Reinforcement Learning-Supervised LLM Question Generation from Educational Textbooks

A Comparative Study of Prompt Engineering and Post-Hoc Filtering

Fardani Annisa Damastuti
Department of Creative Multimedia Technology, Electronic Engineering Polytechnic Institute of Surabaya, Surabaya, Indonesia
fardani@pens.ac.id (corresponding author)

Agustinus Bimo Gumelar
Department of Informatics, School of Information Technology, Universitas Ciputra, Surabaya, Indonesia
bimo.gumelar@ciputra.ac.id

Kenan Firmansyah
Independent Researcher
realkenanfir@gmail.com

Received: 1 February 2026 | Revised: 4 March 2026 and 16 March 2026 | Accepted: 18 March 2026

 DOI: https://doi.org/10.48084/etasr.17900

ABSTRACT

Large Language Models (LLMs) show promise for generating educational questions from textbook content. However, their outputs still need quality control before they can be used in classrooms. This study investigates how prompt constraint design impacts the quality of LLM questions and tests, and whether post-hoc filtering can enhance this process. A total of 566 questions were generated from Indonesian elementary school textbooks using GPT-3.5-turbo and Gemini 2.0-flash, with three different prompt constraint levels (strict, medium, and loose). The experimental results indicate that prompt engineering is the most influential factor. Strict prompts achieved 97.9% answer findability while loose prompts only reached 72.8%, which is a 25% difference. In addition, a Reinforcement Learning (RL)-based supervisor was developed as a proof-of-concept, which achieved 100% findability on accepted questions. The RL-based supervisor demonstrated similar performance compared to a simple rule-based verification method (verifying if the answer appears in the book). The findings suggest that the RL framework could be useful for more complex quality criteria in the future. Moreover, it was also revealed that story problems are approximately 20% harder than factual questions, while GPT-3.5 demonstrated better performance than Gemini 2.0 in terms of findability, achieving 87.5% compared to 84.1%. However, Gemini 2.0 performed better at matching difficulty levels.

Keywords-automatic question generation; large language models; prompt engineering; educational technology; quality control

I. INTRODUCTION

Automatic Question Generation (AQG) has become an important tool for adaptive learning systems. It allows educators to create personalized content at scale [1, 2]. With the rise of Large Language Models (LLMs), question generation quality has improved substantially [4-6]. However, using these models in educational settings has significant challenges. LLMs sometimes hallucinate, generate questions with answers that cannot be found in the source text, or generate questions at the wrong difficulty level [7, 8].

In the context of Indonesian elementary school students, questions have to meet strict requirements. That is, answers must appear within the textbook, and the difficulty level should be appropriate for young learners. Regular LLM prompting cannot guarantee these requirements [9], while bad questions can frustrate students, resulting in a wrong impression. This study addresses the following research question: What is the relative impact of prompt constraint design versus post-hoc filtering on LLM-generated educational question quality?

The present study proposes a framework in which the LLM acts as a teacher, generating candidate questions from textbook content with varying prompt constraints, and a post-hoc filter validates questions based on answer findability. Unlike prior work that relies on simulated data [15], this study evaluates with real LLM outputs from GPT-3.5-turbo and Gemini 2.0-flash, generating 566 questions [28] across three prompt constraint levels. The key contributions of the study are:

- Empirical analysis showing that prompt constraint design is the dominant factor in question quality.
- Comparative evaluation across two commercial LLMs and three question types, revealing story problems as the hardest category.
- Development of proof-of-concept with Reinforcement Learning (RL)-based supervisor that achieves 100% findability on accepted questions, with comparison showing that it matches a simple rule-based baseline.

The quality–quantity trade-off analysis indicates that achieving 100% findability requires rejecting 15.3% of the generated questions.

II. RELATED WORK

A. Automatic Question Generation

Researchers have been extensively working on AQG for about three decades now. Early systems relied on rule-based templates [10], then moved to statistical methods, and eventually to neural approaches [4, 11]. Authors in [14] were among the first to use syntactic transformations for generating factual questions. Their system performed reasonably well; however, it could not produce very diverse questions. Neural sequence-to-sequence models [8] significantly improved performance by learning patterns from reading comprehension datasets.

Authors in [1, 3, 12] revealed the main challenges in this field: keeping answers consistent with the source, controlling question difficulty, and preventing factual mistakes. Authors in [3] studied 93 different AQG systems and reported that most of them only work with English text. Evaluating question quality for teaching is often missing in past works. With the development of LLMs, zero-shot question generation has become possible [5, 6]. LLMs can generate questions without being specifically trained for it. For example, authors in [9] built a benchmark, namely EQGBench, and tested GPT-4, GPT-3.5, and other open-source models. They found that quality varies significantly depending on the subject and difficulty level. The present study builds on this by examining how different prompt styles affect quality and by adding RL-based filtering.

B. Prompt Engineering for Educational Content

LLMs' outputs significantly depend on the prompt [13, 14]. Authors in [13] demonstrated that chain-of-thought prompting helps with reasoning tasks. Authors in [14] reported that mixing reasoning steps with actions is more effective for certain tasks. Educational content has its own challenges. Authors in [8] studied the factors affecting LLM-generated programming questions and highlighted clarity, appropriate difficulty, and avoiding ambiguous wording as key factors. The present study tests three levels of prompt constraints, from very strict to very loose, and measures how each prompt influences whether answers can be found in the source text.

