Constructing Evaluation Datasets for Procedural Reasoning: Balancing Naturalness, Grounding, and Multi-Hop Coverage

Sarah Elshabrawy, Rahul K. Dass, and Ashok K. Goel

Georgia Institute of Technology, Atlanta GA, USA
{selshabrawy3,rdass7,ag25}@gatech.edu

Abstract. Evaluating procedural reasoning in AI-supported learning systems requires question-answer datasets that are both learner-like and grounded in the instructional knowledge the system is expected to use. We study how TMK-based question generation strategies affect dataset quality for procedural and multi-hop reasoning.
We compare three strategies: strict generation from Task–Method–Knowledge (TMK) models, transcript-first generation with post-hoc TMK filtering, and TMK-aware generation that combines transcripts with structured guidance. To evaluate generated items, we introduce a grounding validation framework based on closed-set evidence units extracted from TMK models. The framework measures whether answers are supported by the underlying representation, whether questions are self-contained, and whether they target multi-hop procedural reasoning. Across 23 instructional topics and 690 generated question-answer pairs, strict TMK generation achieves the strongest overall quality, with 96.5% grounded questions and 92.6% usable questions. Transcript-first generation produces more learner-like questions but more context-dependent or weakly grounded items, while TMK-aware generation yields high raw multi-hop coverage but lower grounding. These results show that procedural richness and natural phrasing do not guarantee representational grounding, motivating explicit representation-aware validation for evaluation datasets in AI-supported learning.

Keywords: evaluation datasets · procedural reasoning · multi-hop reasoning · question generation · grounding · structured knowledge representations

Code and artifacts. Prompt templates, validation scripts, aggregation code, aggregate results, and generated QA artifacts are available at: `https://github.com/DILab-Ivy/tmk-procedural-qa-eval`. Restricted course materials are excluded.

1 Introduction

AI-supported learning systems increasingly use generative and hybrid knowledge-based AI approaches to answer student questions, provide explanations, and support reflection [4, 15]. However, evaluating whether such systems can reason procedurally remains difficult. Many question-answer datasets test whether a model can produce a correct or plausible answer, but do not test whether that answer is faithful to the structured instructional knowledge the system is expected to use.

Procedural reasoning is especially challenging because it requires connecting multiple steps, constraints, goals, and domain concepts. In learning settings, a good answer should not only state what is correct, but also explain how a process works, why a step matters, and how different parts of the instructional representation support the answer.

This paper investigates how Task–Method–Knowledge (TMK) models [3, 17] can be used to construct evaluation question-answer pairs for procedural and multi-hop reasoning. We use TMK models as structured sources of procedural knowledge and compare three strategies for deriving questions from TMK and lesson transcripts. We ask: **RQ1** How do generation strategies affect grounding quality? **RQ2** How often are generated questions self-contained rather than dependent on hidden transcript context? **RQ3** Which strategy best balances grounding, self-containedness, and multi-hop procedural coverage?

This paper makes three contributions. First, we compare strict TMK generation, transcript-first generation with TMK filtering, and TMK-aware generation for constructing procedural reasoning evaluation data. Second, we introduce a closed-evidence grounding validation framework for generated question-answer pairs. Third, we evaluate 690 generated items across 23 instructional topics and identify failure modes including transcript leakage, context-dependent questions, partially grounded answers, and inflated multi-hop labels.

2 Background and Related Work

Our work connects three areas: structured procedural knowledge for learning, multi-hop and procedural reasoning benchmarks, and grounding-based validation of generated question-answer pairs.

2.1 Structured Procedural Knowledge for Learning

Knowledge-based learning systems have long used explicit representations of domain knowledge, skills, and problem-solving procedures. Work on generic tasks and task-structure analysis argued that expertise should be modeled not only as facts, but also as goals, procedures, and the knowledge needed to support them [2, 3]. This view is closely related to intelligent tutoring systems, where explicit models of knowledge components and problem-solving steps support feedback and learning [12, 13, 21].

TMK models represent procedural skills through tasks, methods, and knowledge: tasks define goals and success or failure conditions, methods describe procedures, and knowledge grounds those procedures in concepts, relations, and assertions. Prior work has used TMK and related representations to model agent behavior, translate process models into hierarchical task networks, and ground AI coaching systems for procedural explanation [17, 8, 14, 4, 15]. Recent text-to-model work also studies how LLMs can draft TMK models from instructional materials for expert refinement [5].

