Truth-Maintained Memory Agent: Proactive Quality Control for Reliable Long-Context Dialogue

Om Phadke
University of California, Los Angeles
omphadke2005@g.ucla.edu

Jeffrey Guo
Western Washington University
guoj3@wwu.edu

Nolan Koch
Slippery Rock University
kochnolan376@gmail.com

Andy Xu
University of Maryland
andyxu@terpmail.umd.edu

Shanduojiao Jiang
Stanford University
sj99@stanford.edu

Kevin Zhu
Algoverse AI Research
kevin@algoverseairesearch.org

Abstract

Large Language Models (LLMs) are prone to false memory formation during long, multi-turn interactions, incorporating incorrect, irrelevant, or contradictory information. Traditional methods such as enlarging context windows, summarizing memory, or selective retrieval, are often computationally expensive and reactive, which allows errors to accumulate. We propose the *Truth-Maintained Memory Agent (TMMA)*, a proactive multi-agent framework that enforces write-time quality control. In the TMMA system, input context undergoes token-gating, complexity evaluation, and truth-verification through a four-tier hierarchical system consisting of Working Memory, Summarized Memory, Archival Memory, and a Flagged Bin for contested content. This structure balances context specificity with long-term retention, reduces the accumulation of noise, and preserves the coherence of the LLM more efficiently than simply expanding the context. Our research indicates that TMMA significantly reduces the incidence of false memories and enhances response quality on existing benchmarks. It offers a pathway to scalable and reliable long-context management in LLMs.

1 Introduction

Large language models (LLMs) are increasingly deployed in applications that require sustained dialogue, such as personal assistants, research copilots, and customer support systems. In these longer settings, a recurring challenge is **false memory formation**. Once an incorrect detail is stored, an agent may retrieve and reuse it as fact, leading to contradictions, unstable reasoning, and cascading reliability issues over the course of interaction. While multi-agent pipelines show promise for complex, long-context tasks [Zhang et al., 2024], they can often struggle with error propagation as inaccuracies from one agent cascade through the system. Similarly, while LLMs exhibit memory-like behavior, Schrödinger's Memory [Wang and Li, 2024] provides evidence of false memory formation where models absorb contradictory or irrelevant details. Work on context management has focused primarily on efficiency. For example, methods like Compressing Context [Li et al., 2023] and sparse-attention architectures like Longformer [Beltagy et al., 2020] and BigBird [Zaheer et al., 2020] improve efficiency by pruning tokens or expanding context windows. However, these methods generally lack safeguards for truthfulness or provenance. Taken together, these approaches preserve

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

or retrieve information first and filter later, leaving systems vulnerable to persistent errors once they enter memory.

We propose the **Truth-Maintained Memory Agent (TMMA)**, a proactive architecture that enforces quality control at write time. TMMA combines a five-stage pipeline—Planner, Context Filter, Truth Verifier, Memory Curator, and Responder—with a hierarchical four-tier memory (Working, Summarized, Archival, and Flagged). Candidate content is evaluated for evidential support, contradiction risk, and utility before storage, while retrieval privileges credibility-weighted records over superficially similar but unverified spans. This study is guided by three questions: **(i)** Can proactive write-time control reduce false memory formation in LLMs? **(ii)** How should memory be structured to balance recency, reliability, and scalability? **(iii)** Can stress tests expose weaknesses overlooked by standard benchmarks? To address these, we pair TMMA's design with dual evaluation: established dialogue benchmarks for conversational quality and controlled injection tests for memory resilience.

Our contributions include **(1)** a multi-agent architecture that integrates write-time screening with hierarchical storage, **(2)** methods for reducing error propagation in long contexts, including token-level filtering, structured truth signals, and credibility-aware retrieval, and **(3)** an evaluation framework that unifies dialogue benchmarks with stress tests designed to probe resilience against false memories.

