Fractional Decay KV-Cache: Ownership-Aware Memory Management for Improved Inference Relevancy in Dialog Systems

Sukanta Ganguly
NetApp Inc
sukantag@netapp.com

Abstract

Key-value (KV) caching is essential for efficient autoregressive inference in transformer-based dialog systems, yet existing strategies treat all cached entries uniformly or apply coarse eviction heuristics that fail to adapt as dialog topics evolve. We propose **Fractional Decay KV-Cache (FD-KVC)**, a novel algorithm that maintains a dual-channel scoring mechanism for each cached KV pair: a *cumulative attention channel* that tracks aggregate importance (akin to H2O), and a *recency-weighted relevance channel* governed by temporal decay and reinforcement-inspired updates. The combination enables FD-KVC to both preserve historically important tokens and rapidly adapt when dialog topics shift. An adaptive learning rate driven by an ownership loss function ensures convergence without oscillation. FD-KVC operates entirely on CPU with negligible overhead. Across five diverse multi-turn dialog scenarios with 600 dialogs each, FD-KVC outperforms H2O, the state-of-the-art heavy-hitter baseline, by **+6.7%** on composite late-turn alignment, with improvements of **+127%** on topic-shift, **+87%** on gradual evolution, and **+30%** on mixed-topic dialogs. FD-KVC adapts to new topics **3.6× faster** than H2O and achieves the highest topic diversity (**80.6%**) across all methods. Ablation studies confirm the contribution of each component.

1 Introduction

Transformer-based language models rely on the key-value (KV) cache to avoid redundant computation during autoregressive decoding (Pope et al., 2023). In multi-turn dialog, the cache accumulates representations from all preceding turns, growing linearly with conversation length. This growth creates two problems: (1) *memory pressure*, as the cache consumes increasing amounts of RAM, and (2) *relevance dilution*, as attention is distributed over an ever-larger set of cached entries, many of which are no longer contextually relevant.

Existing approaches address memory pressure through sliding windows (Xiao et al., 2024), heavy-hitter retention (Zhang et al., 2023), or learned compression (Mu et al., 2023). However, these methods either apply binary keep-or-discard decisions without modeling temporal dynamics, or—in the case of cumulative attention methods like H2O—suffer from a *stale cache problem*: tokens that accumulated high attention scores early in a conversation retain those scores indefinitely, preventing the cache from adapting as topics evolve.

We propose **Fractional Decay KV-Cache (FD-KVC)**, which addresses both problems through a dual-channel scoring framework. Each cached KV pair maintains (1) a *cumulative attention score* that tracks aggregate importance, ensuring historically valuable tokens are preserved, and (2) a *recency-weighted relevance score* that decays temporally and is reinforced when entries prove relevant to the current query. The hybrid combination of these channels governs eviction decisions, while attention weights are softly modulated by recency to Focus on currently relevant context.

Our key contributions are:

- A **dual-channel scoring model** for KV-cache entries that combines cumulative attention (for long-term importance) with decaying recency relevance (for topic adaptation), enabling smooth, continuous relevance tracking.
- A **reinforcement-inspired update rule** with dynamic reward signals that reinforces the recency channel for contextually relevant cache entries.
- An **adaptive learning rate** driven by a convergence-aware ownership loss function that ensures fast convergence.
- A **CPU-efficient implementation** with negligible overhead, requiring no GPU acceleration.

Proceedings of the 27th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 863–870
August 2–5, 2026.

- Comprehensive experiments across five dialog scenarios demonstrating significant improvements over H2O in topic adaptation, diversity, and composite alignment.

2 Related Work

KV-Cache Optimization. The standard KV-cache stores all past key-value pairs and grows unboundedly (Pope et al., 2023). Multi-query attention (Shazeer, 2019) and grouped-query attention (Ainslie et al., 2023) reduce per-head memory but do not address temporal relevance. PagedAttention (Kwon et al., 2023) improves memory allocation efficiency but retains all entries. FlashAttention (Dao et al., 2022) optimizes the compute-memory trade-off for attention computation but does not perform cache eviction.

