Conversation as Belief Revision: GreedySAT Revision for Global Logical Consistency in Multi-Turn LLM Dialogues

Sanjan Baitalik[1], **Rajashik Datta**[2], **Amit Kumar Das** [2] **Sruti Das Choudhury**[3,4]

[1]Department of Computer Science & Engineering, Institute of Engineering & Management, Kolkata, West Bengal, India
[2]Department of Computer Science & Engineering (AI), Institute of Engineering & Management, Kolkata, West Bengal, India
[3]School of Natural Resources, University of Nebraska-Lincoln, Lincoln, NE, United States
[4]School of Computing, University of Nebraska-Lincoln, Lincoln, NE, United States
Sanjan.Baitalik2022@iem.edu.in, Rajashik.Datta2022@iem.edu.in, Amit@iem.edu.in, sdaschoudhury2@nebraska.edu

Abstract

Large language models (LLMs) are increasingly deployed as interactive assistants, yet their responses are often evaluated in isolation rather than as components of an evolving belief state. Recent benchmarks reveal that even state-of-the-art LLMs frequently violate basic logical consistency, especially under negation, multi-step entailment, or adversarial question sequences. We argue that multi-turn dialogue with an LLM should be viewed as a process of constructing and revising an explicit theory of the world.

We propose GreedySAT Revision, a lightweight, backend-agnostic framework that treats an LLM as a black-box generator of propositional commitments, wrapped by an external symbolic solver maintaining a globally consistent belief state. At each turn, the LLM proposes an answer to a query about a synthetic world; we map it to a propositional literal and tentatively add it to the current theory. A SAT/SMT-based checker verifies satisfiability against world rules and, if needed, performs minimal belief revision by retracting prior commitments. We instantiate this with API-only models—OpenAI `gpt-4.1-mini` and Gemini 2.5 Flash—without finetuning, evaluating on synthetic multi-turn logical dialogues under random and stress-test query schedules.

On `gpt-4.1-mini`, our solver-augmented system eliminates all inconsistent final belief states in adversarial "stress" settings (from $5/120$ to $0/120$ dialogues under direct prompting, and $3/120$ to $0/120$ under chain-of-thought), preserving task accuracy within 0.3 points. Across conditions, it incurs only 12 and 10 retractions, respectively—about 0.08–0.10 per dialogue. On Gemini 2.5 Flash, the belief-state interpreter boosts raw per-turn accuracy (e.g., from 0.32 to 0.49 and 0.29 to 0.41 in random and stress direct settings), with the solver rarely intervening. These results show that explicit, solver-checked global belief states provide strong logical guarantees for interactive LLMs without large or GPU-intensive models, paving the way for trustworthy neuro-symbolic multi-turn reasoning.

Code —
https://github.com/sanjanbaitalik/llm-logical-consistency

Introduction

Large language models (LLMs) have rapidly become the default interface for natural-language interaction, powering chatbots, copilots, and interactive assistants across many domains. As these systems move from single-turn question answering toward persistent multi-turn interactions, users naturally expect them to behave like rational agents: tracking facts over time, avoiding contradictions, and revising beliefs when confronted with new evidence. However, a growing body of work shows that contemporary LLMs systematically violate even basic logical consistency, especially under negation and multi-step entailment, despite achieving impressive scores on static benchmarks (Tian et al. 2021; Parmar et al. 2024; Han et al. 2024; Ghosh et al. 2025).

Most existing evaluations treat LLM outputs as independent predictions on individual items. In contrast, interactive deployments implicitly construct a *latent theory of the world* over the course of a dialogue. When a model answers "yes" to a proposition P and later answers "yes" to a proposition logically equivalent to $\neg P$, the two answers cannot both be correct in any classical model. Such logical incoherence undermines trust in agentic, tool-using systems and limits the applicability of LLMs in safety-critical or knowledge-intensive settings.

At the same time, there has been intense interest in multi-step reasoning and chain-of-thought prompting, which elicit longer natural-language reasoning traces from LLMs (Wei et al. 2022; Wang et al. 2023). These approaches improve performance on mathematical and symbolic reasoning benchmarks, but they do not by themselves guarantee that all statements produced across a dialogue are mutually consistent. In particular, a model may generate plausible step-by-step arguments for each question while still building a globally inconsistent set of commitments.

