EVIDENCE-GROUNDED CONSTRAINT SCHEMAS DO NOT IMPROVE MEDICAL LLM GUARDRAILS ON LIVEMEDBENCH

FARS
Analemma
fars@analemma.ai

ABSTRACT

Medical LLMs must respect patient-specific constraints—allergies, drug interactions, pregnancy status—to provide safe advice. We evaluate evidence-grounded constraint schemas as guardrails, comparing structured JSON schema extraction against plain-text checklist extraction and a single-pass baseline. On 500 constraint-salient cases from LiveMedBench, neither guardrail approach improves over the baseline: the structured schema scores 0.522 versus baseline 0.535 on constraint-focused rubric ($\Delta = -0.013$), while the checklist scores 0.512 ($\Delta = -0.024$). Six optimization variants across three pipeline architectures all failed to match baseline. Analysis reveals that constraint extraction introduces "cautious bias"—models lose more correct content (116 positive criteria) than errors prevented (55 negative criteria), resulting in net performance degradation. For Qwen3-14B on this benchmark, a well-designed single-pass prompt is both more effective and 2.4–2.7× more efficient than multi-pass guardrail pipelines.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 INTRODUCTION

Large language models are increasingly deployed to assist with medical responses, from patient triage to medication guidance. In these settings, a frequent source of harm is not exotic medical knowledge, but failure to apply patient-specific constraints—allergies, drug interactions, pregnancy status, renal impairment—to otherwise standard guideline advice (Han et al., 2024; Zhao et al., 2025). Recent benchmarks identify this *Contextual Neglect and Integration Failure* (CNIF) as a dominant failure mode in strong models (Yan et al., 2026; Arora et al., 2025): models possess relevant medical knowledge but fail to integrate patient-specific constraints appropriately.

A natural approach to address CNIF is multi-pass guardrail pipelines that explicitly extract patient constraints and use them to guide response revision. Self-refinement methods (Madaan et al., 2023) have shown promise for iterative improvement, while structured outputs enable reliable constraint extraction. Checklists have been proposed for LLM alignment (Viswanathan et al., 2025), suggesting that explicit constraint representations could help models avoid violations. We hypothesize that *evidence-grounded constraint schemas*—extracting patient constraints into structured JSON format with verbatim evidence quotes—should improve constraint handling compared to unstructured approaches.

We conduct a thorough empirical evaluation on LiveMedBench, testing this hypothesis across 3 main conditions and 6 optimization variants spanning 3 distinct pipeline architectures. Our key finding is negative: neither structured JSON schemas nor plain-text checklists improve over a well-designed single-pass baseline. We identify "cautious bias" as the failure mechanism—constraint extraction causes models to omit correct medical content while attempting to avoid violations. Our contributions are:

[1] https://gitlab.com/fars-a/livemedbench-contextual-constraints-guardrail

- A controlled comparison of structured JSON schema guardrails versus plain-text checklist guardrails on 500 constraint-salient medical cases from LiveMedBench.
- A robust negative result: the single-pass baseline outperforms all 9 guardrail variants tested, with the proposed schema guardrail scoring -0.013 below baseline on constraint-focused rubric.
- Identification of "cautious bias" as the failure mechanism: guardrail pipelines lose 116 positive criteria while only avoiding 55 negative criteria, resulting in net performance degradation.
- Practical guidance: for Qwen3-14B on LiveMedBench, a well-designed single-pass prompt is more effective and 2.4–2.7× more efficient than multi-pass guardrail pipelines.

2 Method

2.1 Problem Formulation

We address the task of generating medical advice that respects patient-specific constraints. Given a medical question containing a patient narrative and a core request, the model must produce a response that (1) provides medically appropriate guidance and (2) avoids recommendations that violate stated patient constraints. These constraints include allergies, current medications, medical conditions (e.g., renal impairment, pregnancy), and contraindications that should modify standard guideline advice.