In contrast, we use TMK models not as a runtime tutoring substrate or authoring target, but as structured ground truth for constructing and validating evaluation questions.

2.2 Evaluating Multi-hop and Procedural Reasoning

Multi-hop question-answering benchmarks such as HotpotQA, MuSiQue, and QASC show that dataset construction choices affect whether a benchmark actually tests multi-step reasoning rather than shortcut retrieval [22, 20, 11]. More recent procedural benchmarks, including ProcBench and PKR-QA, move beyond isolated facts toward ordered steps, dependencies, and procedural knowledge graphs [6, 18].

Our work is complementary: rather than proposing a new model-performance benchmark alone, we study the construction process for procedural evaluation data. In learning systems, a question can be natural and pedagogically plausible but still unsuitable if its answer depends on information outside the target instructional representation.

2.3 Grounding and Evidence-Based Validation

LLMs can generate fluent outputs that are unsupported or only partially supported by their sources [7, 23]. Prior work on attribution and factual consistency evaluates whether generated answers are supported by explicit evidence, including AIS, TRUE, FActScore, and attributed question answering [19, 9, 16, 1, 10].

We adapt this source-based view to procedural dataset construction. Instead of validating answers against open-ended documents or retrieved passages, we validate question-answer pairs against closed evidence units extracted from a structured procedural model. This lets us distinguish natural transcript-like questions from questions whose answers are actually supported by the intended representation.

3 Method

Figure 1 summarizes the dataset construction and validation pipeline. For each instructional topic, we use a lesson transcript and a corresponding TMK model. The transcript provides learner-facing phrasing and examples, while the TMK model provides the structured procedural representation used for generation guidance and grounding validation.

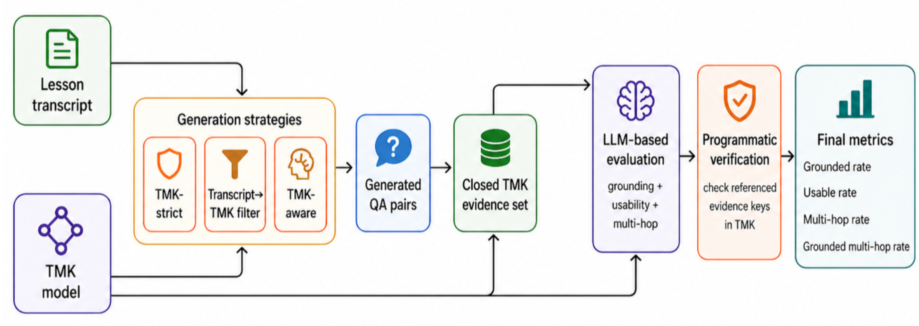

Fig. 1. Question-answer generation and validation pipeline. Transcripts and TMK models are used to generate candidate items under three strategies. Items are validated against a closed TMK evidence set, followed by a programmatic evidence membership check and aggregate metric computation.

3.1 Generation Strategies

We compare three strategies. **TMK-aware generation** uses both the transcript and TMK model during generation: the transcript supports natural student-like wording, while TMK provides procedural structure. **Strict TMK generation** treats TMK as the primary source of truth and uses the transcript only for wording; because TMK is encoded as structured JSON with formal task, method, state, transition, and condition fields, this strategy may produce more schematic questions. **Transcript-first generation with TMK filtering** first generates natural questions from the transcript, then keeps or rewrites only items whose answers are supported by TMK.

A pilot analysis showed that some generated items were grounded but weak as benchmark questions because they relied on hidden classroom context or over-labeled questions as multi-hop. We therefore refined the prompts to require self-contained wording and conservative reasoning-type labels. The final results use these refined prompts.

3.2 Grounding Validation

To validate grounding, we extract a closed set of evidence units from each TMK model. Each unit corresponds to a field in a task, method, concept, instance, relation, assertion, or property. The validator classifies each question-answer pair as *grounded*, *partially grounded*, or *not grounded*, and selects supporting evidence units from this closed set. We then verify programmatically that selected evidence identifiers belong to the closed evidence set. This verifies evidence membership, but not evidence sufficiency; we return to this limitation in Section 5.

3.3 Experimental Setup and Metrics

We evaluate 23 topics from Georgia Tech's Knowledge-Based AI course, including classification, planning, semantic networks, frames, constraint propagation, case-based reasoning, diagnosis, scripts, production systems, version spaces, commonsense reasoning, and explanation-based learning. For each topic, we use a lesson transcript drawn from the course materials and a corresponding TMK model.