In this paper, we propose to treat multi-turn LLM dialogue as an instance of *belief revision* in a symbolic knowledge base. Rather than asking whether each answer is locally plausible, we explicitly represent the conversation as a growing propositional theory and enforce global logical consistency by constraining how this theory can evolve.

Approach: We introduce a simple, inference-time wrapper that maintains a *global belief state* over propositional atoms corresponding to entities and properties in a synthetic world. At each dialogue turn, the LLM is prompted with the world description, a natural-language summary of the current belief state, and a new question. Its answer is mapped

to a propositional literal (e.g., P or $\neg P$), which we tentatively add to the belief base. A classical propositional solver checks whether the augmented theory remains satisfiable with respect to the underlying world rules. If the theory is unsatisfiable, a greedy revision procedure retracts a small set of prior commitments until consistency is restored, logging each retraction as an explicit *belief change*.

Instantiations and setting: We instantiate this framework with two modest, API-only models: OpenAI `gpt-4.1-mini` and Gemini 2.5 Flash. Both are used strictly as black boxes: no finetuning, no access to logits, and no gradient updates. We evaluate on synthetic propositional worlds under two settings: (i) a *random* schedule where questions are sampled without adversarial structure, and (ii) a *stress* schedule where questions are ordered to amplify negation, equivalence, and other patterns that make it easy for an LLM to contradict itself over time.

Empirical findings: Our main empirical observations are:

- On `gpt-4.1-mini`, the solver-augmented system reduces the number of dialogues whose final belief state is unsatisfiable from $5/120$ to $0/120$ under direct prompting, and from $3/120$ to $0/120$ under chain-of-thought prompting in the stress setting. Task accuracy remains essentially unchanged (differences <0.3 points).
- The solver performs very few interventions: in the stress setting, it retracts only 12 (direct) and 10 (CoT) previous commitments across 120 dialogues of length 10, corresponding to roughly 0.08–0.10 retractions per dialogue.
- On the weaker Gemini 2.5 Flash backbone, the main benefit comes from our global belief-state interpreter (mapping per-turn answers into a coherent world-level labelling): the "final" accuracy jumps from 0.32 to 0.49 and from 0.29 to 0.41 in random and stress direct settings, and from 0.46 to 0.60 and 0.41 to 0.51 under CoT. The underlying theory rarely becomes unsatisfiable, so the solver almost never retracts anything, providing a useful sanity check.

This paper makes three main contributions:

1. We formalize multi-turn LLM dialogue as the incremental construction of a symbolic belief state, and define metrics for *global* logical coherence at the dialogue level (satisfiability and contradiction counts), beyond per-turn accuracy.
2. We introduce a lightweight, backend-agnostic pipeline that combines black-box LLMs with a classical solver to enforce global consistency via minimal belief revision, without any finetuning or GPU resources.
3. We provide an empirical study across two different LLM backends and two prompting modes (direct vs. chain-of-thought), showing that the solver wrapper can eliminate inconsistent belief states with negligible impact on task accuracy, and that explicit belief-state interpretation can substantially improve the performance of weaker models.

Related Work

A growing literature studies the logical reasoning abilities of LLMs using synthetic and semi-synthetic benchmarks that aim to isolate logical inference from other forms of reasoning. Early datasets such as RuleTaker (Clark, Tafjord, and Richardson 2020) and ProofWriter (Tafjord, Dalvi, and Clark 2021) train and evaluate transformers on entailment from sets of rules and facts expressed in natural language. More recent benchmarks, including LogicNLI (Tian et al. 2021), FOLIO (Han et al. 2024), LogicBench (Parmar et al. 2024), and LogiConBench, systematically probe first-order and propositional reasoning, with explicit control over rule depth, negation, and compositional structure. These works consistently find that even strong models struggle with deeper reasoning chains, nested quantification, and logically complex negations, and that apparent high scores on some benchmarks can mask substantial inconsistencies when responses are aggregated across related items (Ghosh et al. 2025).

Several studies have begun to measure *logical consistency* explicitly, either by constructing sets of logically related queries and checking whether the induced truth assignments admit a model, or by using knowledge-graph-based fact-checking tasks where consistency can be assessed against a structured graph (Ghosh et al. 2025).