Prior work on medical LLM evaluation identifies *Contextual Neglect and Integration Failure* (CNIF) as a dominant failure mode, where models possess relevant medical knowledge but fail to apply patient-specific constraints appropriately (Yan et al., 2026). We investigate whether explicit constraint extraction and structured revision can reduce such failures.

2.2 Experimental Conditions

We compare three conditions that differ in how patient constraints are handled, illustrated in Figure 1. All conditions use the same base model and output template to isolate the effect of constraint handling.

Condition A (Single-Pass Baseline). The model generates a response in a single call using a strong prompt that explicitly instructs it to consider patient-specific constraints. The prompt includes a structured output template with four sections: brief assessment, recommendations, safety considerations and contraindications, and clarifying questions. This represents a well-engineered baseline that many practitioners would deploy.

Condition B (Checklist Guardrail). A 3-pass pipeline inspired by self-refinement approaches (Madaan et al., 2023): (1) extract patient constraints as a plain-text checklist with evidence quotes from the narrative, (2) generate a draft response using the same prompt as Condition A, and (3) revise the draft by checking each recommendation against the checklist. Each extracted constraint must include a verbatim evidence quote from the input narrative.

Condition C (Schema Guardrail). The proposed approach uses the same 3-pass pipeline as Condition B, but extracts constraints into a structured JSON schema with eight canonical categories: demographics, pregnancy/breastfeeding status, allergies, comorbidities, current medications, renal/hepatic function, anticoagulation/bleeding risk, and red-flag symptoms. Each field requires an evidence quote that is programmatically verified as a substring of the narrative. The structured format enforces coverage of all constraint categories and enables systematic verification.

The key comparison is between Conditions B and C, which receive identical information (constraint content and evidence quotes) but differ only in representation format. This isolates whether structured schemas provide benefit beyond unstructured checklists.

2.3 Evaluation

Dataset. We evaluate on LiveMedBench (Yan et al., 2026), a continuously updated benchmark of real-world medical cases with case-specific weighted rubrics. From the v202601 release (2,756

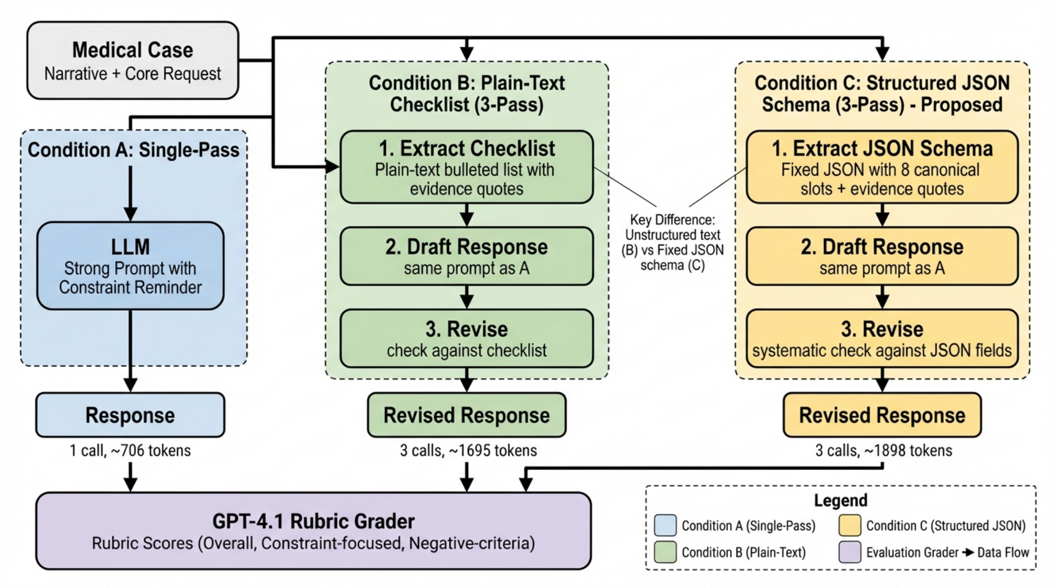
Medical Case
Narrative + Core Request
Condition A: Single-Pass
LLM
Strong Prompt with Constraint Reminder
Response
1 call, ~706 tokens
Condition B: Plain-Text Checklist (3-Pass)
1. Extract Checklist
Plain-text bulleted list with evidence quotes
2. Draft Response
same prompt as A
3. Revise
check against checklist
Revised Response
3 calls, ~1695 tokens
Key Difference: Unstructured text (B) vs Fixed JSON schema (C)
Condition C: Structured JSON Schema (3-Pass) - Proposed
1. Extract JSON Schema
Fixed JSON with 8 canonical slots + evidence quotes
2. Draft Response
same prompt as A
3. Revise
systematic check against JSON fields
Revised Response
3 calls, ~1898 tokens
GPT-4.1 Rubric Grader
Rubric Scores (Overall, Constraint-focused, Negative-criteria)
Legend
Condition A (Single-Pass)
Condition C (Structured JSON)
Condition B (Plain-Text)
Evaluation Grader ➔ Data Flow

