Why Large Language Models Cannot Directly Write PSR Reports: A Framework for Learning from Expert Revisions while Preserving Value Integrity

Sang-Beom Kang[a*], Seung-Hyeok Yang[a], Dae-Young Lee[a]
[a]*FNC Technology Co., Ltd., 13 Heungdeok 1-ro, 32F, Giheung-gu, Yongin-si, Gyeonggi-do, 16954, Korea*
[*]*Corresponding author: blueso55@fnctech.com*

Keywords : Periodic Safety Review, Large Language Model, Value Integrity, Learning from Revisions, Multi-Agent

1. Introduction

A Periodic Safety Review (PSR) report is a high-assurance regulatory document, prepared under the safety assessment framework prescribed by nuclear safety legislation and regulatory guidelines [1]. Each report spans several thousand pages per plant, and every statement in it must remain traceable to the underlying assessment basis documents. For each safety factor, authors cross-check measured data, inspection results, and safety performance indicators, and transcribe the required values into the body text and tables, a process prone to mismatched figures, incorrect notation, and omitted items. As continued operation expands and the number of units under assessment grows, improving both the efficiency and the reliability of report preparation has become a pressing need. Large language models (LLMs) are an increasingly attractive option, yet delegating the entire drafting process to an LLM yields results that are hard to trust: the reference values that underpin safety judgments can be subtly altered during generation [2].

We address this limitation by exploiting the report preparation procedure itself. In practice, a PSR report undergoes multiple rounds of expert revision before a draft is submitted. Motivated by this workflow, we propose a framework in which an LLM generates a draft, an expert revises it, and the expert's writing rules are derived and accumulated from the differences between the two versions. Because the accumulated rules inform the generation of subsequent drafts, the drafts converge toward the expert's output as revisions are repeated. Numerical and reference values, meanwhile, are extracted directly from the source documents along a path separate from LLM generation, so rule learning introduces no value distortion and the reliability required of a regulatory document is preserved.

2. Background and Approach

2.1 Limitations of LLM-based Report Generation

Several studies have noted that LLMs struggle with reports centered on tables and numerical values [3,4]. The root cause is the generation mechanism itself: an LLM composes text by probabilistically predicting the next token, so numbers and unit symbols present in the source can drift during generation. Despite strong performance on natural-language narration, LLMs remain brittle on structured tables, with misaligned columns and altered cell values reported in prior work [4]. Our preliminary experiments show the same failure mode: when sentences containing numerical values were passed through as-is, particles and sentence endings were rewritten, and numbers and unit symbols were transformed. In an ordinary document such changes would be minor; in a regulatory document, a single deviant value undermines the basis of a safety judgment and compromises the credibility of the entire report. Value accuracy must therefore be guaranteed outside the LLM's generation process, with the LLM confined to the auxiliary role of composing and polishing prose.

2.2 Accumulating Revision Knowledge versus Fine-tuning

Fine-tuning is the conventional route for injecting expert knowledge into a model, but it is poorly matched to this setting. Training is costly and must be repeated whenever regulations or writing conventions change, and because the acquired knowledge is dispersed across model weights, one cannot inspect which rules the model has internalized. For regulatory documents, where the rules being applied must be explicit and verifiable, this opacity, combined with the training cost, makes fine-tuning ill-suited to PSR report preparation.

We instead accumulate the expert's revision knowledge as explicit rules. Throughout this paper, a *draft* denotes the initial version generated by the LLM, and a *revision* denotes the version reviewed and corrected by an expert. We define *revision learning* as deriving rules by comparing a draft with its revision, accumulating them, and referencing them in subsequent generation. Learning here means the explicit accumulation of rules, not the updating of model weights. Weight-free approaches that improve performance by accumulating natural-language feedback and experience have recently proven effective [5,6]. Whereas those methods learn from self-generated feedback, our framework learns from expert revisions, a direct signal of what a domain expert changed and why

[7], and recent LLMs are capable of analyzing such differences to identify the regularities underlying the corrections. The benefits are twofold: knowledge accumulates continuously without retraining, and because rules are managed explicitly, they can be reviewed and verified.

3. Proposed Framework

The proposed framework is a cycle of four stages, drafting, revision analysis, rule verification, and rule accumulation, each performed by a separate agent with a distinct role, as shown in Fig. 1 [8]. This separation localizes errors to the stage where they arise and allows the rationale of each stage to be examined independently. The cycle repeats with every revision, and the writing rules grow more refined with each iteration [9].



Fig. 1. Iterative improvement cycle of the expert-revision learning framework.

3.1 Drafting Agent

The drafting agent structures the assessment basis documents while preserving the standard table of contents (the chapter-section-paragraph hierarchy) defined for each PSR safety factor in the regulatory guidelines. It then assembles the evidence required for each item from multiple documents and generates a draft, consulting the accumulated rules to decide how the narrative is organized and how tables are composed. Value accuracy is guaranteed outside this generation process: numbers and reference values are extracted directly from the source text and inserted verbatim, never passing through LLM generation. The LLM participates only in semantic judgments, i.e., whether differently notated names across documents denote the same value, and whether two sentences carry the same meaning. Table filling likewise prioritizes LLM-free methods, attempted in descending order of reliability: transferring values while preserving the merge structure of the source table, copying whole columns by aligning source and draft columns, and importing the corresponding paragraph verbatim from the body text. Only cells these methods leave unfilled receive LLM assistance, and even then the filled values are independently verified against the source. The principle that values are never delegated to LLM generation holds no matter how many rules accumulate.

