SEMPIC: LEARNING SEMANTIC POSITION-INDEPENDENT KV CACHES

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-context retrieval and agentic workloads repeatedly reuse the same documents under changing instructions, histories, and document orders. Prefix caching cannot exploit this reuse, while position-independent caching (PIC) remains unreliable because independently compiled KV states lack the future context in which they will be consumed. Our diagnostics show that a learned boundary-conditioned baseline sharply reduces attention deviation near reusable-block boundaries but leaves interior and task-level residuals, motivating adaptation of the document representation itself. We present *SemPIC*, which trains a LoRA-enabled Writer to compile native per-layer document KVs through behavioral distillation while retaining the pretrained decoder as an unchanged Reader. Adaptation is confined to offline cache construction, preserving the standard KV interface and cache-hit decoding path. We further introduce KV Gradient Checkpointing, which reduces peak training memory without severing gradients through cached KVs. Across three models and four tasks, SemPIC raises mean micro-F1 over KV Packet from 0.53 to 0.60, approaching Full Recompute at 0.62.

1 INTRODUCTION

Retrieval-augmented generation supplies external documents as model context (Lewis et al., 2020), while tool-using and memory-augmented agents create analogous reusable units. These contexts are long, yet highly repetitive: the query changes from request to request, while many of the underlying semantic units recur. A conventional decoder processes every selected token during prefill before producing the first output token (Vaswani et al., 2017; Kwon et al., 2023). Prefix caching removes this work only when the reused content appears behind an identical prefix; tree-structured and modular prompt caches broaden sharing, but still rely on constrained layouts (Zheng et al., 2024; Gim et al., 2024). They cannot fully exploit a document that may be retrieved after different instructions, histories, or neighboring documents.

Position-independent caching (PIC) offers a more general abstraction: prepare the KV cache of each reusable document once, relocate it to its request-time position, and compose it with other cached documents. RoPE re-rotation corrects the positional phase of cached keys (Su et al., 2024), but position is only half of the problem. A token's KV state also summarizes the prefix visible when that state was created. A document compiled in isolation never observed the documents that precede it in a later request. Consequently, under a standard causal decoder, naively stitched caches are position aligned but contextually incomplete, and their quality can collapse.

Existing PIC methods address contextual incompleteness at different loci (Figure 1): CacheBlend and EPIC repair selected document states online (Yao et al., 2025; Hu et al., 2025), KV Packet learns reusable-block boundary states (Chen et al., 2026), and COMB adds an auxiliary cache encoder and Reader interface (Zhao et al., 2026). Yet it remains unclear whether the native document KVs themselves can be learned offline for reuse by an unchanged decoder under unseen document compositions.

To locate where offline adaptation should act, we analyze KV Packet (Section 4). Its learned boundary embeddings reduce near-boundary attention deviation to 1.6–8.1% of No Recompute's, yet the interior retains 66.4–125.7%, and a task-level gap to Full Recompute remains in 11 of 12 settings. We therefore ask whether adapting the document compiler itself can close the remaining gap without changing the online cache interface.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

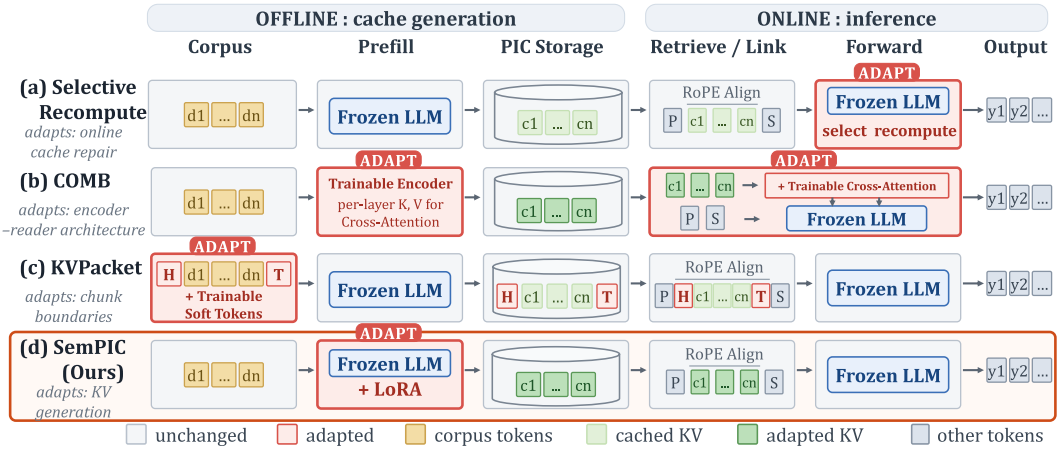

Figure 1: Design space of position-independent cache reuse. Representative designs shown here address PIC's contextual mismatch through online recomputation, offline boundary adaptation, or an auxiliary cache interface. SemPIC broadens the offline trainable locus from boundary embeddings to the native document Writer while retaining standard KVs and an unchanged Reader.

SemPIC broadens offline adaptation from block boundaries to the document Writer. LoRA (Hu et al., 2022) is active only while the Writer compiles each document, whereas the pretrained Reader consumes the resulting native per-layer KVs with the adapters disabled. Behavioral distillation (Chen et al., 2026) matches the Reader's output distributions to those of Full Recompute, allowing gradients to reshape the Writer through the cache interface while leaving online decoding unchanged.

Making every layer's KVs a differentiable Writer–Reader interface creates a training-memory bottleneck. **KV Gradient Checkpointing** reconstructs Writer intermediates during backward while preserving the cached KVs and their gradient path, reducing peak training memory without changing cache-hit serving.

Our contributions are:

- We identify a consistent **boundary–interior asymmetry**: boundary conditioning sharply reduces near-boundary attention deviation yet leaves interior and task-level residuals. This observation motivates document-wide cache adaptation.
- We propose SemPIC: a trainable native document-to-KV **Writer** feeds an unchanged pretrained **Reader** through standard per-layer KVs. Adaptation is offline, and cache-hit decoding remains unmodified.
- We introduce **KV Gradient Checkpointing**, which reconstructs Writer activations while preserving differentiable cache outputs, reducing peak memory enough to train long-context settings that otherwise run out of memory.
- Across three models and four tasks, SemPIC achieves higher micro-F1 than KV Packet in 10 of 12 settings (one tie, one loss), and lowers Full-relative interior attention error in all 12 settings.

2 Background: Position-Independent Caching

Let a request contain an inline prefix P, independently reusable document blocks $D_1, \ldots, D_m$, post-document inline spans $S_1, \ldots, S_m$, request-specific inline query tokens Q, and an output prefix $Y_{<t}$. As in Figure 3, the canonical order is $P, D_1, S_1, \ldots, D_m, S_m, Q$. The S_i denote dynamic request-time inline spans; unlike D_i, they are computed online and are not part of the reusable caches. Full recomputation applies the base model M_θ to this prompt and defines the teacher distribution

$$p_T^t = p_\theta(y_t \mid P, D_1, S_1, \ldots, D_m, S_m, Q, Y_{<t}). \tag{1}$$

A cache compiler C processes each document separately and produces per-layer states $C(D_i) = \{K_i^\ell, V_i^\ell\}_{\ell=1}^L$. At request time, a link operator R assigns these states their final logical positions, interleaves them with $S_1, \ldots, S_m$, and applies the causal mask. The resulting student distribution is

$$p_S^t = p_\theta(y_t \mid P, R(C(D_1), S_1, \ldots, C(D_m), S_m), Q, Y_{<t}). \tag{2}$$

PIC succeeds when linked caches reproduce the teacher distribution p_T under the prefixes and document compositions they meet at request time, whether or not their KVs match any particular full-context cache. We call cache compilation *semantic* when it targets this predictive behavior; SemPIC trains the Writer for it directly (Section 5.1).