2 Methodology

2.1 Problem Setup

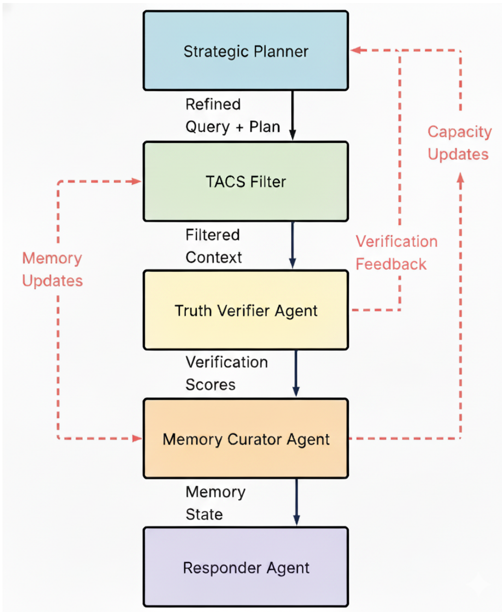
Modern dialogue agents need to handle long conversations, sometimes stretching across dozens or even hundreds of turns. The challenge is not just responding well in the moment but also keeping track of what has been said in a way that remains accurate and useful as the discussion evolves. Many systems today take the simple route of storing everything first and only checking for quality later when information is retrieved. The problem is that this lets errors and misleading details slip into memory unchecked, which can snowball into contradictions, confusion, and eventually a breakdown in coherence.

Our work takes a different approach. Instead of checking memory only when it is retrieved, we enforce quality control at write time, making sure that only reliable, well-supported, and useful information is stored long-term. By shifting quality assurance to the point of ingestion, we reduce error cascades and improve the stability of long conversations. To do this, we introduce a pipeline that combines structured verification, clear curation policies, and a layered memory design.

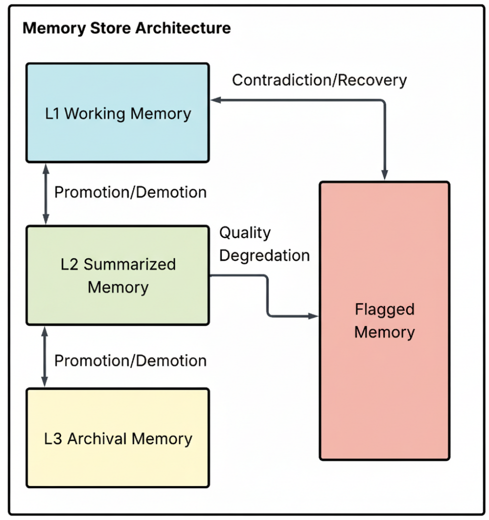
2.2 Architecture

TMMA pairs a five-stage control stack with a typed, hierarchical memory (Figure 1). The **Planner** manages retrieval and verification budgets, and when confidence is low it can trigger retrieve $\rightarrow$ verify $\rightarrow$ refine loops. The **Typed Adaptive Context Selector (TACS)** filters out distractors using lexical and embedding similarity, recency, and learned utility, producing a compact and focused context window. The **Verifier** evaluates candidate units (snippets, tuples, summaries) by assigning a truth score $s_{\text{truth}} \in [0, 1]$, contradiction flags, evidentiality measures, and a calibration label. Its interface is model-agnostic and can be implemented with rules, language models, or hybrid approaches. The **Curator** then applies explicit policies for admission, consolidation, promotion or demotion, and quarantine, with optional input from micro-agents such as redundancy or contradiction checkers. Finally, the **Responder** produces outputs strictly from curated memory, using deterministic templates for factual lookups and grounded generation for more complex queries.

The memory hierarchy is organized into four distinct tiers, each serving a complementary role in balancing recency, reliability, and long-term retention. **L1** Working acts as a short-term buffer for recent turns and momentary notes, enabling rapid access at the cost of relaxed admission criteria and faster expiry. **L2** Summarized holds normalized entities, abstractive summaries, and canonical tuples, compressing volatile details into stable and reusable representations. **L3** Archival functions as the system's authoritative repository, admitting only high-confidence records with evidential support. Finally, **FLAGGED** serves as a quarantine zone where contradicted or uncertain content is isolated but preserved for transparency, future audits, or potential reinstatement. Each memory record is stored as an immutable, typed object with metadata. The storage layer maintains a primary dictionary keyed by ID with tier-aware indices to support efficient operations across L1, L2, L3, and FLAGGED.



(a) **Module boundaries and evidence:** Planner → TACS → Verifier → Curator → Responder; dotted feedback edges for budget updates. Edge labels enumerate evidence at each handoff.



(b) **Typed memory lifecycle.** L1↔L2↔L3 transitions plus a **FLAGGED** quarantine path; glyphs indicate triggers (utility, stability, contradictions, provenance changes).

Figure 1: Architecture at a glance. Panel (a) summarizes responsibilities and evidence flow; Panel (b) shows tier structure and movement criteria.