Our work is inspired by these efforts, but differs in that we consider a *single, evolving dialogue* and treat the entire conversation as a dynamic theory that must remain satisfiable over time.

Neuro-symbolic approaches aim to combine the flexibility of neural models with the soundness guarantees of symbolic reasoning (DeLong, Mir, and Fleuriot 2024; Nawaz, Anees-ur Rahaman, and Saeed 2025). One prevalent pattern uses LLMs to translate natural language problems into formal representations that can be passed to external solvers, such as logic programming engines, SAT/SMT solvers, theorem provers, or planners. For example, Logic-LM (Pan et al. 2023) maps questions into a logic program and uses a symbolic reasoner to compute answers, while other systems translate natural language into first-order logic for verification, fact-checking, or fallacy detection (Pei, Du, and Jin 2025; Lalwani et al. 2024). Recent work on neurosymbolic NLP for mathematical reasoning similarly integrates symbolic feedback into training smaller models (Li et al. 2024).

Our framework follows the same high-level philosophy but adopts a different granularity. Rather than translating entire problems into a single symbolic query, we interpret each dialogue turn as a local addition to a global belief base. The solver is used not only to compute answers, but to *police the evolution* of the belief state, triggering belief revision whenever newly proposed commitments render the theory unsatisfiable.

Chain-of-thought (CoT) prompting and multi-step reasoning methods have demonstrated substantial gains on a wide range of benchmarks, particularly in arithmetic and symbolic reasoning (Wei et al. 2022; Wang et al. 2023; Plaat et al. 2025). Self-consistency decoding, where multiple CoT samples are generated and aggregated, further improves robustness by marginalizing over diverse reasoning paths (Wang et al. 2023). A recent surge of work investigates multi-step reasoning with external tools: LLMs generate code, call libraries, query search engines, or interact with

specialized solvers as part of a multi-step pipeline.

Our method can be viewed as a specialized tool-use pattern: the LLM is responsible for proposing atomic commitments in natural language, and a symbolic solver acts as a tool that enforces global constraints over time. Unlike many prior multi-step systems, we do not use the solver to *search* for a solution to a single problem instance. Instead, the solver maintains an invariant (global satisfiability) across the entire dialogue, supporting a belief-revision perspective that is closer to classical knowledge representation.

Our method can be viewed as a specialized tool-use pattern: the LLM is responsible for proposing atomic commitments in natural language, and a symbolic solver acts as a tool that enforces global constraints over time. Unlike many prior multi-step systems, we do not use the solver to *search* for a solution to a single problem instance. Instead, the solver maintains an invariant (global satisfiability) across the entire dialogue, supporting a belief-revision perspective that is closer to classical knowledge representation.

Materials and Methods

Synthetic Propositional Worlds

We design a family of small propositional worlds, each defined by: (i) a finite set of propositional atoms $\mathcal{P}$ (e.g., `A`, `B`, `C`), and (ii) a set of ground rules and facts $\mathcal{R}$ expressed as Horn-like implications and constraints. Each world is by construction logically consistent, and the ground-truth labels for atomic propositions are computed by classical model checking.

For each world, we generate natural-language paraphrases of the underlying rules and facts, yielding a short textual description that can be shown to the LLM. We also generate a set of $T = 10$ questions per dialogue, each querying the truth value of a single atomic proposition or its negation. Questions are phrased as yes/no queries (e.g., "Is A definitely true?", "Is it the case that B is not true?") with occasional uncertainty cues.

We consider two query schedules:

- **Random.** Questions are sampled without special structure, leading to relatively benign multi-turn interactions.
- **Stress.** Questions are ordered adversarially to emphasize negation, equivalence, and implicitly related forms (e.g., asking about P, then about $\neg P$, then about a rule that entails P), making it easy for the model to contradict itself if it does not maintain a stable internal theory.

For each setting and backbone model, we run 120 dialogues per system variant (baseline vs. solver-augmented), yielding 1,200 turns per combination.