For RoPE models, a key created at position a can be moved to b by

$$\widetilde{k}_b^\ell = R_\ell(b) R_\ell(a)^{-1} k_a^\ell, \tag{3}$$

where $R_\ell(\cdot)$ is the layer-specific rotary transform. Values are reused without rotation. Equation 3 corrects positional phase; it cannot recreate attention to context that was absent during independent compilation.

3 Related Work

Selective-recomputation methods repair context-dependent states online. CacheBlend selects tokens by cache deviation, EPIC refreshes leading tokens, and Cache-Craft applies targeted chunk-cache repair (Yao et al., 2025; Hu et al., 2025; Agarwal et al., 2025). Query-aware selectors refine this choice under different budgets (Yang et al., 2025b; Wang et al., 2026; Yan et al., 2026), but every refreshed token adds work before generation.

Other designs change linking or offline representations. Block-Attention and TurboRAG reuse passage KVs through independent block attention, while KVLink jointly fine-tunes the model and trainable link tokens (Ma et al., 2025; Lu et al., 2025; Yang et al., 2025c). KV Packet learns boundary embeddings through self-distillation, COMB adds a document encoder and cross-attention, and recent C^{2}KV learns a compressed cache manifold through a sidecar extractor (Chen et al., 2026; Zhao et al., 2026; Du et al., 2026). SemPIC instead places adaptation in the native document generator: every document token can change its representation, yet the stored object and online Reader remain standard.

This problem is distinct from reducing KV-cache footprint through token eviction or selection, cross-layer merging, quantization, or head-specific caching (Zhang et al., 2023; Li et al., 2024; Liu et al., 2024; Hooper et al., 2024; He et al., 2024; Xiao et al., 2025). SemPIC instead addresses the behavioral mismatch introduced by independent compilation and later composition.

4 Motivating Analysis: Boundary Conditioning Leaves an Interior Gap

Question. KV Packet's boundary-only adaptation suggests an implicit hypothesis: correcting reusable-block interfaces may suffice to restore independently compiled caches. We ask where its gains appear and what mismatch remains.

Diagnostic. We compare KV Packet with Full Recompute and No Recompute on Qwen3-4B-Instruct-2507, Qwen3-8B, and Llama-3.1-8B-Instruct across four tasks. Let $D_r(A, B)$ denote the mean absolute deviation between the post-softmax attention probabilities of configurations A and B within normalized block region r. We use $r = \mathrm{pre}$ for positions $[0, 0.1)$ and $r = \mathrm{int}$ for $[0.1, 0.9)$. For $m \in \{\mathrm{Packet}, \mathrm{SemPIC}\}$, define its Full-relative error as

$$R_r(m) = \frac{D_r(m, \mathrm{Full})}{D_r(\mathrm{NR}, \mathrm{Full})}. \tag{4}$$

Values below one indicate that method m is closer to Full Recompute than No Recompute under this deviation metric. To measure end-task recovery separately, let Vanilla denote direct reuse with No Recompute and define

$$\mathrm{Recovery}(m) = \frac{\mathrm{F1}_m - \mathrm{F1}_{\mathrm{Vanilla}}}{\mathrm{F1}_{\mathrm{Full}} - \mathrm{F1}_{\mathrm{Vanilla}}}. \tag{5}$$

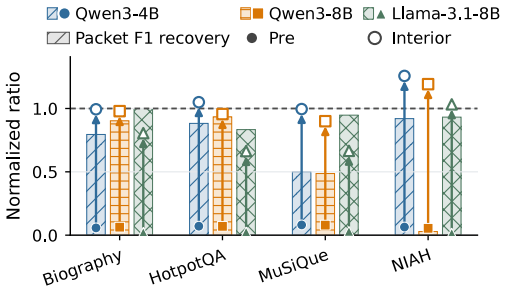

Figure 2: **Boundary adaptation leaves an interior gap.** Bars show KV Packet's F1 recovery; arrows connect its pre-region and interior attention errors relative to Full Recompute. The dashed line marks No Recompute for attention error and Full Recompute for recovery. Three models, four tasks, and 100 samples per configuration.

Recovery is zero for Vanilla and one for Full Recompute; values above one indicate higher micro-F1 than Full Recompute. D_r locates where linked caches attend differently from Full Recompute; task quality is measured separately by recovery (Eq. 5).

Observation and design implication. Figure 2 reveals a consistent asymmetry. Across all 12 settings, $R_{\text{pre}}(\text{Packet})$ ranges from 0.016 to 0.081 and remains below one, whereas $R_{\text{int}}(\text{Packet})$ ranges from 0.664 to 1.257 and falls below one in only 8 of 12 settings. KV Packet's mean F1 recovery is 0.763, and its gap to Full Recompute remains positive in 11 settings. Boundary conditioning therefore corrects the near-boundary region without reliably eliminating the interior discrepancy or closing the task-level gap. A reusable-cache compiler should consequently be able to reshape representations throughout the document while preserving an unchanged Reader and the native KV interface.

5 SemPIC

SemPIC trains an offline document Writer through native per-layer KVs consumed by an unchanged Reader. We describe this interface, behavioral supervision, and KV Gradient Checkpointing for memory-efficient training.

5.1 Writer–Reader Learning through the Native KV Interface

SemPIC separates document representation construction from its subsequent use. A *Writer* processes each reusable document independently, while a *Reader* consumes the compiled documents together with request-specific tokens. Both stages use the same pretrained decoder parameters θ, but only the Writer activates trainable LoRA parameters ϕ. For document D_i, the Writer produces

$$\mathcal{K}_\phi(D_i) = W_{\theta,\phi}(D_i) = \{K^\ell_{i,\phi}, V^\ell_{i,\phi}\}_{\ell=1}^{L}. \tag{6}$$

The pretrained parameters remain frozen, and ϕ denotes the switchable LoRA adaptation activated during Writer compilation. Although only keys and values are exported, the adaptation reshapes the layerwise hidden-state trajectory from which deeper-layer KVs are formed. Equation 6 produces the model's native per-layer cache representation, which can be relocated, linked, and consumed by the pretrained decoder without an auxiliary cache format or cross-attention interface.

The Reader is the frozen base model G_θ. After the linker assigns final positions to the compiled caches, it predicts

$$p^t_S = G_\theta\big(y_t \mid P, R(\mathcal{K}_\phi(D_1), S_1, \ldots, \mathcal{K}_\phi(D_m), S_m), Q, Y_{<t}\big). \tag{7}$$

Crucially, ϕ is disabled throughout query prefill, target-token computation, and generation. This asymmetry isolates learning to the document Writer: SemPIC changes the representation placed in the cache without adapting the model that later reads it. The Reader therefore retains the pretrained model's cache layout, positional relocation, and autoregressive decoding semantics.

