Investigating Tool-Memory Conflicts in Tool-Augmented LLMs

Anonymous Authors[1]

Abstract

Tool-augmented large language models (LLMs) have powered many applications. However, they are likely to suffer from knowledge conflict. In this paper, we propose a new type of knowledge conflict – Tool-Memory Conflict (TMC), where the internal parametric knowledge contradicts with the external tool knowledge for tool-augmented LLMs. We find that existing LLMs, though powerful, suffer from TMC, especially on STEM-related tasks. We also uncover that under different conditions, tool knowledge and parametric knowledge may be prioritized differently. We then evaluate existing conflict resolving techniques, including prompting-based and RAG-based methods. Results show that none of these approaches can effectively resolve tool-memory conflicts.

1. Introduction

Tool-augmented large language models (LLMs) are LLMs that can use external tools, such as function calling and APIs. By integrating external tools, the LLMs exhibit enhanced problem-solving capabilities, especially in applications that require interact with physical world and access knowledge bases. This augmentation extends their functional scope, enabling interaction with dynamic and domain-specific information sources.

Despite the advantages, the integration of external tools introduces potential epistemic inconsistencies, as tool-generated outputs may contradict the parametric knowledge encoded within the parameters of LLMs. For instance, Temporal Discrepancy arises when external tools provide updated information that conflicts with the static, pre-trained knowledge of an LLM to a specific cutoff date. Additionally, mechanistic differences between internal LLM memory and external tools complicate the resolution of such inconsistencies, raising critical concerns regarding knowledge reliability and coherence.

While prior research has examined several types of knowledge conflicts in LLMs, including context-memory conflict and inter-context conflict, limited attention has been given to discrepancies arising between parametric memory and external tool outputs. This study introduces the concept of tool-memory conflict, a novel category of knowledge inconsistency in LLMs, wherein the internal parametric knowledge of the model diverges from the outputs of external tools when addressing the same query. Specifically, we aim to answer the following research questions:

- RQ1: Under what conditions (task, scale of LLM) do tool-memory conflicts appear in tool-augmented LLMs?
- RQ2: When confronted with a tool-memory conflict, do LLMs prioritize parametric knowledge or tool-generated outputs?
- RQ3: What methodologies can effectively reconcile tool-memory conflicts across varying contexts?

By addressing these questions, this study seeks to advance the understanding of knowledge integration in tool-augmented LLMs and develop strategies for mitigating epistemic inconsistencies, ultimately enhancing the reliability and interpretability of tool-augmented language models.

Through experiments and analysis, we find that LLMs can have significant amount of tool-memory conflicts across a variety of tasks, including math, QA. Although LLMs are trained to incorporate external tools during tool-augmentation to complement their memory, especially on professional or time-sensitive tasks, they can still prioritize internal memory over tools.

The contributions of this paper are

- formulating tool-memory conflict, a new type of knowledge conflict in LLMs, and discussing its importance, causes, and differences to existing knowledge conflicts;
- investigating how LLMs prioritize knowledge under tool-memory conflict, and the bias of LLMs;

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

- evaluating a wide range of methods to resolve tool-memory conflict.

2. Related work

Knowledge Conflicts in LLMs Existing works focus on three types of knowledge conflicts (Xu et al., 2024), 1) context-memory conflict, where the output of internal parametric knowledge differs from the contextual knowledge (Longpre et al., 2021; Tan et al., 2024; Gekhman et al., 2023; Wang et al., 2024), 2) inter-context conflict, where conflicts happens between various pieces of contextual information (Zhang & Choi, 2023; Chen et al., 2022; Wan et al., 2024; Jin et al., 2024), and 3) inter-memory conflict, where conflicts happen internally in the parametric knowledge of LLMs (Lee et al., 2022b; Wang et al., 2023; Qi et al., 2023; Hase et al., 2023).

Tool-Augmented LLMs Tool-Augmented LLMs are LLMs that know how to call external tools to answer questions the their internal memory is insufficient at Schick et al. (2023). These LLMs are usually explicitly trained using tool using demonstrations that are synthesized with seed samples and LLMs (Patil et al., 2023; Li et al., 2023; Tang et al., 2023). modifying existing datasets (Basu et al., 2024), and dataset development with GPT-4 (Qin et al., 2024). TAML (Parisi et al., 2022) used self-play to boost LLMs' performance on math and reasoning tasks.

3. Tool-Memory Conflict

Problem Definition Let f denote a tool-augmented LLM capable of calling external tools from tool set $\mathcal{T}$. The *tool-memory conflict* (TMC) occurs if the parametric knowledge of f conflicts with the external tool knowledge of $\mathcal{T}$, more formally

$$f(q) \neq f(q; \mathcal{T}), \tag{1}$$

where q is a query.

3.1. Examples of tool-memory Conflict

There exist numerous scenarios where conflicts may arise between the internal memory of LLMs and external tools.

Example 1: Mathematics Problem When posed with a mathematical question, an LLM may respond using its generative capabilities by recalling patterns from its training data. Alternatively, the model can employ external computational tools, such as a calculator, to ensure precise numerical accuracy. The choice between these approaches can lead to discrepancies, particularly when the model's internal heuristics produce an answer that deviates from the exact computation provided by an external tool.

Example 2: Factual Data Retrieval An LLM responding to fact-based queries may generate responses based on its internalized knowledge, which is constrained by the temporal limitations of its training data. Conversely, it may retrieve real-time information from external databases or search engines. Discrepancies emerge when the internally stored knowledge contradicts newly updated facts, leading to potential inconsistencies in responses.

