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

Sukanta Ganguly | NetApp Inc. | sukantag@netapp.com |

SIGDIAL 2026

Motivation

Transformer dialog systems rely on a **key-value (KV) cache** to avoid redundant computation during autoregressive decoding. As conversations grow, the cache accumulates representations from every turn, creating two problems:

- **Memory pressure** — the cache grows unboundedly with dialog length.
- **Relevance dilution** — attention is spread over an ever-larger set of cached entries, many no longer contextually relevant.

Existing eviction strategies (sliding windows, heavy-hitter retention, learned compression) apply *binary* keep-or-discard decisions. Cumulative-attention methods such as **H2O** suffer from a **stale-cache problem**: once a token accumulates a high attention score, that score can never decrease — so the cache cannot adapt when the dialog topic shifts.

Related Work

- **KV-Cache Optimization:** MQA (Shazeer 2019), GQA (Ainslie et al. 2023), PagedAttention (Kwon et al. 2023), FlashAttention (Dao et al. 2022) — improve memory/compute layout but retain *all* entries.
- **Cache Eviction:** StreamingLLM sliding window + sinks (Xiao et al. 2024); **H2O** heavy-hitter retention (Zhang et al. 2023); Scissorhands (Liu et al. 2023); FastGen (Ge et al. 2024); SnapKV (Li et al. 2024) — all use *binary* eviction, no fractional/temporal modeling.
- **Long-Context Methods:** RMT (Bulatov et al. 2023), Unlimiformer (Bertsch et al. 2023), Gemini 1.5 — modify architecture rather than cache policy.
- **Learned Compression:** Gist tokens (Mu et al. 2023) — requires training specialized modules.

FD-KVC is a *training-free*, drop-in replacement for the standard cache that models continuous, temporally-aware relevance.

Key Contributions

1. A **dual-channel scoring model**: cumulative attention (long-term importance) + decaying recency relevance (topic adaptation) for smooth, continuous relevance tracking.
2. A **reinforcement-inspired update rule** with dynamic reward signals that reinforces the recency channel for contextually relevant entries.
3. An **adaptive learning rate** driven by a convergence-aware ownership loss for fast, stable convergence.
4. A **CPU-efficient implementation** with negligible overhead — no GPU acceleration required.
5. Comprehensive experiments across **5 dialog scenarios** (600 dialogs each) showing large gains over H2O in adaptation, diversity, and composite alignment.

Retention, Latency & Convergence

Table 4: Retention & latency

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

FD-KVC's **+0.078 ms** overhead is negligible next to 50–500 ms per-token LLM decoding.

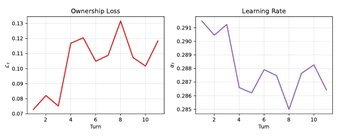

Fig. 2 — Ownership loss stabilizes within 2–3 turns; learning rate adapts responsively.

Discussion & Limitations

Why dual-channel scoring works: disentangling long-term importance (cumulative) from recent relevance (recency) lets FD-KVC adapt 3.6× faster than H2O while the cumulative floor still prevents premature eviction of important tokens (explaining its diversity advantage). The recency update can be viewed as a **policy-gradient / REINFORCE-style** reward signal, with the adaptive learning rate acting as a variance-reduction mechanism.

Limitations: (1) synthetic 64-d embeddings, not real transformer representations; (2) adaptation–persistence tradeoff — decayed tokens cannot be recovered on topic return (H2O wins here by 39%); (3) single-head evaluation only; (4) 7 hyperparameters, task-dependent tuning; (5) latency is 3.9× FIFO (still negligible vs. LLM decoding); (6) alignment/retention/diversity are proxies for downstream task quality; (7) channel weights w_c, w_ρ are static.

Fractional Decay KV-Cache: Dual-Channel Scoring

Problem Formulation. At turn t the model attends over cached pairs $C_t = \{(k_i, v_i)\}_{i=1}^{|C_t|}$. The standard cache accumulates $C_t = C_{t-1} \cup \{(k_j, v_j)\}_{j \in \text{new}}$ with truncation once $|C_t|$ exceeds budget B. FD-KVC instead *selectively retains* relevant entries and gracefully degrades stale ones.