Behavioral supervision. Following KV Packet's self-supervised formulation for cache reuse (Chen et al., 2026), training applies knowledge distillation (Hinton et al., 2015) using the base model's full-context behavior as the target. The teacher path supplies detached logits $z^{i,r,t}_T$ for

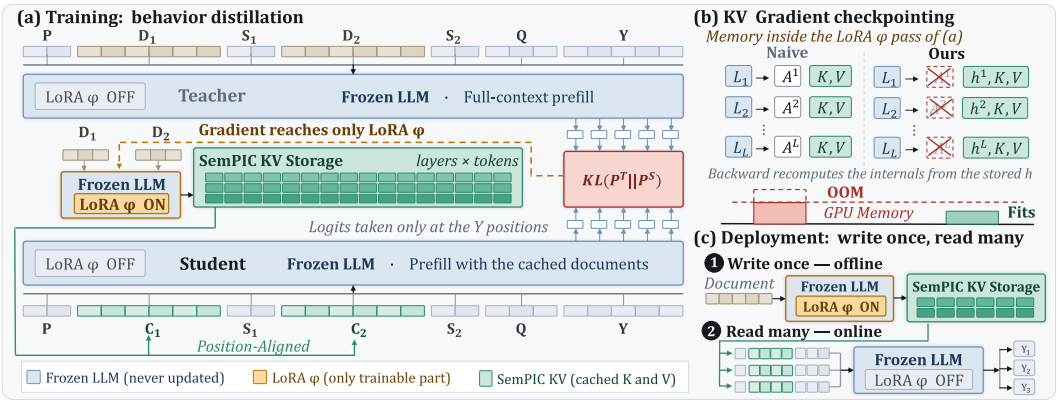

Figure 3: SemPIC trains a document Writer through the model's native KV interface while preserving the pretrained decoder as an unchanged Reader. LoRA is enabled only when the Writer compiles each document; the Reader consumes the differentiable caches with LoRA disabled and matches full-context teacher predictions. Once compiled, the caches enter the standard cache-hit path without document recomputation or auxiliary model execution.

sample i, teacher target r, and target position t. In the student path, the Writer first enables ϕ to construct the document caches; the Reader then disables ϕ and computes $z_S^{i,r,t}$ from those caches. Disabling the adapter does not detach the KV tensors, so the Reader remains unchanged while its loss still reaches the Writer.

With temperature τ, let $q_T = \mathrm{softmax}(z_T/\tau)$ and $q_S = \mathrm{softmax}(z_S/\tau)$. The implemented objective is

$$\mathcal{L}_{\mathrm{KD}} = \tau^2 \sum_i \sum_r \sum_t D_{\mathrm{KL}}\left(q_T^{i,r,t} \,\|\, q_S^{i,r,t}\right). \tag{8}$$

The reported runs use $\tau = 1$ and sum the token-level losses. The objective supervises output distributions rather than attention maps, hidden states, or a designated block-interior region. SemPIC's distinction is not the distillation loss itself, but its gradient path and adaptation locus. In particular,

$$\nabla_\phi \mathcal{L}_{\mathrm{KD}} = \sum_{i,b,\ell} \left(\frac{\partial \mathcal{L}}{\partial K_{i,b}^\ell} \frac{\partial K_{i,b}^\ell}{\partial \phi} + \frac{\partial \mathcal{L}}{\partial V_{i,b}^\ell} \frac{\partial V_{i,b}^\ell}{\partial \phi} \right), \tag{9}$$

where b indexes reusable document blocks within sample i. Every layer's KVs therefore act simultaneously as the reusable representation consumed by the Reader and the differentiable interface through which the Writer is optimized.

5.2 KV Gradient Checkpointing for Writer–Reader Training

KV-mediated Writer–Reader training retains both the all-layer KV interface and the Writer activations needed to differentiate it. The Reader consumes $\{K^\ell, V^\ell\}_{\ell=1}^L$, with gradients connected to ϕ through Equation 9. A naive Writer also retains attention and feed-forward intermediates for backward. Freezing θ removes backbone optimizer state, but these activations still grow with document length.

Activation checkpointing trades retained intermediates for backward recomputation (Chen et al., 2016). The Transformers checkpointing wrapper in our implementation environment disables KV caching during training. In SemPIC, however, each layer's KVs are outputs that the Reader consumes and through which the Writer is trained, so each checkpoint must return them rather than discard them. We therefore use the standard PyTorch non-reentrant checkpoint primitive with an explicit hidden-state and KV output interface:

$$(h^{\ell+1}, K^\ell, V^\ell) = C_{\theta,\phi}^\ell(h^\ell). \tag{10}$$

The hidden state $h^{\ell+1}$ advances the Writer, while K^ℓ and V^ℓ remain available to the Reader. Each layer invocation creates a fresh cache, so backward recomputation replays cache construction without appending to the original forward cache. Checkpointing retains layer inputs and reconstructs

the required internal activations during backward; gradients from both the next Writer layer and the Reader's KV inputs reach the same layer. The retained KVs and Reader activations continue to occupy memory.

Backward runs after the Writer's LoRA context has exited and the Reader has used the base model with LoRA disabled. Each checkpointed layer therefore re-enters the LoRA-enabled Writer context during both original execution and recomputation, restoring the previous adapter state on exit. This preserves the Writer computation and its gradient path through the native KVs. KV Gradient Checkpointing adapts activation checkpointing to this differentiable Writer–Reader interface, reducing retained Writer activations while preserving the Reader path and training objective.

Cache-hit serving. Once compiled, SemPIC KVs are ordinary model caches. A cache hit assigns their final logical positions, re-rotates their keys using Equation 3, and processes only request-specific inline tokens through the pretrained decoder. Prepared and inline KVs are interleaved under the causal mask before standard autoregressive generation. Query- and layout-independent compilation allows the same cache to be reused across histories, document orders, and retrieved sets, including layouts that place the query at different absolute positions. Semantic adaptation is therefore amortized over cache construction and introduces neither a document-side forward pass nor an auxiliary model on the request critical path.

Composition with boundary adaptation. To test whether document-wide and boundary adaptation are complementary, *Joint* co-trains shared Header and Trailer embeddings with the LoRA-enabled Writer under the same behavioral-distillation loss while the pretrained model remains frozen. The Writer adapts the document representation, while the boundary states provide a learned block interface. Once compiled, Joint follows SemPIC's recomputation-free cache-hit path.

6 Experiments

We evaluate task quality, cross-domain transfer, quality–efficiency tradeoffs, interior attention error, residual block-local behavior, and the memory–time tradeoff of KV Gradient Checkpointing.

6.1 Setup

We evaluate Llama-3.1-8B-Instruct (Llama Team, AI @ Meta, 2024), Qwen3-4B-Instruct-2507, and Qwen3-8B (Yang et al., 2025a) on Synthetic Biographies (Karev, 2025), HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), and a Needle-in-a-haystack (NIAH) task. We choose these tasks to span complementary cache-composition regimes: Biography tests factual retrieval from short reusable records, HotpotQA and MuSiQue require multi-hop QA over evidence distributed across multiple documents, and NIAH isolates long-context retrieval of sparse answer-bearing evidence. The latter is a controlled retrieval diagnostic of the type included in long-context suites such as RULER (Hsieh et al., 2024). We train SemPIC on training partitions and evaluate it on held-out test partitions. Each model–task–method cell contains 100 examples selected with a fixed dataset seed, greedy decoding, and bfloat16 execution; task-specific generation limits are reported in the supplementary material. SemPIC adapters are trained per model and domain for five epochs with LoRA rank 8, scale 16, and a learning rate of 5×10^{-4} with linear decay; complete settings are in the supplementary material.