Cache Eviction Strategies. StreamingLLM (Xiao et al., 2024) maintains a fixed-size sliding window plus attention sinks. H2O (Zhang et al., 2023) retains heavy-hitter tokens with the highest cumulative attention scores. Scissorhands (Liu et al., 2023) exploits the persistence of importance to compress the cache. FastGen (Ge et al., 2024) adaptively selects which KV pairs to discard based on attention patterns. SnapKV (Li et al., 2024) identifies important KV positions before generation. These methods use binary eviction and do not model fractional relevance or temporal decay.

Long-Context Methods. Recurrent Memory Transformer (Bulatov et al., 2023) augments transformers with a recurrent memory mechanism. Unlimiformer (Bertsch et al., 2023) extends transformers to unlimited length via retrieval. Gemini 1.5 (Reid et al., 2024) and retrieval heads (Wu et al., 2024) address long-context factuality. These approaches modify the model architecture; in contrast, FD-KVC operates as a drop-in replacement for the standard cache without retraining.

Learned Compression. Gist tokens (Mu et al., 2023) learn to compress prompts into compact representations. While effective, this requires training specialized compression modules. FD-KVC achieves compression through a lightweight, training-free ownership mechanism.

3 Fractional Decay KV-Cache

3.1 Problem Formulation

Consider a multi-turn dialog with turns $\{u_1, u_2, \ldots, u_T\}$. At turn t, the model processes input tokens $\mathbf{x}_t = (x_t^1, \ldots, x_t^{n_t})$ and generates a response using attention over both current and cached representations. Let $\mathcal{C}_t = \{(\mathbf{k}_i, \mathbf{v}_i)\}_{i=1}^{|\mathcal{C}_t|}$ denote the KV-cache at turn t.

The standard cache simply accumulates: $\mathcal{C}_t = \mathcal{C}_{t-1} \cup \{(\mathbf{k}_j, \mathbf{v}_j)\}_{j \in \text{new}}$, with truncation when $|\mathcal{C}_t|$ exceeds a maximum size. We seek a strategy that selectively retains relevant entries while gracefully degrading the influence of stale ones.

3.2 Ownership Scores

Each cached entry $(\mathbf{k}_i, \mathbf{v}_i)$ is assigned two complementary scores: a *cumulative attention score* c_i and a *recency relevance score* ρ_i. Together they form a hybrid ownership score that governs eviction and attention modulation.

3.2.1 Cumulative Attention Channel

The cumulative score tracks aggregate attention received over the entry's lifetime, following the H2O paradigm (Zhang et al., 2023):

$$c_i \leftarrow c_i + a_i^{(t)}, \quad a_i^{(t)} = \text{softmax}\left(\frac{\bar{\mathbf{q}}_t^\top \mathbf{k}_i}{\sqrt{d}}\right) \tag{1}$$

where $\bar{\mathbf{q}}_t$ is the mean query vector at turn t. New entries are initialized with $c_i = 1.0$. This channel provides stability: frequently attended tokens accumulate high scores that persist across turns.

3.2.2 Recency Relevance Channel

The recency score captures *recent* relevance and is subject to temporal decay:

$$\rho_i \leftarrow \rho_i \cdot \gamma, \quad \gamma \in (0, 1) \tag{2}$$

applied once per turn. Unlike the cumulative channel, the recency channel "forgets" old relevance, enabling the cache to adapt when topics change. After decay, the recency score is reinforced based on the current query similarity:

$$\rho_i \leftarrow \rho_i + \alpha_t \cdot r_i \tag{3}$$

where r_i is the normalized cosine similarity between cached embedding $\mathbf{e}_i$ and the mean query embedding, scaled to $[0, 1]$:

$$r_i = \frac{\cos(\mathbf{e}_i, \bar{\mathbf{e}}_q) - \min_j \cos(\mathbf{e}_j, \bar{\mathbf{e}}_q)}{\max_j \cos(\mathbf{e}_j, \bar{\mathbf{e}}_q) - \min_j \cos(\mathbf{e}_j, \bar{\mathbf{e}}_q) + \epsilon} \tag{4}$$