3.2 Revision Analysis Agent

The revision analysis agent aligns the draft with the expert revision and locates every change. Each difference is classified as a value replacement, an expression correction, an item addition, a regulation incorporation, and so on. The agent then derives candidate rules capturing why the expert made each change, in a form such as 'ratio-type formulas are not entered in tables.' The LLM does not produce answers of its own at this stage; it only observes the expert's corrections and infers the regularities within them. Because generalizing from a single correction is risky, derived rules remain candidates and are adopted only after passing the verification of the next stage.

3.3 Rule Verification Agent

The rule verification agent tests whether a candidate rule actually holds. It applies the candidate to already-completed reports and checks whether the outcome matches the values in those reports. Only rules that hold consistently across multiple cases without compromising value integrity are adopted. Because this verification is grounded in completed reports and their assessment basis documents rather than in the model's self-assessment, it runs automatically, and incorrect rules are prevented from propagating into subsequent documents without case-by-case human inspection. The explicitness and verifiability demanded of regulatory documents are secured at this stage.

3.4 Rule Accumulation and Iteration

Rules that pass verification are added to the rule set and inform the next draft. As revisions accumulate, the rule set grows more refined, and draft quality progressively approaches the expert's revised output, without retraining the model.

4. Preliminary Application and Analysis

While this paper proposes the framework as a whole, our first application verifies its prerequisite: the value integrity principle. If numerical values are distorted at the draft stage, the expert revision becomes

contaminated with one-off corrections of simple numerical errors; only when the draft's numerical fidelity is guaranteed can pure writing rules be correctly derived from draft-revision differences. We therefore conducted a preliminary application on the '8.1 Safety Performance' report of the second PSR for Hanul Units 5 and 6 and its assessment basis documents.

To characterize how the completed report's narrative relates to the assessment basis documents, we aligned and compared the two documents paragraph by paragraph. The narrative fell into four types: verbatim copies of sentences or values from the basis documents (~43%), notation-only substitutions (~3%), restatements that rewrite basis-document data as prose (~23%), and new narrative covering content absent from the basis documents (~31%). The first two types, about 46% combined, can be reproduced mechanically from the basis documents, so automation pays off immediately. The remaining ~54%, restatement and new narrative, is where expert judgment intervenes, and is precisely the region where revision learning can contribute once value integrity is secured.

Table filling confirmed that the value integrity principle operates in practice. Tables with merged cells, such as centrifugal pump test acceptance criteria, were populated without breaking the merge structure. Tables of performance grades and calculation formulas were cross-checked cell by cell against the designated safety performance indicator data, cleanly separating verified values from those requiring further confirmation. Because values never passed through LLM generation, no case of numerical transformation occurred; where notation styles or inequality directions differed across documents, they were consolidated into a single format at the verification stage. Even when a formula in the completed report was embedded as an equation object and thus unreadable as text, the same formula was available as text in the assessment basis document and was restored verbatim from it. Indicators missing from any single basis document were filled by consulting multiple documents jointly. Coverage can thus be extended while still transferring reference values verbatim from the source.

This application, however, establishes value integrity and the validity of draft composition; how much the revision learning cycle actually improves draft quality remains to be verified quantitatively in future work, once real revision cases are obtained.

5. Conclusions

We addressed the problem that LLMs may alter reference values, which makes end-to-end PSR report writing untrustworthy. Our solution derives and accumulates rules from the actual workflow in which experts revise drafts: candidate rules are extracted from draft-revision differences, and only those that pass automatic cross-checking against existing completed reports are adopted. This blocks the accumulation of incorrect rules while preserving the expert's writing knowledge in explicit form. Numbers and reference values are transferred verbatim from the source along a path separate from LLM generation, so value integrity is maintained. Because only rules accumulate and model weights never change, the cost is lower than fine-tuning and the incorporated rules remain inspectable at any time. Our preliminary application confirmed that the value integrity principle operates on a real report and its assessment basis documents. In future work, we will collect real revision cases in which experts iteratively correct drafts, apply the full framework, and quantitatively evaluate how draft quality improves over successive revisions; we also plan to extend the application to other safety factors to establish the framework's generality.

ACKNOWLEDGMENTS

This work was supported by Korea Hydro & Nuclear Power Co., Ltd. (No. 2025-AI-001-01).

REFERENCES

[1] International Atomic Energy Agency, Periodic Safety Review for Nuclear Power Plants, IAEA Safety Standards Series No. SSG-25, IAEA, Vienna, 2013.
[2] L. Huang et al., A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions, ACM Transactions on Information Systems, 2024.
[3] C. Li et al., TReB: A Comprehensive Benchmark for Evaluating Table Reasoning Capabilities of Large Language Models, arXiv:2506.18421, 2025.
[4] Rajmohan C, S. Harne, A. Agarwal, LLM driven Text-to-Table Generation through Sub-Tasks Guidance and Iterative Refinement, arXiv:2508.08653, 2025.
[5] N. Shinn et al., Reflexion: Language Agents with Verbal Reinforcement Learning, Advances in Neural Information Processing Systems (NeurIPS 36), 2023.
[6] A. Zhao et al., ExpeL: LLM Agents Are Experiential Learners, Proceedings of the AAAI Conference on Artificial Intelligence, 2024.
[7] A. Xie, C.-Y. Cheng, F. Huang, Y. Li, Leveraging Human Revisions for Improving Text-to-Layout Models, arXiv:2405.13026, 2024.
[8] S. Hong et al., MetaGPT: Meta Programming for a Multi-Agent Collaborative Framework, International Conference on Learning Representations (ICLR), 2024.
[9] A. Madaan et al., Self-Refine: Iterative Refinement with Self-Feedback, Advances in Neural Information Processing Systems (NeurIPS 36), 2023.