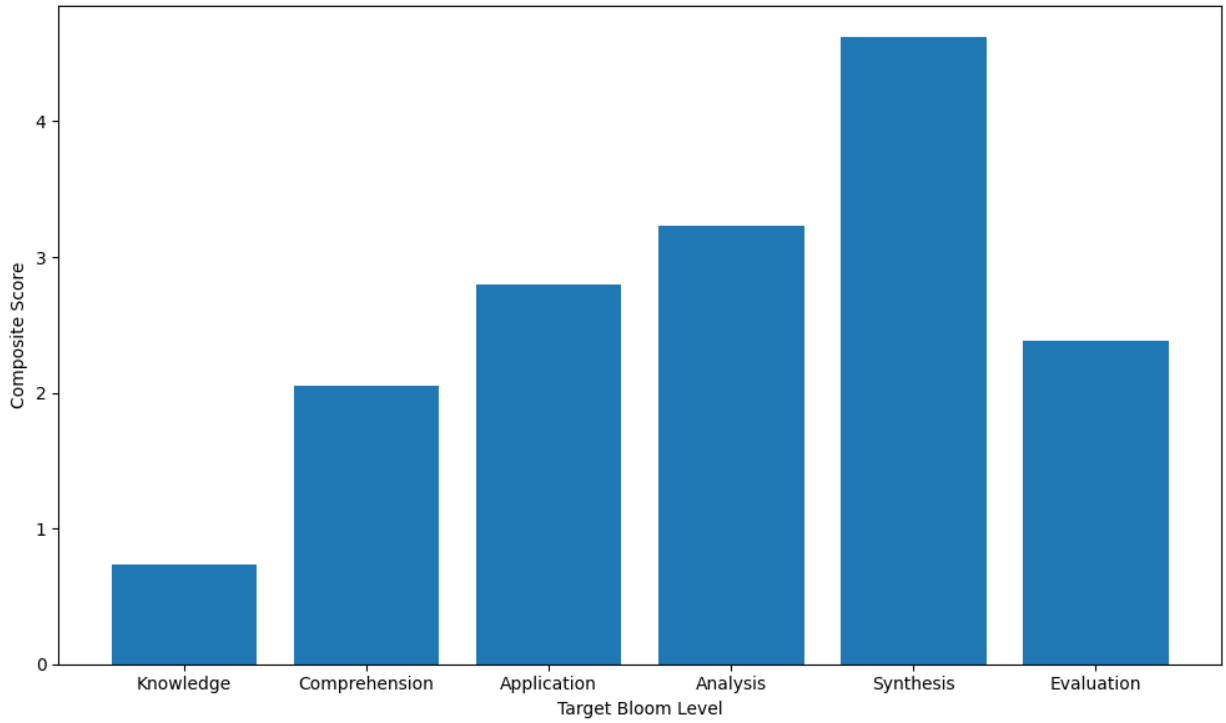


Fig. 6. Comparison of each evaluation criterion for 150 questions generated with the overall best method of topics and using simple iteration, evenly spread across all Bloom levels.

Figure 7 shows how many of the 25 questions targeting each Bloom level were classified into the other Bloom's levels. Only 36% of the generated questions were found to be in the target Bloom level, with most questions being at a lower level than intended.

