KV Packet: Recomputation-Free Context-Independent KV Caching for LLMs

Chuangtao Chen
Technical University of Munich
Munich, Germany
chuangtao.chen@tum.de

Grace Li Zhang
Technical University of Darmstadt
Darmstadt, Germany
grace.zhang@tu-darmstadt.de

Xunzhao Yin
Zhejiang University
Hangzhou, China
xzyin1@zju.edu.cn

Cheng Zhuo
Zhejiang University
Hangzhou, China
czhuo@zju.edu.cn

Bing Li
Technische Universität Ilmenau
Ilmenau, Germany
bing.li@tu-ilmenau.de

Ulf Schlichtmann
Technical University of Munich
Munich, Germany
ulf.schlichtmann@tum.de

Abstract

Large Language Models (LLMs) rely heavily on Key-Value (KV) caching to minimize inference latency. However, standard KV caches are context-dependent: reusing a cached document in a new context requires recomputing KV states to account for shifts in attention distribution. Existing solutions such as CacheBlend, EPIC, and SAM-KV mitigate this issue by selectively recomputing a subset of tokens; however, they still incur non-negligible computational overhead (FLOPs) and increased Time-to-First-Token (TTFT) latency. In this paper, we propose KV Packet, a recomputation-free cache reuse framework that treats cached documents as immutable "packets" wrapped in light-weight trainable soft-token adapters, which are trained via self-supervised distillation to bridge context discontinuities. Experiments on Llama-3.1 and Qwen-3 demonstrate that the proposed KV Packet method achieves near-zero FLOPs and lower TTFT than recomputation-based baselines, while retaining F1 scores comparable to those of the full recomputation baseline. Code and experimental results are available at https://github.com/ChuangtaoChen-TUM/KVPacket.

1 Introduction

Large Language Models (LLMs) (Grattafiori et al., 2024; Jiang et al., 2023; Liu et al., 2024; Wang et al., 2025) are increasingly deployed in Retrieval-Augmented Generation (RAG) systems, where a user query is answered by first retrieving a set of relevant documents from a knowledge base and then prompting the LLM with those documents as context. In production settings, the same documents are often retrieved repeatedly across many different user queries. A natural optimization is therefore to *precompute* and *cache* the Key-Value (KV) states of these documents offline, so that at inference time, the system can simply load the cached values rather than reprocessing the raw text from scratch. This avoids the expensive 'prefill' phase (Kwon et al., 2023; Dao et al., 2022; Lin et al., 2026), where input tokens are processed to generate the KV cache, and can substantially reduce Time-to-First-Token (TTFT) latency.

However, this optimization breaks down in practice due to the *contextual dependence* of KV caches. Because the KV states for each token are conditioned on its entire preceding context, a cache precomputed for a document in isolation is no longer valid when that document is placed alongside other retrieved chunks at inference time. Naively concatenating independently precomputed KV caches causes catastrophic performance degradation, since each cache encodes attention distributions that

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

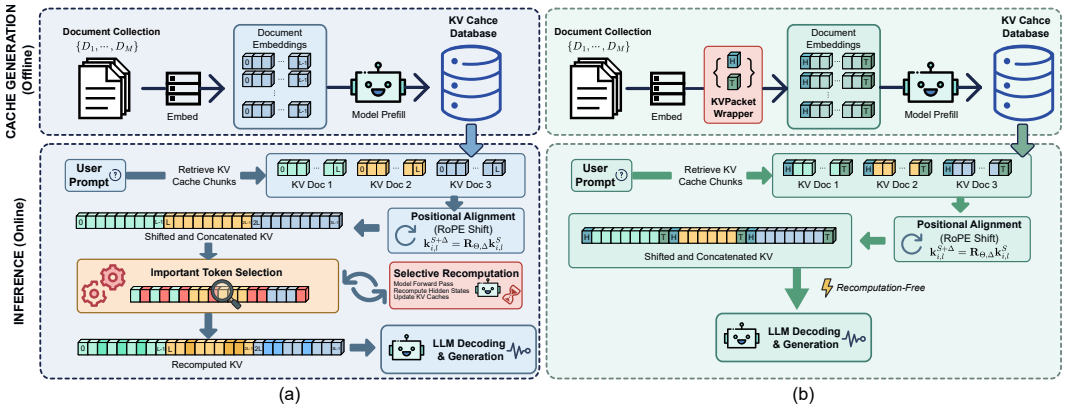

Figure 1: Comparison of KV cache reuse architectures. (a) Recomputation-based approaches require inference-time algorithms to select and recompute important tokens to repair contextual staleness; (b) the proposed KV Packet approach wraps documents with global adapters.

are oblivious to the surrounding documents. Consequently, the attention computation for each document is performed over an incomplete prefix, failing to capture the cross-document dependencies that would arise during full inference. Existing solutions fall into two categories: fine-tuning-based adaptation (KVLink (Yang et al., 2025c), Block-Attention (Ma et al., 2025), CacheClip (Yang et al., 2025b)), which risks degrading the base model's general capabilities, and inference-time selective recomputation (CacheBlend (Yao et al., 2025), EPIC (Hu et al., 2025), A3 (Zhou et al., 2025)), which reintroduces TTFT latency by performing partial forward passes at serving time. Neither approach achieves truly recomputation-free, plug-and-play cache reuse.

To address these limitations, we propose KV Packet, a framework that enables context-independent KV caching without model fine-tuning or inference-time recomputation. By wrapping document blocks with learned shared adapters, KV Packet ensures that precomputed caches are intrinsically compatible for direct concatenation. This decoupling enables seamless composition of retrieved documents at serving time with zero additional computational overhead.

The key contributions of this paper are as follows:

1. **Zero-Recomputation Architecture**: We introduce the KV Packet abstraction, which wraps frozen document caches with trainable soft-token adapters. This design eliminates the need for model parameter modifications and expensive cache recomputation during inference.

2. **Self-Supervised Alignment**: We propose a self-distillation training objective for the trainable fillers that aligns disjointed KV packets with full-attention representations. This approach requires no human-labeled data and restores contextual connectivity without modifying the base model's parameters.

3. **Efficiency and Performance**: We demonstrate that KV Packet reduces computational overhead (FLOPs) by approximately 4 orders of magnitude compared to recomputation-based methods like CacheBlend and EPIC, while maintaining competitive performance on various tasks.

4. **Compatibility with KV Compression**: As a recomputation-free architecture, KV Packet natively supports existing KV compression techniques seamlessly, while recomputation-based methods struggle with unstructured compression with varying token preservation across layers.

The remainder of this paper is organized as follows: Section 2 reviews KV caching mechanics and related work. Section 3 details the KV Packet architecture and self-supervised distillation objective. Section 4 evaluates performance, latency, and computational efficiency against recomputation-based baselines, including compatibility experiments and ablation studies. Section 5 summarizes our findings and discusses limitations and future directions.

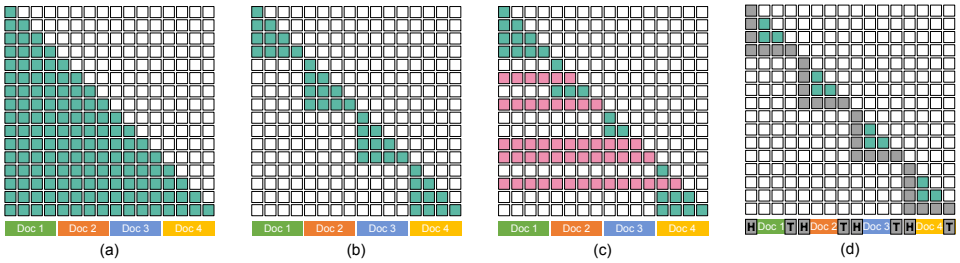

Figure 2: Comparison of mathematically equivalent attention maps. (a) Full Recomputation. (b) Naive Concatenation. (c) Selective Recomputation (dashed blocks for recomputed tokens). (d) KV Packet: Attention with lightweight adapters (**H**eader and **T**railer).

2 Background and Related Work

2.1 Key-Value Caching and Positional Encoding

Efficient LLM inference relies on a two-phase execution: a *prefill* stage that processes the input prompt in parallel to produce the Key-Value (KV) states, and an auto-regressive *decoding* stage that reuses cached states to avoid redundant computation (Vaswani et al., 2017). Industrial providers exploit this by caching recurring contexts across sessions (Cheng et al., 2025; Gim et al., 2024; OpenAI, 2024; Anthropic, 2024; Google, 2026).