Retrieval is handled by an adaptive retriever that ranks candidates using a composite relevance function combining tier signals (e.g., boosts for L1/L2), verification status, aggregated confidence across dimensions, and lightweight semantic similarity. The retriever then returns the top candidates to the downstream agents. Maintenance routines enforce tier limits, perform soft deletions, and record all tier movements, detections, and decision rationales. These logs guarantee reproducibility, enable end-to-end audits, and provide transparent system-grade accountability for memory evolution over time.

2.3 Write-Time Control and Memory Management

Building on the architectural design, the core responsibility of the Curator is to decide, at the moment of commitment, whether and where a record should reside in the memory hierarchy. To accomplish this, each record is evaluated against a confidence function that integrates multiple signals: truth score, evidentiality, recency, utility, and source credibility. These dimensions ensure that the decision is not based on a single heuristic, but instead reflects both factual reliability and practical usefulness. Records are then routed by different thresholds: those with confidence greater than 0.9 are promoted to L3 Archival, those above 0.8 but below 0.9 are placed in L2 Summarized, and all others default to L1 Working. Items that fail trust checks by the false-memory gate are moved to the FLAGGED tier with suppressed confidence, preserving them for transparency without allowing them to influence active reasoning.

To maintain bounded memory footprints, each tier enforces strict capacity constraints (L1 = 100, L2 = 500, L3 = 1000, FLAGGED = 200). When a tier reaches its limit, the system utilizes a Least Recently Used (LRU) eviction strategy, ensuring that stale and low-value records are retired first while recently accessed or high-utility content is preserved. Every eviction is accompanied by a rationale and a set of cross-links to keep it auditable, preventing silent data loss. Contradictions discovered by the Verifier or through background sweeps trigger a structured arbitration process. Rather than relying on a single module's judgment, multiple micro-agents vote on the outcome, and the decision is formalized through a *weighted retention score* which ensures that well-supported records are preserved while conflicted entries are demoted to FLAGGED and linked to the surviving record. Finally, during

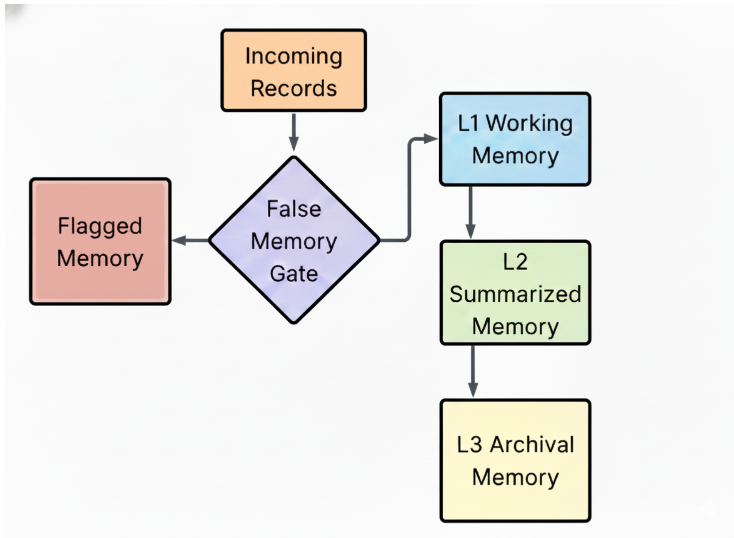


Figure 2: Write-time control. Records are routed to different tiers of the memory hiererarchy based on confidence thresholds.

answer time, the retriever combines similarity-based ranking with credibility signals to guide the Responder toward well-sourced and stable records. Formally, each candidate record is ranked against a query using a composite scoring function described in the Appendix as well. This scoring rule biases retrieval toward content that is both contextually relevant and credibility-weighted. By default, entries in the FLAGGED tier are excluded unless the query explicitly requests conflict resolution.