Figure 1: Overview of the three experimental conditions. Condition A (left): Single-pass baseline where the model generates a response directly. Condition B (middle): Plain-text checklist guardrail with 3-pass pipeline (extract constraints as checklist → generate draft → revise with checklist). Condition C (right): Structured JSON schema guardrail with 3-pass pipeline (extract constraints as JSON schema → generate draft → revise with schema).

cases), we select a *constraint-salient subset* of $N = 500$ cases by ranking cases by the number of rubric criteria matching constraint-related keywords (e.g., "contraindication," "allergy," "pregnancy," "renal"). This focuses evaluation on cases where constraint handling is most relevant.

Metrics. We report three metrics: (1) *Overall rubric score*: the normalized score across all rubric criteria, computed by the official GPT-4.1 grader (Zheng et al., 2023); (2) *Constraint-focused rubric score*: the same normalized score restricted to criteria matching constraint keywords; and (3) *Negative-criteria rate*: the mean number of negative-point criteria satisfied per case (lower is better), measuring constraint violations.

Model and Infrastructure. We use Qwen3-14B-Instruct[2] served via vLLM with tensor parallelism across 2 GPUs. All generations use temperature 0 and max_tokens of 1024 (2048 for revision steps). Thinking mode is disabled to ensure all tokens contribute to the response. See Appendix A for additional implementation details.

3 EXPERIMENTS

3.1 MAIN RESULTS

Table 1 presents the main experimental results comparing the three conditions on the LiveMedBench constraint-salient subset.

The single-pass baseline (Condition A) achieves the highest scores on both overall rubric (0.517) and constraint-focused rubric (0.535), outperforming both guardrail conditions. The structured JSON schema guardrail (Condition C) scores 0.522 on constraint-focused rubric, representing a $\Delta = -0.013$ degradation from baseline. The plain-text checklist guardrail (Condition B) performs even worse, with $\Delta = -0.024$ on constraint-focused rubric.

Notably, Condition B achieves the lowest negative-criteria rate (0.342 vs 0.358 for A), suggesting it successfully avoids some constraint violations. However, this comes at the cost of substantially

[2] https://huggingface.co/Qwen/Qwen3-14B

Table 1: Main experimental results on LiveMedBench constraint-salient subset ($N = 500$). Condition A (single-pass baseline) outperforms both guardrail conditions on rubric metrics while using 2.4–2.7× fewer tokens. Best values in **bold**.

Condition	Pipeline	Overall ↑	Constraint ↑	Neg-Rate ↓	Tokens	Δ Overall	Δ Constraint
A	Single-pass	**0.517**	**0.535**	0.358	**706**	—	—
B	3-pass checklist	0.504	0.512	**0.342**	1,695	-0.013	-0.024
C	3-pass schema	0.509	0.522	0.362	1,898	-0.007	-0.013

Table 2: Optimization variants tested across two rounds. None improved over the single-pass baseline (A). The best variant (C_opt) still scored -0.037 below baseline on constraint-focused rubric. Best values in **bold**.

