Two-Level Learning-Augmented Caching for Long-Context LLM Inference: A Beyond Worst-Case Analysis

Rany Stephan
Institute for Computational and Mathematical Engineering, Stanford University
ranycs@stanford.edu

Abstract

Large Language Models (LLMs) with extended context windows face a critical challenge: recent empirical studies show accuracy degradation in multi-turn conversations compared to single-turn settings, even within the model's context limit. We formalize this *long-chat degradation* as a hierarchical online caching problem and introduce LATEM (Learning-Augmented Two-level Marking), a learning-augmented algorithm that manages both message-level and token-level caches. We prove that LATEM achieves consistency ratio $2 + O(\sqrt{\eta/\mathrm{OPT}})$ when predictions have error η, and $O(\frac{\log k_m + \log k'_t}{\varepsilon})$-robustness in the worst case, where k_m is the message cache size, $k'_t = \lfloor k_t/k_m \rfloor$ is the effective token cache per message, and ε is the trust parameter. Our analysis extends the learning-augmented framework to hierarchical caches with interdependent eviction decisions, offering, to our knowledge, the first formal consistency–robustness guarantees for hierarchical KV-cache management in LLMs.

1 Introduction

The rapid advancement of Large Language Models (LLMs) has led to systems capable of processing increasingly long contexts. Recent models support context windows ranging from 128K to 1M tokens, enabling sophisticated applications such as multi-document reasoning and extended conversational interactions. However, practitioners have observed a puzzling phenomenon: model performance degrades significantly as conversations grow longer, even when the total token count remains well within the advertised context limit.

Recent empirical studies have quantified this degradation. Laban et al. [2025] demonstrated that LLMs exhibit significantly lower performance in multi-turn conversations than single-turn settings, with an average drop of 39% across six generation tasks. This degradation cannot be attributed solely to absolute context length, as the same models perform well on single long documents of comparable length. The root cause lies in how LLMs manage conversational memory during inference, where models must maintain key-value (KV) pairs for attention computation across all previous tokens.

In multi-turn conversations, this creates a two-level hierarchy: messages (user queries and model responses) contain multiple tokens, and the relevance of tokens is inherently tied to the relevance of their parent messages. When memory pressure forces the system to discard information, poor eviction decisions at either level can cascade into significant performance degradation.

This naturally leads us to formulate the problem as a hierarchical online caching challenge. Traditional online algorithms provide worst-case guarantees but ignore the inherent predictability in conversational patterns. Attention scores provide natural predictions about which messages and to-

Preprint. Under review.

kens will be important for future responses, motivating us to apply the learning-augmented algorithms framework.

Our Contributions. We attempt to make several contributions:

First, we formalize long-chat retention as a two-level weighted paging problem with strict dependencies: tokens can only be cached if their parent message is cached.

Second, we design LATEM, extending the classical Marking algorithm to hierarchical caches with predictions, maintaining separate marking phases and prediction quotas for each cache level.

Third, we prove that LATEM achieves consistency ratio $2 + O(\sqrt{\eta/\mathrm{OPT}})$ when predictions have error η, approaching $2 \cdot \mathrm{OPT} + O(\sqrt{\eta \cdot \mathrm{OPT}})$ absolute cost as prediction quality improves. We also establish $O(\frac{\log k_m + \log k'_t}{\varepsilon})$-robustness in the worst case, where $k'_t = \lfloor k_t/k_m \rfloor$ is the effective token cache size per message. Our analysis adapts phase-quota techniques to dependent caches, a setting not covered in prior work.

Finally, we establish matching lower bounds showing our competitive ratios are optimal up to constant factors. We prove that any algorithm achieving consistency ratio $(2 + \delta)$ for small $\delta > 0$ must have robustness at least $\Omega(\log k_m + \log k'_t)$, demonstrating that LATEM achieves an optimal trade-off between consistency and robustness for this problem.

2 Related Work

Classical competitive analysis. The foundations of online caching date back to the competitive-ratio framework of Sleator and Tarjan [1985], who proved that any deterministic paging policy is at least k-competitive and that LRU meets this bound. Randomisation broke the linear barrier: the *Marking* algorithm of Fiat et al. [2002] and its refinement by Karlin et al. [1994] achieve an $H_k = \Theta(\log k)$ ratio against an oblivious adversary by partitioning requests into phases and evicting only unmarked pages. These ideas underpin virtually every later paging algorithm, including the learning-augmented variants discussed below.