Example 3: Code Execution For programming-related inquiries, an LLM may either generate code snippets based on its learned distribution of syntax and semantics or execute the code using an external interpreter. If the generated code contains errors or outdated syntax, whereas the executed code produces a different, verifiable output, conflicts may arise in determining which response is most reliable.

Example 4: Medical Diagnosis In medical applications, an LLM may generate responses based on training data comprising past medical literature and case studies. Alternatively, it may leverage external medical databases containing the latest research findings, treatment protocols, and clinical guidelines. Conflicts may arise when a model's outdated medical knowledge contradicts contemporary best practices or real-time patient data.

In each of these examples, the underlying issue revolves around the reconciliation of an LLM's static memory with its dynamic ability to engage external tools. Understanding and addressing these conflicts is crucial to ensuring reliability, accuracy, and transparency in AI-assisted decision-making systems.

3.2. Causes of Tool-Memory Conflict

The core of tool-memory conflict stems from a discrepancy between tool-based and parametric knowledge. We identify several causes of tool-memory conflicts.

Temporal Information Mismatch Discrepancies occur when tools provide outdated, inconsistent, or delayed information compared to the LLM's internal knowledge. Since LLMs rely on a combination of learned and real-time tool-based data, mismatches can create confusion when attempting to synthesize responses (Jang et al., 2022).

Misinformation by Tools Errors arise when tools return incorrect, biased, or misleading data, leading to conflicts with the model's stored knowledge. These issues can stem from tool limitations, external data inaccuracies, or adversarial manipulation of information sources, making it difficult for the LLM to discern reliable information.

Incorrect Tool Usage Although an LLM may recognize the need for external tools, it can misuse them by selecting

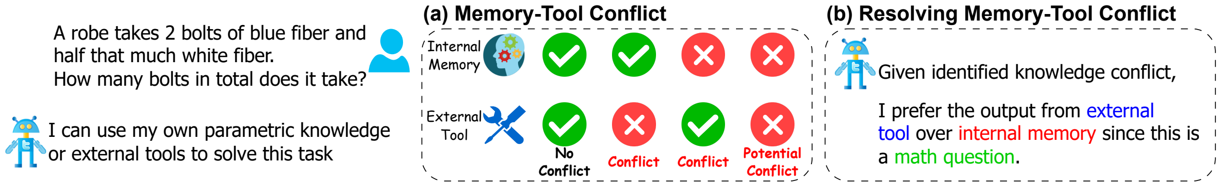
A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take?
I can use my own parametric knowledge or external tools to solve this task
(a) Memory-Tool Conflict
Internal Memory
External Tool
No Conflict
Conflict
Conflict
Potential Conflict
(b) Resolving Memory-Tool Conflict
Given identified knowledge conflict,
I prefer the output from external tool over internal memory since this is a math question.

Figure 1. Illustration of tool-memory Conflict (MCT). ✓and ✗mean the output is correct or wrong respectively compared to the ground truth answer.

an inappropriate tool, misinterpreting output, or passing incorrect arguments. This can result in incomplete or erroneous task execution, especially when tool documentation is ambiguous or when the model's interpretation of tool parameters is flawed.

3.3. Difference to Existing Knowledge Conflicts in LLMs

Though look similar, tool-memory conflict is different from existing knowledge conflicts, especially context-memory conflict (Lee et al., 2022a; Zhou et al., 2023; Tan et al., 2024) and inter-context conflict (Zhang & Choi, 2021; Kasai et al., 2023; Wan et al., 2024).

Differences in Knowledge Representation Contextual knowledge, such as in-context examples and retrieved documents from RAG, is processed as input tokens. The LLM treats it like any prompt, embedding it into the current context and influencing token-by-token generation. However, it is constrained by the context window and fades unless explicitly reintroduced.

In contrast, tool knowledge, such as API calls or external computations, is acquired on demand. The model does not ”know” the result until execution. Tool outputs are typically appended or referenced post-execution rather than embedded in the token sequence. The model does not ”remember” these results unless explicitly stored or re-fed into the prompt. Thus, while contextual knowledge is integrated into the model's generation sequence, tool knowledge operates externally.

Information Flow and Processing Pipeline Retrieved contextual knowledge is fed into the transformer architecture before token generation, shaping the model's latent representation. Tool knowledge, however, is injected mid-process—retrieved results are incorporated after initial processing, often as separate entities. Unlike contextual knowledge, tool outputs bypass the attention mechanism unless explicitly reintroduced.

Epistemological and Practical Implications Contextual knowledge follows a retrieval-and-reasoning paradigm, allowing reinterpretation across prompts. Tool knowledge, however, is authoritative—once executed (e.g., querying a database), the model does not reassess multiple sources.

Time-Sensitivity and Dynamism Contextual knowledge is static at retrieval, only updating when a new query is issued. Tool knowledge, however, is dynamic and real-time, fetching the latest data upon each call (e.g., live weather updates or stock prices).

Error Handling For contextual knowledge, the model can analyze inconsistencies and adjust responses. With tool knowledge, however, the model must accept outputs as-is, reinforcing the idea that tool knowledge functions as an externalized truth source, distinct from contextual knowledge that the model internalizes and processes differently.

3.4. Importance

Understanding the knowledge conflict between LLM memory and external tools is essential from various aspects.