However, KV caches are fragile due to **prefix dependency** and **positional dependency**: KV states are conditioned on both the full preceding context and absolute position indices, therefore, a precomputed cache is only valid at an identical position with an identical prefix. Even shifting a document by one token invalidates the entire cache. This sensitivity prevents the modular reuse of document caches in dynamic environments like multi-document RAG, necessitating expensive full recomputation for every unique configuration.

The positional dependency admits an efficient closed-form solution. Modern LLMs (*e.g.*, Llama 3 (Grattafiori et al., 2024), Mistral (Jiang et al., 2023), Qwen 3 (Yang et al., 2025a)) adopt Rotary Positional Embeddings (RoPE) (Su et al., 2024), which encode position by rotating Query and Key vectors by a position-dependent angle. Since the attention score between any two tokens depends only on their relative displacement, a precomputed Key state $\mathbf{k}_i^S$ can be efficiently realigned to a new position $s + \Delta$ via a single rotation (Su et al., 2024; Yang et al., 2025c):

$$\mathbf{k}_i^{S+\Delta} = \mathbf{R}_{\Theta,\Delta}\mathbf{k}_i^S, \quad \forall \Delta \in \mathbb{Z}, \tag{1}$$

where $\mathbf{R}_{\Theta,m}$ is the RoPE rotation matrix encoding offset Δ. This operation is element-wise and computationally negligible compared to a full forward pass, effectively resolving positional dependency at near-zero cost.

However, positional alignment alone is insufficient. The model still suffers from catastrophic performance degradation due to context dependency. Fig. 2 illustrates this phenomenon through the lens of attention maps. In an ideal full-prefill scenario (Fig. 2(a)), the model computes hidden states within a continuous, full attention map where every token has visibility of the entire preceding stream. In contrast, naive concatenation (Fig. 2(b)) creates isolated attention blocks. Because each document's KV cache was generated independently, its internal representations were computed using an incomplete attention map that lacks the semantic influence of the global prefix.

2.2 Context-Independent KV Caching

Model-Modification Approaches: To address the challenges of contextual dependency in cache reuse without the cost of a full prefill, several recent works have proposed extending context capabilities by updating model parameters or introducing auxiliary networks. KVLink (Yang et al., 2025c), for instance, introduces trainable link tokens coupled with explicit fine-tuning of the base model, while Block-Attention (Ma et al., 2025) modifies the attention mechanism itself to accommodate block-wise processing. Similarly, CacheClip (Yang et al., 2025b) relies on fine-tuning another language model to guide token selection.

Despite their effectiveness, we distinguish these fine-tuning and model-modification approaches from our work due to several practical drawbacks. Fine-tuning the base models or auxiliary supervisors requires massive computational resources. Furthermore, methods like CacheClip (Yang et al., 2025b) increase deployment complexity by requiring the simultaneous serving of multiple models, which significantly raises memory pressure. Most importantly, modifying the base model's weights can lead to catastrophic forgetting, potentially degrading the model's performance on general tasks.

Recomputation-Based Approaches: A complementary line of work addresses cache staleness through *selective recomputation*: recomputing only a critical subset of tokens at inference time to repair the missing prefix context, as illustrated in Fig. 2(c). CacheBlend (Yao et al., 2025) identifies high-deviation tokens across layers; A3 (Zhou et al., 2025) selects tokens by real-time query-document attention scores; EPIC (Hu et al., 2025) recomputes anchor tokens at document boundaries; and SAM-KV (Cao et al., 2026) applies hierarchical compression for multi-context scenarios. Despite their efficiency gains over full prefill, these methods share the following drawbacks:

- **Increased TTFT**: These methods must perform additional forward passes to repair the KV cache before generation, which can be further exacerbated by the complex token selection algorithms.
- **Computational overhead**: While recomputing only a subset of tokens is lighter than a full prefill, the cumulative FLOPs of these methods remain substantial.
- **Engineering burden**: These strategies are inherently invasive, requiring deep integration with the model's internal attention mechanisms and forward-pass logic. Consequently, adapting these methods to different model architectures requires extensive code modifications.

2.3 Soft-token Adapter Methods

Another line of work conceptually related to our method is parameter-efficient adaptation via learnable soft tokens prepended to the input. Prefix-tuning (Li & Liang, 2021) and prompt tuning (Lester et al., 2021) optimize continuous vectors that act as virtual prompt tokens while keeping the backbone LLM frozen, enabling efficient task conditioning without updating model weights. KV Packet similarly employs a small set of trainable tokens; however, instead of task adaptation from scratch, our goal is *cache composition*: training Header/Trailer tokens that absorb boundary artifacts and reconcile disjoint KV caches. Therefore, independently prefetched documents can be stitched together without inference-time recomputation.

3 Methods

When concatenating precomputed KV caches with RoPE realignment, the cached keys for each document remain contextually stale because they were computed without regard for the preceding documents. Prior recomputation-based methods address this by selectively recomputing a subset of tokens to restore cross-document attention. We hypothesize that a significant contributor to performance degradation in naive concatenation is not solely the absence of cross-document attention, but rather **boundary artifacts**, specifically, the disruption of attention sinks and abrupt shifts in token distributions at block boundaries, which interfere with the model's attention mechanism. This is partly motivated by the observation that retrieved documents in typical RAG settings are often semantically independent units, suggesting that the model's primary challenge at inference time is integrating information across structurally discontinuous boundaries rather than resolving deep semantic dependencies between documents. This perspective motivates wrapping each document cache with lightweight learned adapters that act as smooth delimiters at block boundaries, enabling the query to integrate information from independent cached packets without being disrupted by structural discontinuities.

3.1 Formulation of KV Packet

KV Packet Abstraction: We define a KV Packet $\mathcal{P}(D;\phi)$ as the KV cache of a frozen document D encapsulated with learnable boundary adapters $\phi = \{\mathbf{H}, \mathbf{T}\}$, where the header $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_{N_h}] \in \mathbb{R}^{N_h \times d}$ and the trailer $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_{N_t}] \in \mathbb{R}^{N_t \times d}$ are sequences of N_h and N_t continuous vectors, respectively. Denoting $\mathbf{e}_i$ as the embedding of the i-th document token, the packet representation for a single document is:

$$\mathcal{P}(D;\phi) = [\mathbf{h}_1, \ldots, \mathbf{h}_{N_h}, \mathbf{e}_1, \ldots, \mathbf{e}_L, \mathbf{t}_1, \ldots, \mathbf{t}_{N_t}] \tag{2}$$

At inference time, each document is independently wrapped into a packet according to Eq. (2) and its KV cache is precomputed offline. The resulting caches are then directly concatenated at serving time without any recomputation. The adapters ϕ are trained to act as shared smooth delimiters at block boundaries, such that the concatenated packets yield output quality comparable to full-attention inference, without requiring cross-document attention.

Fig. 1 shows the pipelines of recompute-based KV cache reuse methods and the proposed KV Packet approach. Recomputation-based methods (Fig. 1(a)) require an online token selection and partial forward pass step before generation, incurring significant FLOPs and latency. In comparison, as shown in Fig. 1(b), KV Packet confines all adapter-related work to the offline cache generation stage: document embeddings are wrapped with $\mathbf{H}$ and $\mathbf{T}$ at the embedding level prior to KV generation, requiring no architecture-specific implementation. At serving time, the recomputation step is eliminated entirely; only lightweight positional alignment and KV concatenation remain, effectively constructing global context without re-executing attention or FFN layers over document tokens. This results in an equivalent attention map illustrated in Fig. 2(d) where the Header and Trailer adapters serve as structural boundaries between document blocks, enabling full cross-document visibility for the query without recomputing any document tokens.

Beyond the serving-time efficiency gains described above, the parameters ϕ are global and document-independent: learned once on a representative text distribution and shared across all documents. New documents are wrapped with pre-trained adapters offline, incurring a constant $O(1)$ cost of a few kilobytes for the $N_h + N_t$ vectors, regardless of the size of the retrieval corpus. Furthermore, the total adapter length is very small relative to the typical document length; the additional KV cache storage and computation introduced by the adapters is also negligible.