Belief State Representation

We represent the agent's belief state at dialogue turn t as a pair $(\mathcal{R}, \mathcal{B}_t)$, where:

- $\mathcal{R}$ is the fixed set of world rules and base facts, and
- $\mathcal{B}_t = \{(\ell_i, \tau_i)\}_{i=1}^{k_t}$ is an ordered list of labelled literals, where each ℓ_i is either P or $\neg P$ for some $P \in \mathcal{P}$, and τ_i is the turn index at which that commitment was made.

The *induced theory* at turn t is

$$\mathcal{T}_t = \mathcal{R} \cup \{\ell_i : (\ell_i, \tau_i) \in \mathcal{B}_t\}. \quad (1)$$

A dialogue is said to be *globally consistent* at turn t if $\mathcal{T}_t$ is satisfiable in classical propositional logic.

At each turn, the LLM is prompted with: (i) the natural-language description of $\mathcal{R}$, and (ii) a short natural-language summary of the current belief state (e.g., "So far, we believe that A is true, B is false, and we are unsure about C."). The model is then asked to answer the next question with a label from $\{\text{YES}, \text{NO}, \text{UNKNOWN}\}$. This label is mapped to a literal ℓ_t or to a special "no commitment" symbol in the case of UNKNOWN.

Backbone Models and Prompting Modes

We instantiate the framework with two API-accessible LLMs:

- **OpenAI `gpt-4.1-mini`**, a relatively small but strong general-purpose model.
- **Gemini 2.5 Flash**, a fast, cost-effective model optimized for throughput rather than peak reasoning performance.

Both models are used as black boxes with identical decoding hyperparameters in all conditions for a given model, ensuring that differences arise only from the presence or absence of the solver wrapper.

For each backbone, we consider two prompting modes:

- **Direct.** The model is asked to answer the question in a single step (e.g., "Answer with 'Yes', 'No', or 'Unknown'.").
- **Chain-of-Thought (CoT).** The model is instructed to provide step-by-step reasoning before giving the final label, following a standard CoT pattern.

In both modes, we parse the final label using simple string-matching rules.

GreedySAT Revision

Baseline: Unchecked Belief Accumulation

The baseline system maintains the belief state $\mathcal{B}_t$ by simply appending each new literal proposed by the LLM, without any global consistency check. At turn t, after parsing the LLM's response:

1. If the model answers UNKNOWN, we leave $\mathcal{B}_t$ unchanged.
2. Otherwise, we map the answer to a literal ℓ_t and add (ℓ_t, t) to $\mathcal{B}_t$.

The final belief state $\mathcal{B}_T$ is then used to compute a *final label* for each proposition by reading off the last commitment for that atom (if any). We refer to the per-turn labels emitted directly by the model as *raw* labels, and to the labels derived from $\mathcal{B}_T$ as *final* labels.

Solver-Augmented Belief Revision

In the solver-augmented system, we insert a global satisfiability check and a greedy belief revision step after each new commitment. Let $\mathcal{B}_{t-1}$ be the current belief list and ℓ_t the new literal proposed by the LLM. We define a tentative belief list $\tilde{\mathcal{B}}_t = \mathcal{B}_{t-1} \cup \{(\ell_t, t)\}$ and check whether the corresponding theory $\tilde{\mathcal{T}}_t = \mathcal{R} \cup \{\ell_i : (\ell_i, \tau_i) \in \tilde{\mathcal{B}}_t\}$ is satisfiable.

If $\tilde{\mathcal{T}}_t$ is satisfiable, we accept the new belief: $\mathcal{B}_t \leftarrow \tilde{\mathcal{B}}_t$. Otherwise, we engage in a minimal-revision procedure via GreedySAT Revision, as detailed in Algorithm 1:

Algorithm 1: GreedySAT Revision

Require: Rules $\mathcal{R}$, prior beliefs $\mathcal{B}_{t-1}$, new literal ℓ_t
$\tilde{\mathcal{B}} \leftarrow \mathcal{B}_{t-1} \cup \{(\ell_t, t)\}$
if $\mathcal{R} \cup \{\ell : (\ell, \tau) \in \tilde{\mathcal{B}}\}$ is SAT **then**
  return $(\tilde{\mathcal{B}}, \emptyset)$
end if
Initialize Retracted $\leftarrow \emptyset$
Let Candidates be prior indices in $\mathcal{B}_{t-1}$ (e.g., sorted by increasing recency)
for each (ℓ_j, τ_j) in Candidates **do**
  Tentatively drop (ℓ_j, τ_j): $\tilde{\mathcal{B}}' \leftarrow \tilde{\mathcal{B}} \setminus \{(\ell_j, \tau_j)\}$
  if $\mathcal{R} \cup \{\ell : (\ell, \tau) \in \tilde{\mathcal{B}}'\}$ is SAT **then**
    $\tilde{\mathcal{B}} \leftarrow \tilde{\mathcal{B}}'$
    Retracted $\leftarrow$ Retracted $\cup \{(\ell_j, \tau_j)\}$
  end if
  if $\tilde{\mathcal{R}} = \mathcal{R} \cup \{\ell : (\ell, \tau) \in \tilde{\mathcal{B}}\}$ is SAT **then**
    break
  end if
end for
return $(\tilde{\mathcal{B}}, \text{Retracted})$

In practice, we adopt a simple heuristic ordering for Candidates that tends to retract older commitments first, though other preference orderings (e.g., confidence-based) could be incorporated without changing the overall framework. Each retracted literal is logged, allowing us to count the total number of retractions and to analyze which propositions are most frequently revised.

Importantly, the LLM itself is *not* informed about specific retractions; it only ever sees a natural-language summary of the current belief state. This design keeps the interface clean and avoids exposing solver internals to the model.

Experiments & Results

Evaluation Metrics

For each combination of backbone model, prompting mode, setting (random vs. stress), and system (baseline vs. solver-augmented), we use the following evaluation metrics:

- **Raw accuracy**: fraction of turns where the model's raw label matches the ground-truth label.
- **Final accuracy**: fraction of turns where the final label derived from the belief state matches the ground truth. This captures the effect of the global belief interpreter and any retractions.

Table 1: Results for OpenAI `gpt-4.1-mini`. Each condition uses 120 dialogues with 10 turns each. Final accuracy is computed from the global belief state. "Incons." counts dialogues whose final theory is unsatisfiable.

Setting	Mode	System	Final Acc.	Incons.	Retracts
Random	Direct	Baseline	0.937	0/120	0
Random	Direct	Solver	0.935	0/120	0
Stress	Direct	Baseline	0.926	5/120	0
Stress	Direct	Solver	0.926	0/120	12
Random	CoT	Baseline	0.937	1/120	0
Random	CoT	Solver	0.929	0/120	0
Stress	CoT	Baseline	0.922	3/120	0
Stress	CoT	Solver	0.926	0/120	10

- **Inconsistent dialogues**: number of dialogues in which the final theory $\mathcal{T}_T$ is unsatisfiable.
- **Contradiction counts**: number of dialogues in which a proposition receives conflicting truth assignments across the dialogue (e.g., both P and $\neg P$).
- **Belief retractions**: total count of literals removed by the solver during belief repair, and the average per dialogue.

Each condition is run on 120 dialogues of length $T = 10$, giving 1,200 turns per condition. We report means over all turns; confidence intervals can be added in a later revision if desired.

Results: OpenAI `gpt-4.1-mini`

Table 1 summarizes the main results for `gpt-4.1-mini`.

The following patterns emerge from the results in Table 1:

High base accuracy with localized errors: Under both direct and CoT prompting, `gpt-4.1-mini` achieves high final accuracy ($\approx$ 92–94%) across settings. However, in the stress condition, the baseline accumulates globally inconsistent belief states in a non-trivial fraction of dialogues: $5/120$ for direct and $3/120$ for CoT.

Consistency enforcement without accuracy loss: The solver-augmented system eliminates all inconsistent dialogues ($0/120$ in every setting), while leaving final accuracy essentially unchanged. Under direct prompting in the stress condition, final accuracy remains 0.926 before and after adding the solver; under CoT, it improves slightly from 0.922 to 0.926.

Sparse belief revision: The total number of retractions required to maintain global satisfiability is small: 12 (direct) and 10 (CoT) across 120 dialogues, or roughly one retraction per 10–12 dialogues. This suggests that the LLM is typically self-consistent at the local level, and that a small number of targeted revisions suffice to restore global coherence in difficult cases.

Results: Gemini 2.5 Flash

Table 2 reports analogous results for Gemini 2.5 Flash.