C. RL in Education

RL is applied across educational technology for adaptive learning [15-17]. Authors in [15] reviewed RL for instructional sequencing, highlighting challenges in reward specification and sample efficiency. Authors in [16] used RL to optimize learning curricula, while others applied deep RL to personalized learning path recommendation. Additionally, authors in [18] utilized Q-learning for dynamic difficulty adjustment in games. Most prior work applied RL to learner modeling, which adapts content presentation based on the student's state. The proposed approach differs by applying RL to content validation, filtering generated questions to ensure quality. This post-hoc filtering approach is more practical for deployments without model fine-tuning access.

D. LLM Quality Control and Alignment

The alignment and reliability of outputs generated by LLMs were a key focus of [20, 22, 23]. Reinforcement Learning from Human Feedback (RLHF) is one of the most effective approaches for LLM quality control. [19, 20]. This approach has shown promising results; however, it requires human annotation and access to the model weights. Constitutional AI [21] constitutes a different approach where the model critiques itself, with self-consistent decoding [22] running multiple samples, resulting in the best answer. In contrast, the present study does not modify the LLM. Instead, a separate small network is trained to filter the outputs after they are generated.

E. Reading Comprehension and Answer Extraction

The SQuAD benchmark [23, 24] has established answer span extraction as a standard task in machine reading comprehension. Inspired by this benchmark, the findability constraint requires that questions have answers extractable from the source text, ensuring consistency between materials and assessments [25]. Unlike question-answer systems that find answers, the present study confirms that generated answers are extractable from the source text.

III. METHODOLOGY

A. System Overview

The proposed framework comprises three components: (1) a book corpus containing Indonesian elementary school textbook content, (2) LLM question generators with configurable prompt constraints, and (3) an RL-based supervisor that filters generated questions. Figure 1 illustrates the architecture of the proposed framework.

B. Application Context: Back-to-School Game

The Back-to-School game is an educational quiz application designed for Indonesian early childhood education (ages 4–12). The game covers three subjects aligned with the national curriculum: Mathematics (numbers, addition, subtraction), Indonesian Language (alphabet, vowels, reading), and Islamic Religion (Hijaiyah letters, daily prayers). Figure 2

shows the game interface. Each level presents students with learning material followed by a 5-question multiple-choice quiz. Students have 120 s to complete each quiz, earning 20 points per correct answer, and must score at least 80 points (4/5 correct) to advance. The game achieved a System Usability Scale (SUS) score of 84.7 (Excellent) during evaluation with 50 students. The proposed supervised question generation system is designed to automatically generate quiz questions for this game, ensuring that answers are extractable from the source material, and difficulty levels match the target age group.

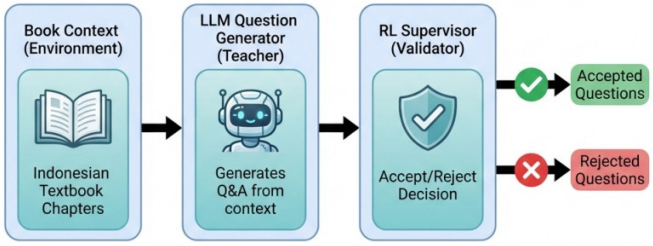

Fig. 1. Architecture of the proposed framework.

Fig. 2. Back-to-School game interface: (a) home screen, (b) subject selection, (c) quiz, (d) result screen.

C. LLM Question Generator

The LLM generator receives textbook content and produces questions in structured JSON format:

```
Algorithms 1: LLM output format
{
"question": " What is 2 + 3?",
"answer": "5",
"difficulty": 1,
"source_span": "2 + 3 = 5",
"reasoning": "Basic addition"
}
```

Two commercial LLMs were evaluated: OpenAI GPT-3.5-turbo and Google Gemini 2.0-flash. Each model generates questions with three prompt constraint levels:

- Strict: Explicit rules requiring exact source quotes, findable answers, and difficulty matching (7 constraints)
- Medium: Standard guidelines with findability emphasis (4 constraints)
- Loose: Minimal constraints allowing creative freedom (2 constraints)

D. RL-Based Supervisor Design

The RL-based supervisor is trained to accept or reject questions based on observable features. Unlike RLHF approaches that modify the LLM, the proposed supervisor operates as an independent filter.

1) State Representation

Each question is encoded as a 6-dimensional feature vector:

$$s = [l_q, l_a, d, f, s_v, q_m]^T \quad (1)$$

where l_q represents the normalized question length, l_a represents the normalized answer length, d is the claimed difficulty, f is the findability indicator, s_v is the source validity, and q_m is the question mark present.

2) Policy Network

The policy is modeled as a Multilayer Perceptron (MLP) and is defined as:

$$\pi_\theta(a|s) = \text{softmax}(W_3 \cdot \text{ReLU}(W_2 \cdot \text{ReLU}(W_1 s))) \quad (2)$$

where the network follows a $6 \to 32 \to 32 \to 2$ architecture, outputting probabilities for accept/reject actions.