For each topic and each of the three generation strategies, we generate 10 question-answer pairs, yielding 690 total items.

Each item is evaluated for grounding, self-containedness, and reasoning type. A question is self-contained if it can be understood without the transcript, slides, previous questions, or classroom discussion. A question is multi-hop only if answering it requires connecting at least two distinct pieces of evidence, such as procedural steps, constraints, concepts, or representations. We define an item as usable when it is both grounded and self-contained:

$$\text{Usable}(q, a) = \text{Grounded}(q, a) \wedge \text{SelfContained}(q).$$

We report grounded, partially grounded, unsupported, self-contained, usable, multi-hop, grounded multi-hop, and usable multi-hop rates.

4 Results and Discussion

We report results across 690 generated question-answer pairs, with 230 items per generation strategy.

4.1 Overall Dataset Quality

Table 1 summarizes grounding, self-containedness, and usability. Strict TMK generation performs best overall, producing the highest grounding rate, self-contained rate, and usable rate. It generated no unsupported items and only 8 partially grounded items.

Transcript-first generation performs second-best: post-hoc TMK filtering recovers much of the grounding quality, but context-dependent phrasing and unsupported transcript details remain. TMK-aware generation performs weakest on grounding, suggesting that combining transcripts and TMK during generation can lead the model to blend supported structure with unsupported transcript-only details.

4.2 Multi-Hop Procedural Coverage

Table 2 shows that raw multi-hop coverage can be misleading. TMK-aware generation has the highest raw multi-hop rate, but its usable multi-hop rate drops sharply because many items are not grounded or self-contained. Strict TMK generation has slightly lower raw multi-hop coverage but the highest usable

Table 1. Overall quality of generated question-answer pairs by generation strategy. Each strategy generated 230 items across 23 instructional topics.

Strategy	**Grounded**	**Partial**	**Unsup.**	**Self-cont.**	**Usable**
Strict TMK generation	**96.5%**	**3.5%**	**0.0%**	**96.1%**	**92.6%**
Transcript-first + TMK filter	90.9%	8.7%	0.4%	86.5%	78.7%
TMK-aware generation	52.6%	33.0%	14.3%	84.3%	51.7%

Table 2. Multi-hop procedural coverage by generation strategy. Usable multi-hop items are both multi-hop and usable.

Strategy	**Multi-hop rate**	**Grounded multi-hop**	**Usable multi-hop**	**Usable MH items**
Strict TMK generation	79.6%	**76.5%**	**73.0%**	**168**
Transcript-first + TMK filter	78.7%	70.0%	61.7%	142
TMK-aware generation	**82.2%**	41.7%	40.9%	94

multi-hop rate, producing 168 usable multi-hop items. This makes it the strongest strategy for evaluation data that is both procedurally rich and representationally grounded.

4.3 Naturalness, Self-Containedness, and Grounding

Self-containedness is a major source of quality loss. A question may be grounded but still weak as a benchmark item if it depends on hidden lesson context, such as references to "the example" or "the process we discussed." The transcript-first strategy produced the most non-self-contained questions, showing that learner-like phrasing can preserve classroom context that is unsuitable for standalone evaluation.

At the same time, strict TMK generation should not be interpreted as the most natural strategy. Because it is anchored in formal JSON structures of tasks, methods, states, transitions, and conditions, it can produce more schematic questions. For example, a strict TMK item for Classification asked: "What sequence of successful checks must occur for the overall animal-to-bird classification process to finish successfully? Describe each step's required condition in order." Although grounded, the wording mirrors the procedural control structure of the TMK model rather than the way a learner would typically ask about classification. A more learner-like transcript-first item asked: "Can you walk me through the steps the agent actually takes to go from the observed animals to a finished list of birds, and what has to be true to move past each step?"

The central tradeoff is therefore not between good and bad generation, but between natural student-like language and faithful alignment with the structured representation.

Table 3. Representative failure modes from TMK-aware generation.

Failure mode	**Example pattern**	**Why validation rejects it**
Unsupported causal links	The answer claims that "hasConcavity" implies "carriesLiquids," and that "carriesLiquids" and "isLiftable" imply "enablesDrinking."	The TMK evidence lists object features, but does not contain explicit instantiated implication rules connecting these predicates. The item therefore treats feature co-occurrence as causal structure.
Hidden prior context	The question begins from a previous Mug scenario and assumes that "limitsHeatTransfer," "isStable," and "enablesDrinking" have already been proven.	The item may be understandable in a sequence, but it is not self-contained as a standalone benchmark question. The validator marks it for rewrite rather than direct use.
Transcript-only detail	The question refers to causal characterizations of examples such as Brick, Bowl, Briefcase, Glass, or Pot when those concrete causal rules are not present in the TMK evidence set.	The answer partly follows the EBL procedure, but its concrete proof steps depend on details outside the closed TMK evidence set. This makes the item only partially grounded.