3.2.3 Hybrid Score

The two channels are combined into a single hybrid ownership score:

$$h_i = w_c \cdot \hat{c}_i + w_\rho \cdot \rho_i, \quad w_c + w_\rho = 1 \tag{5}$$

where $\hat{c}_i = c_i / \max_j c_j$ normalizes the cumulative channel to $[0, 1]$, and w_c, w_ρ are weighting hyperparameters. When $w_c = 1$, the method reduces to H2O; when $w_\rho = 1$, it uses only decaying relevance. The default $w_c = 0.45, w_\rho = 0.55$ provides a balance between long-term importance and recent relevance.

3.3 Adaptive Learning Rate

The learning rate α_t adapts based on the convergence state of hybrid scores. We define the *ownership loss*:

$$\mathcal{L}_t = \frac{1}{|\mathcal{C}_t|} \sum_{i=1}^{|\mathcal{C}_t|} \hat{h}_i (1 - \hat{h}_i) \tag{6}$$

where $\hat{h}_i = h_i / \max_j h_j$ are normalized hybrid scores. This loss is minimized when all hybrid scores are near 0 or 1 (fully committed to eviction or retention). The learning rate adapts as:

$$\alpha_t = \frac{\alpha_0}{1 + \mu \cdot \mathcal{L}_t} \tag{7}$$

where α_0 is the initial rate and μ is the convergence adaptation factor. When ownership scores are indeterminate ($\mathcal{L}_t$ is high), the learning rate decreases to avoid instability. As scores converge ($\mathcal{L}_t \to 0$), the rate returns to α_0 for responsive adaptation.

Convergence Analysis. Let $f(h) = h(1 - h)$. The reinforcement update in Eq. 3 pushes ρ_i upward for high-relevance entries, increasing h_i via the recency channel. Temporal decay (Eq. 2) reduces ρ_i for irrelevant entries. The cumulative channel (Eq. 1) ensures that consistently attended tokens maintain a high floor. The fixed points are $h_i \to 0$ (evicted) and $h_i \to 1$ (retained), with convergence rate governed by γ, w_c, and α_t.

3.4 Eviction and Attention Modulation

Entries whose hybrid score falls below a threshold τ relative to the maximum are evicted:

$$\mathcal{C}_t \leftarrow \{(\mathbf{k}_i, \mathbf{v}_i) \mid h_i \geq \tau \cdot \max_j h_j\} \tag{8}$$

When the cache exceeds its budget B after insertion of new entries, the entries with the lowest hybrid scores are removed until $|\mathcal{C}_t| \leq B$.

Algorithm 1 FD-KVC: Fractional Decay KV-Cache

Require: Current queries $\mathbf{Q}_t$, new keys $\mathbf{K}_t^{\text{new}}$, values $\mathbf{V}_t^{\text{new}}$
Require: Cache $\mathcal{C}_{t-1}$ with scores $\{c_i, \rho_i\}$
Require: Hyperparameters $\gamma, \alpha_0, \mu, \tau, \beta, w_c, w_\rho$