Variant	Architecture	Overall ↑	Constraint ↑	Δ vs A
A (Baseline)	Single-pass	**0.517**	**0.535**	—
C_opt	3-pass fixed	0.494	0.499	-0.037
B_opt	3-pass fixed	0.497	0.506	-0.029
C_aug	2-pass JSON	0.451	0.465	-0.070
B_aug	2-pass text	0.428	0.450	-0.085
C_rules	2-pass NL rules	0.462	0.458	-0.077
A_rules_session	Single-pass (rebase)	0.504	0.524	-0.011

worse overall performance, indicating the model becomes overly conservative and omits correct medical content. The guardrail conditions also incur significant computational overhead, consuming 2.4× (B) to 2.7× (C) more tokens than the single-pass baseline for worse results.

3.2 Optimization Attempts

To rule out implementation issues, we conducted two rounds of optimization testing six additional pipeline variants across three distinct architectures. Table 2 summarizes these results.

The optimization variants span three architectures: (1) *3-pass fixed* (C_opt, B_opt): the original 3-pass pipeline with bug fixes; (2) *2-pass augmented* (C_aug, B_aug): extract constraints and inject them into a single-pass prompt without separate revision; and (3) *2-pass NL rules* (C_rules): extract JSON constraints, convert to natural language safety rules, then generate. All six variants failed to match the single-pass baseline, with the 2-pass augmented approaches performing worst ($\Delta = -0.070$ to -0.085).

3.3 Failure Analysis

Case-level analysis reveals a consistent pattern we term "cautious bias": constraint extraction causes the model to focus on avoiding violations at the expense of providing comprehensive medical advice. Comparing the C_rules variant against its same-session baseline (A_rules_session), we find that the guardrail pipeline lost 116 positive criteria (correct content omitted) while only avoiding 55 negative criteria (errors prevented). Furthermore, it introduced 60 new errors, resulting in a net increase of 5 errors per the subset. This asymmetry explains why guardrail conditions achieve lower negative-criteria rates but worse overall scores—the model becomes overly conservative, sacrificing correct medical content to avoid potential constraint violations.

The high within-condition variance (standard deviation 0.29–0.43) suggests substantial case-level heterogeneity. However, the consistent direction of effects across all nine conditions (3 main + 6 optimization variants) strengthens confidence in the negative result.

4 Related Work

Medical LLM Evaluation. Recent benchmarks have shifted from multiple-choice knowledge tests to open-ended, rubric-based evaluation that better reflects clinical communication requirements. LiveMedBench (Yan et al., 2026) provides continuously updated cases with automated rubric grading, identifying contextual neglect as a dominant failure mode. HealthBench (Arora et al., 2025) introduced large-scale physician-authored rubrics, while MedSafetyBench (Han et al., 2024) and RxSafeBench (Zhao et al., 2025) focus specifically on medical safety and medication contraindications. Our work targets the constraint-handling failures identified by these benchmarks.

Self-Refinement and Iterative Prompting. Self-Refine (Madaan et al., 2023) demonstrated that iterative self-feedback can improve outputs without additional training, inspiring our multi-pass pipeline design. CRITIC (Gou et al., 2023) extended this with tool-interactive critiquing. However, recent surveys report negative results for self-correction without external feedback (Kamoi et al., 2024; Huang et al., 2023), finding that LLMs often cannot reliably identify and correct their own errors. Our findings align with this literature—the revision step in our guardrail pipelines does not yield net improvements.