3.2 Training via Self-Supervised Distillation

To enable the KV packets to function cohesively, we propose a self-supervised knowledge distillation framework to optimize the adapters $\phi = \{\mathbf{H}, \mathbf{T}\}$ to mimic the behavior of a model with full-context visibility, where the model $\mathcal{M}$ serves as its own teacher. The framework consists of two passes: a **reference pass**, in which the model processes the full context with standard causal attention to produce the target distribution; and a **student pass**, in which the model processes the same context using the KV Packet architecture to produce the approximated distribution. The adapters are then optimized by minimizing the divergence between the two distributions.

3.2.1 Reference Generation and Student Pass

1. **Input Construction**: We randomly sample a set of documents $\{D_1, \ldots, D_M\}$ and a corresponding query Q from the training corpus. These are concatenated to form the context: $X_{context} = [D_1, \cdots, D_M, Q]$.
2. **Auto-regressive Generation**: The model processes $X_{context}$ using standard causal attention, generating a continuation sequence $G = [g_1, \cdots, g_T]$ auto-regressively. The resulting token distribution $\mathbf{P}_{teacher} \in \mathbb{R}^{|G| \times |\mathcal{V}|}$ serves as the golden reference.
3. **Packet Construction**: Each document D_i is wrapped to form $\mathcal{P}(D_i;\phi)$ according to Eq. (2). Then, we construct the KV caches with adapters for all $\mathcal{P}(D_i;\phi)$ independently.
4. **Forward with KV Packets**: KV caches with adapters are realigned via Eq. (1) and then concatenated into a single KV cache. A forward pass is performed on $[Q, G]$ given the KV cache with causal attention, where $[Q, G]$ can attend to all previous tokens. The probability distribution for G is recorded as $\mathbf{P}_{student} \in \mathbb{R}^{|G| \times |\mathcal{V}|}$.
5. **Loss Calculation**: We minimize the Kullback-Leibler Divergence between the student's output distribution and the teacher's reference distribution:

$$\mathcal{L} = \frac{1}{|G|} \sum_{t=1}^{|G|} D_{KL}\left(\mathbf{P}^t_{teacher} \parallel \mathbf{P}^t_{student}\right),$$

where $\mathbf{P}^t$ represents the probability distribution to predict token g_t in the generated sequence.

Utilizing the model's own full-attention generation as a teacher signal offers several advantages over standard fine-tuning. Since only the small adapter tensors receive gradients while the base model $\mathcal{M}$ and all document embeddings remain frozen, memory and FLOPs are kept significantly below those of fine-tuning methods, and catastrophic forgetting is bypassed by design. Furthermore, no human-annotated data is required: adapters can be trained on diverse unlabeled corpora by treating random text segments as independent documents, and distilling the model's own distribution ensures that the adapters learn only the structural patterns needed to recover full-prefill information flow, rather than memorizing external facts.

4 Experiments

4.1 Experimental Setup

We evaluate KV Packet on four datasets spanning two task types: simple information retrieval (Needle-in-a-Haystack (Jegou, 2024), Biography (Karev, 2025)) and multi-step reasoning (HotpotQA (Yang et al., 2018), MusiQue (Trivedi et al., 2022)). All experiments use a single NVIDIA A100 (80 GB) GPU with 8 adapter tokens ($N_h = N_t = 8$), float32 adapters, and models in bfloat16. Adapters are trained for 30 epochs with AdamW, a linear decay schedule, and 256–512 training samples (batch size 64); learning rates are 5×10^{-4} for Biography, NIAH, and MusiQue, and 1×10^{-3} for HotpotQA. Cached KV tensors are stored in CPU memory and loaded to the GPU upon retrieval.

We benchmarked against a comprehensive suite of baselines: Full Recompute (upper bound), No Recompute (lower bound, RoPE realignment only), No Cache (no documents, control baseline using only model knowledge), and the following recomputation-based methods: CacheBlend (Yao et al., 2025), A3 (Zhou et al., 2025), and Random recomputation with recomputation ratios: 10%, 30%, 50%, 70%, and 90% of document tokens; EPIC (Hu et al., 2025) with 10, 30, 50, 70, and 90 recomputed tokens (and 100–500 for NIAH). SAM-KV (Cao et al., 2026) evaluated under two configurations: small: block/initial/local tokens = 8/8/16; large: 64/64/128. The configurations listed above correspond to the left-to-right order of data points in Fig. 3.

4.2 Efficiency Metrics

FLOPs measure operations required to align and prepare retrieved caches before generation: for KV Packet, this is solely the RoPE position realignment; for recompute methods, it additionally includes the partial forward pass over selected tokens. **TTFT** is the end-to-end wall-clock latency from query receipt to first token, including cache transfer from CPU to GPU, RoPE shifts or recomputation, and query processing. Offline cache generation costs are excluded from inference metrics.

4.3 Results and Analysis

We present the performance of Llama-3.1-8B-Instruct and Qwen-3-4B-Instruct on the evaluated datasets in Fig. 3.

Generation Quality: For most configurations, KV Packet achieves comparable F1 on information retrieval tasks (Needle-in-a-Haystack, Biography) and multi-step reasoning (HotpotQA, MusiQue), substantially outperforming No Recompute in all settings. Recomputation methods such as EPIC and CacheBlend struggle at low recompute ratios, particularly in long-context settings. For the Qwen model on the MusiQue dataset, the KV Packet method demonstrates a performance gap when compared with the Full Recompute baseline, but still maintains a favorable Pareto trade-off when considering the FLOPs and TTFT.

Computational Efficiency: With the recomputation-free nature of the proposed method, we achieve significant reductions compared with the Full Recompute baseline, with FLOPs by 5–6 orders of magnitude lower (6.50×10^{-6} to 1.04×10^{-5}), matching the No Recompute baseline. As shown in Fig. 3, the KV Packet is located at the upper-left of the F1 score vs. FLOPs plot, achieving high generation quality while retaining low computational cost.

Latency (TTFT): KV Packet consistently delivers one of the lowest Time-to-First-Token (TTFT) values, similar to No Recompute and only slightly higher than No Cache (which lacks the necessary context). For experiments on the Llama model, the KV Packet method achieves $1.36\times$ and $3.3\times$