3.1 Cumulative Attention Channel (stability, akin to H2O):

$$c_i \leftarrow c_i + a_i, \qquad a_i = \text{softmax}\left(\frac{\bar{q}_t^\top k_i}{\sqrt{d}}\right) \tag{1}$$

3.2 Recency Relevance Channel (adaptation, decays every turn):

$$\rho_i \leftarrow \rho_i \cdot \gamma, \quad \gamma \in (0,1) \tag{2} \qquad \rho_i \leftarrow \rho_i + \alpha_t \cdot r_i \tag{3}$$

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

3.3 Hybrid Ownership Score:

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

w_c=1 reduces to H2O; w_ρ=1 uses only decaying relevance. Default: w_c=0.45, w_ρ=0.55.

3.4 Adaptive Learning Rate. Ownership loss $\mathcal{L}_t = \frac{1}{|C|}\sum_i \hat{h}_i(1-\hat{h}_i)$ (Eq. 6) is minimized when scores are near-binary (fully committed). The rate adapts as

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

so the cache slows down when ownership is indeterminate and speeds up as it converges.

3.5 Eviction & Attention Modulation. Entries below threshold are dropped, $C_t \leftarrow \{(k_i, v_i) \mid h_i \geq \tau \cdot \max_j h_j\}$ (Eq. 8), and surviving attention weights are softly modulated by recency:

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

guaranteeing low-recency entries still retain at least half their unmodulated weight.

Algorithm

Require: Queries Q_t, new keys/values K_t^{new}, V_t^{new}; cache C_{t-1} with $\{c_i, \rho_i\}$; hyperparams $\gamma, \alpha_0, \mu, \tau, \beta, w_c, w_\rho$

1: **Cumulative update:** $a_i^{(t)} \leftarrow \text{softmax}(\bar{q}_t^\top k_i/\sqrt{d})$; $c_i \leftarrow c_i + a_i^{(t)}$ (Eq.1)
2: **Recency decay:** $\rho_i \leftarrow \rho_i \cdot \gamma \ \ \forall i \in C_{t-1}$ (Eq.2)
3: **Relevance reinforcement:** $r_i \leftarrow \text{norm}(\cos(e_i, \bar{e}_q))$; $\rho_i \leftarrow \rho_i + \alpha_t r_i$ (Eq.3-4)
4: **Adaptive rate:** $h_i \leftarrow w_c \hat{c}_i + w_\rho \rho_i$; $\mathcal{L}_t \leftarrow$ Eq.6; $\alpha_t \leftarrow \alpha_0/(1+\mu\mathcal{L}_t)$ (Eq.5-7)
5: **Soft+hard eviction:** remove $h_i < \tau \max_j h_j$; insert new entries (Eq.8)
6: **if** $|C_t| > B$ **then** keep top-B entries by h_i
7: **end if**
8: **Modulated attention:** $\hat{w}_i \leftarrow$ Eq.9; **return** $\sum_i \hat{w}_i v_i$

Complexity: $O(|C| \cdot d)$ per turn for relevance (dominated by attention itself), fully vectorized on **CPU** — no GPU required.

Experimental Setup

Synthetic multi-turn benchmark: 64-d embeddings, single-head projections $W_Q, W_K, W_V \in \mathbb{R}^{64\times16}$, 32 tokens/turn, Gram–Schmidt orthogonalized topic vectors. **5 scenarios** (600 dialogs each): *Topic Shift* (A→B), *Topic Return* (A→B→A), *Mixed 3-Topic*, *5-Topic Complex*, *Gradual Evolution* (A⇝B).
Baselines (cache budget = 56 tokens ≈1.75 turns): FIFO, Sink+Window (Xiao et al. 2024), H2O (50% heavy-hitters, Zhang et al. 2023). FD-KVC: γ=0.88, α_0=0.3, μ=0.4, τ=0.02, β=0.25, w_c=0.45, w_ρ=0.55.
Metrics: Late-Turn Alignment (*LateAlign*, primary), Late Topic Retention (*LateRet*), Late Topic Diversity (*LatDiv*), Adaptation Speed (turns to 80% new-topic retention), Latency (ms/turn).