Hierarchical and multilevel caches. Beyond a single cache, theoreticians have explored models in which memory consists of several levels with increasing capacity and access cost. Aggarwal et al. [1987] introduced a hierarchy model, and Chrobak and Noga [2000] analysed the relaxed list-update formulation, showing that move-to-front (equivalently, independent LRU at each level) is $\Theta(k)$-competitive in the worst case. These results provide a baseline for our two-level setting but do not offer prediction-aware refinements.

Learning-augmented algorithms. A modern line of work augments online decision-making with machine-learned advice. Purohit et al. [2018] formalised *consistency* and *robustness* for problems such as ski-rental and scheduling, proving algorithms that interpolate between near-optimality under accurate predictions and classic competitive guarantees when predictions fail. Focusing on paging, Lykouris and Vassilvitskii [2020] introduced PREDICTIVE MARKER, which uses next-use predictions but caps the number of advice-driven evictions per phase to preserve an $O(\log k)$ worst-case ratio while attaining $2 + O(\sqrt{\eta/\mathrm{OPT}})$ consistency. Follow-up papers simplified and sharpened these bounds: Rohatgi [2020] obtained near-optimal trade-offs via a potential-function argument, and Wei and Zhang [2020] gave a black-box combiner that mixes an oracle follower with LRU.

Bridging to two-level caches. Our work extends the learning-augmented framework to a hierarchical cache with strict coherence constraints. While Rohatgi [2020] and Wei and Zhang [2020] hint that phase-based quotas generalise to "structured" state spaces, we are not aware of earlier algorithms that address the strict parent–child coherence constraint. Moreover, existing competitive results for multilevel caching do not exploit predictions and thus remain linear in cache size.

LLM KV-cache management. Token-level heuristics such as Heavy-Hitter Oracle (H2O) [Zhang et al., 2023] and SAGE-KV [Wang et al., 2025], and structural tricks like attention sinks in StreamLLM [Xiao et al., 2024] dynamically prune keys and values based on attention statistics, but they provide no adversarial guarantees. Our algorithm LATEM is complementary: it can ingest those

same attention-based scores as predictions while providing consistency–robustness guarantees for hierarchical KV eviction in long-context inference.

In summary, LATEM combines the phase-quota design of predictive paging with insights from multilevel caching, matching single-level bounds up to a logarithmic factor while suggesting paths toward practical LLM workloads.

3 Model and Preliminaries

3.1 Two-Level Cache Structure

We consider a system managing conversations consisting of messages and their constituent tokens. Let $\mathcal{M} = \{m_1, m_2, \ldots\}$ denote the universe of possible messages. Each message m_i contains tokens $T(m_i) = \{t_{i,1}, t_{i,2}, \ldots, t_{i,|m_i|}\}$, where tokens are uniquely associated with their parent message.

At any time, the system maintains:

1. A message cache $\mathcal{C}_m \subseteq \mathcal{M}$ with $|\mathcal{C}_m| \leq k_m$
2. A token cache $\mathcal{C}_t$ with $|\mathcal{C}_t| \leq k_t$

Assumption: Throughout our analysis, we assume $k_t > k_m$, ensuring that the effective token allocation per message $k'_t = \lfloor k_t/k_m \rfloor \geq 1$.

Remark 1. *The working assumption $k_t > k_m$ is used only for notational simplicity. All subsequent arguments remain valid when the two capacities are comparable or even identical. Formally, one may set*

$$k'_t \;=\; \max\bigl\{1,\; \bigl\lfloor k_t/k_m \bigr\rfloor\bigr\}$$

and observe that every lemma and competitive-ratio bound carries over unchanged.

This assumption is reasonable in practice for LLM systems where token-level storage significantly exceeds message-level storage requirements.

The crucial *coherence constraint* is:

$$\mathcal{C}_t \subseteq \bigcup_{m \in \mathcal{C}_m} T(m)$$

This ensures tokens can only be cached if their parent message is cached.

3.2 Request Model and Cost Structure

A request sequence $\sigma = r_1, r_2, \ldots, r_n$ consists of requests for either messages or tokens. The cost structure is:

For message request m_j:

- If $m_j \in \mathcal{C}_m$: cost 0 (hit)
- If $m_j \notin \mathcal{C}_m$: cost 1 (miss)

For token request $t_{j,\ell}$:

- If $m_j \notin \mathcal{C}_m$: cost 1 (must fetch entire message)
- If $m_j \in \mathcal{C}_m$ but $t_{j,\ell} \notin \mathcal{C}_t$: cost β where $\beta \in (0, 1)$ (token miss)
- If both cached: cost 0 (hit)

The parameter $\beta \in (0, 1)$ represents the relative cost of fetching a single token versus an entire message, assumed to be a known constant.