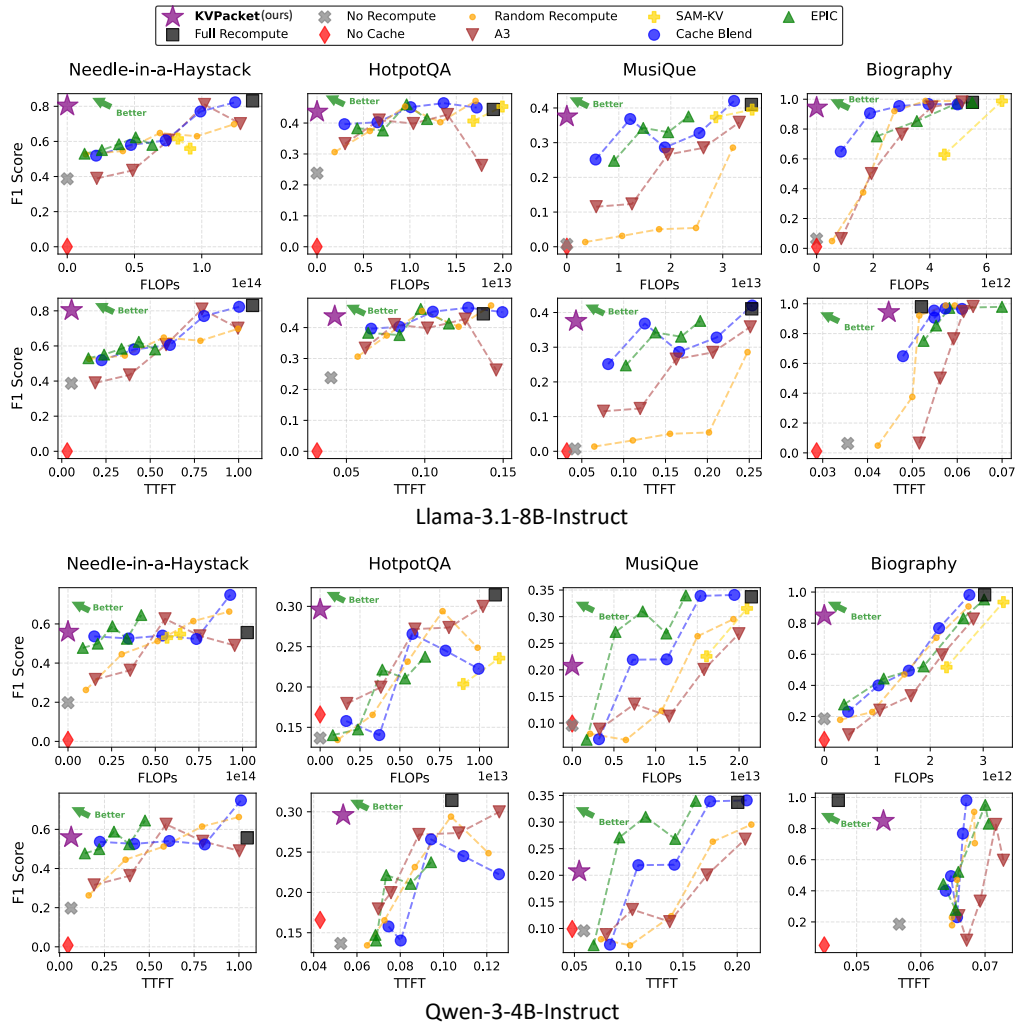

Figure 3: Evaluation results (F1 score, FLOPs, Time-to-First-Token in seconds) of Llama-3.1-8B / Qwen-3-4B on datasets: Needle-in-a-Haystack, Biography, HotpotQA, and MusiQue.

speedup on the Biography and HotpotQA tasks, respectively. The TTFT reduction is especially significant in long-context settings: we observe a $19.45\times$ reduction in TTFT on Needle-in-a-Haystack and a $5.81\times$ reduction on MusiQue. As shown in the F1 score vs. TTFT plots in Fig. 3, KV Packet occupies the upper-left region, indicating high generation quality at low latency.

Overall, the experimental results validate KV Packet as an effective alternative to recomputation-based strategies, recovering much of the same base model's Full Recompute accuracy on both extraction and reasoning tasks while eliminating the recomputation overhead entirely at inference time. A direct comparison with KVLink (Yang et al., 2025c) is reported in Appendix A.5.

4.4 Compatibility with KV Compression

Recomputation-based methods face significant practical challenges when combined with modern KV compression: partial forward passes require complete, contiguous caches with known positional indices. However, compression algorithms produce unstructured caches that preserve different token subsets per layer, breaking RoPE realignment and selective attention assumptions. Furthermore, current compression algorithms are strictly optimized for auto-regressive generation rather than historical KV recomputation, and their efficacy and numerical stability in recomputation settings remain unverified. KV Packet natively bypasses these bottlenecks entirely by treating each document

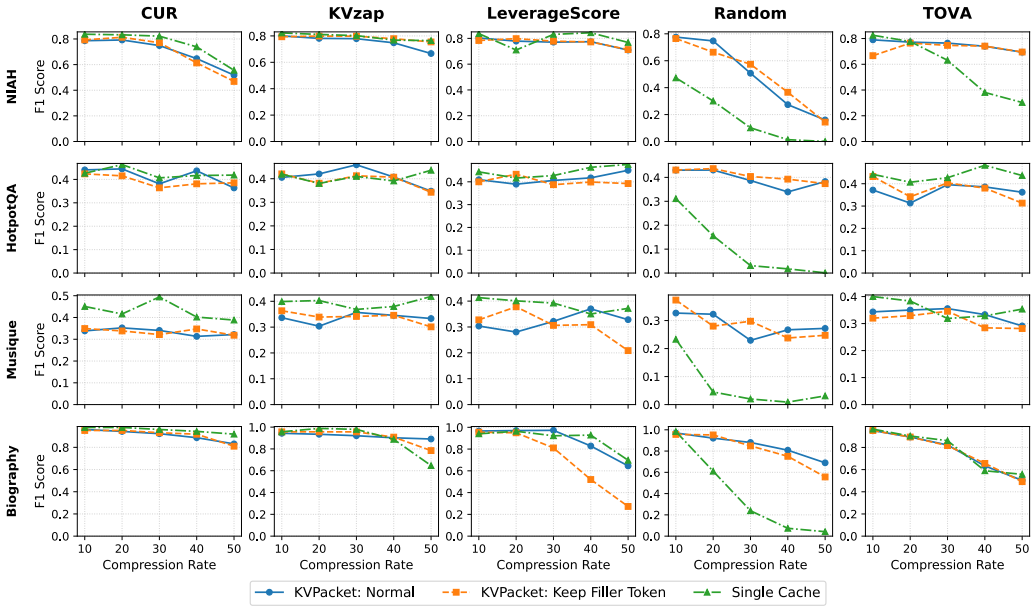

Figure 4: F1 Score vs. Compression rate of Llama-3.1-8B-Instruct model on four datasets with different compression methods

cache as an opaque unit and never re-evaluating compressed hidden states; it integrates seamlessly with off-the-shelf unstructured KV pruning.

Fig. 4 evaluates Llama-3.1-8B-Instruct under five compression methods (CUR (Sengupta et al., 2025), KVzap (Jegou & Jeblick, 2026), LeverageScore (Chari & Durme, 2025), TOVA (Oren et al., 2024), and random pruning) from the KVPress library (Devoto et al., 2025), at compression rates of 10%–50%, under three configurations: **KVPacket Normal** (compression over full wrapped cache), **KVPacket Keep Filler** (filler tokens excluded from pruning), and **Single Cache** (compression over Full Recompute cache). Notably, KV Packet proves significantly more robust than the baseline under random pruning, maintaining a much flatter performance profile. Furthermore, the KVPacket Normal setting generally outperforms the Keep Filler Token variant. This suggests that allowing the compression algorithm a more flexible pruning strategy over the entire wrapped cache is beneficial and indicates that our trained fillers are inherently resilient to KV compression.

Table 1: Cross-domain generalization analysis. Rows represent the dataset used to train the KV Packet adapters for Llama-3.1-8B-Instruct, while columns indicate the evaluation performance (F1 score) across different domains.

	Evaluation Dataset			
Training Dataset	Biography	HotpotQA	MusiQue	NIAH
No Recompute	0.06	0.24	0.01	0.39
Biography	**0.96**	0.18	0.16	0.44
HotpotQA	0.88	**0.44**	0.39	0.36
MusiQue	0.87	0.36	0.37	0.34
Needle-in-a-Haystack (NIAH)	0.54	0.28	0.07	**0.80**
Cross-domain (Mixture)	**0.95**	**0.42**	**0.43**	0.71

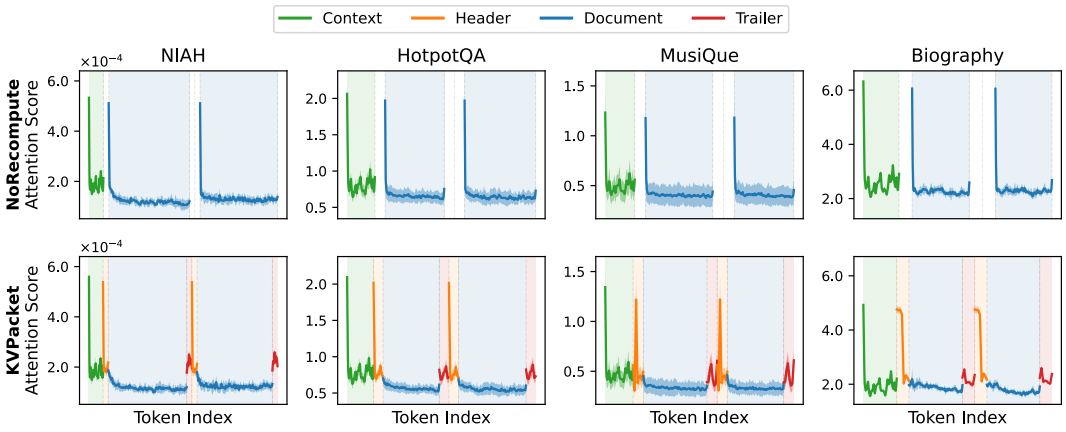

Figure 5: Query-to-context attention scores of the No Recompute and KV Packet methods. The attention scores are averaged by all layers, all query tokens, and 50 randomly selected samples from each dataset. The token positions for the No Recompute method in the figure are adjusted to align with those of the KV Packet method with filler tokens.