Identifying LLM Limitations LLMs are constrained by training data limitations, outdated knowledge, and biases. Analyzing conflicts helps pinpoint deficiencies, guiding improvements in model design, dataset curation, and fine-tuning.

Assessing Tool Reliability External tools vary in accuracy and trustworthiness. Conflicts with LLM responses highlight potential misinformation, enabling better fact-checking, source prioritization, and AI-assisted decision-making.

3.5. Extracting Tool-Memory Conflict

Eliciting Tool-Memory Conflict For a given query, we prompt the LLM twice, restricting the LLM to only use its internal parametric knowledge (memory) and to only use external tools respectively. To make sure the LLMs are indeed only using memory or tools as we desired, we add keywords in the prompt such as “only using your internal memory / external tools”, an approach similar to Xie et al. (2024); Wang et al. (2024). Additionally, we ask the LLMs to document the process of using tools, including what and how tools are used. We exclude all responses where LLMs fail to follow these instructions of solely using tools or memory.

Identifying Bias of Tool-Augmented LLMs Given both output based on internal memory and external tools, we prompt the LLM to resolve the conflict by itself, shown in Figure 1 (b). This can reveal the bias and tendency of LLMs towards specific type of knowledge on different tasks. Following Wu et al. (2024), we define memory bias and tool bias as

- Memory Bias = $Pr[\texttt{LLM}(q|t) = 0|\texttt{LLM}(q|t;T) = 1, \texttt{LLM}(q|t) = 0]$ measures the probability the model uses the memory while the tool-based output is correct.

- Tool Bias = $Pr[\texttt{LLM}(q|t;T) = 0|\texttt{LLM}(q|t;T) = 0, \texttt{LLM}(q|t) = 1]$ measures the probability the model prefers tool-based output while the memory is correct.

Resolving conflicts We also evaluate whether existing knowledge resolving methods can resolve tool-memory conflict. Opinion-based prompting (Zhou et al., 2023) reformulates the input query as opinion-based prompting, which demonstrates strong performance to alleviate inter-context conflicts. Vigilant prompting (Pan et al., 2023) instructs the LLM to beware of potential misinformation, which is shown to significantly alleviate misinformation and inter-context conflicts. We also examine if incorporating additional sources of information using Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) can alleviate knowledge conflict.

4. Experimental Setup

Datasets We evaluate TMC on the following widely-used benchmarks: MMLU (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), MATH-500 (Lightman et al., 2023), AIME 2024 (aim), GPQA Diamond (Rein et al., 2024). These benchmarks cover a diverse range of tasks, including STEM, humanities & social science, and long-tail world knowledge.

LLMs We conduct the experiments across a wide range of LLMs which are capable of calling external tools, 1) GPT-4o (Hurst et al., 2024), 2) DeepSeek-V3 (Liu et al., 2024), 3) LLaMA-3 (3.3 70B, 3.1 8B) (Grattafiori et al., 2024), 4) QWen-2.5 (72B) (Yang et al., 2024), 5) QwQ (32B) (qwq), 6) Groq-LLaMA-3 8B (gro), 7) Watt (8B) (wat).

5. Results

5.1. Under What Conditions Do Tool-Memory Conflict Occur?

Prevalence of Tool–Memory Conflict Across Models and Tasks Our comprehensive evaluation across seven state-of-the-art LLMs and a broad spectrum of tasks reveals that Tool–Memory Conflict (TMC) is pervasive. On average, 49.8% of all test instances exhibit a discrepancy between the model's internally generated answer and the result obtained from an external tool. Specifically, as shown in Table 1, GPT-4o demonstrates a TMC rate of 14.1%, DeepSeek-v3 records 15.3%, LLAMA-3.3 70B yields 15.5%, QWen-2.5 72B registers 26.9%, QwQ exhibits a pronounced conflict rate of 75.4%, Groq-LLAMA-3 8B shows 83.2%, and Watt 8B records 48.6%. These figures indicate that even the most advanced LLMs frequently produce outputs that clash with external tool responses. The high conflict rates underscore a fundamental challenge in integrating retrieval or computation tools: the model's internal knowledge often diverges from—and sometimes directly contradicts—the external source.

Effect of Model Scale on Conflict Frequency We observe a clear correlation between model size and TMC incidence. In particular, 70B parameters appear to mark a threshold above which models exhibit substantially lower conflict rates. For example, LLAMA-3.3 70B (15.5%) and GPT-4o (14.1%) both outperform their smaller counterparts, Groq-LLAMA-3 8B (83.2%) and Watt 8B (48.6%), by wide margins. This suggests that larger models possess richer internal representations, enabling better alignment with external tool outputs. Below the 70B parameter range, LLMs rely more heavily on memorized approximations and heuristics, which are vulnerable to divergence when precision is required or when the tool's result contradicts the learned distribution.

Domain-Specific Sensitivities to Tool–Memory Conflict Figure 2 illustrates how TMC varies across different domains. Domains such as *Math* and *STEM* exhibit the highest conflict probability, indicating that tasks requiring precise numerical computation or specialized technical knowledge are particularly susceptible to discrepancies. In contrast, *Humanities & Social Sciences* and *Other* tasks show much lower conflict rates, suggesting that, for interpretive or open-ended tasks, the model's internal reasoning often remains consistent with external cues or that the task does not demand exactitude. *Long-Tail Knowledge* tasks—questions about obscure or infrequently discussed entities—also show moderate TMC levels, likely reflecting mismatches between the model's outdated or incomplete training data and the tool's up-to-date information.