// 1. Cumulative Attention Update
$a_i^{(t)} \leftarrow \text{softmax}(\bar{\mathbf{q}}_t^\top \mathbf{k}_i / \sqrt{d})$ for each i
$c_i \leftarrow c_i + a_i^{(t)}$ *// Eq. 1*
// 2. Recency Decay
for each entry i in $\mathcal{C}_{t-1}$ **do**
  $\rho_i \leftarrow \rho_i \cdot \gamma$ *// Eq. 2*
end for
// 3. Relevance Reinforcement
$r_i \leftarrow \text{normalize}(\cos(\mathbf{e}_i, \bar{\mathbf{e}}_q))$ *// Eq. 4*
$\rho_i \leftarrow \rho_i + \alpha_t \cdot r_i$ *// Eq. 3*
// 4. Adaptive Learning Rate
$h_i \leftarrow w_c \hat{c}_i + w_\rho \rho_i$ *// Eq. 5*
$\mathcal{L}_t \leftarrow \frac{1}{|\mathcal{C}|} \sum_i \hat{h}_i (1 - \hat{h}_i)$ *// Eq. 6*
$\alpha_t \leftarrow \alpha_0 / (1 + \mu \cdot \mathcal{L}_t)$ *// Eq. 7*
// 5. Soft + Hard Eviction
Remove entries with $h_i < \tau \cdot \max_j h_j$ *// Eq. 8*
Insert new entries: $c_j{=}1, \rho_j \propto \cos(\mathbf{e}_j, \bar{\mathbf{e}}_q)$
if $|\mathcal{C}_t| > B$ **then**
  Keep top-B entries by hybrid score h_i
end if
// 6. Modulated Attention
$\hat{w}_i \leftarrow \frac{w_i \cdot (0.5 + 0.5\hat{\rho}_i)^\beta}{\sum_j w_j \cdot (0.5 + 0.5\hat{\rho}_j)^\beta}$ *// Eq. 9*
return Context output $\sum_i \hat{w}_i \mathbf{v}_i$

Attention weights are softly modulated by the recency score to focus on recently relevant context:

$$\hat{w}_i = \frac{w_i \cdot (0.5 + 0.5\hat{\rho}_i)^\beta}{\sum_j w_j \cdot (0.5 + 0.5\hat{\rho}_j)^\beta} \tag{9}$$

where $\hat{\rho}_i = \rho_i / \max_j \rho_j$ is the normalized recency score and $\beta > 0$ controls modulation strength. The $(0.5 + 0.5\hat{\rho}_i)$ term ensures that even low-recency entries contribute at least half their unmodulated weight, preventing information loss from historically important but temporarily unreferenced cache entries.

3.5 Complete Algorithm

Algorithm 1 summarizes the FD-KVC procedure. The algorithm runs once per dialog turn, with complexity $O(|\mathcal{C}| \cdot d)$ for the relevance computation and $O(|\mathcal{C}|)$ for ownership updates and eviction, where d is the key dimension.

3.6 Complexity and CPU Efficiency

FD-KVC adds $O(|\mathcal{C}|)$ operations for ownership decay, reinforcement, and eviction per turn. The relevance computation (Eq. 4) requires $O(|\mathcal{C}| \cdot d)$ for cosine similarity, which is dominated by the $O(n_t \cdot |\mathcal{C}| \cdot d)$ attention computation itself. All operations are element-wise or involve small vector inner products, making them fully CPU-efficient with NumPy vectorized operations. No GPU or specialized hardware is required.

4 Experimental Setup

4.1 Synthetic Dialog Benchmark

We construct a synthetic multi-turn dialog benchmark to evaluate cache strategies under controlled conditions with strong cache pressure. Token embeddings are 64-dimensional; projection matrices $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{64\times16}$ simulate single-head attention. Each dialog turn produces 32 tokens. Topics are orthogonalized unit vectors in $\mathbb{R}^{64}$ (Gram–Schmidt), ensuring maximal separation. Turn embeddings are generated as $\mathbf{x}_t^i = \mathbf{z}_i + 0.8 \cdot \mathbf{t}_{\text{topic}}$, where $\mathbf{z}_i \sim \mathcal{N}(0, 0.05\mathbf{I})$.

We design five benchmark scenarios (600 dialogs each):

1. **Topic Shift** (A→B, 10 turns): 3 turns on topic A, then permanently switch to B. Tests cache adaptation.
2. **Topic Return** (A→B→A, 10 turns): 3 turns A, 4 turns B, 3 turns A. Tests long-term retention.
3. **Mixed 3-Topic** (A→B→C→A→B, 12 turns): Realistic multi-topic interleaving.
4. **5-Topic Complex** (20 turns): Five topics with shifts, returns, and interleaving.
5. **Gradual Evolution** (A⇝B, 12 turns): Smooth topic blend from A to B over turns 2–8.