4.5 Ablation Study

4.5.1 Cross-Domain Generalization and Mixture Training

To evaluate the transferability and robustness of our learnable adapters, we conducted a cross-domain experiment in which KV Packets were trained on a specific source domain and evaluated across all other target domains. As detailed in Table 1, the No Recompute baseline fails consistently across all benchmarks (0.01–0.39), while adapters trained on specialized tasks like Biography achieve near-perfect in-domain performance (0.96), but struggle with the logical complexity of multi-hop reasoning tasks such as HotpotQA (0.18) and MusiQue (0.16). Conversely, the NIAH-trained adapter—while highly effective at synthetic retrieval (0.80)—shows limited generalization to natural-language nuances. The cross-domain adapter, trained on a diverse mixture of all four datasets, performs well across the evaluated domains. It effectively bridges the gap between specialized domains, approaching the performance of the Biography-specific model (0.95) and outperforming the single-domain adapters on the MusiQue reasoning benchmark (0.43). This suggests that heterogeneous training can teach cache-stitching patterns that transfer across related tasks. A cross-domain adapter requires representative unlabeled training traffic and validation of its task mixture; its applicability to arbitrary unseen domains is not established.

4.5.2 Attention Dynamics with KV Packet

To understand how KV Packet resolves contextual dependencies, we analyze the attention distributions of the Llama model across various datasets. We measure the averaged attention scores of a query token directed toward its preceding context, comparing the naive cache concatenation baseline (NoRecompute) with our KV Packet approach. Fig. 5 visualizes these scores by context: the task prompt, the document tokens, and the learned adapters. Since the adapters are absent in the baseline, we manually shifted the token positions to align with the KV Packet's positions. Under naive concatenation, sharp attention spikes emerge at the start of every document block, because the model perceives each cached document's first token as a sequence-initial token, triggering the attention sink effect (Xiao et al., 2024) at mid-sequence positions and fragmenting global reasoning.

KV Packet resolves this: the Headers and Trailers absorb the sink mass that would otherwise concentrate on document tokens, as evidenced by the higher attention scores on adapter tokens than on document tokens. This redirects attention away from non-semantic boundaries and restores a smooth distribution over document content. These observations explain the strong empirical performance of even a small number of adapter tokens and strongly validate our boundary artifact hypothesis.

5 Conclusion

We presented KV Packet, a recomputation-free framework for context-independent KV caching. By wrapping immutable document caches with lightweight trainable Headers and Trailers, we address boundary artifacts that arise from naive cache concatenation, as validated by attention score analysis. Our self-supervised distillation objective trains these adapters without human-labeled data, keeping base model weights frozen. KV Packet integrates seamlessly with existing KV compression techniques, a compatibility that recomputation-based methods fundamentally cannot match, and reduces inference-time FLOPs by 5–6 orders of magnitude while recovering much of the same base model's Full Recompute accuracy on retrieval and reasoning benchmarks.

Limitations and Future Work

1. Adapter effectiveness assumes the retrieval corpus is reasonably aligned with the training distribution; generalization to highly out-of-distribution domains remains an open question.
2. Current evaluations cover only a limited set of model families due to the engineering complexity of validating recomputation baselines for each architecture.
3. Highly fragmented retrieval can limit absolute multi-hop accuracy. Grouping documents into larger packets mitigates this in the controlled MuSiQue experiment, but packet granularity remains an important cache-construction parameter (Appendix A.9).
4. Long-context performance remains limited: raw recovery decreases from 91.5% at 8k/4 packets to 76.5% at 32k/16 packets. Context length and packet count increase together, and we have not evaluated 128k contexts (Appendix A.7).
5. Long-form generation retains a quality gap relative to Full Recompute. Although ROUGE and reference-summary perplexity are close, order-balanced pairwise judgments favor Full Recompute (Appendix A.8).

Acknowledgments and Disclosure of Funding

This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 504518248 and in part by the European Union - European Research Council (ERC) Starting Grant -Project-ID 101219243. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

References

Anthropic. Prompt Caching. `https://docs.anthropic.com/en/docs/build-with-claude/prompt-caching`, 2024. Accessed: 2026-01-21.

Ziyi Cao, Qingyi Si, Jingbin Zhang, and Bingquan Liu. Sparse Attention across Multiple-context KV Cache. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30165–30173, 2026.

Vivek Chari and Benjamin Van Durme. Compactor: Calibrated query-agnostic kv cache compression with approximate leverage scores. *arXiv preprint arXiv:2507.08143*, 2025.

Yihua Cheng, Yuhan Liu, Jiayi Yao, et al. LMCache: An Efficient KV Cache Layer for Enterprise-Scale LLM Inference. *arXiv preprint arXiv:2510.09665*, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL `https://arxiv.org/abs/2110.14168`.

Tri Dao, Dan Fu, Stefano Ermon, et al. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.

Alessio Devoto, Maximilian Jeblick, and Simon Jégou. Expected Attention: KV Cache Compression by Estimating Attention from Future Queries Distribution. *arXiv preprint arXiv:2510.00636*, 2025.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. Multi-News: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1074–1084. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1102. URL https://aclanthology.org/P19-1102/.

In Gim, Guojun Chen, Seung-seob Lee, et al. Prompt Cache: Modular Attention Reuse for Low-Latency Inference. *Proceedings of Machine Learning and Systems*, 6:325–338, 2024.

Google. Context Caching | Gemini API. https://ai.google.dev/gemini-api/docs/caching, 2026. Last updated: 2026-01-16. Accessed: 2026-01-21.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024.

Junhao Hu, Wenrui Huang, Weidong Wang, et al. EPIC: Efficient Position-Independent Caching for Serving Large Language Models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 24391–24402, 2025.

Simon Jegou. Needle-in-a-Haystack: Ruler. https://huggingface.co/datasets/simonjegou/ruler, 2024.

Simon Jegou and Maximilian Jeblick. KVzap: Fast, Adaptive, and Faithful KV Cache Pruning. *arXiv preprint arXiv:2601.07891*, 2026.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Alex Karev. Synthetic Biographies. https://huggingface.co/datasets/alex-karev/biographies, 2025.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, et al. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Brian Lester, Rami Al-Rfou, and Noah Constant. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021.

Xiaolin Lin, Jingcun Wang, Olga Kondrateva, Yiyu Shi, Bing Li, and Grace Li Zhang. CompressKV: Semantic-retrieval-guided KV-cache compression for resource-efficient long-context LLM inference, 2026. URL https://arxiv.org/abs/2606.24467.

Aixin Liu, Bei Feng, Bing Xue, et al. Deepseek-v3 Technical Report. *arXiv preprint arXiv:2412.19437*, 2024.

Dongyang Ma, Yan Wang, and Tian Lan. Block-Attention for Efficient Prefilling. *arXiv preprint arXiv:2409.15355*, 2025.

OpenAI. Prompt Caching. https://platform.openai.com/docs/guides/prompt-caching, 2024. Accessed: 2026-01-21.

Matanel Oren, Michael Hassid, Nir Yarden, Yossi Adi, and Roy Schwartz. Transformers are Multi-State RNNs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18724–18741, 2024.

Ayan Sengupta, Siddhant Chaudhary, and Tanmoy Chakraborty. Value-Guided KV Compression for LLMs via Approximated CUR Decomposition. *arXiv preprint arXiv:2509.15038*, 2025.

Jianlin Su, Murtadha Ahmed, Yu Lu, et al. Roformer: Enhanced Transformer with Rotary Position Embedding. *Neurocomputing*, 568:127063, 2024.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multi-hop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics*, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention Is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6000–6010, 2017.

Jingcun Wang, Yu-Guang Chen, Ing-Chao Lin, Bing Li, and Grace Li Zhang. Basis sharing: Cross-layer parameter sharing for large language model compression. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, et al. Efficient Streaming Language Models with Attention Sinks. *arXiv preprint arXiv:2309.17453*, 2024.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025a.

