RSCE: TRAINING-FREE RESIDUAL STREAM ENCODING FOR PERSISTENT CONTEXT AMORTIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose Residual Stream Context Encoding (RSCE), a training-free method that eliminates redundant long-context prefill costs in retrieval-augmented generation. Given a context document ctx, RSCE extracts a vector $C \in \mathbb{R}^{d_M}$ by mean-pooling residual stream activations at a calibrated intermediate layer $f(M)$, then injects it as an additive shift at query time—replacing $O(|T(\mathrm{ctx})|)$ attention prefill with an $O(1)$ operation with zero per-query context forward pass. For tasks requiring factual precision, we pair C with a compact explicit fact block F, forming a dual-channel representation amortized across $N \geq 2$ queries. We evaluate five decoder-only architectures (7B–70B) on multi-document QA (LongBench, $n = 108$) and six architectures on cross-file code completion (RepoBench-C), comparing against LongLLMLingua and EHPC. A key mechanistic finding: vector injection alone suppresses parametric recall *below* the question-only baseline—a dual-pathway interference effect absent in behavioral steering that motivates the dual-channel design. At extreme compression ($\sim$99% token reduction), RSCE Vec+F is competitive with EHPC on smaller architectures (LLaMA-8B F1 0.333 vs. EHPC 0.334; DeepSeek-14B both 0.214) while both substantially outperform LongLLMLingua (0.209, 0.172). On larger models, EHPC's capacity-scaling token selection widens the gap, reaching F1 0.539 vs. RSCE 0.365 on LLaMA-70B—a finding we explain through model capacity scaling of in-context reasoning. On RepoBench-C, LongLLMLingua substantially improves over baseline via compression-as-retrieval; RSCE is the only method achieving 81% compression at 100% operational reliability. Code is available at `https://anonymous.4open.science/r/RSCE-2E4C/`.

1 INTRODUCTION

Retrieval-augmented generation architectures prepend long context documents ctx to user queries. Processing 1,000–8,000 tokens dominates inference compute. For repeated queries over static contexts—API documentation, codebase modules, legal corpora—this cost is incurred redundantly on every interaction.

We propose Residual Stream Context Encoding (RSCE), which encodes ctx into a fixed vector $C \in \mathbb{R}^{d_M}$ via mean-pooling the residual stream at an empirically calibrated layer $f(M)$, then injects C as an additive shift at query time, bypassing explicit token prefill entirely. A dual-channel design pairs C with a minimal explicit fact block F to compensate for mean-pooling's loss of token-level precision.

Hard-prompt compressors (Jiang et al., 2023; 2024; Fei et al., 2025) select and delete tokens but still require a per-query forward pass and face a quality floor at extreme ratios. Trained soft-compression methods (Cheng et al., 2024; Ge et al., 2024; Chevalier et al., 2023) require auxiliary supervision and restrict plug-and-play deployment. Activation steering methods (Liu et al., 2024b; Todd et al., 2024) inject computed vectors to encode task demonstrations or behavioral abstractions—but not factual document content. RSCE applies the same injection mechanism to a qualitatively different problem. The failure mode we uncover (Vec $<$ Q-only) has no analog in behavioral steering, where injected vectors augment performance monotonically. It arises from dual-pathway interference: the vector engages contextual retrieval circuitry without providing token-level grounding, suppressing parametric recall without adequate compensation. This motivates the complementary fact block.

We make the following contributions. (1) RSCE: a training-free, $O(1)$ amortized context encoding method with zero per-query context prefill and 100% operational reliability. (2) We confirm across five architectures that vector injection alone *suppresses* parametric recall; the fact block is constitutive, not supplementary. (3) We provide a full multi-model, multi-method comparison at matched extreme compression, revealing a capacity-scaling effect where EHPC's advantage over RSCE grows with model size—while RSCE retains its per-query compute and reliability guarantees. (4) We offer a compression-as-retrieval explanation for LongLLMLingua's strong code performance, grounded in the lost-in-the-middle literature (Liu et al., 2024a).

2 Related Work

Jiang et al. (2023) and Li et al. (2023b) use perplexity-based token scoring. Jiang et al. (2024) adds question-aware contrastive perplexity and document reordering. Fei et al. (2025) identifies "evaluator heads" whose attention scores locate important tokens at 0.88s latency—current state-of-the-art training-free hard-prompt compression. All still require a per-query prefill and face a quality floor at extreme compression (Li et al., 2025).

Zhang et al. (2023), Xiao et al. (2024), and Li et al. (2024) evict low-importance KV entries within a single pass but do not produce persistent reusable representations.

Ge et al. (2024), Chevalier et al. (2023), and Mu et al. (2023) train encoder modules for soft embeddings. Cheng et al. (2024) projects dense retriever embeddings through a trained MLP bridge. Feldman & Artzi (2025) validates mean-pooling of hidden states as superior to alternative soft-compression architectures—but requires training, whereas RSCE is entirely training-free.

Turner et al. (2023), Li et al. (2023a), and Zou et al. (2023) steer behavior via hidden state perturbations. Liu et al. (2024b) and Todd et al. (2024) demonstrate that intermediate-activation vectors encode task abstractions when injected additively, exploiting the residual stream's role as a shared additive communication channel (Elhage et al., 2021). The linear representation hypothesis (Park et al., 2024) formalizes why additive injection shifts downstream computation: high-level concepts are encoded as linear directions in activation space, making additive injection structurally equivalent to shifting the model's belief state (Shai et al., 2024). RSCE shares this mechanism but targets factual document content, uncovering a parametric-memory interference effect absent in behavioral steering.