Full Recompute processes the complete prompt and serves as the quality reference. No Recompute directly links independently prepared KVs after positional realignment, while No Cache omits the retrieved documents as a model-only control. KV Packet learns Header and Trailer tokens around document tokens (Chen et al., 2026); Joint is the co-trained composition of these boundary embeddings with the SemPIC LoRA described above. We report corpus-level token micro-F1.

6.2 Semantic Compilation Recovers PIC Quality

Table 1 shows higher micro-F1 for SemPIC than KV Packet in 10 of 12 settings, lifting the overall mean from 0.53 to 0.60, close to Full Recompute at 0.62. SemPIC matches or exceeds KV Packet across all eight Qwen3 settings, improving the 4B and 8B model averages by 0.10 and 0.13. The

Table 1: Task-level micro-F1. NR: No Recompute; Packet: KV Packet. Gray columns identify SemPIC and Joint; bold marks the best learned-cache result. Averages cover four tasks or all 12 task rows.

Task	Full	NR	Packet	**SemPIC**	**Joint**
Llama-3.1-8B					
Biography	0.96	0.06	0.96	**0.99**	**0.99**
HotpotQA	0.48	0.24	0.44	**0.45**	0.41
MuSiQue	0.39	0.01	0.37	0.28	**0.38**
NIAH	0.83	0.39	0.80	**0.82**	**0.82**
Average	0.67	0.18	0.64	0.64	**0.65**
Qwen3-4B					
Biography	0.97	0.19	0.81	**0.98**	0.96
HotpotQA	0.31	0.14	0.29	0.29	**0.30**
MuSiQue	0.41	0.07	0.24	0.31	**0.35**
NIAH	0.67	0.17	0.63	**0.76**	0.74
Average	0.59	0.14	0.49	**0.59**	**0.59**
Qwen3-8B					
Biography	0.96	0.24	0.89	**0.98**	0.97
HotpotQA	0.28	0.13	0.27	0.28	**0.29**
MuSiQue	0.45	0.08	0.26	0.34	**0.38**
NIAH	0.72	0.37	0.38	**0.70**	**0.70**
Average	0.60	0.21	0.45	0.58	**0.59**
Overall Average	0.62	0.17	0.53	0.60	**0.61**

Table 2: Cross-domain micro-F1 on Qwen3-8B. Row groups identify training sources and columns identify evaluation targets. Average is the four-target mean; bold marks the best average per source. Gray rows identify SemPIC and Joint.

Method	**Biography**	**HotpotQA**	**MuSiQue**	**NIAH**	**Average**
Train: Biography					
Packet	0.89	0.17	0.07	0.44	0.39
SemPIC	0.98	0.16	0.12	0.32	0.40
Joint	0.97	0.22	0.16	0.39	**0.44**
Train: HotpotQA					
Packet	0.73	0.27	0.29	0.41	0.43
SemPIC	0.86	0.28	0.38	0.46	**0.50**
Joint	0.85	0.29	0.37	0.37	0.47
Train: MuSiQue					
Packet	0.75	0.24	0.26	0.47	0.43
SemPIC	0.76	0.31	0.34	0.45	**0.47**
Joint	0.83	0.27	0.38	0.40	**0.47**
Train: NIAH					
Packet	0.53	0.18	0.19	0.38	0.32
SemPIC	0.25	0.21	0.22	0.70	0.35
Joint	0.36	0.21	0.18	0.70	**0.36**
Train: Mixture					
Packet	0.94	0.23	0.31	0.56	0.51
SemPIC	0.96	0.26	0.36	0.70	**0.57**
Joint	0.97	0.25	0.36	0.71	**0.57**

exception is Llama MuSiQue (0.28 versus 0.37), where Joint reaches 0.38; Joint's overall mean of 0.61 indicates that document-wide and boundary adaptation can be complementary. Because relocation, linking, and online decoding are shared, the central experimental contrast is the adaptation locus: document-wide for SemPIC and boundary-only for KV Packet.

6.3 Cross-Domain Transfer

We next test whether a Writer transfers across document domains.

On Qwen3-8B, SemPIC improves KV Packet's four-target average for every training source. Joint attains the highest average for four of the five sources, counting ties on MuSiQue and Mixture. Mixture training yields the highest average for each method: 0.51 for KV Packet and 0.57 for both SemPIC and Joint (Table 2). Across all three models, SemPIC improves the corresponding KV Packet average under 14 of 15 training sources; the complete Llama-3.1-8B and Qwen3-4B matrices are reported in the supplementary material.

The gains extend beyond the training domain: with HotpotQA training on Qwen3-8B, SemPIC raises Biography F1 from 0.73 to 0.86 and MuSiQue F1 from 0.29 to 0.38 over KV Packet. This Writer therefore improves cache reuse across distinct document domains.

6.4 Quality–Efficiency Tradeoff

Following KV Packet's comparison grid (Chen et al., 2026), CacheBlend (Yao et al., 2025), A^3 (Zhou et al., 2025), and Random Recompute sweep 10%, 30%, 50%, 70%, and 90% of document tokens. EPIC (Hu et al., 2025) recomputes 10, 30, 50, 70, and 90 tokens, with budgets from 100 to 500 tokens for NIAH. Figure 4 orders each sweep by increasing recomputation budget.

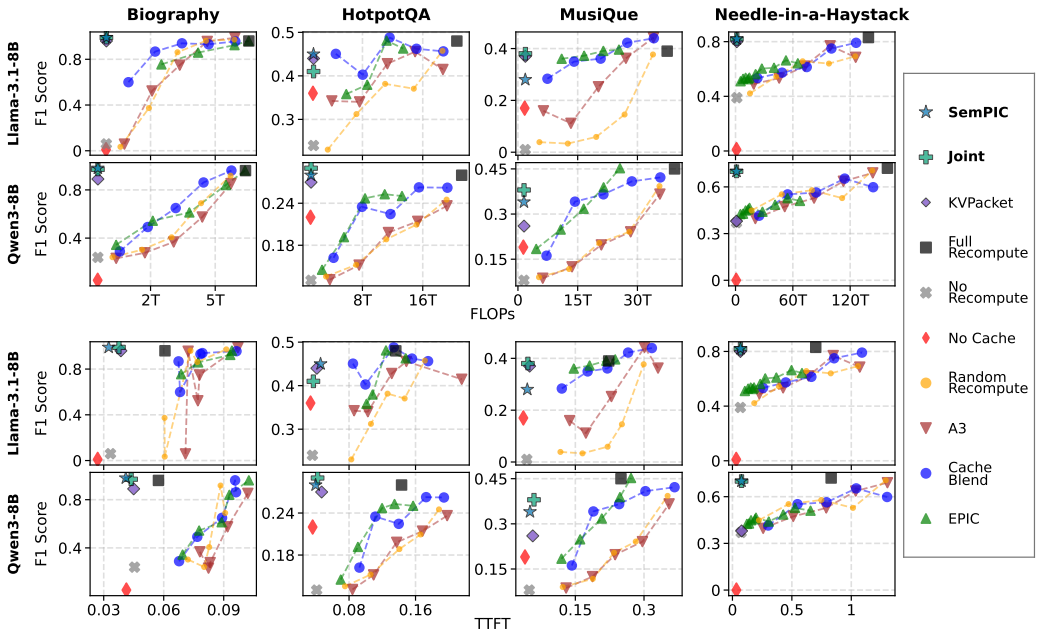

Figure 4: Quality–efficiency tradeoffs on Llama-3.1-8B and Qwen3-8B across four tasks. The upper two rows plot corpus micro-F1 against mean online prefill FLOPs; the lower two rows plot F1 against cache-hit time to first token (TTFT). Curves sweep selective-recomputation budgets, while isolated markers denote learned recomputation-free methods or references. Qwen3-4B and an enlarged three-model view are reported in the supplementary material.