Table 1. Proportion of tool-memory conflict (TMC) across all evaluated tasks of LLMs.

Model	No Conflict			Conflict			
	Total	Both=1	Both=0	Total	Tool=1	Mem=1	Both=0
GPT-4o	83.6	74.2	9.4	14.1	7.2	2.7	4.2
DeepSeek-v3	80.5	72.7	7.8	15.3	6.2	3.3	5.8
LLAMA-3.3 70B	84.5	72.2	12.3	15.5	9.1	3.0	3.4
QWen-2.5 72B	73.1	62.3	10.8	26.9	8.3	11.6	7.0
QwQ	24.6	21.1	3.4	75.4	5.6	33.4	36.4
Groq-LLAMA-3 8B	16.8	10.7	6.1	83.2	1.2	38.1	44.0
Watt 8B	51.4	34.3	17.1	48.6	4.8	18.9	24.9

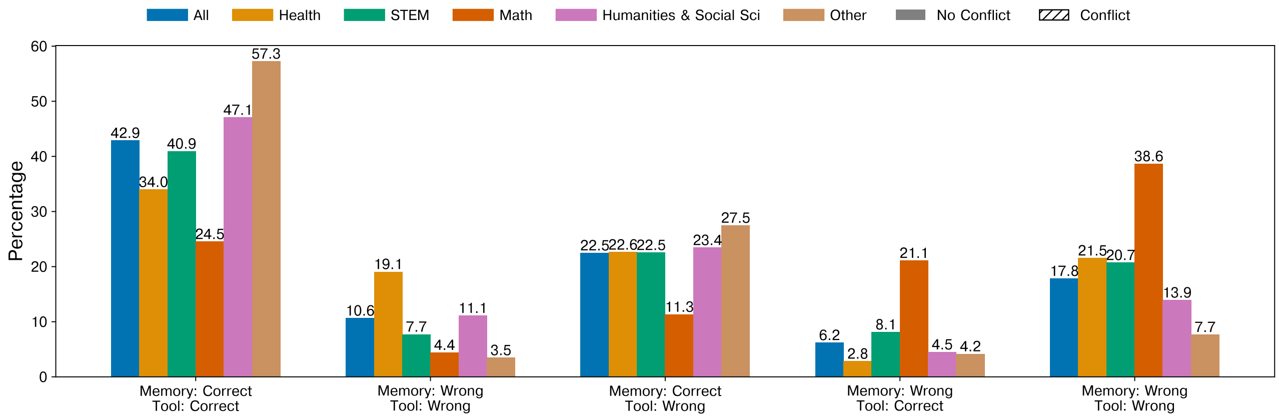

Figure 2. Tool-Memory Conflict across different domains.

Impact of Conflict on Model Confidence and Accuracy Whenever a conflict arises, the model must reconcile competing signals: its internal knowledge and the tool's output. We measure the downstream effect on task performance by comparing accuracy under "No Conflict" versus "Conflict" conditions. In *Math* tasks, accuracy drops by an average of 4.5 absolute points when a tool–memory conflict occurs, indicating that numerical discrepancies severely undermine the model's confidence and ability to produce correct solutions. For *STEM* and *Health* tasks, we observe a smaller but still notable decline, reflecting the dependence on precise or domain-specific knowledge that external tools often supply. By contrast, *Humanities & Social Sciences* tasks see only a 0.5 difference between "No Conflict" and "Conflict" conditions, suggesting that models can often override or reinterpret minor contradictions when dealing with more subjective or contextual content. Even for *Long-Tail Knowledge*, the accuracy gap remains modest, implying that models and tools occasionally share similar erroneous assumptions, thereby muting the overall effect of conflict.

Task Type Analysis: Fine-Grained View of Conflict Figure 3 provides a breakdown of TMC by task type (e.g., arithmetic, algebra, logical reasoning, multi-hop retrieval, etc.). *Arithmetic* and *Algorithmic* tasks exhibit the highest conflict frequencies (often exceeding 70–80%), underscoring that deterministic, step-by-step computations are prone to mismatch when LLMs rely on learned heuristics rather than exact algorithms. In *Multi-Hop Retrieval* tasks—where the model must chain multiple facts—TMC rates hover around 50–60%, indicating that each retrieval step compounds the chance of divergence. *Fact Verification* tasks, which require confirming or refuting a given statement, show lower conflict rates, implying that model predictions and tool retrievals tend to align more often. For *Common-Sense Reasoning* tasks, conflict rates fall in the 30–40% range, suggesting that while external knowledge occasionally contradicts the model's intuition, the impact is less severe than in strictly quantitative tasks.