3.3 Prediction Model

Following Lykouris and Vassilvitskii [2020], we assume access to predictions $\hat{p}(x)$ of the next request time for each element x. Let $p(x)$ denote the true next request time. The prediction error is defined as the total L_1 error:

$$\eta = \sum_{x \in \sigma} |\hat{p}(x) - p(x)|$$

4 Algorithm Design: Learning-Augmented Two-level Marking

4.1 Algorithm Overview

We extend the classical Marking algorithm to handle two cache levels with dependencies. The key insights arise from the need to carefully balance prediction usage with worst-case guarantees in a hierarchical setting. First, we run separate marking phases at each cache level while maintaining coherence constraints between them. This separation allows us to isolate prediction errors at each level and prevent cascading failures from propagating across the hierarchy.

Second, we use predictions for eviction decisions only up to carefully chosen quotas, after which we fall back to arbitrary evictions that preserve the marking structure. This quota-based approach ensures that even when predictions are adversarially bad, the algorithm maintains the logarithmic competitive ratio of classical marking algorithms.

Third, we maintain separate prediction quotas for each message's token cache to prevent error propagation across different parts of the conversation. This isolation is crucial because poor predictions about one message's token importance should not affect the caching decisions for tokens belonging to other messages.

4.2 Token Cache Allocation

A critical design decision in our algorithm is how to allocate the total token cache capacity k_t among the messages currently stored in the message cache. We adopt a simple but effective equal allocation strategy: each message $m \in \mathcal{C}_m$ gets at most $k'_t = \lfloor k_t / k_m \rfloor$ token slots from the global k_t pool, where k'_t represents its allocated share.

Worst-case impact of idle token slots. When fewer than k_m messages are resident, at most $(k_m - |\mathcal{C}_m|)\, k'_t \leq k_t$ token positions go unused. Consequently, the equal-allocation rule can incur at most one additional token miss per phase. Every competitive bound therefore worsens by no more than an additive $O(\beta k_t)$ term, which is already subsumed by the $O(k_m + \beta k_t)$ constant appearing in Theorem 1.

The equal allocation strategy has several advantages. First, it prevents any single message from monopolizing the token cache, which could lead to poor performance when multiple important messages are active simultaneously. Second, it creates a natural separation of concerns where token-level decisions for different messages are independent, preventing error propagation across message boundaries. Third, the uniform allocation simplifies the competitive analysis by allowing us to treat each message's token cache as an independent caching problem of size k'_t.

4.3 Quota Management

A fundamental challenge in learning-augmented algorithms is determining how much to trust predictions versus falling back to worst-case guarantees. We address this through a quota-based system that limits the number of prediction-based evictions per phase. The message quota is $\tau_m = \lceil \varepsilon k_m \rceil$, where $\varepsilon \in (0, 1)$ is a trust parameter that controls the trade-off between consistency and robustness.

For token-level caching, we maintain separate quotas for each message to prevent error propagation. Each message m gets its own token quota $\tau'_t = \lceil \varepsilon k'_t \rceil$, where k'_t is the token allocation per message. This independence is crucial because poor predictions about one message's tokens should not affect the caching quality for other messages.