At the online cost of No Recompute, SemPIC achieves near-Full-Recompute quality on average, with micro-F1 of 0.60 versus 0.62 (Table 1). Both methods reuse document KVs and process only request-specific inline tokens during cache-hit prefill; SemPIC's learned adaptation is confined to offline cache construction (Figure 4).

The Biography and NIAH panels illustrate this tradeoff on both displayed models: SemPIC reaches quality near or above Full Recompute with substantially fewer online FLOPs. The corresponding TTFT panels also show lower cache-hit latency than Full Recompute. These settings demonstrate how offline representation learning can recover task quality while avoiding request-time document recomputation.

6.5 Interior Attention Analysis

Across all 12 settings, SemPIC reduces $R_{\text{int}}(\text{SemPIC})$ below both the No Recompute reference and $R_{\text{int}}(\text{Packet})$, ranging from 0.564 to 0.985 versus 0.664 to 1.257 for KV Packet (Figure 5). Its unclipped recovery toward Full Recompute ranges from 0.703 to 1.180, with a mean of 0.921. The lower interior error and stronger task-level results relative to KV Packet move in the same favorable direction, but the evidence remains descriptive and does not show that the attention change causes the F1 gain.

6.6 Residual Block-Local Behavior and Scope

Attention sinks often concentrate on sequence-initial tokens (Xiao et al., 2024; Gu et al., 2025). Independent compilation places every document's first token at block-local offset one, so we test for a corresponding concentration after linking. For block token offset j, let $\text{density}_{\text{SemPIC}}(j)$ denote its attention density; we normalize it by the width-weighted mean density over the block interior $[0.1, 0.9)$:

$$T_j = \frac{\text{density}_{\text{SemPIC}}(j)}{\text{density}_{\text{SemPIC}}([0.1, 0.9))}. \tag{11}$$

Across all 12 settings, T_1 is the largest of the first eight ratios, ranging from 132.5 to 3248.3 versus 0.61 to 23.49 for offsets 2–8 (Figure 6). The attention peak is localized at block offset one rather than extending across neighboring offsets. The concentration survives Writer adaptation even as task quality improves. Separating the roles of block position and block-start content is left to future work.

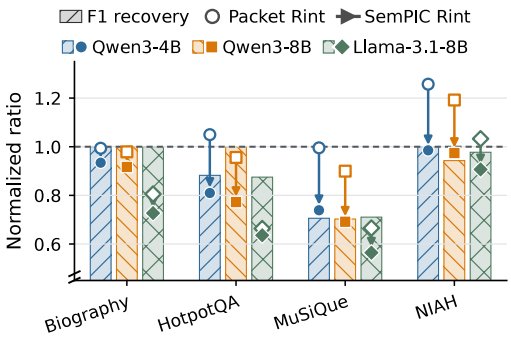

Figure 5: **SemPIC lowers interior error in all 12 settings.** Arrows connect $R_{\text{int}}(\text{Packet})$ to $R_{\text{int}}(\text{SemPIC})$; lower is better. Bars show F1 recovery, capped at one; the main text reports unclipped values. Three models, four tasks, and 100 samples per configuration.

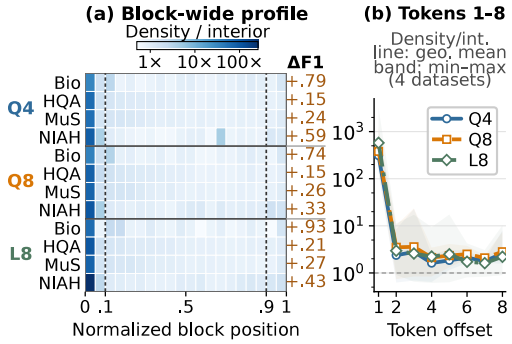

Figure 6: **A block-local first-token attention peak persists.** (a) Attention density and ΔF1 over No Recompute. (b) T_j at offsets 1–8; lines and bands denote geometric means and task ranges. Q4/Q8/L8 identify the three models; $n = 100$ per configuration.

Figures 5 and 6 show that improved quality and lower interior error coexist with a persistent first-token attention peak. This is consistent with SemPIC's behavioral objective: it supervises Reader outputs, without explicitly requiring attention-map matching or suppression of local peaks.

Table 3: Qwen3-8B SemPIC peak training memory (GiB) and wall-clock time (h:mm) on one A100 80GB GPU. The document-forward batch size (FBS) is 1 unless marked FBS 8; Ours denotes KV Gradient Checkpointing.

	Naive		Ours	
Dataset	Mem.	Time	Mem.	Time
Biography	23.8	0:34	18.0	1:55
Biography (FBS 8)	75.8	0:23	33.2	0:25
HotpotQA	OOM		26.4	5:40
MuSiQue	OOM		33.1	6:46
NIAH	OOM		43.4	13:26

6.7 Training Memory and Time

Table 3 compares KV Gradient Checkpointing with the naive training path in peak device memory and wall-clock time.

KV Gradient Checkpointing reduces Qwen3-8B peak memory from 23.8 to 18.0 GiB at FBS 1 and from 75.8 to 33.2 GiB at FBS 8. Biography training time rises from 34 to 115 minutes and from 23 to 25 minutes, respectively: overhead is substantial at FBS 1 but small at FBS 8. Checkpointing also enables HotpotQA, MuSiQue, and NIAH training where the naive path runs out of memory. This memory–time tradeoff affects training only, not cache-hit serving.

7 Conclusion

Position-independent reuse asks a document cache to function in contexts absent during construction. SemPIC addresses this mismatch by training a Writer to produce native KVs for an unchanged Reader. Across the evaluated settings, SemPIC raises overall mean micro-F1 over KV Packet from 0.53 to 0.60 and reduces Full-relative interior attention error under our metric. Joint suggests that document-wide and boundary adaptation are complementary, while KV Gradient Checkpointing makes long-document Writer training feasible. Together, these results show that document caches can be learned offline while preserving the pretrained Reader and native KV interface.

AI USE STATEMENT

We used generative AI tools to assist with method implementation and code development, as well as information search, manuscript editing, and LaTeX preparation. The authors take responsibility for the final manuscript, methods, code, and reported results.