Conflict Cases Where Both Memory and Tool Are Incorrect A nontrivial fraction of conflict instances occur when both the model's internal response and the external tool's output are incorrect but disagree with each other. These "Both=0" cases (Table 1, last column under "Conflict") illustrate scenarios in which neither internal memory nor the external tool holds the correct answer. For example, on QwQ, 36.4% of all test cases fall into this category; for

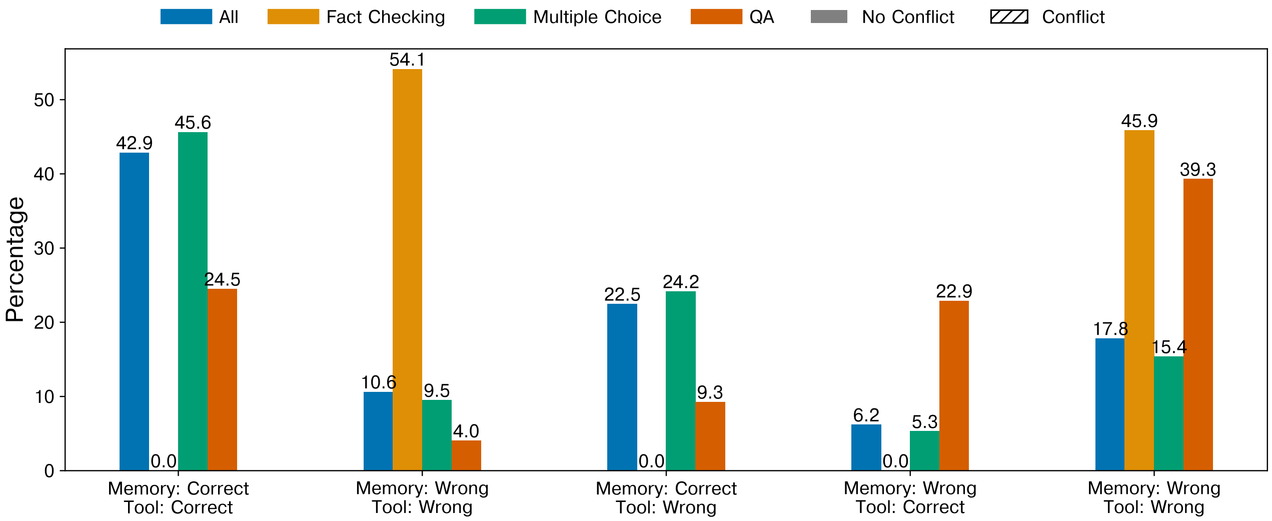

Figure 3. Tool-Memory Conflict across different types of tasks.

Groq-LLAMA-3 8B, 44.0% are “Both=0.” Such phenomena highlight that simply combining two erroneous sources does not guarantee improved performance, and that conflict detection alone does not resolve fundamental knowledge gaps.

Mathematical Tasks Exacerbate Tool–Memory Conflicts TMC is markedly more pronounced for strictly mathematical tasks. Table 1 shows that, for arithmetic problems, the majority of models exhibit conflict rates well above 70%. This is attributable to the inherently precise nature of mathematics: external calculators or symbolic solvers provide exact results, whereas LLMs often produce approximate answers based on statistical patterns. Consequently, whenever an LLM’s heuristic estimate deviates by even a small margin, the conflict is detected. Moreover, the discrepancy is magnified in multi-step problems (e.g., solving systems of equations), where compounding rounding errors or misapplied rules lead to large divergences from the tool’s output.

Tool–Memory Conflict Exposes Limitations of Tool-Augmented LLMs One of the primary motivations for augmenting LLMs with external tools is to compensate for gaps in the model’s internal knowledge—particularly for long-tail or recently emerged entities. However, our analysis in the *Long-Tail World Knowledge* domain reveals that, even when a tool retrieves the correct answer, the LLM’s internal memory often contains similar inaccuracies. As a result, conflict does not always correspond to a scenario where the tool corrects the model; rather, both sources can share the same outdated or erroneous information. In Figure 2, the modest difference between “Tool=1” and “Mem=1” under Long-Tail conditions indicates that neither memory nor tool consistently provides a clear advantage. This underscores that tool-augmentation alone cannot rectify deep-seated knowledge deficiencies without also addressing the quality of the underlying retrieval mechanism and ensuring that the external knowledge base is regularly updated.

Domain-Aware Conflict Mitigation Strategies Given the varying sensitivity to TMC across domains, we propose that conflict mitigation should be domain-specific. For *Math* and *STEM* tasks, integrating high-precision computational modules (e.g., symbolic solvers, exact arithmetic engines) directly into the LLM’s inference pipeline can drastically reduce discrepancies. In *Humanities & Social Sciences*, where conflicts are rare and often inconsequential, a simple “favor memory” or “favor tool” heuristic may suffice. For *Long-Tail Knowledge*, a hybrid retrieval strategy that cross-verifies multiple external sources before presenting an answer could minimize the propagation of outdated information. Ultimately, these domain-aware approaches seek to reconcile internal representations with external data in a more principled manner, rather than relying on a one-size-fits-all fallback.

These insights pave the way for future work aimed at harmonizing internal model knowledge with external tool outputs, thereby enhancing the reliability and accuracy of tool-augmented language systems.

5.2. Are LLMs Biased Towards Tools or Memory?

Tool Bias Tool bias is defined here as the fraction of outputs in which the model overly relies on an external tool (for instance, invoking or favoring API-driven lookup routines) instead of integrating information directly. LLAMA-3.3

Table 2. Tool Bias and Memory Bias of LLMs.

Model	Tool Bias	Memory Bias
GPT-4o	41.7	41.9
DeepSeek-v3	39.2	41.3
LLAMA-3.3 70B	44.3	40.2
QWen-2.5 72B	35.8	37.3
QwQ	0.1	24.5
Groq-LLAMA-3 8B	0.2	16.4
Watt 8B	0.0	51.4

70B exhibits the highest tool bias at 44.3%, suggesting that in nearly half of its responses, it defaults to leveraging the attached tool rather than grounding its output solely in internal reasoning. GPT-4o and DeepSeek-v3 have intermediate tool biases of 41.7% and 39.2%, respectively. QWen-2.5 72B manifests a lower degree of tool dependence (35.8%). The three lower-accuracy models—QwQ (0.1%), Groq-LLAMA-3 8B (0.2%), and Watt 8B (0.0%)—demonstrate virtually no tool bias, indicating that they almost never defer to external tools and instead rely exclusively on internal parameters (even though this may come at the cost of accuracy). See Table 2 for details.