Algorithm 1 LATEM: Learning-Augmented Two-level Marking

Parameters: k_m, k_t with $k_t > k_m$; trust parameter $\varepsilon \in (0, 1)$
Initialize: $\mathcal{C}_m \leftarrow \emptyset$, $\mathcal{C}_t \leftarrow \emptyset$, $k'_t \leftarrow \lfloor k_t/k_m \rfloor \geq 1$
Quotas: $\tau_m \leftarrow \lceil \varepsilon k_m \rceil$, $\tau'_t \leftarrow \lceil \varepsilon k'_t \rceil$ for each message
Invariant: $\mathcal{C}_t \subseteq \bigcup_{m \in \mathcal{C}_m} T(m)$ (coherence)
procedure HANDLEMESSAGEREQUEST(m)
    if $m \in \mathcal{C}_m$ **then**
        Mark m if unmarked; **return** 0
    end if
    if $|\mathcal{C}_m| = k_m$ **then**
        if all messages in $\mathcal{C}_m$ are marked **then**
            Start new phase: unmark all; reset pred_count$_m \leftarrow 0$
        end if
        $U_m \leftarrow \{m' \in \mathcal{C}_m : m' \text{ unmarked}\}$
        if pred_count$_m < \tau_m$ **then**
            $m_{\text{evict}} \leftarrow \arg\max_{m' \in U_m} \hat{p}(m')$
            pred_count$_m \leftarrow$ pred_count$_m + 1$
        else
            $m_{\text{evict}} \leftarrow$ arbitrary element from U_m
        end if
        Evict m_{evict} and all its tokens from both caches
    end if
    Insert m into $\mathcal{C}_m$ and mark it; **return** 1
end procedure
procedure HANDLETOKENREQUEST(m, t)
    if $m \notin \mathcal{C}_m$ **then return** HANDLEMESSAGEREQUEST(m)
    end if
    if $t \in \mathcal{C}_t$ **then**
        Mark t in m's token phase; **return** 0
    end if
    $T_m \leftarrow T(m) \cap \mathcal{C}_t$
    if $|T_m| = k'_t$ **then**
        if all tokens in T_m are marked **then**
            Start new token phase for m: unmark all in T_m
            Reset pred_count$'_t[m] \leftarrow 0$
        end if
        $U_t \leftarrow \{t' \in T_m : t' \text{ unmarked}\}$
        if pred_count$'_t[m] < \tau'_t$ **then**
            $t_{\text{evict}} \leftarrow \arg\max_{t' \in U_t} \hat{p}(t')$
            pred_count$'_t[m] \leftarrow$ pred_count$'_t[m] + 1$
        else
            $t_{\text{evict}} \leftarrow$ arbitrary element from U_t
        end if
        Evict t_{evict} from $\mathcal{C}_t$
    end if
    Insert t into $\mathcal{C}_t$ and mark it; **return** β
end procedure

The algorithm maintains the coherence invariant $\mathcal{C}_t \subseteq \bigcup_{m \in \mathcal{C}_m} T(m)$ throughout execution by explicitly evicting all tokens of a message whenever that message is evicted from the message cache (line 20).

5 Theoretical Analysis

We now analyze LATEM's performance, proving bounds on both consistency and robustness. Our analysis extends the techniques of Lykouris and Vassilvitskii [2020] to handle the hierarchical structure and dependencies.

5.1 Spread Function Analysis

We first establish the key technical tool for analyzing prediction errors.

Definition 1 (Spread Function)**.** *For a sequence of items with increasing true next-access times $a_1 < a_2 < \cdots < a_\ell$ but non-increasing predicted times $b_1 \geq b_2 \geq \cdots \geq b_\ell$, the spread function $S(\eta)$ is the maximum length ℓ such that the total prediction error $\sum_{i=1}^{\ell} |a_i - b_i| = \eta$.*

Lemma 1 (Spread Bound)**.** *For absolute loss, $S(\eta) \leq \sqrt{5\eta}$.*

Proof. Consider a sequence of items with true next-access times $a_1 < a_2 < \cdots < a_\ell$ and predicted times $b_1 \geq b_2 \geq \cdots \geq b_\ell$. To find the maximum length ℓ for given error η, we minimize the error for each sequence length. Without loss of generality, let $a_i = i - 1$ for $i = 1, 2, \ldots, \ell$. The error-minimizing prediction choice is $b_i = c$ for all i, where c is the median.

For odd $\ell = 2n + 1$, the optimal choice is $c = n$, giving total error $\eta = n(n + 1) = \frac{\ell^2 - 1}{4}$. Solving for ℓ: $\ell = \sqrt{4\eta + 1} \leq \sqrt{5\eta}$ for $\eta \geq 1$. The complete derivation is provided in Appendix A.1.

Therefore, $S(\eta) \leq \sqrt{5\eta}$. □

5.2 Message-Level Analysis

Lemma 2 (Message Phase Structure)**.** *In any message phase i with $Q_m^{(i)}$ clean messages:*

1. *OPT incurs cost $\geq Q_m^{(i)}/2$*
2. LATEM *incurs exactly $Q_m^{(i)}$ misses for clean messages*
3. *Additional misses from stale messages $\leq \min\{S(\eta_m^{(i)}), \tau_m\}$*

Proof. Parts (1) and (2) follow from standard marking analysis [Fiat et al., 2002].