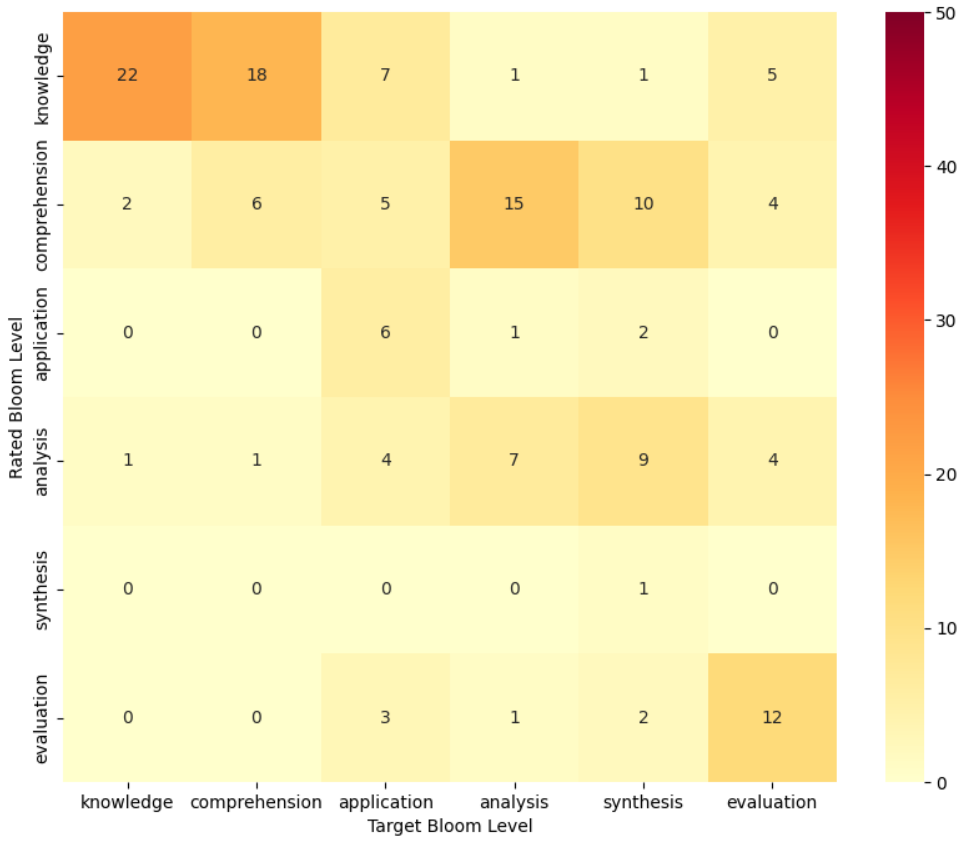


Fig. 7. Comparison of target Bloom-level with the rated Bloom-level

## 5 discussion

Thus, simple iteration substantially improves MCQ quality, with most of the benefit occurring in the first two passes, which would likely be sufficient in practice. Meanwhile, feedback-driven iteration produced unstable results, with question quality improving on one iteration but regressing on the next, producing an oscillating pattern. This suggests that feedback-driven iteration is effective when the question is actually flawed, but introduces issues when no real flaws exist, possibly due to “fixing” hallucinated flaws. Future research may examine constraining iteration to occur based on programmatically generated feedback to avoid this issue. These findings align with prior research showing that multi-stage prompting and iterative refinement can improve question quality in educational LLM applications [5], [9], [17].

Topics improved question quality for small batch sizes (1 or 5), likely because they anchored the LLM to distinct, relevant topics. However, at batch size 25, performance degraded: distributing attention across too many topics appears to strain the LLM, increasing the number of IWFs. This suggests that breaking large batches into smaller, topic guided sets or even individually generating questions may improve performance. The effect was also Bloom-level dependent, suggesting that further work is required to determine what kinds of topics effectively guide question generation at different Bloom's Taxonomy levels.

Finally, preprocessing the lecture transcripts had little impact on the final question quality, possibly even leading to a slight degradation. This suggests that Gemini is robust to minor transcription noise and that additional preprocessing may result in the loss of useful information.

Therefore, agentic workflows can allow better questions to be generated with LLMs, but care must be taken to ensure that each step is effective. Impressive performance was achieved at the knowledge level of Bloom's Taxonomy, but the system struggled to generate quality questions that reached higher cognitive levels. This may be partially due to the prompts being originally finetuned for generation at the knowledge level, then extended to higher Bloom's Taxonomy levels.

Our research was subject to several limitations. Only a single LLM family (Google Gemini) was used for both question generation and evaluation, which may lead to model-specific biases, reducing external validity. Prompts were tuned to this model, meaning performance may be reduced if they are directly transferred to a different LLM. The evaluation framework only achieved a 77.4% accuracy on Bloom-level classification and may miss other flaws. Finally, all experiments were conducted on lecture transcripts from a single third-year engineering course. Further research may use different models for generation and evaluation to reduce shared bias. More prompts and agentic steps could be investigated to further narrow down on the best MCQ generation method and improve Bloom's Taxonomy alignment at higher cognitive levels. Additionally, our research suggests that providing rules-based feedback for iteration may allow higher quality questions to be generated. How LLM-based preprocessing of lecture transcripts and other materials impacts LLM performance on comprehension tasks could also be investigated. Finally, extending the workflow to additional domains and course formats, while incorporating subject-matter-expert evaluations will be essential to establish generalizability.

## 6 conclusion

This study systematically evaluated agentic large language model workflows for automated multiple-choice question generation in engineering education. Through controlled experiments

across Bloom's Taxonomy levels, we found that iterative refinement substantially improves MCQ quality, with the majority of gains achieved within the first two iteration passes. Topic-guided generation improved question quality for small batch sizes but degraded performance at larger batch sizes, suggesting an attention-diffusion trade-off when too many topics are processed concurrently. In contrast, LLM-based preprocessing of lecture transcripts had negligible benefit, indicating that modern models are robust to moderate transcription noise.

Using an automated evaluation framework combining item writing flaws, diversity, content alignment, and Bloom's Taxonomy classification, we showed that over half of the generated questions met established pedagogical quality thresholds, with strongest performance at lower Bloom levels and reduced reliability for higher-order cognitive tasks. These findings demonstrate that agentic workflows can meaningfully enhance automated MCQ generation when carefully constrained, while also highlighting clear limits and trade-offs. Collectively, this work provides empirically grounded guidance for integrating LLM-based MCQ generation into large-scale engineering assessment pipelines and establishes a reproducible foundation for future improvements in higher-level cognitive alignment.

references


[1] T. M. Haladyna, Developing and Validating Multiple-Choice Test Items, 3rd ed. Routledge, 2004. doi: 10.4324/9780203825945.