Memory Bias Memory bias represents the probability of and LLM prioritizing where the model disproportionately leverages cached internal knowledge (e.g., memorized facts, pretrained tokens) over dynamic reasoning or tool-assisted querying. GPT-4o and DeepSeek-v3 report memory biases of 41.9% and 41.3%, respectively, closely paralleling their tool biases. LLAMA-3.3 70B incurs a slightly lower memory bias (40.2%), indicating that its reliance on internal memory is marginally less pronounced than its inclination toward tool usage. QWen-2.5 72B's memory bias (37.3%) is also lower, reflecting a more balanced distribution between tool usage and memory recall. Conversely, QwQ and Groq-LLAMA-3 8B have memory biases of 24.5% and 16.4%, respectively, which—when combined with their negligible tool bias—suggests that these smaller models tend to produce outputs rooted almost entirely in internal knowledge, albeit with far lower overall correctness. Watt 8B exhibits a memory bias of 51.4%, which exactly matches its overall accuracy score; this suggests that whenever Watt 8B answers correctly, it does so purely via internal recall, never invoking external tools. See Table 2 for details.

Comparisons across LLMs The two highest-performing models (LLAMA-3.3 70B and GPT-4o) achieve accuracy rates above 83%, but they differ subtly in how they allocate "cognitive" effort between external tool calls and internal memory. LLAMA-3.3 70B slightly favors the tool (44.3% tool bias vs. 40.2% memory bias), whereas GPT-4o displays nearly equal reliance on tools (41.7%) and memory (41.9%), reflecting a more balanced hybrid strategy.

Table 3. Resolving Tool-Memory Conflicts

Model	Conflict	Vig Prompt	Op Prompt	RAG
GPT-4o	14.1	13.8	14.7	11.5
DeepSeek-v3	15.3	14.6	15.3	12.2
LLAMA-3.3 70B	15.5	14.4	16.3	13.6
QWen-2.5 72B	26.9	22.6	25.1	17.9
QwQ	75.4	71.7	69.3	55.7
Groq-LLAMA-3 8B	83.2	81.5	82.2	74.3
Watt 8B	48.6	44.7	46.1	39.1

DeepSeek-v3 (80.5% accuracy) also exhibits a nearly balanced tool/memory split (39.2% vs. 41.3%), suggesting that its architectural design equally privileges pretrained knowledge and external lookups to achieve high performance. In contrast, QWen-2.5 72B, despite having a 72-billion-parameter backbone, achieves only moderate accuracy (73.1%) and shows a moderately lower dependence on both tools and memory, implying potential architectural or training differences that reduce over-reliance on either resource.

Smaller models (QwQ, Groq-LLAMA-3 8B, Watt 8B) uniformly demonstrate low tool bias (less than 1.0%), indicating they are effectively "tool-agnostic." Their primary—and sometimes sole—decision driver is memorized knowledge, as evidenced by their nonzero memory biases. However, the trade-off is clear: these models achieve sub-optimal accuracy (ranging from 16.8% to 51.4%). Watt 8B is the only low-capacity model that manages to achieve accuracies above 50%, but it does so entirely through memorized content, with no tool assistance.

The close correspondence between accuracy and memory bias in Watt 8B, along with the negligible tool bias, illustrates a scenario in which model capacity is sufficient to store a limited subset of facts (achieving correct responses on roughly half of the dataset) but insufficient to generalize beyond static memorization or effectively integrate external tools. On the other hand, models such as LLAMA-3.3 70B and GPT-4o achieve higher task coverage by intelligently deciding when to call upon external tools versus internal representations, thereby balancing recall and dynamic retrieval.

5.3. Can Tool-Memory Conflicts Be Resolved?

Table 3 provides a quantitative comparison of seven large language models (LLMs) on their propensity to exhibit "tool-memory conflicts" under different mitigation strategies. In each row, the first column lists a particular LLM, ordered roughly from highest-capacity (GPT-4o, LLAMA-3.3 70B) to more compact configurations (Groq-LLAMA-3 8B, Watt 8B). The subsequent four columns report the measured conflict rate—expressed as a percentage—under four distinct configurations:

- Conflict: This is the raw conflict probability when the model is allowed to choose freely between relying on its internal memorized knowledge and invoking an external tool.
- Vigilant prompting: In this case, specialized "vigilance" wording in the prompt is prepended to each query, explicitly instructing the model to detect and avoid contradictory signals between its memory and the tool's output.
- Opinion-based prompting: Opinion-based prompting reformulates the output of LLMs as someone's opinion.
- RAG: Additional informaiton is retrieved from external knowledge source and incorporated to answer the query. The model is thus expected to defer to retrieved facts, thereby reducing contradictory reliance on stale internal parameters.

Impact of prompt engineering Across all architectures, appending a vigilance prompt consistently reduces conflict by approximately 1–4 percentage points relative to the baseline. This suggests that explicit meta-instructions—e.g., "If your internal knowledge conflicts with the tool's information, defer to the tool"—do encourage better alignment. However, the magnitude of improvement is modest for the smallest models, which likely lack the representational capacity to fully internalize the meta-instruction.