4.4 Qualitative Failure Modes

Table 3 shows representative validation failures from the TMK-aware strategy. These examples illustrate why raw procedural richness was not sufficient for benchmark quality. In several cases, generated items appeared to require multi-step reasoning, but the answer relied on information that was not fully available in the TMK evidence set or depended on context from a previous question.

4.5 Implications

These results suggest that procedural evaluation datasets should report combined metrics such as grounded multi-hop and usable multi-hop rates, rather than raw multi-hop coverage alone. For AI-supported learning interfaces, evaluation questions should be validated against the same instructional representation the system is expected to use; otherwise, evaluation may reward fluent answers to questions outside the modeled knowledge.

5 Limitations and Future Work

This study has four main limitations. First, grounding validation still relies on LLM judgment. Although the validator is constrained by closed TMK evidence units and we programmatically verify evidence membership, we do not independently verify evidence sufficiency. Second, we do not include full manual expert annotation of all generated items, which would provide a stronger gold standard for grounding and self-containedness. Third, all topics come from one symbolic AI course, so future work should test other domains, instructional styles, and knowledge representations. Finally, we evaluate dataset quality rather than downstream learner outcomes or AI tutor performance.

Future work will extend this framework toward larger-scale procedural reasoning benchmarks in which each item is explicitly linked to structured evidence. Such benchmarks could evaluate whether models can explain why steps are needed, how substeps interact, what happens when constraints are violated, and how dependencies unfold across multiple reasoning hops. Another direction is interactive evaluation: procedural learning often involves clarification, follow-up questions, and guided questioning, so future datasets could include learner-like follow-ups or Socratic tutoring exchanges.

6 Conclusion

We compared three TMK-based strategies for constructing evaluation datasets for procedural and multi-hop reasoning. Using closed-set evidence units extracted from TMK models, we evaluated whether generated question-answer pairs were grounded, self-contained, usable, and multi-hop.

Strict TMK generation produced the most reliable evaluation data, while transcript-first generation produced more learner-like but more context-dependent items. TMK-aware generation produced high raw multi-hop coverage, but often blended grounded content with unsupported transcript details. These findings show that natural phrasing and procedural richness are not sufficient for high-quality evaluation data. Evaluation datasets for AI-supported learning should include explicit representation-aware validation so that questions test the knowledge the system is expected to use.

References

1. Bohnet, B., Tran, V.Q., Verga, P., Aharoni, R., Andor, D., Soares, L.B., Ciaramita, M., Eisenstein, J., Ganchev, K., Herzig, J., Hui, K., Kwiatkowski, T., Ma, J., Ni, J., Saralegui, L.S., Schuster, T., Cohen, W.W., Collins, M., Das, D., Metzler, D., Petrov, S., Webster, K.: Attributed question answering: Evaluation and modeling for attributed large language models (2023), https://arxiv.org/abs/2212.08037
2. Chandrasekaran, B.: Generic tasks in knowledge-based reasoning: High-level building blocks for expert system design. IEEE Intelligent Systems **1**(3), 23–30 (1986)
3. Chandrasekaran, B., Johnson, T.R., Smith, J.W.: Task-structure analysis for knowledge modeling. Communications of the ACM **35**(9), 124–137 (1992)
4. Dass, R.K., Madhusudhana, R.H., Deye, E.C., Verma, S., Bydlon, T.A., Brazil, G., Goel, A.K.: Ivy: A hybrid knowledge-based and generative ai coach for explaining procedural skills. In: Artificial Intelligence in Education. Lecture Notes in Computer Science, vol. 15878, pp. 233–246. Springer (2025). https://doi.org/10.1007/978-3-031-98417-4_17