[2] M. G. Simkin and W. L. Kuechler, "Multiple-Choice Tests and Student Understanding: What Is the Connection?," Decision Sciences Journal of Innovative Education, vol. 3, no. 1, pp. 73–98, 2005, doi: 10.1111/j.1540-4609.2005.00053.x.

[3] B. S. Bloom (Ed.), Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook I: Cognitive Domain. New York, NY, USA: David McKay Company, 1956.

[4] M. Tarrant, A. Knierim, S. K. Hayes, and J. Ware, "The frequency of item writing flaws in multiple-choice questions used in high stakes nursing assessments," Nurse Education Today, vol. 26, no. 8, pp. 662–671, Dec. 2006, doi: 10.1016/j.nedt.2006.07.006.

[5] S. Moore, E. Costello, H. A. Nguyen, and J. Stamper, "An Automatic Question Usability Evaluation Toolkit," in Artificial Intelligence in Education (AIED 2024), Proceedings, Part II, Lecture Notes in Computer Science, vol. 14830, Cham, Switzerland: Springer, 2024, pp. 31–46, doi: 10.1007/978-3-031-64299-9_3.

[6] A. Plaat et al., "Agentic Large Language Models, a survey," arXiv:2503.23037, 2025, doi: 10.48550/arXiv.2503.23037.

[7] J. Moncada-Ramirez et al., "Agentic Workflows for Improving Large Language Model Reasoning in Robotic Object-Centered Planning," Robotics, vol. 14, no. 3, Art. no. 24, Mar. 2025, doi: 10.3390/robotics14030024.

[8] R. Zhang et al., "A Review on Question Generation from Natural Language Text," ACM Transactions on Information Systems, vol. 40, no. 1, Art. no. 14, pp. 1–43, 2022, doi: 10.1145/3468889.

[9] S. Bulathwela, H. Muse, and E. Yilmaz, “Scalable Educational Question Generation with Pre-trained Language Models,” in Artificial Intelligence in Education (AIED 2023), Lecture Notes in Computer Science, vol. 13916, Cham, Switzerland: Springer, 2023, pp. 327–339, doi: 10.1007/978-3-031-36272-9_27.
[10] Y. Kumar et al., “Optimizing Large Language Models for Auto-Generation of Programming Quizzes,” in Proc. 2024 IEEE Integrated STEM Education Conf. (ISEC), Princeton, NJ, USA, Mar. 2024, pp. 1–5, doi: 10.1109/ISEC61299.2024.10665141.
[11] S. Luo et al., “Generating Multiple Choice Questions from Scientific Literature via Large Language Models,” in Proc. 2024 IEEE Int. Conf. on Knowledge Graph (ICKG), Abu Dhabi, UAE, Dec. 2024, pp. 219–226, doi: 10.1109/ICKG63256.2024.00035.
[12] A. J. Winata et al., “Utilizing Large Language Models for Developing Automatic Question Generation in Education,” in Proc. 2025 Int. Conf. on Advancement in Data Science, E-learning and Information System (ICADEIS), Bandung, Indonesia, Feb. 2025, pp. 1–6, doi: 10.1109/ICADEIS65852.2025.10933227.
[13] C. N. Hang et al., “MCQGen: A Large Language Model-Driven MCQ Generator for Personalized Learning,” IEEE Access, vol. 12, pp. 102261–102273, 2024, doi: 10.1109/ACCESS.2024.3420709.
[14] W. Chan et al., “A Case Study on ChatGPT Question Generation,” in Proc. 2023 IEEE Int. Conf. on Big Data (BigData), Sorrento, Italy, Dec. 2023, pp. 1647–1656, doi: 10.1109/BigData59044.2023.10386520.
[15] K. Hwang et al., “Towards Automated Multiple Choice Question Generation and Evaluation: Aligning with Bloom’s Taxonomy,” in Artificial Intelligence in Education (AIED 2024), Proceedings, Part II, Lecture Notes in Computer Science, vol. 14830, Cham, Switzerland: Springer, 2024, pp. 389–396, doi: 10.1007/978-3-031-64299-9_35.
[16] S. S. Mucciaccia et al., “Automatic Multiple-Choice Question Generation and Evaluation Systems Based on LLM: A Study Case With University Resolutions,” in Proc. 31st Int. Conf. on Computational Linguistics (COLING), Abu Dhabi, UAE, Jan. 2025, pp. 2246–2260.
[17] S. Maity et al., “A Novel Multi-Stage Prompting Approach for Language Agnostic MCQ Generation using GPT,” arXiv:2401.07098, 2024, doi: 10.48550/arXiv.2401.07098.
[18] K. Wang et al., “Multi-Agent Interactive Question Generation Framework for Long Document Understanding,” arXiv:2507.20145, 2025, doi: 10.48550/arXiv.2507.20145.