For opinion-based prompting, the conflict probability that either match or slightly exceed the baseline—particularly in LLAMA-3.3 70B and QWen-2.5 72B—implying that not every prompt-based or fine-tuning approach uniformly aids consistency. Without explicit documentation, one might tentatively infer that the third intervention either over-restricts the model (e.g., too aggressive a filter) or fails to provide enough context to override entrenched internal biases.

Effectiveness of RAG : Incorporating a retrieval stage proves to be the most effective single intervention: on average, RAG reduces conflict rates by 2–6 percentage points in high-capacity models and by 8–15 percentage points in mid- and low-capacity models. This indicates that augmenting model "context" with up-to-date, externally retrieved evidence not only enriches factual accuracy but also resolves contradictions between stale memorization and the most current tool output. In other words, RAG effectively "grounds" the model's predictions, irrespective of its size.

Scaling effects on model consistency The baseline conflict rates correlate inversely with model size. Larger models (GPT-4o, LLAMA-3.3 70B, DeepSeek-v3) demonstrate stronger internal coherence between memorized knowledge and the outputs of any connected tool. By contrast, smaller models often encode more fragmented or less-robust knowledge representations, leading to pronounced tool-memory mismatch.

These findings suggest that more principled approaches, potentially involving model fine-tuning or architectural adjustments, may be necessary to robustly resolve TMC without sacrificing performance or coherence.

6. Conclusion

We propose a new type of knowledge conflicts for LLMs – Tool-Memory Conflict (TMC), where external tools contradicts the internal parametric knowledge of LLMs. Through experiments on diverse datasets, we find that existing LLMs, including powerful proprietary models, suffer from TMC, especially in STEM tasks. Under different conditions, LLMs may be biased towards internal memory or external tools. We evaluate existing tools of resolving conflicts, where prompting-based methods have limited contribution. Incorporating additional external knowledge, such as RAG, may help alleviating the conflicts.

Limitations and Future Work We did not experiment with all LLMs, such as Claude. Future work can extend this analysis to a broader range of LLMs. In addition, many conflicting resolving techniques remain not evaluated. We plan to develop novel mechanisms to resolve tool-memory conflicts.

Impact Statement

Our work investigates the trustworthiness of integrating external tools into internal parametric knowledge of tool-augmented Large Language Models (LLMs). We focus on identifying a key issue, where the interal memory and external tools conflict with each other. Tool-memory conflict undermines model reliability in several ways: (1) it erodes user trust, since end-users cannot anticipate whether the LLM will defer to its memorized parameters (which may be outdated or imprecise) or to real-time tool outputs (which may also be noisy or incomplete); (2) it introduces inconsistency in downstream applications—ranging from automated fact-checking to decision-support systems—where conflicting signals can lead to erroneous or unsafe conclusions; and (3) it complicates model evaluation and calibration, since standard accuracy metrics fail to capture the nuanced interplay between static knowledge and dynamic retrieval. By systematically quantifying and characterizing the prevalence of tool–memory contradictions, our work illuminates how even state-of-the-art LLMs can exhibit unpredictable behavior when faced with conflicting evidence. In doing so, we highlight the necessity of developing robust conflict-resolution mechanisms (e.g., vigilance prompting, retrieval-augmented grounding, and calibrated confidence scoring) that can reconcile or at least surface divergent sources of information. Addressing this gap is essential not only for improving the factual correctness of LLM responses, but also for ensuring transparent, explainable, and trustworthy deployment in high-stakes domains such as healthcare, finance, and legal assistance.

References

URL https://huggingface.co/datasets/HuggingFaceH4/aime_2024.

URL https://groq.com/introducing-llama-3-groq-tool-use-models/.

URL https://qwenlm.github.io/blog/qwq-32b/.

URL https://huggingface.co/watt-ai/watt-tool-8B.

Basu, K., Abdelaziz, I., Chaudhury, S., Dan, S., Crouse, M., Munawar, A., Austel, V., Kumaravel, S., Muthusamy, V., Kapanipathi, P., and Lastras, L. API-BLEND: A comprehensive corpora for training and benchmarking API LLMs. 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)*, pp. 12859–12870, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.694. URL https://aclanthology.org/2024.acl-long.694/.

Chen, H.-T., Zhang, M., and Choi, E. Rich knowledge sources bring complex knowledge conflicts: Recalibrating models to reflect conflicting evidence. In Goldberg, Y., Kozareva, Z., and Zhang, Y. (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 2292–2307, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.146. URL https://aclanthology.org/2022.emnlp-main.146/.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Gekhman, Z., Herzig, J., Aharoni, R., Elkind, C., and Szpektor, I. TrueTeacher: Learning factual consistency evaluation with large language models. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2053–2070, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.127. URL https://aclanthology.org/2023.emnlp-main.127/.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Hase, P., Diab, M., Celikyilmaz, A., Li, X., Kozareva, Z., Stoyanov, V., Bansal, M., and Iyer, S. Methods for measuring, updating, and visualizing factual beliefs in language models. In Vlachos, A. and Augenstein, I. (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2714–2731, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.199. URL https://aclanthology.org/2023.eacl-main.199/.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Hurst, A., Lerer, A., Goucher, A. P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Jang, J., Ye, S., Lee, C., Yang, S., Shin, J., Han, J., Kim, G., and Seo, M. TemporalWiki: A lifelong benchmark for training and evaluating ever-evolving language models. In Goldberg, Y., Kozareva, Z., and Zhang, Y. (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6237–6250, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.418. URL https://aclanthology.org/2022.emnlp-main.418/.