For part (3): Prediction-based evictions create a sequence where evicted messages have non-increasing predicted next-access times. If these messages return in increasing order of true next-access times, we have a spread-function scenario. By Lemma 1, at most $S(\eta_m^{(i)})$ such premature misses occur. The quota τ_m limits total prediction-based evictions. □

Proposition 1 (Message Cost)**.** *The expected message cost in phase i is:*

$$\mathbb{E}[\textit{cost}_m^{(i)}] \leq Q_m^{(i)} + \min\{S(\eta_m^{(i)}), \tau_m\} + R_m^{(i)} \cdot H_{k_m}$$

where $R_m^{(i)}$ is the number of stale messages handled after quota exhaustion.

Proof. Clean messages: $Q_m^{(i)}$ misses (Lemma 2)
Prediction-based stale misses: $\leq \min\{S(\eta_m^{(i)}), \tau_m\}$ (Lemma 2)
Remaining stale messages: Expected H_{k_m} cost each by standard marking analysis

The total follows by summing these components. □

5.3 Token-Level Analysis

Lemma 3 (Token Independence)**.** *Token phases for different messages are independent. Prediction errors for message m's tokens don't affect eviction decisions for message $m' \neq m$.*

Proof. Each message maintains its own token marking phase, quota τ_t', and eviction counter $\text{pred\_count}_t'[m]$. This independence holds because LATeM pins each message to exactly k_t' token slots. □

Proposition 2 (Token Cost per Message)**.** *For message m in phase i, the expected token cost is:*

$$\mathbb{E}[\textit{cost}_t^{(i,m)}] \leq \beta \cdot [Q_t^{(i,m)} + \min\{S(\eta_t^{(i,m)}), \tau_t'\} + R_t^{(i,m)} \cdot H_{k_t'}]$$

where $Q_t^{(i,m)}$ is clean tokens of m, $\eta_t^{(i,m)}$ is prediction error for m's tokens, and $R_t^{(i,m)}$ is stale tokens handled after quota exhaustion.

Proof. Follows the same logic as Proposition 1, with cost β per token miss and effective cache size k'_t for each message. □

5.4 Main Results

Performance metrics. An online algorithm is said to be α-*consistent* if, under perfect predictions ($\eta = 0$), its cost is at most $\alpha \cdot \mathrm{OPT}$ on every sequence. It is γ-*robust* if, for arbitrary predictions, its cost is at most $\gamma \cdot \mathrm{OPT}$.

Theorem 1 (Performance of LATEM)**.** LATEM *with trust parameter* $\varepsilon \in (0, 1)$ *achieves:*

1. ***Consistency****:* $\mathit{cost}_{\mathrm{LATEM}}(\sigma) \le 2 \cdot \mathit{OPT}(\sigma) + O(\sqrt{\eta \cdot \mathit{OPT}(\sigma)}) + O(k_m + \beta k_t)$
2. ***Robustness****:* $\mathit{cost}_{\mathrm{LATEM}}(\sigma) \le \left(\frac{4H_{k_m}}{\varepsilon} + \frac{4\beta H_{k'_t}}{\varepsilon}\right) \cdot \mathit{OPT}(\sigma)$

where $k'_t = \lfloor k_t/k_m \rfloor$ *and* H_k *denotes the* k*-th harmonic number.*

Proof. **Consistency Analysis:**

When prediction errors are small, $S(\eta_m^{(i)}) \le \tau_m$ and $S(\eta_t^{(i,m)}) \le \tau'_t$ for most phases, making the post-quota terms negligible.

Message-level cost: Summing Proposition 1 over all phases:

$$\mathbb{E}[\mathrm{cost}_m] \le \sum_i Q_m^{(i)} + \sum_i S(\eta_m^{(i)})$$

$$\sum_i S(\eta_m^{(i)}) \le \sum_i \sqrt{5\eta_m^{(i)}} \le \sqrt{5N}\sqrt{\sum_i \eta_m^{(i)}} = \sqrt{5N}\sqrt{\eta_m}$$

where N is the number of message phases. Since each phase contributes at least $1/2$ to OPT_m (Lemma 2), we have $N \le 2\mathrm{OPT}_m$.

Therefore: $\sum_i S(\eta_m^{(i)}) \le \sqrt{5 \cdot 2\mathrm{OPT}_m}\sqrt{\eta_m} = \sqrt{10\mathrm{OPT}_m \cdot \eta_m}$.

From Lemma 2: $\sum_i Q_m^{(i)} \le 2\mathrm{OPT}_m$.

Hence: $\mathbb{E}[\mathrm{cost}_m] \le 2\mathrm{OPT}_m + \sqrt{10\mathrm{OPT}_m \cdot \eta_m}$.