3) Reward Function

During training, the reward combines five components designed for question quality. To ensure scalability, the reward evaluation was fully automated using deterministic rule-based algorithms. Specifically, R_{find} is scored via exact string matching that checks whether the generated answer exists in the source_span or the textbook corpus. R_{span} is verified programmatically by confirming that the cited substring literally appears in the textbook. While the automated scoring formed the backbone of the training pipeline, generated questions were manually spot-checked using the digital textbook search functionality and direct curriculum knowledge

to validate that the automated scoring rules were behaving correctly.

The findability reward as defined in (4), evaluates automatically via exact string-matching if the answer exists in the source text. In contrast, the source span validity function, as defined in (5), checks whether the cited substring appears exactly in the textbook.

$$R = R_{find} + R_{span} + R_{diff} + R_{len} + R_{triv} \quad (3)$$

$$R_{find} = \begin{cases} +1.0 & \text{if answer} \in \text{book content} \\ -2.0 & \text{otherwise (harsh penalty)} \end{cases} \quad (4)$$

Source span validity is defined as:

$$R_{span} = \begin{cases} +0.5 & \text{if source span quote is valid} \\ -0.5 & \text{otherwise} \end{cases} \quad (5)$$

Difficulty match is given by:

$$R_{diff} = -0.3 \times |d_{claimed} - d_{target}| \quad (6)$$

Additional components include length heuristics ($R_{len} = +0.2$ for 5–30-word questions) and non-trivial answer bonuses ($R_{triv} = +0.3$ for difficulty ≥ 3 with non-trivial answers).

4) Training Algorithm

The model is trained using the policy gradient method (REINFORCE [26]), defined as:

$$\nabla J(\theta) = \mathbb{E}[\nabla_\theta \log \pi_\theta(a|s) \cdot R] \quad (7)$$

During evaluation, the reward is simplified to a binary signal: +1 for correctly accepting findable questions or rejecting unfindable ones, and −1 otherwise. This formulation allows efficient learning of the findability boundary. At inference, questions where $\pi_\theta(\text{accept}|s) > 0.3$are accepted to balance precision and recall.

IV. EXPERIMENTS

A. Dataset and Experimental Setup

Content based on official Indonesian Buku Sekolah Elektronik (BSE) textbooks [27] was selected for experiments. The dataset covers the following subjects:

- Mathematics: Grades 1–2 (numbers, addition, subtraction, place value).
- Indonesian Language: Alphabet, vowels, and simple sentences.
- Islamic Education: Hijaiyah letters, daily prayers.

Table I summarizes the corpus statistics.

TABLE I. DATASET STATISTICS

Subject	Chapters	Words	Questions
Mathematics	5	1,650	188
Indonesian language	5	1,820	189
Islamic education	5	1,530	189
Total	15	5,000	566

Questions were generated using OpenAI GPT-3.5-turbo and Google Gemini 2.0-flash, with Strict, Medium, and Loose constraint levels. Furthermore, the questions were divided into three types: factual, story (soal cerita), and reasoning. Difficulty levels were defined on a three-point scale (1-3) appropriate for elementary education. Questions were evaluated based on findability and accuracy:

$$FR = \frac{\sum_{i=1}^{N} \mathbb{1}[a_i \in book]}{N} \quad (8)$$

$$DA = \frac{\sum_{i=1}^{N} \mathbb{1}[d_i = d_{target}]}{N} \quad (9)$$

where *FR* and *DA* represent findability and accuracy, respectively. Additional evaluation metrics include source span validity (percentage of valid source quotes) and RL acceptance rate (percentage of questions passing the supervisor).

B. Results and Discussion

1) Impact of Prompt Constraints

Table II presents the findability score evaluated across different prompt constraint levels and the two tested LLMs. The results indicate that strict prompts consistently achieve higher findability scores compared to looser constraints. This performance gap is much more pronounced for the Gemini model. A difference of 25% between strict prompts, which achieved 97.9% findability, and loose prompts, which reduced to 72.8% findability, demonstrates that dedicated prompt engineering has a significant impact on the overall quality of questions generated by the model. Figure 3 provides a visual representation of this comparison and highlights the importance of prompt design.

2) Question Type Comparison

Table III provides a detailed breakdown of findability performance categorized by the question type. The results indicate that story problems (often referred to as "soal cerita") exhibit the lowest findability scores across both language models. This decline in performance can be attributed to the creative nature of story problems, which require models to creatively invent narrative contexts while simultaneously maintaining a high findability score. Balancing these two requirements is a difficult constraint that frequently leads to the generation of hallucinated or ungrounded content. Consequently, this outcome identifies story problems as a significant priority area requiring robust post-processing supervision.