ETHICS STATEMENT

SemPIC reuses document-derived KV states. Deployments should apply the same access controls and retention policies to these states as to the source documents, and prevent cache sharing across unauthorized users. Cache adaptation does not remove the pretrained model's potential biases or factual errors.

REPRODUCIBILITY STATEMENT

The method section defines the Writer–Reader interface, distillation objective, and checkpointing mechanism. Appendices A–D report data splits, adapter and optimization settings, generation limits, the quality metric, cache-hit timing and FLOP boundaries, the computing environment, and the memory-measurement protocol. Additional transfer matrices and quality–efficiency plots provide the complete reported comparisons.

REFERENCES

Shubham Agarwal, Sai Sundaresan, Subrata Mitra, Debabrata Mahapatra, Archit Gupta, Rounak Sharma, Nirmal Joshua Kapu, Tong Yu, and Shiv Saini. Cache-Craft: Managing chunk-caches for efficient retrieval-augmented generation. *Proceedings of the ACM on Management of Data*, 3 (3):1–28, 2025. doi: 10.1145/3725273.

Chuangtao Chen, Grace Li Zhang, Xunzhao Yin, Cheng Zhuo, Bing Li, and Ulf Schlichtmann. KV Packet: Recomputation-free context-independent KV caching for LLMs. *arXiv preprint arXiv:2604.13226*, 2026.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

Chuheng Du, Junyi Chen, Hanlin Tang, Kan Liu, Tao Lan, Lin Qu, Chaoyue Niu, Shengzhong Liu, Guihai Chen, and Fan Wu. C^2KV: Compressed and composable KV cache reuse for efficient LLM inference. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 991–1002, 2026.

In Gim, Guojun Chen, Seung-seob Lee, Nikhil Sarda, Anurag Khandelwal, and Lin Zhong. Prompt cache: Modular attention reuse for low-latency inference. In *Proceedings of Machine Learning and Systems*, volume 6, 2024.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view. In *International Conference on Learning Representations*, 2025.

Yefei He, Luoming Zhang, Weijia Wu, Jing Liu, Hong Zhou, and Bohan Zhuang. ZipCache: Accurate and efficient KV cache quantization with salient token identification. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. KVQuant: Towards 10 million context length LLM inference with KV cache quantization. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Junhao Hu, Wenrui Huang, Weidong Wang, Haoyi Wang, Tiancheng Hu, Zhang Qin, Hao Feng, Xusheng Chen, Yizhou Shan, and Tao Xie. EPIC: Efficient position-independent caching for serving large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24391–24402, 2025.

Alex Karev. Synthetic biographies. Hugging Face dataset: `https://huggingface.co/datasets/alex-karev/biographies`, 2025. Accessed: 2026-07-29.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0722.

Akide Liu, Jing Liu, Zizheng Pan, Yefei He, Gholamreza Haffari, and Bohan Zhuang. MiniCache: KV cache compression in depth dimension for large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-4443.

Llama Team, AI @ Meta. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Songshuo Lu, Hua Wang, Yutian Rong, Zhi Chen, and Yaohua Tang. TurboRAG: Accelerating retrieval-augmented generation with precomputed KV caches for chunked text. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6588–6601. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.334.

Dongyang Ma, Yan Wang, and Tian Lan. Block-attention for efficient prefilling. In *International Conference on Learning Representations*, 2025.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063.

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*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Shihao Wang, Jiahao Chen, Yanqi Pan, Hao Huang, Yichen Hao, Xiangyu Zou, Wen Xia, Wentao Zhang, Chongyang Qiu, and Pengfei Wang. ProphetKV: User-query-driven selective recomputation for efficient KV cache reuse in retrieval-augmented generation. *arXiv preprint arXiv:2602.02579*, 2026.

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

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. DuoAttention: Efficient long-context LLM inference with retrieval and streaming heads. In *International Conference on Learning Representations*, 2025.

Jianxin Yan, Wangze Ni, Zhenxin Li, Jiabao Jin, Zhitao Shen, Haoyang Li, Jia Zhu, Peng Cheng, Xuemin Lin, Lei Chen, and Kui Ren. QCFuse: Query-aware cache fusion via compressed view for efficient RAG serving. *arXiv preprint arXiv:2606.05875*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, 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, Yujia Bao, and Shiyu Chang. KVLink: Accelerating large language models via efficient KV cache reuse. In *Advances in Neural Information Processing Systems*, volume 38, pp. 133797–133824, 2025c. doi: 10.52202/085713-4461.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018. doi: 10.18653/v1/D18-1259.

Jiayi Yao, Hanchen Li, Yuhan Liu, Siddhant Ray, Yihua Cheng, Qizheng Zhang, Kuntai Du, Shan Lu, and Junchen Jiang. CacheBlend: Fast large language model serving for RAG with cached knowledge fusion. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 94–109, 2025. doi: 10.1145/3689031.3696098.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Shiju Zhao, Junhao Hu, Jiaqi Zheng, and Guihai Chen. You need an encoder for native position-independent caching. *arXiv preprint arXiv:2602.01519*, 2026.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. SGLang: Efficient execution of structured language model programs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2000.

Yuechi Zhou, Yi Su, Jianxin Zhang, Juntao Li, Qingrong Xia, Zhefeng Wang, Xinyu Duan, and Baoxing Huai. A^3: Attention-aware accurate KV cache fusion for fast large language model serving. *arXiv preprint arXiv:2511.17560*, 2025.

This supplement provides implementation, measurement, and reproducibility details for the main paper. It does not add claims beyond the reported experimental scope.

A SemPIC Training Details

A.1 Document and Student Paths

For each training sample, SemPIC identifies prompt spans designated as reusable context blocks. The frozen token embedding maps every block to input embeddings, and the document adapter encodes the blocks independently. Independence is maintained under batching: padding is masked, and a block cannot attend to another block in the same batch.

Prepared KVs are assigned to each associated teacher target and linked with the inline query tokens and teacher-forced output prefix. Student logits are collected only at target-token positions. The base-model parameters remain frozen, the adapter is disabled during student computation, and cached teacher logits are detached. No teacher forward is executed inside the measured student-training process.

A.2 Dataset Splits and Sample Construction

Examples are split before prompt construction. Biography and NIAH use disjoint project-level training and test partitions; HotpotQA and MuSiQue use the official training and validation splits. Distillation contains 1,024 Biography records—256 each with 2, 3, 5, and 7 documents—and 2,048 records per remaining task; evaluation uses 100 held-out zero-shot examples per task.

A.3 Loss and Optimization

The teacher cache contains one or more generated sequences per sample and a floating-point logit tensor over the complete vocabulary for every generated token. The temperature-scaled forward KL objective defined in the main paper is implemented by summing over the complete vocabulary, target-token, teacher-target, and sample dimensions. The evaluated runs use $\tau = 1$.

Table 4 records the common adapter settings. Adapters are trained separately by model and domain.

Table 4: Common SemPIC training and evaluation settings for the evaluated campaign.