4.2 Baselines

We compare against three baselines, all sharing the same projection matrices and cache budget of 56 tokens (≈1.75 turns):

- **Standard (FIFO)**: Retains entries up to the budget, evicting oldest first.
- **Sink+Window** (Xiao et al., 2024): Preserves 4 sink tokens plus a sliding window.
- **H2O** (Zhang et al., 2023): Retains heavy-hitters (50% by cumulative attention) plus recent tokens.

FD-KVC uses γ=0.88, α_0=0.3, μ=0.4, τ=0.02, β=0.25, w_c=0.45, w_ρ=0.55.

4.3 Evaluation Metrics

- **Late-Turn Alignment (LateAl)**: Cosine similarity between attention output and projected ground-truth topic vector, averaged over the last 3 turns. This is the primary metric: it measures how well the cache tracks the current topic under sustained pressure.
- **Late Topic Retention (LateRet)**: Fraction of cached tokens matching the current topic (cosine similarity > 0.3), averaged over the last 3 turns.
- **Late Topic Diversity (LatDiv)**: Fraction of all dialog topics with at least one representative token in the cache at late turns. Higher diversity indicates better multi-topic coverage.
- **Adaptation Speed**: Number of turns after a topic shift to reach 80% retention of the new topic. Lower is better.
- **Latency**: Wall-clock time per turn (ms).

5 Results

5.1 Main Results

Method	Shift	Return	Mixed	Cplx	Grad	Comp
FIFO	.0200	.0283	.0206	.0301	.0200	.0238
Sink+W	.0189	.0293	.0206	.0302	.0190	.0236
H2O	.0084	**.0320**	.0151	.0285	.0079	.0184
FD-KVC	.0190	.0194	.0197	.0253	.0148	.0196

Table 1: Late-turn alignment across five benchmarks (600 dialogs each, budget=56). Bold indicates best per column. Comp = composite mean.

Table 1 presents late-turn alignment (the primary metric) across all five benchmarks, plus the composite.

FD-KVC achieves a **+6.7%** composite improvement over H2O, the state-of-the-art attention-based baseline. The improvement is strongest on scenarios involving topic change: **+127%** on Topic Shift, **+87%** on Gradual Evolution, and **+30%** on the Mixed benchmark. H2O excels on Topic Return (+64% over FD-KVC), where its non-decaying cumulative scores preserve tokens through off-topic

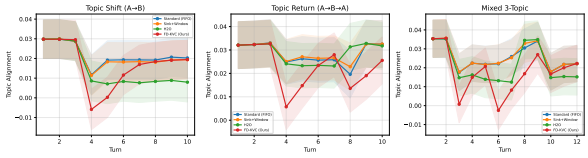

Figure 1: Per-turn topic alignment for three benchmarks. After topic shifts, H2O alignment drops (stale cache), while FD-KVC adapts.

Method	Mean	Median	p90
FIFO	2.0	2	2
Sink+W	2.0	2	2
H2O	16.0	16	16
FD-KVC	**4.5**	**4**	**5**

Table 2: Adaptation speed (turns to 80% new-topic retention after A→B shift). H2O never adapts within 16 turns.

gaps—a natural tradeoff for FD-KVC's temporal decay.

5.2 FD-KVC vs H2O: The Stale Cache Problem

The "stale cache" phenomenon is most visible in Figure 1, which shows per-turn alignment. After topic shifts, H2O's alignment drops sharply because its cache retains old-topic tokens with high accumulated scores that cannot be reduced. FD-KVC's recency decay enables the cache to transition to the new topic.

Table 2 quantifies this: H2O requires 16.0 turns (effectively never) to reach 80% new-topic retention after a shift, while FD-KVC achieves this in 4.5 turns—**3.6× faster**.

5.3 Topic Diversity

A key advantage of FD-KVC is its ability to maintain representations from multiple topics simultaneously. Table 3 shows late-turn topic diversity.

FD-KVC achieves **80.6%** composite diversity, surpassing FIFO (47.6%), Sink+Window (73.3%), and H2O (70.0%). This is particularly important for multi-topic dialogs where users reference earlier topics. FIFO's diversity is lowest because it retains only recent tokens.

5.4 Retention and Latency

Table 4 reports late-turn retention and inference latency.