TABLE II. FINDABILITY SCORE BY PROMPT CONSTRAINT LEVEL

Constraint	GPT-3.5	Gemini 2.0	Mean
Strict	95.8%	100.0%	97.9%
Medium	90.3%	73.6%	82.0%
Loose	76.4%	69.2%	72.8%
Overall	87.5%	84.1%	85.8%

TABLE III. FINDABILITY SCORE BY QUESTION TYPE

Type	GPT-3.5	Gemini 2.0	Mean
Factual	98.6%	89.8%	94.2%
Reasoning	90.3%	83.0%	86.7%
Story	73.6%	66.7%	70.2%

3) LLM Comparison

In terms of overall answer findability, GPT-3.5-turbo outperforms Gemini 2.0-flash, scoring 87.5% compared to Gemini's 84.1%. This performance gap becomes even wider when the models are given loose prompts, where GPT-3.5-turbo scores 76.4% against Gemini, which achieved a findability score of 69.2%. On the other hand, Gemini demonstrates a significant advantage in another area. It achieves a much higher difficulty match accuracy, reaching 95.9% compared to the 69.4% observed for GPT-3.5-turbo. This contrast in performance suggests that the two models have very different underlying training emphases, with one prioritizing findability and the other focusing on the difficulty level.

4) Performance of RL-Based Supervisor

For question filtering, the RL-based supervisor was trained on an initial set of 396 questions, which represents 70% of the dataset. The performance of the supervisor was evaluated on the remaining 170 questions. Table IV presents a comparative analysis of three distinct approaches: a scenario with no filtering, a standard rule-based baseline filter, and a proposed RL-based supervisor.

The rule-based baseline operates by filtering out questions whenever the required text match condition is not met. Both the rule-based approach and the RL-based supervisor achieve identical performance metrics on a specific test set. Although the quantitative results are similar in this controlled environment, the true potential value of the RL approach lies in its mathematical flexibility. This is because the agent learns to assign varying weights across multiple distinct features, and it is better equipped to dynamically adapt to complex and evolving failure patterns over time.

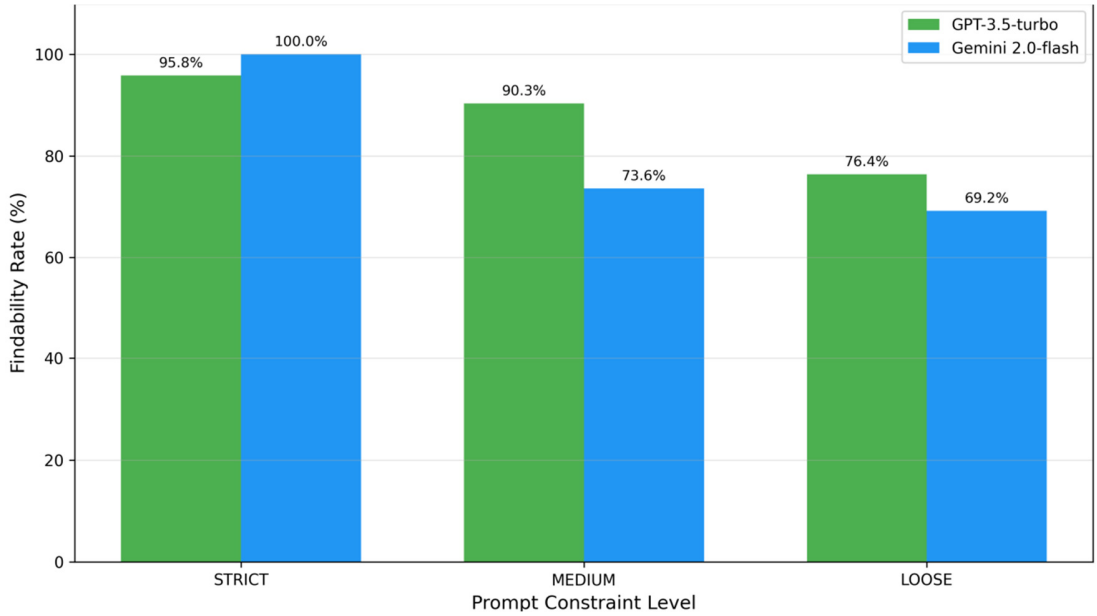

Fig. 3. LLM performance comparison: GPT-3.5-turbo vs Gemini 2.0-flash findability across prompt constraint levels.

TABLE IV. COMPARATIVE ANALYSIS OF NO FILTERING, RULE-BASED, AND RL-BASED SUPERVISION APPROACHES

Metric	No filter	Rule-based	RL
Total questions	170	144	144
Findability score	84.7%	100.0%	100.0%
Acceptance rate	100%	84.7%	84.7%
Rejected questions analysis			
Rejected count	0	26	26
Rejected findability	–	0.0%	0.0%

The RL-based supervisor achieves 100% findability on accepted questions by correctly rejecting all 26 unfindable questions in the test set. It rejects only unfindable questions (0% findability among rejected), demonstrating that it has learned the findability boundary rather than arbitrary filtering. This represents a +15.3% improvement in findability for accepted questions. The RL-based supervisor rejects 15.3% of questions to achieve 100% findability. For a system requiring N questions, this implies generating approximately 1.18N questions, a modest cost increase compared to the 63% rejection rate in prior simulated experiments. The lower rejection rate reflects that real LLMs with proper prompting produce higher-quality output than simulated error injection would suggest.

C. Question Examples

Table V shows representative accepted, rejected, and story problem questions. The examples illustrate key patterns: (1) factual questions with direct text matches achieve the highest acceptance, (2) rejection occurs when LLMs extrapolate beyond source content (multiplication in Grade 1-2) or hallucinate narrative details, (3) story problems require careful construction to maintain answer findability.