5. Dass, R.K., Puri, S., Khandelwal, A., Jin, X., Goel, A.K.: Developing models of procedural skills using an ai-assisted text-to-model approach (2026), preprint
6. Fujisawa, I., Nobe, S., Seto, H., Onda, R., Uchida, Y., Ikoma, H., Chien, P.C., Kanai, R.: Procbench: Benchmark for multi-step reasoning and following procedure (2024), https://arxiv.org/abs/2410.03117
7. Gao, M., Hu, X., Ruan, J., Pu, X., Wan, X.: Llm-based nlg evaluation: Current status and challenges (2025), https://arxiv.org/abs/2402.01383
8. Goel, A.K., Rugaber, S.: Gaia: A cad-like environment for designing game-playing agents. IEEE Intelligent Systems **32**(3), 60–67 (2017)
9. Honovich, O., Aharoni, R., Herzig, J., Taitelbaum, H., Kukliansky, D., Cohen, V., Scialom, T., Szpektor, I., Hassidim, A., Matias, Y.: True: Re-evaluating factual consistency evaluation. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 3905–3920. Association for Computational Linguistics (2022). https://doi.org/10.18653/v1/2022.naacl-main.287
10. Hu, N., Chen, J., Wu, Y., Qi, G., Wang, H., Bi, S., Chen, Y., Wu, T., Pan, J.Z.: Can llms evaluate complex attribution in qa? automatic benchmarking using knowledge graphs (2025), https://arxiv.org/abs/2401.14640
11. Khot, T., Clark, P., Guerquin, M., Jansen, P., Sabharwal, A.: Qasc: A dataset for question answering via sentence composition. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 8082–8090 (2020)
12. Koedinger, K.R., Corbett, A.T.: Cognitive tutors: Technology bringing learning science to the classroom. In: Sawyer, R.K. (ed.) The Cambridge Handbook of the Learning Sciences, pp. 61–78. Cambridge University Press, New York, NY (2006)
13. Koedinger, K.R., Corbett, A.T., Perfetti, C.: The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. Cognitive Science **36**(5), 757–798 (2012)
14. Lee-Urban, S.M.: TMK Models to HTNs: Translating Process Models into Hierarchical Task Networks. Ph.d. thesis, Lehigh University (2005)
15. Lum, C., Deye, E., Brazil, G., Bydlon, T., Verma, S., Madhusudhana, R., Dass, R., Goel, A.: Designing an ai coaching system for interactive video-based skill learning. In: International Conference on Intelligent Tutoring Systems. pp. 281–291. Springer (2025)
16. Min, S., Krishna, K., Lyu, X., Lewis, M., Yih, W.t., Koh, P.W., Iyyer, M., Zettlemoyer, L., Hajishirzi, H.: Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. pp. 12076–12100. Association for Computational Linguistics (2023). https://doi.org/10.18653/v1/2023.emnlp-main.741
17. Murdock, J.W., Goel, A.K.: Meta-case-based reasoning: Self-improvement through self-understanding. Journal of Experimental & Theoretical Artificial Intelligence **20**(1), 1–36 (2008)
18. Nguyen, T.S., Yang, H., Neoh, T.Y., Zhang, H., Yeo Keat, E., Fernando, B.: PKR-QA: A benchmark for procedural knowledge reasoning with knowledge module learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 40, pp. 24549–24557 (2026). https://doi.org/10.1609/aaai.v40i29.39638, https://doi.org/10.1609/aaai.v40i29.39638
19. Rashkin, H., Nikolaev, V., Lamm, M., Aroyo, L., Collins, M., Das, D., Petrov, S., Tomar, G.S., Turc, I., Reitter, D.: Measuring attribution in natural language generation models. Computational Linguistics **49**(4), 777–840 (Dec 2023).

https://doi.org/10.1162/coli_a_00486, https://aclanthology.org/2023.cl-4.2/

20. Trivedi, H., Balasubramanian, N., Khot, T., Sabharwal, A.: Musique: Multihop questions via single-hop question composition. Transactions of the Association for Computational Linguistics **10**, 539–554 (2022). https://doi.org/10.1162/tacl_a_00475
21. VanLehn, K.: The behavior of tutoring systems. International Journal of Artificial Intelligence in Education **16**(3), 227–265 (2006)
22. Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W.W., Salakhutdinov, R., Manning, C.D.: Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 2369–2380. Association for Computational Linguistics (2018). https://doi.org/10.18653/v1/D18-1259
23. Zhang, Y., Li, Y., Cui, L., Cai, D., Liu, L., Fu, T., Huang, X., Zhao, E., Zhang, Y., Chen, Y., et al.: Siren's song in the ai ocean: A survey on hallucination in large language models. Computational Linguistics **51**(4), 1373–1418 (2025)