Bin Yang, Qiuyu Leng, Jun Zeng, and Zhenhua Wu. CacheClip: Accelerating RAG with Effective KV Cache Reuse. *arXiv preprint arXiv:2510.10129*, 2025b.

Jingbo Yang, Bairu Hou, Wei Wei, et al. KVLink: Accelerating Large Language Models via Efficient KV Cache Reuse. *arXiv preprint arXiv:2502.16002*, 2025c.

Zhilin Yang, Peng Qi, Saizheng Zhang, et al. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Jiayi Yao, Hanchen Li, Yuhan Liu, et al. CacheBlend: Fast Large Language Model Serving for RAG with Cached Knowledge Fusion. *arXiv preprint arXiv:2405.16444*, 2025.

Yuechi Zhou, Yi Su, Jianxin Zhang, et al. A^3: Attention-Aware Accurate KV Cache Fusion for Fast Large Language Model Serving. *arXiv preprint arXiv:2511.17560*, 2025.

A Appendix

Table 2: Ablation study on adapter token choice. Rows represent different KV Packet configurations for Llama-3.1-8B-Instruct, including a no-adapter baseline (No Recompute), fixed-token handcrafted adapters, and the trained adapter. Columns indicate evaluation performance (F1 score) across datasets.

	Evaluation Dataset			
KV Packet Configuration	Biography	HotpotQA	MusiQue	NIAH
No Recompute	0.06	0.24	0.01	0.39
`<|begin_of_text|>`	0.01	0.00	0.00	0.17
`.`	0.20	0.21	0.02	0.29
`<eot_id>`	0.25	0.22	0.03	0.33
`\n\n`	0.22	**0.32**	0.03	0.29
`" "`	0.21	0.24	0.08	0.28
`\n`	**0.29**	0.24	0.07	0.33
`skip`	0.15	0.11	**0.11**	0.27
`document`	0.15	0.12	**0.11**	0.26
KV Packet (trained)	0.94	0.44	0.38	0.80

A.1 Ablation Study: Handcrafted vs. Trained Adapters

To validate the importance of the proposed training objective, we compare KV Packet (trained) against a set of handcrafted adapters constructed by repeating a single fixed token from the model's vocabulary. Results are reported in Table 2.

Handcrafted adapters provide inconsistent improvements over the No Recompute baseline. For instance, the single escape configuration `\n` achieves an F1 of 0.29 on Biography and 0.33 on NIAH, outperforming No Recompute (0.06 and 0.39 respectively) on the former but not on the latter. Similarly, double escape `\n\n` improves HotpotQA from 0.24 to 0.32, yet yields no gain on MusiQue. No single token consistently outperforms others across all four datasets, and some tokens, notably `<|begin_of_text|>`, severely degrade performance to near zero on HotpotQA and MusiQue. This instability suggests that the choice of delimiter token is highly dataset-sensitive and cannot be reliably determined without exhaustive search.

A deeper issue is that discrete vocabulary tokens carry inherent semantic and syntactic associations learned during pretraining. When repeated as boundary adapters, these tokens cannot function as purely structural delimiters — the model's attention mechanism responds to their semantic content in addition to their positional role, interfering with the document tokens rather than cleanly absorbing boundary artifacts. In contrast, the trained continuous adapter vectors are optimized directly to absorb attention sinks and smooth token distribution shifts at block boundaries, without any semantic baggage. This is reflected in the results: KV Packet (trained) achieves F1 scores of 0.94, 0.44, 0.38, and 0.80 on Biography, HotpotQA, MusiQue, and NIAH respectively, outperforming the best handcrafted token on every dataset by a substantial margin — for example, exceeding the best handcrafted result on Biography by 0.65 and on MusiQue by 0.27.

A.2 Parallel Training Implementation

The sequential training procedure described in Section 3 — which involves independent KV generation per document, concatenation, and a separate forward pass — can be unified into a single forward pass for efficient batch training. We describe this equivalent parallel implementation here.
Block-Diagonal Attention Mask. The key insight is that the isolated caching environment of the sequential procedure can be exactly simulated by a carefully designed attention mask. As illustrated in Fig. 6, we process the complete sequence $X_{total} = [\mathbf{H}, D_1, \mathbf{T}, \mathbf{H}, D_2, \mathbf{T}, \ldots, \mathbf{H}, D_M, \mathbf{T}, Q, G]$ in a single forward pass using a block-diagonal attention mask. Specifically, the mask enforces the following attention rules:

- Each document block $[\mathbf{H}, D_i, \mathbf{T}]$ attends only within itself, simulating independent offline KV cache generation.
- The query Q and generation tokens G attend to all preceding tokens, including all document packets, consistent with the full-context assumption at inference time.

Loss Calculation. Only the logits corresponding to the generation tokens G are used to compute the KL divergence loss against the teacher reference distribution $\mathbf{P}_{teacher}$, which is obtained from a standard causal forward pass over X_{total}. The hatched region in Fig. 6 denotes logits that are masked out and excluded from the loss computation.

Mathematical Equivalence. This parallel formulation is mathematically equivalent to the sequential procedure: the block-diagonal mask ensures that the KV states of each document block are computed in isolation, identical to what would be produced by independent caching. The query and generation tokens then attend across all packets via the concatenated KV cache, replicating the inference-time composition. This equivalence means the two implementations are interchangeable, and the parallel version is adopted for training due to its significantly higher throughput.

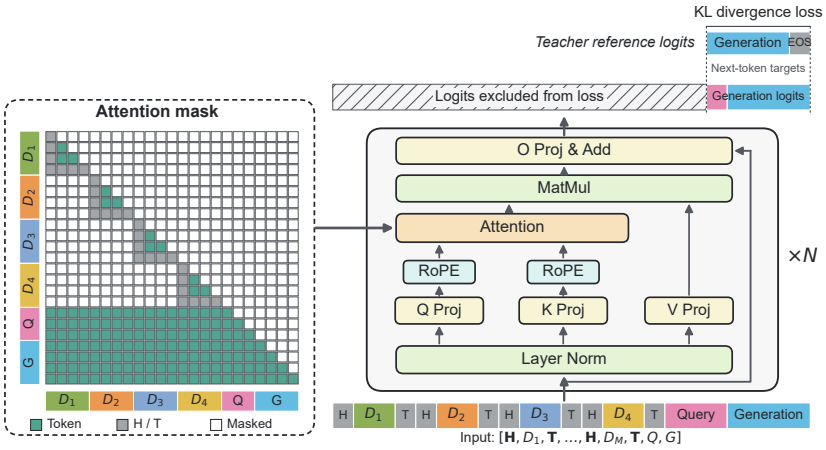

Figure 6: Parallel training via block-diagonal masking. *Left*: The attention mask used during training. Document blocks attend only within themselves (block-diagonal structure), while the query and generation tokens attend to all preceding tokens (full causal attention). Hatched cells indicate logits excluded from the loss computation. *Right*: The transformer layer processes the complete sequence in a single forward pass. The KL divergence loss is computed only between the generation logits and the teacher reference distribution.

A.3 Training Objective

We compare two training objectives for the adapter: KL divergence against the teacher distribution ($\mathcal{L}_{KL}$) and standard cross-entropy against the teacher labels ($\mathcal{L}_{CE}$). As shown in Table 3, the two objectives perform comparably on retrieval tasks: on Biography, label training achieves 0.979 versus 0.943 for logits, and on NIAH, label training achieves 0.833 versus 0.803. However, on reasoning tasks that require generating longer sequences, the gap widens considerably: on MusiQue, logits training achieves 0.375 compared to only 0.226 for label training, and on HotpotQA the difference is smaller but consistent (0.436 vs. 0.430). We attribute this discrepancy to the nature of the two tasks: retrieval tasks require generating short, specific answers where the label signal is sufficient, whereas reasoning tasks require generating longer sequences where the full output distribution carries richer information. Since our objective is to align the adapter's output with the model's own full-attention behavior rather than with external ground truth, the logits-based objective better preserves the fine-grained distributional details necessary for multi-step reasoning, and is therefore adopted as the default training objective. Nevertheless, the precise mechanism underlying this difference, and whether the optimal choice of training objective varies across task types or model families, remains unclear and we leave a more thorough investigation for future work.

Table 3: Ablation study on training objective. F1 scores are reported for KL divergence against teacher logits ($\mathcal{L}_{KL}$) and cross-entropy against teacher-generated labels ($\mathcal{L}_{CE}$) across four datasets.