D. RL Training Dynamics

Figure 4 shows the RL-based supervisor's training progress. The model quickly learns to distinguish findable from unfindable questions, achieving 100% accuracy by epoch 100 and maintaining stable performance thereafter. Figure 5 illustrates the state representation and policy architecture of the RL-based supervisor.

TABLE V. ACCEPTED, REJECTED, AND STORY PROBLEM QUESTIONS

Question (Indonesian)	Question (English)	Answer	Status	Explanation
Accepted questions (high quality)				
Berapa hasil dari 2 + 3?	What is the result of 2 + 3?	5	Accepted	Answer directly found in text: 2 + 3 = 5
Huruf vokal yang bunyinya keluar dari mulut bundar adalah?	Which vowel is pronounced with rounded lips?	U	Accepted	Matches source: U
Jika kamu punya 7 kelereng dan memberikan 3, berapa sisa?	If you have 7 marbles and give away 3, how many are left?	4	Accepted	Calculation uses numbers from the subtraction chapter
Rejected questions (unfindable answers)				
Berapa hasil 15 × 7?	What is the result of 15 × 7?	105	Rejected	Multiplication not in grade 1-2 content
Tuliskan bilangan 300 dalam bentuk ratusan	Write the number 300 in hundreds.	3 hundred	Rejected	Abstract format not in source text
Ibu membeli 3 bungkus permen, berapa total?	Mom bought 3 packs of candies. What is the total?	Varies	Rejected	Hallucinated story with no source basis
Story problems (challenging cases)				
Dina punya 2 jeruk. Ibu memberi Dina 3 jeruk lagi. Berapa jeruk Dina?	Dina has 2 oranges. Mom gives Dina 3 more oranges. How many oranges does Dina have?	5	Accepted	Valid story problem using addition from text
Budi mempunyai 3 pulpen dan menemukan 5 lagi di tas	Budi has 3 pens and finds 5 more in his bag.	8	Rejected	Menemukan (finds) not in the source context

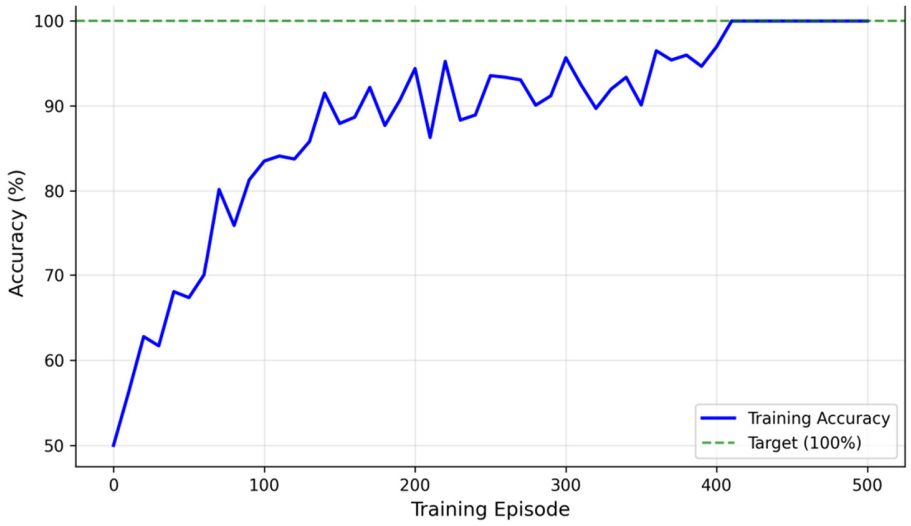

Fig. 4. RL training progress with accuracy rapidly increases and converges to 100% after ~100 episodes.

E. Ablation Study

Table VI presents the impact of individual reward components. Removing the findability reward has the largest negative impact, confirming its importance. Similarly, Table VII shows feature importance for the RL-based supervisor's decisions, measured by policy gradient magnitude.

F. Prompt Templates

An example of the contrast between Strict and Loose prompt constraints is presented below:

Prompt 1: Strict prompt (7 constraints)-English translation

```
Create a quiz for elementary grade
{grade}.
STRICT RULES:
1. Answer must exist in the text
2. Quote source_span from the text
3. Difficulty must match target:
{difficulty}
4-7. [additional constraints...]
```

Prompt 2: Loose prompt (minimal)-English translation

```
Create a {question_type} question with
difficulty level {difficulty} for grade
{grade}.
```

TABLE VI. ABLATION STUDY: REWARD COMPONENT IMPACT

Configuration	Findability	Rejection rate
Full model	100.0%	15.3%
No findability reward	85.2%	8.4%
No source reward	98.6%	12.1%
No difficulty reward	99.1%	14.8%