2.4 False Memory Gate

The routing, capacity protection, and arbitration rules in §2. 3.3 are only as reliable as the records they evaluate. Before any candidate reaches those mechanisms, we pass it through a dedicated **False Memory Gate** that operates prior to commitment and exports normalized risk signals back to the Curator. The goal is to block high-risk content early, reduce contradiction churn, and stabilize long contexts so that §2. 3.3 can apply its thresholds and retention scoring on trustworthy inputs (Figure 3). The gate aggregates evidence from three layers. First, a dictionary checks for known false facts to provide high-precision rejections. Second, pattern detectors identify risky phrasing (e.g., hedging, uncertainty markers, adversarial formulations) that historically correlate with low reliability. These items are not discarded outright, but are routed onward with stricter scrutiny. Third, semantic contradiction checks search for conflicts via embedding-based retrieval and then extract tuples (entities, dates, numbers) to perform direct temporal, logical, and semantic comparisons against active memory and the proposed insert. Layer outputs are combined into a single risk score. Known false matches receive the highest weight (≈ 0.95); pattern-based signals contribute a moderate weight (0.7–0.8); semantic contradictions scale with similarity and the strength of the extracted evidence. Records whose fused score exceeds the risk threshold are moved to FLAGGED with confidence suppressed to 0.05 and a brief rationale. All events log what was detected, why the action was taken, and links to related items so that audits and reversions remain straightforward. The resulting risk score and rationale are consumed by the tier routing of the curator and, when applicable, by the arbitration step in §2. 3.3.

3 Evaluation

3.1 Evaluation Framework

To evaluate the Truth-Maintained Memory Agent (TMMA), we adopt a dual-level framework that balances conventional dialogue performance with the system's novel capacity for truth maintenance. This layered perspective ensures that evaluation captures both the core qualities expected of dialogue systems and the unique contributions TMMA introduces.

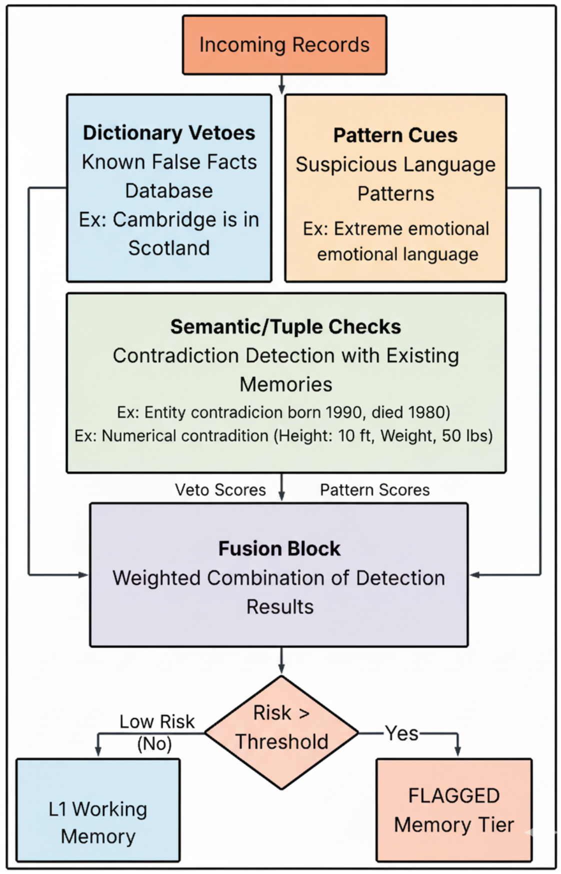


Figure 3: False-memory gate. Three detection layers: dictionary vetoes, pattern cues, and semantic/tuple checks—feed a fusion block; high-risk records are quarantined in **FLAGGED** with suppressed confidence and a short rationale.

Level 1: Dialogue Performance: At the first level, we assess whether TMMA produces fluent, coherent, and contextually appropriate responses across diverse interaction settings. The goal is to confirm that introducing truth maintenance does not come at the expense of conversational quality or task effectiveness.

Level 2: False Memory Prevention: At the second level, we introduce targeted assessments designed to test TMMA's ability to prevent corrupted or contradictory information from entering long-term memory. This layer focuses on stability in extended interactions and highlights the advantages of enforcing write-time quality control.

3.2 Baselines

We first benchmark against widely used open-source and API-based LLMs such as LLaMA-2, Mistral, and GPT-3.5. The second category includes systems that extend standard LLMs with long-context capabilities or retrieval-augmented generation (RAG) mechanisms. These models are designed to improve recall in extended interactions, but they typically rely on permissive storage that admits all content into context or retrieval indices.