Token-level cost: By the same argument:

$$\mathbb{E}[\mathrm{cost}_t] \le \beta(2\mathrm{OPT}_t + \sqrt{10\mathrm{OPT}_t \cdot \eta_t})$$

Total cost: Define the total weighted prediction error as $\eta := \eta_m + \beta\eta_t$.

To combine the error terms, we apply the Cauchy–Schwarz inequality. Let the vectors

$$\mathbf{u} = \big(\sqrt{\mathrm{OPT}_m},\ \sqrt{\beta\,\mathrm{OPT}_t}\big), \qquad \mathbf{v} = \big(\sqrt{\eta_m},\ \sqrt{\beta\,\eta_t}\big).$$

Then the cross-level error term satisfies

$$\sqrt{10}\big(\sqrt{\mathrm{OPT}_m\,\eta_m} \ +\ \beta\sqrt{\mathrm{OPT}_t\,\eta_t}\big) \ =\ \sqrt{10}\,\mathbf{u}\cdot\mathbf{v}.$$

By Cauchy–Schwarz,

$$\mathbf{u}\cdot\mathbf{v} \ \le\ \|\mathbf{u}\|\,\|\mathbf{v}\| \ =\ \sqrt{\mathrm{OPT}_m + \beta\,\mathrm{OPT}_t}\,\sqrt{\eta_m + \beta\,\eta_t} \ =\ \sqrt{\mathrm{OPT}\,\eta},$$

so the total contribution is bounded by $\sqrt{10\,\mathrm{OPT}\,\eta}$.

Combining this bound with the baseline term $2\,\mathrm{OPT}(\sigma)$ arising from clean-phase misses—and with the additive $O(k_m + \beta k_t)$ slack established in the token allocation section, we obtain

$$\mathbb{E}[\mathrm{cost}_{\mathrm{LATEM}}] \ \le\ 2\,\mathrm{OPT}(\sigma) \ +\ \sqrt{10\,\mathrm{OPT}(\sigma)\,\eta} \ +\ O\big(k_m + \beta k_t\big),$$

which matches the consistency bound claimed in part (1) of Theorem 1.

This gives consistency ratio $2 + O(\sqrt{\eta/\text{OPT}})$, confirming our main theorem statement.

Robustness Analysis: In the worst case, predictions are adversarially bad. Using the analysis from Proposition 1 and 2, when quotas are quickly exhausted:

$$\text{cost}_{\text{LATEM}}(\sigma) \leq \frac{4H_{k_m}}{\varepsilon} \cdot \text{OPT}_m + \frac{4\beta H_{k'_t}}{\varepsilon} \cdot \text{OPT}_t \leq \left(\frac{4H_{k_m}}{\varepsilon} + \frac{4\beta H_{k'_t}}{\varepsilon}\right) \cdot \text{OPT}(\sigma)$$

Each quota-exhausted phase incurs at most $2H_{k_m}$ additional message-level cost and $2\beta H_{k'_t}$ additional token-level cost. Since predictions may govern only an ε-fraction of evictions, dividing the phase length by ε yields the multiplicative factor $4/\varepsilon$.

□

5.5 Lower Bounds

Theorem 2 (Lower Bound)**.** *Any algorithm for two-level caching with coherence constraints that achieves consistency ratio $(1 + \delta)$ for sufficiently small $\delta > 0$ must have robustness $\gamma \geq \Omega(\log k_m + \log k'_t)$, where $k'_t = \lfloor k_t/k_m \rfloor$.*

Proof. We construct adversarial sequences that force the stated lower bound.

Message-level adversary: Use the standard construction for randomized paging lower bounds with $k_m + 1$ messages in round-robin fashion. Any algorithm must have competitive ratio $\Omega(H_{k_m}) = \Omega(\log k_m)$.

Token-level adversary: For each message m that the algorithm keeps cached, construct a token sequence within $T(m)$ using $k'_t + 1$ tokens in round-robin fashion. This forces competitive ratio $\Omega(H_{k'_t}) = \Omega(\log k'_t)$.

Composition argument: Following the direct-sum theorem approach of Rohatgi [2020], we construct a sequence where both challenges occur with balanced costs. Specifically, we ensure $\text{OPT}_m \approx \beta \cdot \text{OPT}_t \approx \text{OPT}/2$ by carefully interleaving message and token adversarial subsequences.