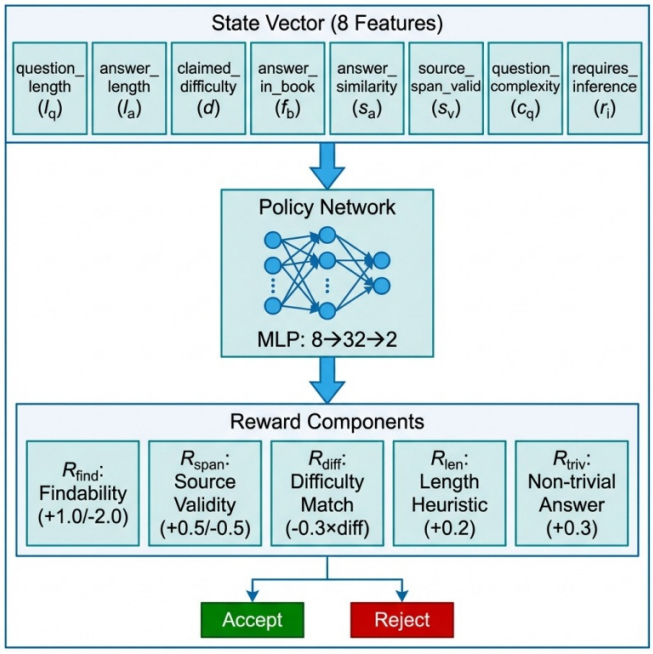

Fig. 5. State representation and policy architecture of the RL-based supervisor.

TABLE VII. FEATURE IMPORTANCE ANALYSIS

Feature	Symbol	Importance
Feature	f	-
Findability indicator	l_a	0.45 (High)
Answer length	s_v	0.22 (Medium)
Source validity	d	0.15 (Medium)
Claimed difficulty	l_q	0.10 (Low)
Question length	q_m	0.05 (Low)

V. DISCUSSION

A. Key Findings

The results indicate that the prompt engineering is highly effective, with strict prompts achieving a findability score of 97.9%, reducing the need for extensive post-hoc filtering. This suggests that investment in prompt design should precede investment in supervision infrastructure. The performance of the RL-based supervisor indicates that even with strong prompts, some unfindable questions are generated (2–15% depending on conditions). The RL-based supervisor filters most of these questions, providing production-grade reliability. In addition, the results indicate that the story problems are the hardest to generate. Across LLMs and constraint levels, story problems show +20% lower findability compared to factual questions. These narrative-contextual questions should receive more attention during post-processing.

B. Comparison with Previous Works

The proposed LLM-based experiments have yielded better performance compared to simulation-based studies. For example, authors in [18] conducted a simulated analysis and reported 63% rejection rates. In contrast, the present study achieved excellent performance with 15% rejection rates. This suggests that simulation may overestimate LLM error rates, potentially justifying unnecessarily complex supervision architectures. Similarly, the findings of the present study align with EQGBench [13] on LLM question quality variation across subjects. The proposed framework demonstrated higher consistency in mathematics compared to language tasks, which often require contextual reasoning.

C. Practical Implications

Using fully automated, deterministic rule-based algorithms (e.g., exact string matching) for acquiring initial decisions brings the risk of false negatives, such as rejecting a perfectly valid answer simply because it is paraphrased (e.g., 'lima' vs '5'). However, the core advantage of utilizing an RL-based supervisor instead of relying solely on hardcoded rules in question generation is that the RL agent can learn to weigh continuous and diverse features (such as question length, claimed difficulty, and structural validity). This allows the system to generalize beyond strict string-matching constraints over time. A human-in-the-loop is only functionally necessary during the initial design phase to validate these heuristic rules on a small baseline, ensuring that the continuous question generation pipeline remains fully automated with minimal human intervention at scale.

Figure 6 displays the theoretical trade-off between acceptance rate and findability. Lower thresholds accept more questions but reduce quality; higher thresholds improve findability with a higher reject rate. The results indicate that a threshold of 0.3 provides an effective balance between these objectives for practical deployment.

For production deployment, the study proposes:

- Using strict prompts as the first quality layer (achieving ~96% findability).
- Applying RL-based supervision for the remaining ~4–15% of problematic questions.
- Prioritizing supervision for story problems and loose-constraint scenarios.
- Planning for ~1.2× overgeneration to account for filtered questions.

D. Limitations

The proposed RL-based supervisor achieves identical performance to a simple rule-based filter ("accept if answer ∈ book"). For the quality metrics studied, learned filtering provides no measurable benefit over explicit rules. The RL framework's value is speculative (extensibility, retainability) rather than empirically demonstrated. In addition, the smaller sample size of 566 questions may not capture all LLM failure modes.

While generated questions were manually verified using the digital textbook search functionality and direct curriculum knowledge, no formal, structured human annotation study was conducted over the entire question dataset. This means that subjective pedagogical quality (e.g., question clarity, age appropriateness) beyond answer findability was not formally measured. Future work will focus on incorporating human annotation for further evaluation. Moreover, the results are specific to Indonesian content, and cross-lingual generalization remains untested. Despite the favorable results, only GPT-3.5 and Gemini were evaluated; performance may differ when using GPT-4 or open-source models.