3.3 Benchmarks and Metrics

We evaluate TMMA across a set of open-source dialogue benchmarks as well as targeted stress tests for false-memory prevention. This unified view ensures that both conversational quality and memory robustness are measured consistently, with each benchmark paired to metrics that capture its core challenges. For dialogue performance, we adopt three widely used benchmarks. **MultiWOZ 2.4**

[Budzianowski et al., 2019] is a large multi-domain task-oriented corpus covering domains such as hotel, restaurant, taxi, train, and attractions. It contains multi-turn, goal-driven dialogues with detailed annotations, making it well-suited for evaluating multi-domain dialogue management. Metrics used here include response diversity, response relevance, information accuracy, and task understanding. We also use **Schema-Guided Dialogue (SGD)** [Rastogi et al., 2020], which spans a broad set of services and intents, each paired with a schema specifying slots and actions. This makes it particularly suitable for testing generalization across unseen services. Evaluation focuses on BLEU for surface-level response quality, slot extraction F1 for information extraction, semantic similarity for alignment with references, and intent accuracy for correct identification of user goals. **Taskmaster** [Byrne et al., 2019] contains multi-turn conversations across domains such as food ordering, movie tickets, and travel. It mixes human–human and human–assistant styles, providing robustness checks against different conversational patterns. Evaluation uses BLEU, ROUGE, semantic similarity, and slot extraction F1.

Beyond dialogue quality, we introduce stress test variants for each benchmark to directly assess the resilience of TMMA against false memory formation. These stressors insert controlled false facts, temporal contradictions, or mixed true/false contexts into the dialogue flow. Evaluation in this setting employs four targeted metrics: **False Memory Rate (FMR)** measures how often the system incorporates false or fabricated information into its responses. Lower FMR indicates stronger prevention against memory corruption. **Disturbance Adaptation Rate (DAR)** - evaluates how well the system maintains precision in conversations that mix true and false information. Higher DAR demonstrates stronger resilience in unstable or adversarial contexts. **Contradiction Detection Rate (CDR)** measures the precision with which contradictions are identified and quarantined rather than stored as reliable memory. Higher CDR indicates more effective check for write-time consistency.

3.4 Experimental Setup

Model Configuration: TMMA is implemented as a multi-agent pipeline orchestrated using LangGraph. Memory is structured into four tiers with strict capacity limits, as specified above. Confidence thresholds follow an increasing progression across tiers (L1 ≥ 0.7, L2 ≥ 0.8, L3 ≥ 0.9), while items falling below 0.5 or flagged by the False Memory Gate are diverted into the FLAGGED tier. All baseline systems are configured with identical retrieval depth, maximum context length, and inference budgets to ensure fairness. Random seeds are fixed (42) to guarantee reproducibility in sampling, injection placement, and conversation ordering.

Evaluation Protocol: Each model is evaluated on 100 randomly selected conversations per benchmark, yielding 300 conversations per model across MultiWOZ 2.4, SGD, and Taskmaster. Conversations are drawn from the official test sets, with care taken to preserve domain balance and dialogue diversity. For every conversation, models generate responses turn by turn. Metrics are then computed using standardized evaluation libraries, and results are reported as averages with standard deviations across conversations. The protocol follows a four-stage pipeline: (1) load benchmark datasets with train/dev/test splits, (2) run inference on each conversation turn, (3) compute standardized metrics, and (4) report aggregate results. In this study we restrict ourselves to direct metric-based evaluation; ablation studies and statistical tests (e.g., significance testing) are deferred to future work, where they will be considered as part of limitations and model improvement.

False Memory Injection Protocol: To evaluate memory robustness, we construct stress-test variants of each benchmark using a controlled false-memory injection system (Figure 4). Exactly one false memory is introduced into each conversation, for a total of 300 injections per model. Injections are drawn from a curated database of 400 validated false facts, evenly distributed across eight injection types: direct false statements, implicit hallucinations, explicit contradictions, temporal inconsistencies, contextual distortions, semantic paraphrases, numerical manipulations, and causal distortions. Injection timing is randomized to a single user turn between turns 2–8 (excluding the first turn), ensuring that context has been established before perturbation occurs. Injected content is adapted to the active domain and phrased to fit natural dialogue flow, preventing reliance on superficial cues for detection. Each injection is logged with full provenance, including injection type, source fact, and placement, enabling detailed traceability of system behavior.