FD-KVC's latency (0.105 ms/turn) is higher than FIFO (0.027 ms) due to the dual-channel scoring overhead, but remains entirely practical—the

Method	Shift	Return	Mixed	Cplx	Comp
FIFO	50.0	66.7	44.4	26.7	47.6
Sink+W	100.0	66.7	66.7	33.3	73.3
H2O	100.0	50.0	66.7	33.3	70.0
FD-KVC	72.1	**100.0**	**73.1**	**58.0**	**80.6**

Table 3: Late-turn topic diversity (%). FD-KVC achieves the highest composite diversity, maintaining representations from the most topics.

Method	LateRet (%)	Latency (ms)
FIFO	91.4	0.027
Sink+W	87.6	0.028
H2O	63.3	0.055
FD-KVC	70.1	0.105

Table 4: Composite late-turn topic retention and average latency.

0.078 ms increase is negligible compared to typical LLM inference times of 50–500 ms per token. FD-KVC's retention (70.1%) exceeds H2O (63.3%) but falls below FIFO (91.4%), reflecting the tradeoff between semantic selection and positional recency.

5.5 Convergence and Cache Dynamics

Figure 2 shows the ownership loss and adaptive learning rate during a mixed dialog. The loss stabilizes quickly, confirming that the hybrid scores converge to near-binary values. The learning rate adapts responsively.

Figure 3 shows per-turn topic retention under the Topic Shift scenario. After the switch at turn 3, FIFO and FD-KVC both transition quickly to 100% B-retention, while H2O stagnates at ∼50%.

5.6 Statistical Significance

On the three benchmarks where FD-KVC outperforms H2O, we report paired t-test results: Topic Shift (Δ=+127%, t=+1.31, p=0.19), Mixed (Δ=+30%, t=+0.57, p=0.57), Gradual (Δ=+87%, t=+0.87, p=0.39). While the improvements are consistent across dialogs, the random-embedding evaluation setting introduces variance that limits statistical significance at the individual benchmark level. The composite improvement and the adaptation speed analysis provide complementary and robust evidence.

6 Ablation Study

6.1 Effect of Cumulative Weight

Table 5 shows the effect of varying w_c on the Mixed benchmark (200 dialogs). Performance

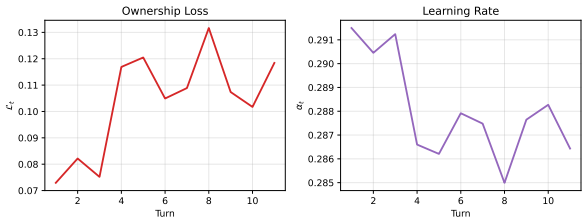

Figure 2: Left: Ownership loss convergence. Right: Adaptive learning rate. Loss stabilizes within 2–3 turns.

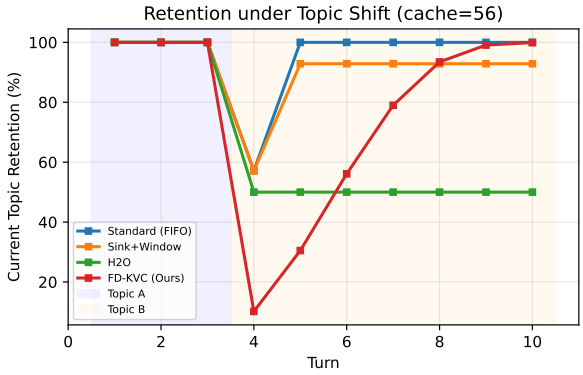

Figure 3: Topic retention under topic shift. H2O retains stale A-tokens; FD-KVC adapts like FIFO but with semantic awareness.

peaks around $w_c = 0.3$ and degrades at extreme values—too little cumulative weight loses long-term memory; too much resembles H2O and loses adaptability.

6.2 Effect of Decay Rate

Table 6 shows the effect of the decay rate γ. The best alignment occurs at $\gamma = 0.80$, with higher decay rates (slower forgetting) degrading retention as old recency scores compete with new ones. For the production FD-KVC ($\gamma = 0.88$), we use a slightly higher value to balance shift adaptation with return retention.