	Evaluation Dataset			
Training Objective	Biography	HotpotQA	MusiQue	NIAH
$\mathcal{L}_{KL}$ (Logits)	0.943	**0.436**	**0.375**	0.803
$\mathcal{L}_{CE}$ (Label)	**0.979**	0.430	0.226	**0.833**

A.4 Sensitivity Analysis and Storage Overhead

We conduct a sensitivity analysis on the NIAH dataset using Llama-3.1-8B-Instruct to study the effect of document length and adapter length on KV Packet's performance and **storage overhead**. We fix the total context length at $8k$ tokens and vary the document length over {256, 512, 1024, 2048} tokens and the adapter length over {4+4, 8+8, 16+16} tokens, training each configuration for 20 epochs. Results are shown in Fig. 7.

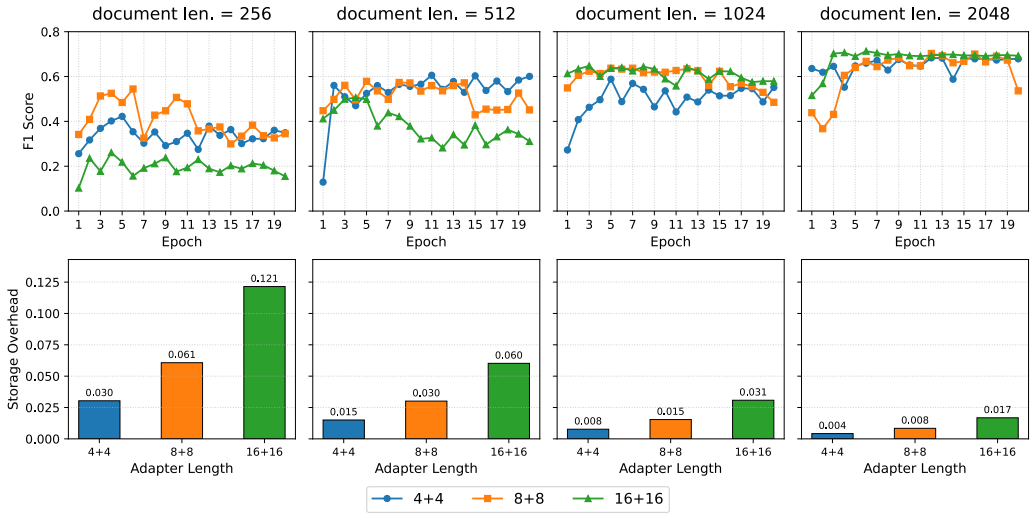

Figure 7: Sensitivity analysis of KV Packet on NIAH (Llama-3.1-8B-Instruct, 8k context). *Top row*: F1 score vs. training epoch for each document length and adapter length configuration. *Bottom row*: storage overhead introduced by each adapter length configuration at each document length.

Convergence speed Across nearly all configurations, the F1 score stabilizes within 3-5 epochs and shows no consistent improvement beyond that point, indicating that the 20-epoch training budget is redundant. This suggests that KV Packet adapters can be trained efficiently with a small number of gradient steps, which is favorable for practical deployment.

Storage overhead The storage overhead introduced by KV Packet is determined jointly by the adapter length and the document length, and remains small in realistic settings. The worst case — document length 256 with adapter length 16+16 — incurs an overhead of 12.1%, but this configuration is also the least performant (see below). For document lengths of 512 or above, which better reflect real-world RAG scenarios, the overhead ranges from 0.4% to 6.0%, which is negligible relative to the KV cache storage of the documents themselves.

Adapter length sensitivity Longer adapters do not always yield better performance. For document lengths of 256 and 512, the 16+16 configuration performs consistently worse than 4+4 and 8+8. We attribute this to the fact that shorter documents result in a larger number of retrieved chunks within the fixed $8k$ context (up to 32 documents for 256-token chunks), and optimizing a large number of adapter tokens jointly across many boundaries makes the training objective harder to satisfy. Better hyperparameter tuning could potentially mitigate this issue, and we leave this to future work.

Effect of document count There is a consistent performance degradation as the number of retrieved documents increases. Across all adapter configurations, average F1 scores are lower for shorter document lengths, since the fixed $8k$ context accommodates more documents (32 for 256-token

chunks versus 4 for 2048-token chunks), increasing the number of block boundaries that the adapters must bridge simultaneously. While retrieving 32 documents in a single query is uncommon in practice, we acknowledge this as a current limitation of KV Packet and leave improving performance under high document counts as a direction for future work.

A.5 Direct Comparison with KVLink

We compare stock Llama-3.2-3B with KV Packet against the released Llama3B-KVLink5 checkpoint (Yang et al., 2025c). Both methods use the same 100 seed-42 examples per dataset, a shared prompt, greedy decoding, BF16 eager attention, and one A100 80 GB. KVLink fine-tunes the backbone, whereas KV Packet keeps it frozen, so their Full Recompute references differ. Table 4 reports absolute F1 together with raw recovery, defined as the cache-reuse F1 divided by Full Recompute F1 for the same checkpoint. Recovery is higher for KV Packet on all four datasets; this is a comparison of accuracy preservation, not an assumption that the checkpoints have identical capabilities.

Table 4: KVLink comparison. Full denotes recomputation using each method's own checkpoint. Recovery is reported as a percentage; the mean averages the four dataset recovery values.

	KV Packet (stock)			KVLink-5 (fine-tuned)		
Dataset	F1	Full	Recovery	F1	Full	Recovery
Biography	0.8044	0.8811	91.3%	0.8811	0.9744	90.4%
NIAH	0.7474	0.8088	92.4%	0.6589	0.7628	86.4%
HotpotQA	0.3336	0.4289	77.8%	0.1320	0.2977	44.3%
MuSiQue	0.1340	0.2797	47.9%	0.0702	0.2727	25.7%
Mean	–	–	77.4%	–	–	61.7%

For serving efficiency, the common preamble and document caches are prepared offline for both methods. KVLink's links attend to preceding segments, so their context-dependent states must be computed online. We evaluate both sequential KVLink and a fused implementation, verifying first-token equivalence on all evaluated examples. Table 5 reports total online FLOPs, including query forward computation; these totals differ from the cache-preparation-only FLOPs in Fig. 3. KV Packet uses 44.2% fewer online FLOPs on HotpotQA and 57.9% fewer on MuSiQue than fused KVLink. Fused TTFT is comparable, with no consistent latency advantage for either method.

Table 5: KVLink serving comparison. TTFT is in milliseconds and online FLOPs in trillions (10^{12}), including query forward computation.

Dataset	Method	TTFT (ms)	Online FLOPs (T)
HotpotQA	KV Packet	90.03	0.3900
HotpotQA	KVLink sequential	340.09	0.6870
HotpotQA	KVLink fused	93.73	0.6983
MuSiQue	KV Packet	173.03	0.4732
MuSiQue	KVLink sequential	635.10	1.0837
MuSiQue	KVLink fused	167.85	1.1252

The methods also differ in supervision and placement: KVLink uses labeled or GPT-4-generated targets and trailer-side link tokens, while KV Packet self-distills the frozen model's logits with Header and Trailer adapters. Consequently, this comparison does not isolate the effect of placement, supervision, or backbone treatment individually. We have not run a decisive header-versus-trailer ablation. Block-Attention and CacheClip were not reproduced in this comparison: we prioritized KVLink as the closest learning-based baseline with released checkpoints, rather than claiming coverage of every related method.

A.6 Expanded Mechanistic Decomposition

We expand the HotpotQA decomposition to 2,000 unique examples using Llama-3.1-8B-Instruct, with matched examples across four conditions. No Recompute concatenates isolated document caches without adapters; KV Packet adds trained H/T while retaining isolated document encoding;

the hybrid restores full cross-document attention with the trained H/T still present; Full Recompute uses ordinary full attention without H/T. Table 6 reports pooled token-count Micro F1, example-averaged Macro F1, and exact match (EM).

Table 6: HotpotQA decomposition on 2,000 unique examples using Llama-3.1-8B-Instruct. The hybrid retains H/T and restores full cross-document attention.