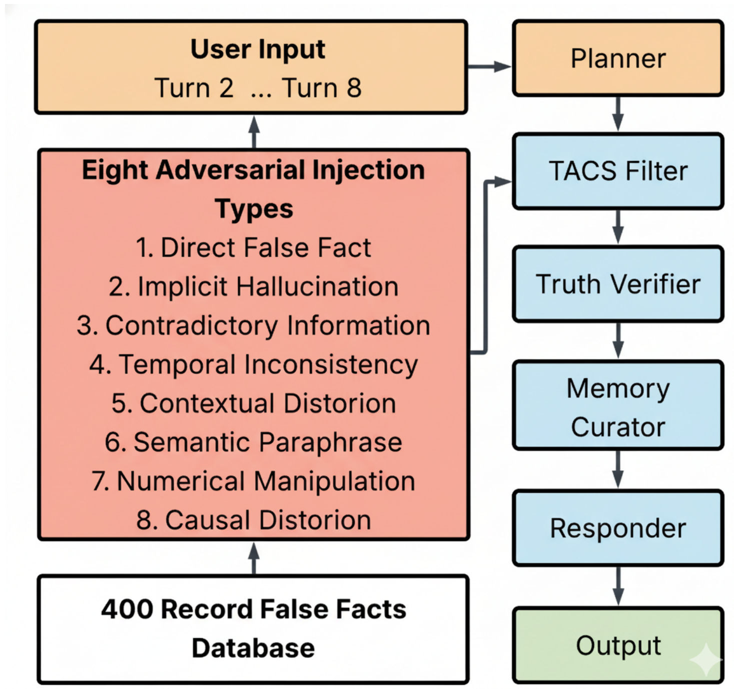


Figure 4: Dynamic false-memory injection framework. User turns are intercepted at randomized points (turns 2–8), and one of eight validated injection types is introduced from a 400-item database. The modified dialogue then flows through the TMMA pipeline, which either quarantines, corrects, or propagates the injected content.

Contradiction Handling Assessment and Reproducibility: Beyond injecting fabricated content, we also test whether systems can resist incorporating contradictory information into long-term memory. Contradiction tests include temporal inconsistencies (changing event times mid-dialogue), logical conflicts (introducing mutually exclusive statements), and semantic contradictions (offering conflicting interpretations of entities or actions). These cases probe whether models propagate the contradiction, suppress it, or correctly quarantine it in memory. The Verifier identifies contradictions and routes to the FLAGGED tier, preventing them from polluting archival memory. All models are evaluated under identical conditions, with random seeds fixed for sampling and injection, balanced coverage across injection types, and uniform distribution of injection points. Memory usage and conversation logs are recorded for every run. Each false memory injection is pre-validated for clear falsehood, contextually adapted to the dialogue domain, and spot-checked during evaluation to ensure quality.

4 Results & Discussion

4.1 Results

We report combined results showing both dialogue performance and false-memory metrics for each benchmark. All entries are computed on 100 conversations per benchmark per model.

4.2 Discussion

We organize the discussion around the research questions motivating this study and interpret the findings in relation to TMMA's core design goals: preserving dialogue quality, reducing false memory formation, and enabling proactive memory control in long-context agents.

Does TMMA preserve dialogue quality? Across all three benchmarks, TMMA sustains or exceeds the dialogue quality of strong baselines, indicating that truth maintenance can coexist with fluent,

Table 1: Results on **MultiWOZ 2.4**. Dialogue metrics: Div. (response diversity), Rel. (relevance), Acc. (information accuracy), Und. (task understanding). False-memory metrics: FMR (False Memory Rate, lower is better), DAR (Disturbance Adaptation Rate, higher is better), CDR (Contradiction Detection Rate, higher is better).

Model	**Div.**↑	**Rel.**↑	**Acc.**↑	**Und.**↑	**FMR**↓	**DAR**↑	**CDR**↑
TMMA	7.26	38.9%	78.2%	94.5%	6.8%	82.4%	41.2%
GPT-3.5 Turbo	2.31	11.5%	37.6%	62.2%	62.3%	20.2%	8.9%
LLaMA-2	1.91	14.8%	25.3%	51.7%	69.8%	13.5%	5.1%
Mistral 7B	1.53	9.5%	17.4%	42.4%	71.1%	9.3%	3.7%
Simple RAG	1.71	7.2%	14.2%	38.3%	81.5%	6.8%	1.6%
Embedded RAG	3.56	23.4%	32.7%	71.2%	52.3%	29.3%	13.5%

Table 2: Results on **Schema-Guided Dialogue (SGD)**. Dialogue metrics: BLEU (surface-level response quality), F1 (slot extraction F1), Sem. (semantic similarity), Int. (intent accuracy). False-memory metrics: FMR (False Memory Rate, lower is better), DAR (Disturbance Adaptation Rate, higher is better), CDR (Contradiction Detection Rate, higher is better).