Setting	Value
Adapter	LoRA on q/k/v/o projections
Rank / scale / dropout	8 / 16 / 0
Epochs	5
Learning rate	5×10^{-4}
Schedule	Linear decay to zero
Weight decay	0
Precision	bfloat16
Loss temperature	1
Evaluation-data seed	42

B KV Gradient Checkpointing

B.1 Layer-Wise Contract

The top-level `kv_gradient_checkpointing` option defaults to true in the training configuration. When enabled, the Writer calls decoder layers individually through PyTorch checkpointing with `use_reentrant=False`. Each layer function creates a fresh `DynamicCache`, uses `use_cache=True`, and returns the next hidden state together with that layer's keys and values. The hidden state feeds the next layer; all per-layer KVs remain connected to autograd and feed

the Reader. Non-reentrant recomputation can stop once the required internal activations have been reconstructed.

The Transformers checkpointing wrapper in the implementation environment disables `use_cache` and clears cache arguments during checkpointed training. Our layer function makes KVs explicit checkpoint outputs and creates its cache locally on every invocation, allowing cache updates to be replayed during backward. A `forward_context_factory` re-enters the LoRA-enabled Writer state for original execution and recomputation, then restores the prior adapter state. This is necessary because the Reader runs with LoRA disabled before backward reaches the Writer. Gradients can also flow to Joint's trainable Header and Trailer embeddings.

When the option is disabled, document compilation calls the complete model and extracts `past_key_values`. That path also executes the final normalization and a small logits computation; the checkpointed path executes only the decoder layers needed to construct KVs. Both paths retain differentiable KVs and use the same Reader and loss. The measured time difference therefore includes this forward-path difference as well as activation recomputation.

The checkpointed Writer receives the same causal mask, position identifiers, cache positions, rotary embeddings, and model-specific sliding-window mask as the direct path. Variable-length blocks are padded during batched construction and trimmed to their valid lengths afterward. The implementation supports the Llama and Qwen3 decoder families used in the paper, requires document input embeddings and an autograd-enabled document-KV forward, and does not combine checkpointed construction with KV compression or index-based KV selection.

B.2 Numerical Validation

Direct and checkpointed paths are tested for per-layer key/value equality, position handling, and input gradients on variable-length Llama and Qwen3 examples. Student-prefill tests verify that LoRA gradients survive checkpointing. Joint-method tests additionally compare losses and gradients when packet parameters and LoRA are trained together. These checks validate the forward and gradient contract; they do not substitute for the memory measurements below.

B.3 Memory Measurement Protocol

The main comparison uses Qwen3-8B LoRA-only KL training on one NVIDIA A100 80GB PCIe GPU per process. Each dataset is measured once with checkpointing enabled and once with it disabled. The forward microbatch contains one sample, while the optimizer batch contains 64 samples accumulated across microbatches. The model executes in bfloat16 with rank-8 LoRA on q/k/v/o projections.

Device memory is sampled with `nvidia-smi` every two seconds. Each GPU is empty before launch and hosts only the measured process. A successful probe enters the real forward/backward training path; selected probes complete at least one optimizer update. A condition is recorded as OOM when the normal configuration cannot complete its forward/backward path. The values include model weights, LoRA state, runtime buffers, and resident teacher-generation data, so they are system-level device observations rather than isolated activation allocations.

The resulting microbatch-1 measurements are summarized in the main paper. A separate Biography probe uses document-forward microbatch size 8. Its direct path operates near device capacity and emits allocation-pressure warnings, so this condition is kept separate from the microbatch-1 cross-dataset comparison.

These are short operational probes rather than repeated full training runs. The two-second sampling period may miss short-lived allocator peaks, and no variance estimate is available. They support the fit-versus-OOM and approximate memory-effect claims, not a statistical characterization of GPU memory.

C Position-Independent Linking

C.1 Logical Layout and Re-Rotation

Prepared blocks can be stored and batched independently of their later prompt order. At link time, the runtime builds a token-level map from physical cache slots to logical prompt positions, applies the key re-rotation defined in the main paper, and reuses values directly. Inline positions are processed layer by layer and their new KVs are interleaved with the prepared document states. The same caches therefore support different queries, histories, document orders, and retrieved sets, with logical positions assigned for each layout.

A frontier causal mask prevents each inline token from observing logical successors, even when physical slots for later blocks have already been assembled. The terminal inline hidden state is projected through the language-model head to obtain the first-token distribution. The completed cache is then used by ordinary autoregressive decoding.

C.2 TTFT Boundary

The paper reports cache-hit online TTFT. The timer starts after prepared document KVs and method-specific wrappers are available, immediately before prompt-layout construction. It ends after the first output token is produced, with device synchronization at both endpoints. It includes:

- logical layout and causal-frontier construction;
- document-key re-rotation;
- inline-token prefill and cache assembly;
- generation setup and first-token generation.

It excludes model loading, data preprocessing, cache lookup and transfer I/O, offline document-KV construction, and cache-miss ingestion. This boundary models repeated cache-hit service rather than first use of a new document.

D Evaluation Details

D.1 Quality Metric

After task-specific answer normalization, each prediction and reference is split on whitespace and treated as a token multiset. True positives, false positives, and false negatives are pooled over the complete 100-example set, with $P = TP/(TP+FP)$, $R = TP/(TP+FN)$, and $F_1 = 2PR/(P+R)$. We choose corpus micro-F1 because these retrieval and open-form QA tasks can return multi-token answers: token overlap credits partially recovered answer content, while precision penalizes extra non-reference tokens, rather than assigning every non-exact generation zero credit. The metric remains lexical, whitespace-sensitive, and order-insensitive.

All methods use greedy decoding. Generation caps are 32/512/128 tokens for Biography/HotpotQA/NIAH; MuSiQue uses 256 for Llama-3.1-8B and 512 for Qwen3.

D.2 Efficiency Metrics

Mean online prefill FLOPs are architecture-aware analytical estimates averaged over the 100 evaluation requests. For cache-reuse methods, accounting begins after reusable KVs and learned wrappers are prepared; Full Recompute instead counts complete-prompt prefill. The estimate includes key re-rotation, transformer arithmetic for inline and recomputed tokens, implemented method-specific scoring or fusion, and the final normalization and output projection that produce first-token logits. It excludes offline cache construction and training, model loading, data preprocessing, cache lookup and transfer, and autoregressive computation after the first-token logits. It also omits system effects not represented by the analytical counters, including data movement, allocation, layout construction, kernel launch, and synchronization overhead.

Table 5: Cross-domain corpus micro-F1 on Llama-3.1-8B.

Method	Bio-graphy	Hotpot QA	Mu SiQue	NIAH	Aver-age
Train: Biography					
Packet	0.96	0.18	0.16	0.44	0.44
SemPIC	0.99	0.32	0.10	0.40	**0.45**
Joint	0.99	0.26	0.08	0.40	0.43
Train: HotpotQA					
Packet	0.88	0.44	0.39	0.36	0.52
SemPIC	0.95	0.45	0.33	0.56	0.57
Joint	0.96	0.41	0.38	0.56	**0.58**
Train: MuSiQue					
Packet	0.87	0.36	0.37	0.34	0.49
SemPIC	0.95	0.44	0.28	0.54	0.55
Joint	0.92	0.37	0.38	0.64	**0.58**
Train: NIAH					
Packet	0.54	0.28	0.07	0.80	0.42
SemPIC	0.53	0.36	0.22	0.82	0.48
Joint	0.85	0.34	0.06	0.82	**0.52**
Train: Mixture					
Packet	0.95	0.42	0.43	0.71	0.63
SemPIC	0.99	0.42	0.38	0.77	0.64
Joint	0.99	0.49	0.39	0.82	**0.67**

Table 6: Cross-domain corpus micro-F1 on Qwen3-4B-Instruct-2507.