Condition	Micro F1	Macro F1	EM
No Recompute	0.207	0.215	0.156
KV Packet	0.382	0.423	0.309
Full Attention + H/T	0.428	0.443	0.328
Full Recompute	0.408	0.460	0.344

We compute paired-bootstrap 95% confidence intervals for the Macro F1 differences between matched conditions (Table 7). KV Packet improves Macro F1 over No Recompute by 0.2089 and recovers 84.9% of the Full-Recompute-over-No-Recompute improvement. Here, recovered improvement is $(F_{\mathrm{KV}} - F_{\mathrm{No}})/(F_{\mathrm{Full}} - F_{\mathrm{No}})$, distinct from raw recovery $F_{\mathrm{KV}}/F_{\mathrm{Full}}$. Restoring cross-document attention adds 0.0198 Macro F1, with a positive confidence interval. Boundary repair therefore accounts for most of the recovered performance in this experiment, while cross-document attention has a smaller, statistically resolved positive contribution. Estimates are reported separately from the rounded scores in Table 6.

Table 7: Paired-bootstrap uncertainty for Macro F1 differences in the expanded HotpotQA decomposition.

Comparison	Estimate	95% CI
KV Packet − No Recompute	+0.2089	$[0.1888,\ 0.2293]$
Full Attention + H/T − KV Packet	+0.0198	$[0.0035,\ 0.0362]$
Full Recompute − Full Attention + H/T	+0.0173	$[0.0013,\ 0.0340]$

To quantify evaluation-set variability at a 100-example test size, we partition the 2,000-example pool into 20 non-overlapping subsets. Mean raw recovery across these subsets is $92.5\% \pm 10.4$ percentage points (mean $\pm$ sample standard deviation), with a reported 95% confidence interval of $[88.2\%, 97.1\%]$ for mean recovery. This subset analysis measures evaluation-set variability; it is distinct from independent adapter training runs and from the paired-bootstrap differences above.

A.7 Long-Context Scaling

On NIAH, we reuse an 8k-trained Llama-3.1-8B adapter without retraining, keep packet length near 2k tokens, and increase packet count from four to sixteen. Each point is evaluated on 100 frozen examples (Table 8). Raw F1 recovery declines from 91.5% at 8k to 76.5% at 32k, although KV Packet remains above No Recompute. At 32k, it recovers 60.1% of the Full-over-No improvement. Context length and packet count increase together, so this experiment does not isolate their individual effects. We have not trained an adapter at 32k or evaluated 128k contexts.

Table 8: NIAH long-context evaluation with an unchanged 8k-trained Llama-3.1-8B adapter. Recovery is raw F1 recovery relative to Full Recompute, as reported.

Context / packets	Full Recompute	No Recompute	KV Packet	Recovery
8k / 4	0.868	0.613	0.795	91.5%
16k / 8	0.845	0.453	0.742	87.9%
32k / 16	0.801	0.328	0.612	76.5%

A.8 Long-Form Generation Quality

We train a task-specific Llama-3.1-8B adapter on 256 self-generated Multi-News examples (Fabbri et al., 2019), select epoch five using a disjoint calibration set, and evaluate 100 frozen validation examples with greedy generation capped at 384 tokens. Table 9 reports reference-based ROUGE, gold-summary perplexity, and repeated trigrams. Gold-summary perplexity changes from 6.592

to 6.603 (+0.18%). Paired-bootstrap intervals for all three ROUGE differences include zero; the ROUGE-1 interval for KV Packet minus Full Recompute is $[-0.0105, 0.0068]$. Mean output length is 198.5 words for Full Recompute and 201.4 for KV Packet, and neither method reaches the generation cap.

Table 9: Multi-News generation evaluation. ROUGE values are reported on a 0–1 scale; gold PPL is reference-summary perplexity.

Method	ROUGE-1	ROUGE-2	ROUGE-L	Gold PPL	Repeated trigrams
Full Recompute	0.4284	0.1344	0.2041	6.592	1.64%
KV Packet	0.4266	0.1294	0.1999	6.603	2.03%

A blinded Qwen3-32B pairwise judge assesses factuality, coverage, integration, coherence, and fluency. To mitigate observed A/B position bias, each pair is judged in both presentation orders. Of the resulting 200 judgments for 100 example pairs, 133 favor Full Recompute (66.5%), 63 favor KV Packet (31.5%), and four are ties. These are 200 order-balanced judgments, not 200 distinct examples. Full Recompute generations have perplexity 1.251 under Full Recompute and 1.402 under KV Packet, showing a distributional difference in actual outputs that gold-summary perplexity does not capture. Thus, close reference-based metrics do not eliminate the residual long-form generation-quality gap. We have not tested whether a larger training budget closes it.

A.9 MuSiQue Packet Granularity

On Llama-3.2-3B, we use all 20 MuSiQue source documents and the same 100 seed-42 examples, prompts, and greedy decoder. We cross the original multi-hop question with a direct-final-hop control, and one document per packet with groups of at most four consecutive documents per packet (Max4). Grouping preserves the concatenated document-token stream, making Full Recompute identical across packetizations. Table 10 evaluates one cross-domain adapter trained across Biography, HotpotQA, NIAH, GSM8K (Cobbe et al., 2021), and all four MuSiQue configurations. This is a separate 3B configuration from the original four-task Llama-3.1-8B mixture in Table 1.

Table 10: MuSiQue Macro F1 for the cross-domain Llama-3.2-3B adapter, with fixed retrieved content. Max4 groups up to four consecutive documents. Recovery percentages are reported alongside rounded F1 scores.

Packing	Question	No Recompute	KV Packet	Full Recompute	Recovery
One/doc	Direct	0.040	0.316	0.449	70.4%
One/doc	Multi-hop	0.036	0.133	0.235	56.7%
Max4	Direct	0.108	0.414	0.449	92.2%
Max4	Multi-hop	0.001	0.208	0.235	88.5%

Grouping raises cross-domain multi-hop raw recovery by 31.8 percentage points, from 56.7% to 88.5%; under Max4, the direct-versus-multi-hop recovery gap is 3.7 points. A task-specific adapter reproduces the improvement (Table 11). Recovery above 100% is a finite-sample point estimate at $n = 100$, rather than evidence of superior underlying capability. These controlled results support packet fragmentation as a major factor in this setting. Larger packets reduce isolated-encoding boundaries and allow attention within the grouped documents, so the experiment does not establish that reasoning depth is irrelevant for all workloads. Packet granularity is a cache-construction parameter: any grouped packet must be built as that unit before reuse.

Table 11: MuSiQue Macro F1 with a task-specific adapter under the same packet-granularity protocol.

Packing	Question	No Recompute	KV Packet	Full Recompute	Recovery
One/doc	Direct	0.040	0.352	0.449	78.3%
One/doc	Multi-hop	0.036	0.118	0.235	50.4%
Max4	Direct	0.108	0.475	0.449	105.8%
Max4	Multi-hop	0.001	0.250	0.235	106.5%

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The contributions and scope are accurately reflected in the abstract and introduction sections.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: A limitation section is provided at the end of the paper.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

Justification: This paper does not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: All the codes and experiment configurations to reproduce the reported results are open-sourced. All the models and datasets used in the experiments are publicly available.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: The code and experiment configurations are available in the public GitHub repository linked in the abstract.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: All the experiment configuration files are available with the codes.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Appendix A.6 reports paired-bootstrap 95% confidence intervals for the expanded HotpotQA decomposition and separately describes evaluation-subset standard deviations. Appendix A.8 reports uncertainty for ROUGE differences. The original benchmark figures report point estimates.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The computing resources information used for the experiments are provided in the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that our research conforms with it in every respect.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This paper proposes a systems-level optimization for LLM inference efficiency and does not introduce new capabilities or applications that would directly lead to negative societal impacts.

Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper releases code for a KV cache optimization framework and does not release new pretrained models or scraped datasets that pose high risks for misuse.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We properly credit all existing assets used in this paper, including models, evaluation datasets, and baseline implementations. All assets are used in accordance with their respective licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The released KV Packet code is documented in the public repository linked in the abstract, which provides installation and usage instructions and an MIT license.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: Section 3 describes self-distillation from the frozen base LLM's generated continuations and output distributions. Appendix A.8 describes the Qwen3-32B pairwise judge and its order-balanced evaluation protocol.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.