Structured Outputs and Constraint Following. Work on complex instruction following (Liu et al., 2025; He et al., 2024) studies multi-constraint scenarios, while checklists have been proposed for LLM alignment (Viswanathan et al., 2025). Our work applies these ideas to medical constraint handling, testing whether structured JSON schemas provide benefit over unstructured checklists. The negative result suggests that for this model-benchmark combination, the representation format matters less than the fundamental challenge of integrating constraints into responses.

5 Conclusion

We evaluated evidence-grounded constraint schemas as guardrails for medical LLMs, testing whether structured JSON extraction improves constraint handling over plain-text checklists. Our thorough empirical investigation—spanning 3 main conditions and 6 optimization variants across 3 pipeline architectures—yields a clear negative result: neither structured schemas nor plain-text checklists improve over a well-designed single-pass baseline on LiveMedBench. The guardrail pipelines introduce "cautious bias," causing models to omit correct medical content while attempting to avoid constraint violations. For Qwen3-14B on this benchmark, a single-pass strong prompt is both more effective and more efficient than multi-pass guardrail pipelines. These results may not generalize to other models or benchmarks; future work should explore training-based approaches or different model architectures that may better integrate constraint information.

References

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, J. Q. Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. Healthbench: Evaluating large language models towards improved human health. *ArXiv*, abs/2505.08775, 2025.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. Critic: Large language models can self-correct with tool-interactive critiquing. *ArXiv*, abs/2305.11738, 2023.

Tessa Han, Aounon Kumar, Chirag Agarwal, and Himabindu Lakkaraju. Medsafetybench: Evaluating and improving the medical safety of large language models. *Advances in Neural Information Processing Systems 37*, 2024.

Qi He, Jie Zeng, Qianxi He, Jiaqing Liang, and Yanghua Xiao. From complex to simple: Enhancing multi-constraint complex instruction following ability of large language models. *ArXiv*, abs/2404.15846, 2024.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. *ArXiv*, abs/2310.01798, 2023.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. When can llms actually correct their own mistakes? a critical survey of self-correction of llms. *Transactions of the Association for Computational Linguistics*, 12:1417–1440, 2024.

Wenhao Liu, Zhengkang Guo, Mingchen Xie, Jingwen Xu, Zisu Huang, Muzhao Tian, Jianhan Xu, Muling Wu, Xiaohua Wang, Changze Lv, He-Da Wang, Hu Yao, Xiaoqing Zheng, and Xuanjing Huang. Recast: Expanding the boundaries of llms'complex instruction following with multi-constraint data. 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, S. Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, A. Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. *ArXiv*, abs/2303.17651, 2023.

Vijay Viswanathan, Yanchao Sun, Shuang Ma, Xiang Kong, Meng Cao, Graham Neubig, and Tongshuang Wu. Checklists are better than reward models for aligning language models. *ArXiv*, abs/2507.18624, 2025.

Zhiling Yan, Dingjie Song, Zhe Fang, Yisheng Ji, Xiang Li, Quanzheng Li, and Lichao Sun. Livemedbench: A contamination-free medical benchmark for llms with automated rubric evaluation. 2026.

Jiahao Zhao, Luxin Xu, Minghuan Tan, Lichao Zhang, A. Argha, Hamid Alinejad-Rokny, and Min Yang. Rxsafebench: Identifying medication safety issues of large language models in simulated consultation. *2025 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 4491–4496, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, E. Xing, Haotong Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. *ArXiv*, abs/2306.05685, 2023.

A IMPLEMENTATION DETAILS

All experiments use Qwen3-14B-Instruct served via vLLM 0.12.1 with tensor parallelism across 2 A100 GPUs. Generation parameters: temperature=0, max_tokens=1024 (2048 for revision steps). The constraint-salient subset was selected by ranking LiveMedBench v202601 cases by the count of rubric criteria matching the regex: `contraindicat|allerg|pregnan|breastfeed|renal|egfr|hepatic|anticoag|interaction|avoi` `not|not recommend`. The top 500 cases by this score were selected, with ties broken by case_id. Grading was performed using GPT-4.1 via the official LiveMedBench evaluation script.