Model	**BLEU**↑	**F1**↑	**Sem.**↑	**Int.**↑	**FMR**↓	**DAR**↑	**CDR**↑
TMMA	5.75	0.654	38.8%	55.5%	10.5%	73.3%	34.2%
GPT-3.5 Turbo	1.60	0.185	14.3%	27.8%	65.4%	32.7%	12.5%
LLaMA-2	2.70	0.191	8.3%	11.2%	71.4%	27.4%	10.8%
Mistral 7B	1.90	0.214	7.3%	14.4%	76.3%	22.2%	8.4%
Simple RAG	1.12	0.223	12.1%	12.2%	84.3%	14.9%	2.4%
Embedded RAG	4.38	0.490	17.4%	31.4%	55.7%	42.6%	18.8%

contextually coherent language generation. On MultiWOZ 2.4 (Table 1), TMMA achieves the highest diversity, relevance, accuracy, and task understanding scores, outperforming both closed- and open-weight baselines, with Embedded RAG emerging as the closest competitor. On Schema-Guided Dialogue (Table 2), TMMA leads by significant margins in BLEU, slot filling, semantic similarity, and intent accuracy. These improvements are especially pronounced in semantic similarity and intent grounding, which measure the agent's ability to maintain consistency across long, multi-turn interactions. Results on Taskmaster (Table 3) reinforce these observations, showing that TMMA's responses remain fluent and well-grounded even as conversations expand in length and complexity.

These results are consistent with TMMA's overall design rather than to any single component. The architecture's combination of tiered consolidation, structured verification, and credibility-weighted retrieval stabilizes context without constraining generative flexibility. Components such as the Typed Adaptive Context Selector (TACS), Verifier, and Curator jointly promote this stability by filtering distractors, structuring truth signals, and regulating information promotion across tiers. While their individual effects cannot yet be disentangled, the aggregate pattern suggests that proactive control of memory formation enhances contextual precision while preserving conversational naturalness. Future ablation studies will clarify how each element contributes to these gains.

Does TMMA reduce false memory formation? We now address how TMMA performs on the primary purpose of this system, which is to reduce the formation of false memories in long-form dialogues and multi-turn conversations. Controlled injection experiments (Section 4) reveal a clear gap between TMMA and baseline systems. On MultiWOZ 2.4 (Table 1), most baselines record false memory rates (FMR) above 50%, while TMMA reduces FMR to 6.75% and simultaneously achieves higher disturbance adaptation and contradiction detection. Comparable trends hold on SGD (Table 2) and Taskmaster (Table 3), where TMMA maintains low FMR (10.50% and 7.60%, respectively) alongside strong adaptation and detection scores. These consistent patterns across structurally distinct

Table 3: Results on **Taskmaster**. Dialogue metrics: BLEU (surface-level response quality), RG (ROUGE), Sem. (semantic similarity), F1 (slot extraction F1). False-memory metrics: FMR (False Memory Rate, lower is better), DAR (Disturbance Adaptation Rate, higher is better), CDR (Contradiction Detection Rate, higher is better).

Model	BLEU↑	RG↑	Sem.↑	F1↑	FMR↓	DAR↑	CDR↑
TMMA	6.50	6.41	22.9%	0.727	7.6%	77.5%	39.0%
GPT-3.5 Turbo	1.25	2.70	17.1%	0.203	59.4%	37.5%	14.7%
LLaMA-2	2.84	1.69	7.4%	0.164	63.2%	31.9%	9.4%
Mistral 7B	1.90	1.11	6.6%	0.097	68.4%	10.4%	3.2%
Simple RAG	1.78	2.65	14.0%	0.137	78.3%	9.9%	2.0%
Embedded RAG	3.91	5.84	15.4%	0.485	46.2%	48.4%	21.6%

datasets demonstrate that the proposed framework generalizes well to varied dialogue domains and interaction styles.

Taken as a whole, these outcomes align with TMMA's proactive control objective. During write-time, uncertain or self-contradictory content is routed to the FLAGGED tier, which quarantines it from active reasoning while preserving it for auditability. During retrieval, the exclusion of flagged records prevents the reinforcement of low-credibility information, breaking the rehearsal loops that typically amplify falsehoods over time. This combination of write-time filtration and retrieval-time containment correlates with a substantial reduction in false memory propagation. Without dedicated ablations, we interpret these findings as evidence that TMMA's integrated design supports robust containment and correction of misinformation during extended dialogue.