The total competitive ratio becomes:

$$\frac{\Omega(\log k_m) \cdot \text{OPT}/2 + \Omega(\log k'_t) \cdot \text{OPT}/2}{\text{OPT}} = \Omega(\log k_m + \log k'_t)$$

Therefore, any algorithm achieving $(1+\delta)$-consistency must have robustness $\Omega(\log k_m + \log k'_t)$. □

6 Extensions and Applications

6.1 Empirical Motivation from Recent Studies

Our theoretical framework is motivated by recent empirical findings. Laban et al. [2025] conducted large-scale simulation experiments comparing LLM performance in single- and multi-turn settings, finding that "all the top open- and closed-weight LLMs we test exhibit significantly lower performance in multi-turn conversations than single-turn, with an average drop of 39% across six generation tasks". Their analysis of 200,000+ simulated conversations confirms that this degradation is systematic across different model architectures and sizes. This phenomenon is termed "getting lost in conversation" and is attributed to models making premature assumptions and overly relying on incorrect previous attempts.

This empirical evidence strongly supports our hierarchical caching model: as conversations grow longer, the challenge of maintaining relevant context across message and token levels becomes critical for performance.

6.2 Practical Predictions from Attention Scores

In practice, the most natural source of predictions for LLM caching comes from the attention mechanisms that are already computed during inference. When processing a sequence of tokens, the

attention mechanism computes attention scores that indicate how much each token in the context contributes to the current prediction. These scores provide a natural measure of token importance that can be used to predict future access patterns.

For a token t_i at position i, the attention score from the most recent token provides an immediate importance measure. However, for more robust predictions, we can accumulate attention scores over multiple recent tokens or even over entire message boundaries. Following the approach introduced in the H2O paper by [Zhang et al., 2023], we can maintain running averages of attention scores to identify "heavy hitter" tokens that consistently receive high attention.

For message-level predictions, we can aggregate token-level attention scores within each message to derive message-level importance scores. Messages containing many high-attention tokens, or messages that receive high attention scores across multiple recent interactions, can be predicted to be accessed sooner than messages with consistently low attention.

6.3 Parameter Selection and Adaptive Trust

The trust parameter ε provides the primary mechanism for controlling the consistency-robustness trade-off in practical deployments. Small values of ε (e.g., 0.1-0.2) make the algorithm more conservative, requiring very accurate predictions to achieve good consistency but providing strong worst-case guarantees. Larger values of ε (e.g., 0.4-0.6) allow the algorithm to rely more heavily on predictions, potentially achieving better performance when predictions are accurate but with weaker robustness guarantees.

An important extension is adaptive trust parameter selection, where ε is adjusted based on observed prediction quality. If recent predictions have been accurate, the algorithm can increase ε to rely more heavily on predictions. This adaptive approach maintains the theoretical guarantees while potentially improving practical performance.

7 Conclusion

We presented a theoretical framework for long-chat degradation in LLM systems through the lens of hierarchical learning-augmented caching. Our algorithm LATEM achieves consistency ratio $2 + O(\sqrt{\eta/\text{OPT}})$ and robustness $O(\frac{\log k_m + \log k'_t}{\varepsilon})$, with matching lower bounds showing optimality up to constant factors.

Our analysis extends existing techniques to handle cache dependencies and offers a formal starting point for KV-cache analysis in LLMs.

A Complete Proof Derivations

A.1 Proof of Lemma 1: Spread Function Bound (Complete)

Proof. Consider a sequence of items with true next-access times $a_1 < a_2 < \cdots < a_\ell$ and predicted times $b_1 \geq b_2 \geq \cdots \geq b_\ell$. The total prediction error is:

$$\eta = \sum_{i=1}^{\ell} |a_i - b_i|$$

Step 1: Optimal prediction configuration. To find the maximum length ℓ for given error η, we minimize the error for each sequence length. Without loss of generality, let $a_i = i - 1$ for $i = 1, 2, \ldots, \ell$.

For the predicted times, the error-minimizing choice is $b_i = c$ for all i, where c is the median of $\{a_1, \ldots, a_\ell\} = \{0, 1, \ldots, \ell - 1\}$.

Step 2: Computing the minimal error. For odd $\ell = 2n + 1$, the median is $c = n$. The error becomes:

$$\eta = \sum_{i=1}^{2n+1} |i - 1 - n| = \sum_{j=0}^{2n} |j - n| \quad \text{(substituting } j = i - 1) \tag{1}$$

$$= \sum_{j=0}^{n-1} (n - j) + 0 + \sum_{j=n+1}^{2n} (j - n) \tag{2}$$

$$= \sum_{k=1}^{n} k + \sum_{k=1}^{n} k = 2 \sum_{k=1}^{n} k = n(n + 1) \tag{3}$$

For $\ell = 2n + 1$, we have $n = (\ell - 1)/2$, so:

$$\eta = \frac{\ell - 1}{2} \cdot \frac{\ell + 1}{2} = \frac{\ell^2 - 1}{4}$$

Step 3: Solving for ℓ. From $\eta = \frac{\ell^2 - 1}{4}$, we get: $\ell^2 = 4\eta + 1$, so $\ell = \sqrt{4\eta + 1}$.