Method	Bio-graphy	Hotpot QA	Mu SiQue	NIAH	Aver-age
Train: Biography					
Packet	0.81	0.18	0.07	0.31	0.34
SemPIC	0.98	0.20	0.13	0.44	**0.44**
Joint	0.96	0.17	0.13	0.21	0.37
Train: HotpotQA					
Packet	0.81	0.29	0.28	0.50	0.47
SemPIC	0.89	0.29	0.30	0.55	**0.51**
Joint	0.78	0.30	0.37	0.49	0.49
Train: MuSiQue					
Packet	0.67	0.26	0.24	0.48	0.41
SemPIC	0.49	0.18	0.31	0.49	0.37
Joint	0.67	0.26	0.35	0.49	**0.44**
Train: NIAH					
Packet	0.56	0.14	0.12	0.63	0.36
SemPIC	0.27	0.26	0.18	0.76	0.37
Joint	0.47	0.25	0.19	0.74	**0.41**
Train: Mixture					
Packet	0.93	0.26	0.24	0.48	0.48
SemPIC	0.94	0.28	0.42	0.67	**0.58**
Joint	0.96	0.27	0.34	0.75	**0.58**

TTFT is the mean synchronized wall-clock time under the boundary defined above; for cache-reuse methods, this is cache-hit latency. We report both metrics because FLOPs isolates algorithmic online arithmetic independently of a particular GPU implementation, whereas TTFT captures realized serving effects such as parallelism, memory traffic, and kernel efficiency. Conversely, the FLOP axis helps distinguish reduced online computation from implementation-specific latency differences.

The reported tradeoff curves do not establish performance at unevaluated recomputation budgets or end-to-end cache-miss latency. Figure 7 presents the complete three-model, four-task overview.

D.3 Computing Environment

All quality and TTFT comparisons use the same server and software environment. The server contains eight NVIDIA A100 80GB PCIe GPUs, two Intel Xeon Gold 6530 CPUs (64 physical cores and 128 hardware threads), 1 TiB of host memory, and Ubuntu 24.04.3 LTS; each process uses one GPU. The environment uses Python 3.12.3, CUDA 13.0, PyTorch 2.13.0+cu130, Transformers 5.2.0, PEFT 0.19.1, Accelerate 1.14.0, and Datasets 5.0.0, with NVIDIA driver 610.43.02. All methods within a TTFT panel are therefore compared under matched hardware and software conditions.

D.4 Cross-Domain Transfer

Tables 5 and 6 complete the main-paper transfer analysis under the same training-source and evaluation-target organization. Packet denotes KV Packet; row groups identify training sources, columns identify evaluation targets, and Average is their arithmetic mean. Bold numeric entries mark the best average within each training source. SemPIC improves the corresponding KV Packet four-target average under all five Llama-3.1-8B training sources and four of five Qwen3-4B sources; the exception is Qwen3-4B trained on MuSiQue.

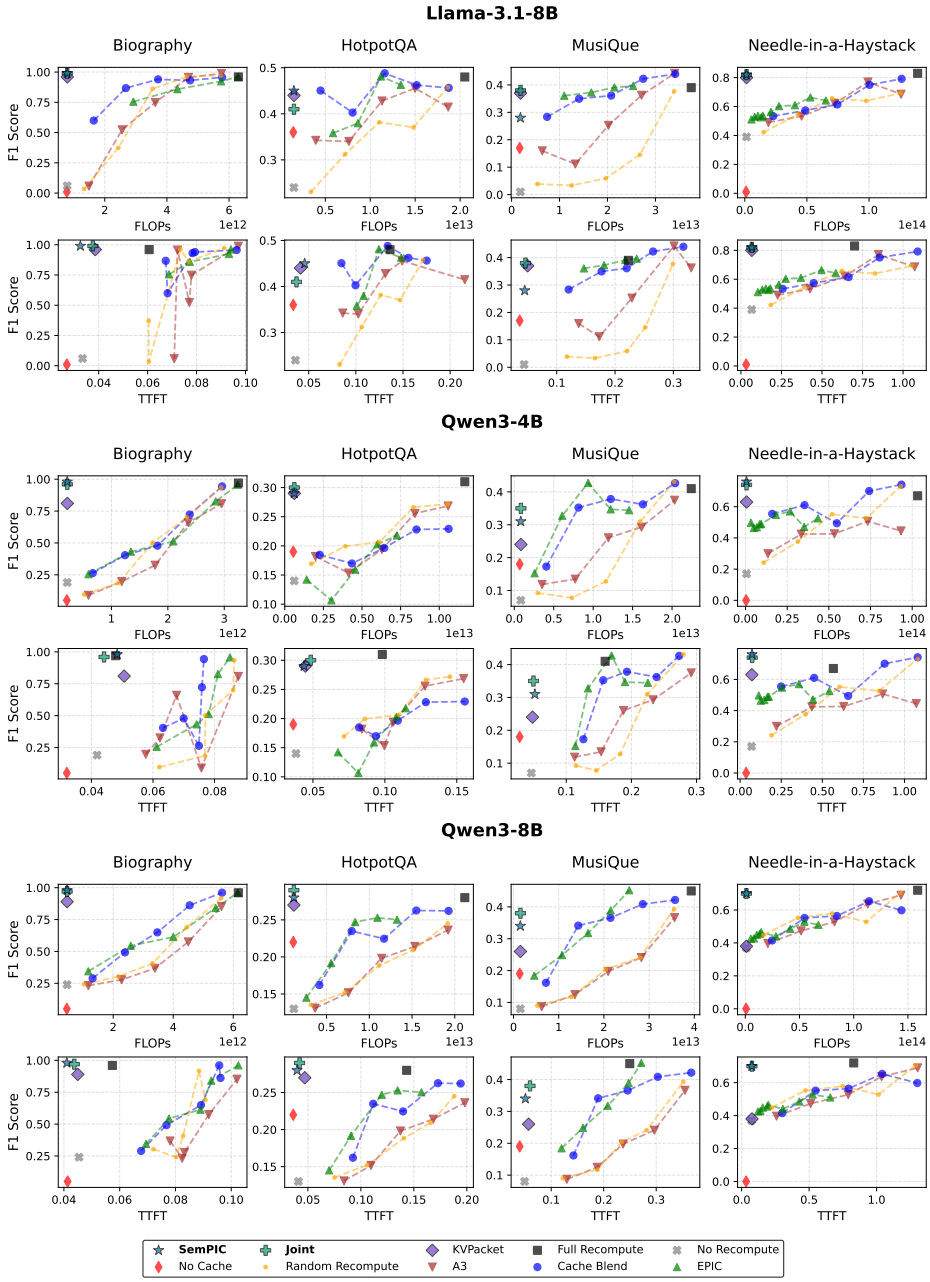

Figure 7: High-resolution overview of quality–efficiency tradeoffs on three models and four tasks, reporting corpus micro-F1 against online prefill FLOPs (upper rows) and cache-hit TTFT (lower rows). TTFT excludes offline cache construction. Connected points show evaluated recomputation budgets, not continuous curves or repeated trials.