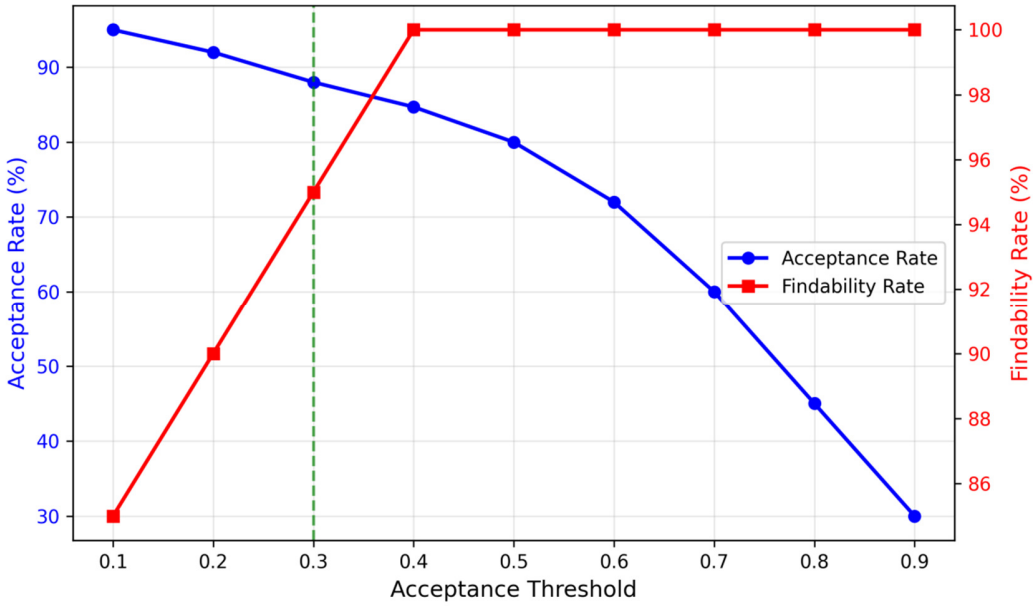

Fig. 6. Threshold sensitivity analysis: parameter tuning trade-off between acceptance rate and findability rate.

VI. CONCLUSION

This study examined the effects of prompt constraints and post-hoc filtering on the quality of Large Language Model (LLM)-generated educational questions. LLM models, including GPT-3.5-turbo and Gemini 2.0-flash, were employed to generate 566 questions based on Indonesian elementary school material. The results indicate that prompt design is the most significant factor in question generation. Strict prompts achieved a findability score of 97.9%, outperforming loose prompts with a findability score of 72.8.

The proposed Reinforcement Learning (RL)-based supervisor and a simple rule-based check achieved 100% findability on accepted questions. This suggests that for basic metrics such as findability, a simple rule-based verification method is as effective as an RL-based supervisor. However, the RL-based approach demonstrates better performance for more complex quality criteria.

Future research will include a formal human evaluation study with Indonesian students and teachers, leveraging their curriculum knowledge to assess question quality beyond automated findability metrics and to explore how curriculum-based knowledge can serve as an effective performance evaluation criterion. In addition, the RL-based model will be evaluated with more sophisticated quality metrics integrated with the Back-to-School educational game.

DECLARATION OF COMPETING INTERESTS

The authors declare no competing interests.

ACKNOWLEDGMENT

The authors express their deepest gratitude to Nur Mira Permatasari, whose dedication and contributions to the Back-to-School educational game project were invaluable. This research is dedicated to her memory. The authors acknowledge the Research Group Multimedia Imaging and Serious Game at Politeknik Elektronika Negeri Surabaya for their continued support throughout this work.

DATA AVAILABILITY

The dataset generated during this study is publicly available and can be accessed at [28].

AI USE AND DECLARATION OF GENERATIVE AI USE

During the preparation of this work, the authors used Large Language Models (OpenAI GPT-3.5-turbo and Google Gemini 2.0-flash) as the primary subjects of the experimental framework to automatically generate educational questions. For the writing process of the manuscript itself, the authors used Generative AI models (e.g., ChatGPT / Gemini) as language assistance tools to improve readability, grammar, and sentence structure. After using these tools, the authors reviewed and edited the text as needed and take full responsibility for the content of the publication.

REFERENCES

[1] S. Guo, L. Liao, C. Li, and T.-S. Chua, "A Survey on Neural Question Generation: Methods, Applications, and Prospects." arXiv, 2024, https://doi.org/10.48550/ARXIV.2402.18267.

[2] S. Alamoudi, L. A. Al Khuzayem, and A. Jamal, "Optimizing Automated Question Generation for Educational Assessments: A Semantic Analysis of LLMs with Structured and Unstructured Ontologies," *Engineering, Technology & Applied Science Research*, vol. 15, no. 3, pp. 23664–23671, Jun. 2025, https://doi.org/10.48084/etasr.10662.

[3] X. Du, J. Shao, and C. Cardie, "Learning to Ask: Neural Question Generation for Reading Comprehension," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1342–1352, https://doi.org/10.18653/v1/P17-1123.

[4] S. Maity and A. Deroy, "The Future of Learning in the Age of Generative AI: Automated Question Generation and Assessment with Large Language Models." arXiv, 2024, https://doi.org/10.48550/ARXIV.2410.09576.

[5] T. Brown *et al.*, "Language Models are Few-Shot Learners," in *Advances in Neural Information Processing Systems*, Vancouver, Canada, Dec. 2020, pp. 1877–1901.

[6] Z. Ji *et al.*, "Survey of Hallucination in Natural Language Generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, Dec. 2023, https://doi.org/10.1145/3571730.