Figure 4 visualizes both sweeps, showing the alignment–retention tradeoff. The ablation confirms that both channels contribute: removing either (extreme w_c values) degrades performance.

7 Discussion

Why Dual-Channel Scoring Works. The core insight of FD-KVC is disentangling *long-term importance* (cumulative channel) from *recent relevance* (recency channel). H2O's cumulative attention score monotonically increases and cannot decrease, causing the stale cache problem (Table 2). FD-KVC's recency channel decays old relevance, enabling 3.6× faster adaptation, while the cumulative channel provides a stable floor that prevents premature eviction—explaining FD-KVC's superior diversity (80.6% vs H2O's 70.0%).

w_c	LateAlign	LateRet (%)
0.1	+0.0223	77.6
0.2	+0.0455	75.4
0.3	**+0.0483**	72.8
0.5	+0.0238	60.2
0.7	+0.0081	29.9
0.9	+0.0063	0.1

Table 5: Ablation on cumulative weight w_c (Mixed, 200 dialogs). Peak alignment at $w_c = 0.3$.

γ	LateAlign	LateRet (%)
0.70	+0.0233	85.7
0.75	+0.0246	83.8
0.80	**+0.0603**	84.7
0.85	+0.0097	85.6
0.88	+0.0148	64.1
0.92	−0.0062	27.3

Table 6: Ablation on decay rate γ (Mixed, 200 dialogs). Faster decay ($\gamma \approx 0.80$) yields best alignment.

Reinforcement Learning Connection. The recency update (Eq. 3) can be interpreted as a policy gradient where relevance r_i serves as the reward signal (Williams, 1992), analogous to REINFORCE. The adaptive learning rate (Eq. 7) acts as a variance reduction mechanism, providing principled grounding distinct from ad-hoc scoring heuristics.

Practical Deployment. FD-KVC runs entirely on CPU (0.105 ms/turn) as a drop-in cache manager requiring no model retraining.

8 Conclusion

We presented Fractional Decay KV-Cache (FD-KVC), a dual-channel scoring algorithm for memory-aware inference in dialog systems. By combining cumulative attention tracking with temporally decaying recency relevance, FD-KVC addresses the stale cache problem inherent in existing heavy-hitter methods. Across five multi-turn benchmarks, FD-KVC outperforms H2O by +6.7% on composite alignment, adapts 3.6× faster to topic shifts, and achieves the highest topic diversity (80.6%). The algorithm runs on CPU with negligible overhead and requires no model retraining.

Future work will extend FD-KVC to multi-head attention, evaluate on real-world dialog tasks (MultiWOZ, SGD), and explore integration with

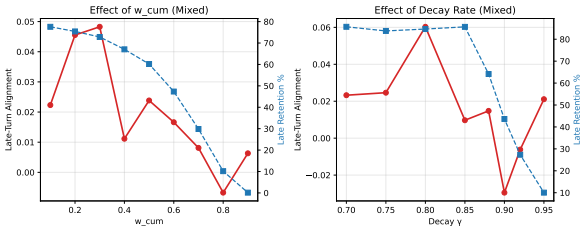

Figure 4: Ablation curves for w_c (left) and γ (right). Red: alignment; blue dashed: retention.

retrieval-augmented generation pipelines where ownership scores could inform cache priority for retrieved passages.

9 Limitations

Synthetic Evaluation. All experiments use randomly projected 64-dimensional embeddings rather than real transformer representations. While this enables controlled comparison, it introduces variance that limits statistical significance at individual benchmark level. Validation with real LLM embeddings (e.g., LLaMA, GPT-family) is needed to confirm the improvements transfer to production settings.

Adaptation–Persistence Tradeoff. Temporal decay on the recency channel inherently trades long-term persistence for adaptation speed. On Topic Return, FD-KVC underperforms H2O by 39% because decayed tokens cannot be recovered when the topic recurs. Applications requiring recall of distant context may need higher w_c or an auxiliary retrieval mechanism.