Conclusion & Future Work

FD-KVC combines cumulative attention with temporally-decaying recency relevance to solve the stale-cache problem of heavy-hitter methods. Across five multi-turn benchmarks it beats H2O by **+6.7%** composite alignment, adapts **3.6× faster** to topic shifts, and reaches the **highest topic diversity (80.6%)** — all on CPU with negligible overhead and *no model retraining*.

Future work: multi-head attention, evaluation on real dialog tasks (MultiWOZ, SGD), and integration with retrieval-augmented generation where ownership scores prioritize retrieved passages.

Main Results: Late-Turn Alignment

FD-KVC achieves +6.7% composite improvement over H2O, strongest on topic-change scenarios: **+127%** Topic Shift, **+87%** Gradual Evolution, **+30%** Mixed. H2O wins on Topic Return (+64%) where non-decaying scores preserve tokens through off-topic gaps.

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**

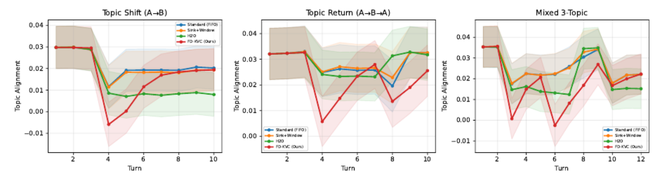

Fig. 1 — Per-turn topic alignment. After a topic shift, H2O's alignment collapses (stale cache) while FD-KVC recovers.

Adaptation Speed & Topic Diversity

Table 2: Adaptation speed (turns to 80% new-topic retention)

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

H2O **never** adapts within 16 turns; FD-KVC is **3.6× faster**.

Table 3: Late-turn diversity (%)

Method	Shift	Ret.	Mix	**Comp**
FIFO	50.0	66.7	44.4	47.6
Sink+W	100.0	66.7	66.7	73.3
H2O	100.0	50.0	66.7	70.0
FD-KVC	72.1	**100.0**	**73.1**	**80.6**

FD-KVC covers the **most topics** simultaneously (highest composite diversity).

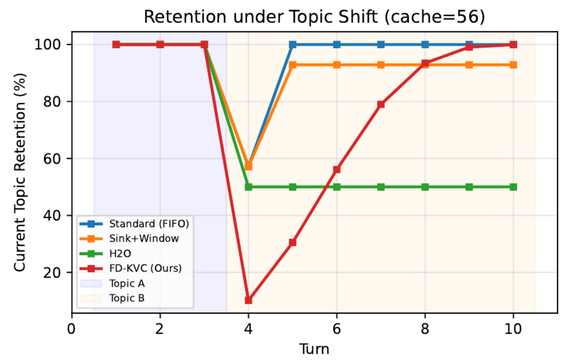

Fig. 3 — Retention under topic shift. H2O stagnates near 50%; FD-KVC recovers like FIFO but with semantic awareness.

Ablation Study

Table 5: w_c (Mixed)

w_c	LateAlign	Ret(%)
0.1	+.0223	77.6
0.3	**+.0483**	72.8
0.5	+.0238	60.2
0.9	+.0063	0.1

Table 6: γ (Mixed)

γ	LateAlign	Ret(%)
0.70	+.0233	85.7
0.80	**+.0603**	84.7
0.88	+.0148	64.1
0.92	−.0062	27.3

Peak alignment at w_c=0.3 and γ=0.80: too little cumulative weight loses long-term memory, too much resembles H2O. Both channels are necessary — removing either degrades performance.

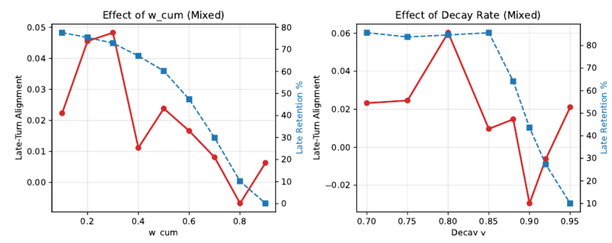

Fig. 4 — Ablation curves: alignment (red) vs. retention (blue dashed) for w_c and γ.