For $\eta \geq 1$, we have $\sqrt{4\eta + 1} \leq \sqrt{4\eta + \eta} = \sqrt{5\eta}$.

For even $\ell = 2n$, similar analysis gives $\eta = n^2 = \ell^2/4$, leading to $\ell = 2\sqrt{\eta} \leq \sqrt{5\eta}$ for all $\eta \geq 1$.

Therefore, $S(\eta) \leq \sqrt{5\eta}$ for all $\eta \geq 1$. □

A.2 Consistency Analysis

Cauchy-Schwarz bound:

$$\sum_{i=1}^{N} S(\eta_i) \leq \sum_{i=1}^{N} \sqrt{5\eta_i} \leq \sqrt{5N} \sqrt{\sum_{i=1}^{N} \eta_i}$$

Phase counting: Since each phase contributes at least $1/2$ to OPT (by potential function analysis), we have $N \leq 2 \cdot \text{OPT}$.

Final bound: This gives us:

$$\text{cost} \leq 2 \cdot \text{OPT} + \sqrt{10 \cdot \text{OPT} \cdot \eta} = \text{OPT}(2 + \sqrt{10\eta/\text{OPT}})$$

which is exactly consistency ratio $2 + O(\sqrt{\eta/\text{OPT}})$ with leading coefficient 2.

References

A. Aggarwal, B. Alpern, A. Chandra, and M. Snir. A model for hierarchical memory. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, page 305–314, New York, NY, USA, 1987. Association for Computing Machinery. ISBN 0897912217. doi: 10.1145/28395.28428. URL https://doi.org/10.1145/28395.28428.

Marek Chrobak and John Noga. Competitive algorithms for relaxed list update and multilevel caching. *Journal of Algorithms*, 34(2):282–308, 2000. ISSN 0196-6774. doi: https://doi.org/10.1006/jagm.1999.1061. URL https://www.sciencedirect.com/science/article/pii/S0196677499910611.

Amos Fiat, Richard M. Karp, Michael Luby, Lyle A. McGeoch, Daniel Dominic Sleator, and Neal E. Young. Competitive paging algorithms. *CoRR*, cs.DS/0205038, 2002. URL https://arxiv.org/abs/cs/0205038.

A. Karlin, Mark Manasse, L. McGeoch, and S. Owicki. Competitive randomized algorithms for nonuniform problems. *Algorithmica*, 11:542–571, 01 1994. doi: 10.1007/BF01189993.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. Llms get lost in multi-turn conversation, 2025. URL https://arxiv.org/abs/2505.06120.

Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice, 2020. URL https://arxiv.org/abs/1802.05399.

Manish Purohit, Zoya Svitkina, and Ravi Kumar. Improving online algorithms via ml predictions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/73a427badebe0e32caa2e1fc7530b7f3-Paper.pdf.

Dhruv Rohatgi. Near-optimal bounds for online caching with machine learned advice. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2020)*, pages 1834–1845, 2020. doi: 10.1137/1.9781611975994.112.

Daniel D. Sleator and Robert E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208, 1985. doi: 10.1145/2786.2793.

Guangtao Wang, Shubhangi Upasani, Chen Wu, Darshan Gandhi, Jonathan Li, Changran Hu, Bo Li, and Urmish Thakker. Llms know what to drop: Self-attention guided kv cache eviction for efficient long-context inference, 2025. URL https://arxiv.org/abs/2503.08879.

Alexander Wei and Fred Zhang. Optimal robustness-consistency trade-offs for learning-augmented online algorithms. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8042–8053. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/5bd844f11fa520d54fa5edec06ea2507-Paper.pdf.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks, 2024. URL https://arxiv.org/abs/2309.17453.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark W. Barrett, Zhangyang Wang, and Beidi Chen. H_2o: Heavy-hitter oracle for efficient generative inference of large language models. *CoRR*, abs/2306.14048, 2023. doi: 10.48550/ARXIV.2306.14048. URL https://doi.org/10.48550/arXiv.2306.14048.