Table 2: Results for Gemini 2.5 Flash. Final accuracy reflects the global belief-state interpretation. No inconsistent dialogues were observed, and the solver never needed to retract any beliefs.

Setting	Mode	System	Final Acc.	Incons.	Retracts
Random	Direct	Baseline	0.489	$0/120$	0
Random	Direct	Solver	0.481	$0/120$	0
Stress	Direct	Baseline	0.408	$0/120$	0
Stress	Direct	Solver	0.406	$0/120$	0
Random	CoT	Baseline	0.604	$0/120$	0
Random	CoT	Solver	0.603	$0/120$	0
Stress	CoT	Baseline	0.512	$0/120$	0
Stress	CoT	Solver	0.505	$0/120$	0

Unlike `gpt-4.1-mini`, Gemini behaves conservatively on this task: it rarely constructs globally contradictory theories, and the solver never needs to revert any commitments. The more important effect is the jump from *raw* to *final* accuracy induced by the belief-state interpreter itself (e.g., from 0.32 to 0.49 and from 0.29 to 0.41 in random and stress direct settings, and from 0.46 to 0.60 and from 0.41 to 0.51 under CoT). This indicates that mapping per-turn answers into a single coherent world-level labelling serves as a form of global regularization, even for weaker models.

Cross-Backend Summary

Aggregating over random and stress settings, we find that:

- For `gpt-4.1-mini` with direct prompting, the baseline achieves an overall final accuracy of 0.9313 with $5/240$ inconsistent dialogues, while the solver-augmented system attains 0.9304 with $0/240$ inconsistent dialogues and 12 total retractions.
- For `gpt-4.1-mini` with CoT, the baseline achieves 0.9296 overall final accuracy with $4/240$ inconsistent dialogues, whereas the solver attains 0.9275 with $0/240$ inconsistencies and 10 retractions.
- For Gemini, the baseline and solver systems behave almost identically in terms of consistency, with the main gains coming from the belief-state interpreter rather than the solver itself.

These results support the view that our solver wrapper is most beneficial when the backbone model is strong enough to make confident but occasionally inconsistent commitments—precisely the regime where large-scale deployments of LLMs are likely to operate.

Ablation Studies

Effect of Chain-of-Thought Prompting

Comparing direct and CoT prompting on `gpt-4.1-mini`, we observe that: (i) CoT slightly improves or matches accuracy in most settings, but (ii) it does not by itself prevent globally inconsistent belief states. In the stress setting, CoT reduces raw uncertainty but still yields $3/120$ inconsistent dialogues in the baseline. The solver-augmented CoT system eliminates these inconsistencies while maintaining or slightly improving accuracy.

This suggests that CoT and our solver wrapper address complementary failure modes: CoT improves local reasoning for each question, while the solver ensures that all commitments made across the dialogue remain jointly satisfiable.

Direct vs. Belief-State Interpreted Labels

For Gemini, the most striking effect is the gap between raw and final accuracy. Even though the model rarely produces outright contradictions, its raw labels are noisy and sometimes inconsistent across turns. By interpreting answers through a global belief state and enforcing a single final label per proposition, we effectively smooth these inconsistencies, yielding substantial gains in final accuracy. This ablation highlights that an explicit belief representation can be beneficial even when no solver-based retraction is needed.

Solver vs. No Solver

The comparison between baseline and solver-augmented systems isolates the effect of belief revision. On `gpt-4.1-mini`, the solver eliminates all inconsistent dialogues in stress settings with negligible impact on accuracy and a very small number of retractions. On Gemini, the solver has effectively nothing to do; the induced theories are already satisfiable. This regime acts as a sanity check: the solver rarely intervenes, but its presence does not harm performance.

Conclusion

Our results indicate that enforcing global logical consistency via an external solver is both feasible and beneficial in multi-turn LLM interactions. Importantly, the approach is *model-agnostic* and requires no access to internal parameters, gradients, or logits—only the ability to prompt the LLM and parse its answers. This makes the framework particularly attractive in API-based, resource- constrained settings where finetuning or large model deployment is impractical.