Interpreting robustness and design implications: Collectively, the results suggest that TMMA addresses a central limitation of long-context dialogue systems: the absence of explicit mechanisms for memory filtration. By sythensizing verification, consolidation, and credibility-weighted retrieval, the framework enforces factual integrity without degrading generative capability. The improvements observed across MultiWOZ, Schema-Guided Dialogue, and Taskmaster indicate that this principle generalizes across task-oriented and mixed-domain dialogue corpora. At the same time, the lack of component-level ablations constrains our ability to draw causal conclusions about specific modules. We therefore view these findings as system-level evidence that proactive memory design is a promising direction for achieving both reliability and coherence in long-context agents.

A notable outcome is that TMMA's write-time curation mitigates the compounding effects of early-stage hallucinations. By restricting what enters long-term memory, the system effectively narrows the error surface exposed during retrieval, resulting in lower downstream distortion. This mechanism also has implications for interpretability: tier transitions, credibility scores, and arbitration logs create an auditable trail of how each record evolves over time. Such transparency could serve as a foundation for human-centered trust and debugging in future conversational AI systems.

5 Conclusion

We introduced the Truth Maintained Memory Agent (TMMA), a proactive architecture that enforces quality control at the point of memory formation. By combining our multi-agentic pipeline with a typed, multi-tier memory, TMMA ensures that dialogue reasoning proceeds over curated rather than fragile context.

Across three dialogue benchmarks, TMMA matched or exceeded strong baselines on measures of fluency, accuracy, and task competence, while substantially reducing falsememory formation under controlled injections. These gains reflect two key design choices: quarantining suspicious content before it enters longterm memory, and excluding flagged items during retrieval. Together, these mechanisms interrupt the rehearsal loops that typically allow falsehoods to persist and compound.

Beyond these empirical results, our contribution is conceptual. We argue that longcontext dialogue agents should treat memory hygiene as a core design principle rather than an afterthought. Structured truth signals, tierspecific policies, and credibilityaware retrieval offer a general framework for building systems that are both capable and reliable in extended interactions. TMMA provides not only a concrete implementation, but also a foundation for future exploration, ranging from adaptive thresholds and lightweight verification, to integration with human centered evaluation, toward conversational agents that are robust, efficient, and aligned with user trust in realworld settings.

References

I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer, 2020. URL `https://arxiv.org/abs/2004.05150`. Preprint; Longformer GitHub: `https://github.com/allenai/longformer`.

P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Riedl, S. Ultes, O. Ramadan, and M. Gasic. Multiwoz 2.4: A multi-domain task-oriented dialogue dataset, 2019. URL `https://github.com/budzianowski/multiwoz`. Multi-domain task-oriented dialogue dataset used for evaluation.

B. Byrne, K. Krishnamoorthy, R. Sarikaya, R. Goel, P. Bennett, K. Chung, D. Fohr, J. Lopes, A. Papangelis, S. Riedl, et al. Taskmaster-1: Toward a realistic and diverse dialog dataset, 2019. URL `https://github.com/google-research-datasets/Taskmaster`. Google's task-oriented dialogue dataset.

Y. Li, B. Dong, C. Lin, and F. Guerin. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. URL `https://arxiv.org/pdf/2310.06201.pdf`. arXiv:2310.06201.

A. Rastogi, X. Zang, S. Sunkara, R. Gupta, and P. Khaitan. Schema-guided dialogue dataset. In *Proceedings of the 8th Dialog System Technology Challenge (DSTC8)*. Association for Computational Linguistics, 2020. URL `https://github.com/google-research-datasets/dstc8-schema-guided-dialogue`. Google's schema-guided dialogue dataset for evaluation.

W. Wang and Q. Li. Schrödinger's memory: Large language models, 2024. URL `https://arxiv.org/abs/2409.10482`.

M. Zaheer, G. Guruganesh, A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, et al. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, 2020. URL `https://arxiv.org/abs/2007.14062`. Sparse-attention transformer for long sequences.

Y. Zhang, R. Sun, Y. Chen, T. Pfister, R. Zhang, and S. Ö. Arik. Chain of agents: Large language models collaborating on long-context tasks, 2024. URL `https://arxiv.org/abs/2406.02818`.

A Technical Appendices and Supplementary Material

Technical appendices with additional results, figures, graphs and proofs may be submitted with the paper submission before the full submission deadline (see above), or as a separate PDF in the ZIP file below before the supplementary material deadline. There is no page limit for the technical appendices.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper briefly discusses potential positive applications of the work performed in the abstract. The nature of the work performed (improving LLM memory and response quality) does not have negative societal impacts and are therefore not mentioned.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [No]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.