Jin, Z., Cao, P., Chen, Y., Liu, K., Jiang, X., Xu, J., Qiuxia, L., and Zhao, J. Tug-of-war between knowledge: Exploring and resolving knowledge conflicts in retrieval-augmented language models. In Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., and Xue, N. (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 16867–16878, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.1466/.

Kasai, J., Sakaguchi, K., yoichi takahashi, Bras, R. L., Asai, A., Yu, X. V., Radev, D., Smith, N. A., Choi, Y., and Inui, K. Realtime QA: What's the answer right now? In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=HfKOIPCvsv.

Lee, K., Han, W., Hwang, S.-w., Lee, H., Park, J., and Lee, S.-W. Plug-and-play adaptation for continuously-updated QA. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 438–447, Dublin, Ireland, May 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.37. URL https://aclanthology.org/2022.findings-acl.37/.

Lee, N., Ping, W., Xu, P., Patwary, M., Fung, P., Shoeybi, M., and Catanzaro, B. Factuality enhanced language models for open-ended text generation. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022b. URL https://openreview.net/forum?id=LvyJX20Rll.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

Li, M., Zhao, Y., Yu, B., Song, F., Li, H., Yu, H., Li, Z., Huang, F., and Li, Y. API-bank: A comprehensive benchmark for tool-augmented LLMs. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3102–3116, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.187. URL https://aclanthology.org/2023.emnlp-main.187/.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Longpre, S., Perisetla, K., Chen, A., Ramesh, N., DuBois, C., and Singh, S. Entity-based knowledge conflicts in question answering. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t. (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7052–7063, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.565. URL https://aclanthology.org/2021.emnlp-main.565/.

Pan, Y., Pan, L., Chen, W., Nakov, P., Kan, M.-Y., and Wang, W. On the risk of misinformation pollution with large language models. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 1389–1403, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.97. URL https://aclanthology.org/2023.findings-emnlp.97/.

Parisi, A., Zhao, Y., and Fiedel, N. Talm: Tool augmented language models. *arXiv preprint arXiv:2205.12255*, 2022.

Patil, S. G., Zhang, T., Wang, X., and Gonzalez, J. E. Gorilla: Large language model connected with massive apis. *arXiv preprint arXiv:2305.15334*, 2023.

Qi, J., Fernández, R., and Bisazza, A. Cross-lingual consistency of factual knowledge in multilingual language models. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10650–10666, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.658. URL https://aclanthology.org/2023.emnlp-main.658/.

Qin, Y., Liang, S., Ye, Y., Zhu, K., Yan, L., Lu, Y., Lin, Y., Cong, X., Tang, X., Qian, B., Zhao, S., Hong, L., Tian, R., Xie, R., Zhou, J., Gerstein, M., dahai li, Liu, Z., and Sun, M. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=dHng2O0Jjr.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Schick, T., Dwivedi-Yu, J., Dessi, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer: Language models can teach themselves to use tools. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Yacmpz84TH.

Tan, H., Sun, F., Yang, W., Wang, Y., Cao, Q., and Cheng, X. Blinded by generated contexts: How language models merge generated and retrieved contexts when knowledge conflicts? 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)*, pp. 6207–6227, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.337. URL https://aclanthology.org/2024.acl-long.337/.

Tang, Q., Deng, Z., Lin, H., Han, X., Liang, Q., Cao, B., and Sun, L. Toolalpaca: Generalized tool learning for language models with 3000 simulated cases. *arXiv preprint arXiv:2306.05301*, 2023.

Wan, A., Wallace, E., and Klein, D. What evidence do language models find convincing? 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)*, pp. 7468–7484, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.403. URL https://aclanthology.org/2024.acl-long.403/.

Wang, F., Mo, W., Wang, Y., Zhou, W., and Chen, M. A causal view of entity bias in (large) language models. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 15173–15184, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.1013. URL https://aclanthology.org/2023.findings-emnlp.1013/.

Wang, Y., Feng, S., Wang, H., Shi, W., Balachandran, V., He, T., and Tsvetkov, Y. Resolving knowledge conflicts in large language models. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=ptvV5HGTNN.

Wu, K., Wu, E., and Zou, J. Clasheval: Quantifying the tug-of-war between an LLM's internal prior and external evidence. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=WGoCZl2itU.

Xie, J., Zhang, K., Chen, J., Lou, R., and Su, Y. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=auKAUJZMO6.

Xu, R., Qi, Z., Guo, Z., Wang, C., Wang, H., Zhang, Y., and Xu, W. Knowledge conflicts for LLMs: A survey. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8541–8565, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.486. URL https://aclanthology.org/2024.emnlp-main.486/.

Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Zhang, M. and Choi, E. SituatedQA: Incorporating extra-linguistic contexts into QA. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t. (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7371–7387, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.586. URL https://aclanthology.org/2021.emnlp-main.586/.

Zhang, M. and Choi, E. Mitigating temporal misalignment by discarding outdated facts. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14213–14226, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.879. URL https://aclanthology.org/2023.emnlp-main.879/.

Zhou, W., Zhang, S., Poon, H., and Chen, M. Context-faithful prompting for large language models. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14544–14556, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.968. URL https://aclanthology.org/2023.findings-emnlp.968/.

A. You *can* have an appendix here.