[7] P. Denny, S. MacNeil, J. Savelka, L. Porter, and A. Luxton-Reilly, "Desirable Characteristics for AI Teaching Assistants in Programming Education," in *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, Milan, Italy, Jul. 2024, pp. 408–414, https://doi.org/10.1145/3649217.3653574.

[8] G. Călugăreanu, H. F. Pop, and A. Vasiu, "Matrix Invertible Extensions Over Commutative Rings. Part III: Hermite Rings." arXiv, Jul. 27, 2025, https://doi.org/10.48550/arXiv.2405.01234.

[9] S. Doroudi, V. Aleven, and E. Brunskill, "Where's the Reward?: A Review of Reinforcement Learning for Instructional Sequencing," *International Journal of Artificial Intelligence in Education*, vol. 29, no. 4, pp. 568–620, Dec. 2019, https://doi.org/10.1007/s40593-019-00187-x.

[10] M. Heilman, "Automatic Factual Question Generation from Text," Carnegie Mellon University, Pittsburgh, PA, USA, 2025.

[11] Q. Zhou, N. Yang, F. Wei, C. Tan, H. Bao, and M. Zhou, "Neural Question Generation from Text: A Preliminary Study," in *Natural Language Processing and Chinese Computing*, vol. 10619, X. Huang, J. Jiang, D. Zhao, Y. Feng, and Y. Hong, Eds. Cham: Springer International Publishing, 2018, pp. 662–671.

[12] N. Mulla and P. Gharpure, "Automatic Question Generation: A Review of Methodologies, Datasets, Evaluation Metrics, and Applications," *Progress in Artificial Intelligence*, vol. 12, no. 1, pp. 1–32, Mar. 2023, https://doi.org/10.1007/s13748-023-00295-9.

[13] G. Kurdi, J. Leo, B. Parsia, U. Sattler, and S. Al-Emari, "A Systematic Review of Automatic Question Generation for Educational Purposes," *International Journal of Artificial Intelligence in Education*, vol. 30, no. 1, pp. 121–204, Mar. 2020, https://doi.org/10.1007/s40593-019-00186-y.

[14] J. Wei *et al.*, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," in *Proceedings of Advances in Neural Information Processing Systems*, Red Hook, NY, USA, Dec. 2022, pp. 24824–24837.

[15] S. Yao *et al.*, "ReAct: Synergizing Reasoning and Acting in Language Models." arXiv, 2022, https://doi.org/10.48550/ARXIV.2210.03629.

[16] S. Reddy, A. Dragan, and S. Levine, "Shared Autonomy via Deep Reinforcement Learning," in *Robotics: Science and Systems XIV*, Pittsburgh, PA, USA, Jun. 2018, https://doi.org/10.15607/RSS.2018.XIV.005.

[17] K. Mo, S. Li, Y. Zhang, J. Li, and Q. Yang, "Personalizing a Dialogue System with Transfer Reinforcement Learning." arXiv, 2016, https://doi.org/10.48550/ARXIV.1610.02891.

[18] F. A. Damastuti, K. Firmansyah, Y. M. Arif, T. Dutono, A. Barakbah, and M. Hariadi, "Dynamic Level of Difficulties Using Q-Learning and Fuzzy Logic," *IEEE Access*, vol. 12, pp. 137775–137789, 2024, https://doi.org/10.1109/ACCESS.2024.3457801.

[19] L. Ouyang *et al.*, "Training Language Models to Follow Instructions with Human Feedback." arXiv, 2022, https://doi.org/10.48550/ARXIV.2203.02155.

[20] Y. Bai *et al.*, "Constitutional AI: Harmlessness from AI Feedback." arXiv, Dec. 15, 2022, https://doi.org/10.48550/arXiv.2212.08073.

[21] X. Wang *et al.*, "Self-Consistency Improves Chain of Thought Reasoning in Language Models," in *International Conference on Learning Representations*, Kigali, Rwanda, 2023.

[22] P. F. Christiano *et al.*, "Deep Reinforcement Learning from Human Preferences," in *Advances in Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 4299–4307.

[23] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, TX, USA, 2016, pp. 2383–2392, https://doi.org/10.18653/v1/D16-1264.

[24] P. Rajpurkar, R. Jia, and P. Liang, "Know What You Don't Know: Unanswerable Questions for SQuAD," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018, pp. 784–789, https://doi.org/10.18653/v1/P18-2124.

[25] S. Sugawara, Y. Kido, H. Yokono, and A. Aizawa, "Evaluation Metrics for Machine Reading Comprehension: Prerequisite Skills and Readability," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 806–817, https://doi.org/10.18653/v1/P17-1075.

[26] R. J. Williams, "Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning," *Machine Learning*, vol. 8, no. 3–4, pp. 229–256, May 1992, https://doi.org/10.1023/A:1022672621406.

[27] "Buku Sekolah Elektronik (BSE)," *Myedisi Interaktif Media*, 2022. https://buku.kemdikbud.go.id.

[28] F. A. Damastuti, A. B. Gumelar, and K. Firmansyah, "Back to School Dataset." GitHub, 2025, [Online]. Available: https://github.com/Kenanfir/BackToSchool-Dataset/blob/main/rl_llm_question_dataset_566.csv.