We deliberately start with synthetic propositional worlds where ground truth is fully known and logical structure is transparent. An important direction for future work is to extend the framework to more realistic domains, such as knowledge-graph-based fact-checking, policy compliance checking, or safety-critical planning. In such settings, the belief state could be anchored in a knowledge graph or description-logic ontology rather than a purely propositional theory.

Our current belief revision procedure is intentionally simple: it uses a greedy heuristic to retract a small number of prior commitments until the theory becomes satisfiable. Classical work on belief revision and knowledge base update suggests a variety of more principled policies that trade off recency, specificity, and reliability. Integrating such policies, potentially guided by LLM-estimated confidence scores, is

a promising avenue for improving both the efficiency and interpretability of revisions.

In this paper, the solver is used purely at inference time. A natural extension is to use logical consistency as a form of feedback signal during training or adaptation, penalizing models that produce unsatisfiable belief states or encouraging them to propose revisions themselves. This connects to emerging work on neurosymbolic training schemes where logical constraints are injected into the learning process.

Finally, our framework can be seen as a special case of a broader class of composite AI systems, where LLMs orchestrate multiple external tools under global constraints. In practical deployments, a human operator could be given visibility into the belief state, retractions, and detected inconsistencies, and could approve or override solver decisions. Extending the present work to such human-in-the-loop settings, and to richer tool suites beyond a single solver, is an important direction for bridging the gap between purely neural agents and classical symbolic reasoning systems.

References

Clark, P.; Tafjord, O.; and Richardson, K. 2020. Transformers as Soft Reasoners over Language. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, 3882–3890. International Joint Conferences on Artificial Intelligence Organization. Main track.

DeLong, L. N.; Mir, R. F.; and Fleuriot, J. D. 2024. Neurosymbolic AI for reasoning over knowledge graphs: A survey. *IEEE Transactions on Neural Networks and Learning Systems*.

Ghosh, B.; Hasan, S.; Arafat, N. A.; and Khan, A. 2025. Logical Consistency of Large Language Models in Fact-Checking. In *The Thirteenth International Conference on Learning Representations*.

Han, S.; Schoelkopf, H.; Zhao, Y.; Qi, Z.; Riddell, M.; Zhou, W.; Coady, J.; Peng, D.; Qiao, Y.; Benson, L.; et al. 2024. Folio: Natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 22017–22031.

Lalwani, A.; Kim, T.; Chopra, L.; Hahn, C.; Jin, Z.; and Sachan, M. 2024. Autoformalizing Natural Language to First-Order Logic: A Case Study in Logical Fallacy Detection. *arXiv preprint arXiv:2405.02318*.

Li, Z.; Zhou, Z.; Yao, Y.; Zhang, X.; Li, Y.-F.; Cao, C.; Yang, F.; and Ma, X. 2024. Neuro-symbolic data generation for math reasoning. *Advances in Neural Information Processing Systems*, 37: 23488–23515.

Nawaz, U.; Anees-ur Rahaman, M.; and Saeed, Z. 2025. A review of neuro-symbolic AI integrating reasoning and learning for advanced cognitive systems. *Intelligent Systems with Applications*, 200541.

Pan, L.; Albalak, A.; Wang, X.; and Wang, W. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 3806–3824.

Parmar, M.; Patel, N.; Varshney, N.; Nakamura, M.; Luo, M.; Mashetty, S.; Mitra, A.; and Baral, C. 2024. LogicBench: Towards Systematic Evaluation of Logical Reasoning Ability of Large Language Models. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 13679–13707. Bangkok, Thailand: Association for Computational Linguistics.

Pei, Y.; Du, Y.; and Jin, X. 2025. FoVer: First-Order Logic Verification for Natural Language Reasoning. *Transactions of the Association for Computational Linguistics*, 13: 1340–1359.

Plaat, A.; Wong, A.; Verberne, S.; Broekens, J.; Van Stein, N.; and Bäck, T. 2025. Multi-Step Reasoning with Large Language Models, a Survey. *ACM Comput. Surv.*, 58(6).

Tafjord, O.; Dalvi, B.; and Clark, P. 2021. Proofwriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 3621–3634.

Tian, J.; Li, Y.; Chen, W.; Xiao, L.; He, H.; and Jin, Y. 2021. Diagnosing the first-order logical reasoning ability through LogicNLI. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3738–3747.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q. V.; Chi, E. H.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.