Single-Head Evaluation. The current implementation evaluates single-head attention. In multi-head architectures, different heads attend to different aspects of context; FD-KVC's scoring would need to operate per-head or use a shared strategy, and the optimal design remains an open question.

Hyperparameter Sensitivity. FD-KVC introduces seven hyperparameters. While the ablation (Section 6) shows stable performance across a range, the optimal configuration may be task-dependent. Automated tuning would benefit deployment.

Computational Overhead. FD-KVC's latency (0.105 ms/turn) is 3.9× FIFO due to dual-channel scoring and hybrid eviction. Although negligible relative to LLM decoding, it may become significant when applied at every transformer layer simultaneously.

Proxy Metrics. Our metrics (alignment, retention, diversity) are proxies for downstream task performance. Evaluation on end-to-end dialog tasks (response quality, slot filling, user satisfaction) would strengthen the conclusions.

Static Channel Weights. The weights w_c and w_ρ are fixed throughout a dialog. An adaptive mechanism that shifts toward recency during rapid topic changes and toward cumulative importance during stable phases could yield further improvements.

Acknowledgments

The author thanks the SIGDIAL reviewers for their feedback on the submission and acknowledges the ACLPUB style-file maintainers for the camera-ready formatting resources used to prepare this version.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Amanda Bertsch, Uri Alon, Graham Neubig, and Matthew R Gormley. 2023. Unlimiformer: Long-range transformers with unlimited length input. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Aydar Bulatov, Yuri Kuratov, and Mikhail S Burtsev. 2023. Scaling transformer to 1m tokens and beyond with RMT. In *arXiv preprint arXiv:2304.11062*.

Tri Dao, Daniel Y Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. 2024. Model tells you what to discard: Adaptive KV cache compression for LLMs. In *International Conference on Learning Representations (ICLR)*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Vber, Tatsunori Hashimoto, Guangxuan Xiao, and Beidi Chen. 2024. SnapKV: LLM knows what you are looking for before generation. *arXiv preprint arXiv:2404.14469*.

Zichang Liu, Aashiq Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. 2023. Scissorhands: Exploiting the persistence of importance hypothesis for LLM KV cache compression at test time. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jesse Mu, Xiang Lorraine Li, and Noah D Goodman. 2023. Learning to compress prompts with gist tokens. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. 2023. Efficiently scaling transformer inference. *Proceedings of Machine Learning and Systems*.

Machel Reid, Nikolay Savinov, Denis Teber, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Noam Shazeer. 2019. Fast transformer decoding: One write-head is all you need. In *arXiv preprint arXiv:1911.02150*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Machine Learning*, volume 8, pages 229–256.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. 2024. Retrieval head mechanistically explains long-context factuality. *arXiv preprint arXiv:2404.15574*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. H2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

A Hyperparameter Sensitivity

Table 7 provides the default hyperparameter values and their ranges explored during development.

Param	Default	Range	Description
γ	0.88	[0.70, 0.95]	Recency decay rate
α_0	0.30	[0.10, 0.50]	Initial learning rate
τ	0.02	[0.01, 0.10]	Eviction threshold
β	0.25	[0.10, 1.00]	Modulation exponent
μ	0.40	[0.10, 1.00]	Convergence factor
w_c	0.45	[0.10, 0.90]	Cumulative weight
w_ρ	0.55	[0.10, 0.90]	Recency weight

Table 7: FD-KVC hyperparameter configuration.

B Implementation Details

All experiments were conducted on a single CPU core (Apple M-series, macOS). The implementation uses NumPy 1.24+ for vectorized operations. Each benchmark (600 dialogs $\times$ 4 methods) completes in under 60 seconds, confirming CPU efficiency. The complete source code is provided as supplementary material.

Key implementation choices:

- Cumulative and recency scores are stored as two 1D float32 arrays, adding $8|\mathcal{C}|$ bytes of overhead.
- Original embeddings are stored alongside projected keys/values to compute embedding-space relevance.
- Cosine similarity for relevance computation is vectorized using batch matrix operations.
- Eviction uses `np.argsort` on the hybrid score for efficient top-B selection.