When models receive external context signals that conflict with or are noisier than parametric knowledge, performance can degrade below the no-context baseline (Longpre et al., 2021; Mallen et al., 2023; Shi et al., 2023). This literature directly motivates our Vec $<$ Q-only finding.

3 Method

3.1 Formal Specification

Let M be a decoder-only transformer, T a tokenizer, and P a prompt string. The residual context encoding is:

$$g(M, \text{ctx}) = \text{mp}(\text{res}_M(\text{ctx}, f(M))) = \frac{1}{|T(\text{ctx})|} \sum_{i=1}^{|T(\text{ctx})|} H_i \ \in \ \mathbb{R}^{d_M}$$

where $H \in \mathbb{R}^{|T(\text{ctx})| \times d_M}$ are the residual stream hidden states at layer $f(M)$. Concepts are encoded as linear directions in residual stream space (Park et al., 2024), making mean-pooling a structure-preserving operation over the document's distributed semantic content. The aggregation produces a document-level representation analogous to a belief state (Shai et al., 2024)—structurally distinct from sentence-level embeddings but grounded in the same geometric principles validated in the sentence embedding literature (Reimers & Gurevych, 2019).

During inference, the input is $T(F \oplus P)$ only. At layer $f(M)$, prior to its attention and feed-forward sublayers, C is added uniformly: $H'_i \leftarrow H_i + \alpha \cdot C \ \forall i$, with $\alpha = 1.0$ (held-out calibration). The transformer residual stream is architecturally designed around additive writes from all components (Elhage et al., 2021); this external injection is structurally indistinguishable from an internal layer's

contribution. The break-even query count is $N^* = T_{\text{setup}}/(T_{\text{baseline}} - T_{\text{query}}) \leq 1.1$ for both evaluated domains; RSCE is net-beneficial after a single additional query.

3.2 Fact Block Construction

Mean-pooling preserves global semantic directions but irreversibly destroys token identities and sequential precision—the factual grounding required for multi-hop entity resolution. The fact block F restores this precision without reintroducing the full document's compute cost. For QA: capitalised multi-word proper nouns, four-digit year tokens, and numeric values; top-15 by appearance prepended as `Facts:` e_1`;` e_2`; ...` For code: BM25Okapi retrieval over function signatures, class declarations, import statements, and SCREAMING_SNAKE_CASE constants; top-5 by relevance to the last 200 local-context characters. The two channels are structurally complementary: C provides the global semantic frame that contextualizes the query, while F supplies precise named-entity anchors that attention heads can resolve against.

3.3 Injection Layer Calibration

We determine $f(M)$ per architecture by sweeping layers in $[n_{\text{layers}}/4,\ 0.85 \cdot n_{\text{layers}}]$ at stride 2 on 10 calibration examples. Table 1 summarizes the results. Optimal depth varies substantially across architectures: Mistral's sliding-window attention forces rapid shallow consolidation of global semantics (25% depth), while larger full-attention models require deeper processing before residual states encode sufficient content (47–62%). This is consistent with observations that attention mechanism design governs how quickly semantic information propagates through depth (Gromov et al., 2024).

Table 1: Calibrated injection layers. †DeepSeek-R1-Distill uses layer 29 (60%) for code. ‡Calibrated on RepoBench-C EditSim ($n = 20$).

Model	n_{layers}	d_M	$f(M)$	Depth	Calib. Score
LLaMA-3.1 8B	32	4096	14	44%	0.096
Qwen2.5 7B	28	3584	17	61%	0.105
Mistral Small 24B	40	5120	10	25%	0.312
DeepSeek-R1 14B†	48	5120	12	25%	0.085
DeepSeek-LLM 67B‡	95	8192	45	47%	0.382
LLaMA-3.1 70B‡	80	8192	50	62%	0.371

4 Experimental Design

We use six instruction-tuned architectures in bfloat16 on NVIDIA H100 80GB GPUs: LLaMA-3.1 8B, Qwen2.5 7B, DeepSeek-R1-Distill 14B, Mistral Small 24B, DeepSeek-LLM 67B Chat, and LLaMA-3.1 70B Instruct. All six are used for RepoBench-C; QA evaluation uses five (all except DeepSeek-LLM 67B, which lacks instruction-following calibration for the QA prompt format). Code is available at `https://anonymous.4open.science/r/RSCE-2E4C/`.

We sample $n = 108$ QA examples (HotpotQA: 17, 2WikiMultiHopQA: 91), filtering for 200–12,000-word contexts (Bai et al., 2024). All five models and all baselines are evaluated on identical samples. Average baseline length is ≈8,700 tokens vs. ≈52 for Vec+F (>99% reduction). Metrics: SQuAD-style Token F1 and Exact Match substring containment.

For RepoBench-C, we use $n = 200$ samples per model from `tianyang/repobench_python_v1.1 cross_file_first`, seed 42 (Liu et al., 2024c). RSCE compresses only the cross-file context; local context passes verbatim. Metric: character-level Edit Similarity.

LongLLMLingua (Jiang et al., 2024) uses a separate `Llama-2-7b-hf` compressor (standard EMI configuration). EHPC (Fei et al., 2025) is implemented in NMI mode with targeted forward hooks and top-8 evaluator heads per model from a 50-probe NIAH pilot. Both are evaluated at 4×, 10×, and token-matched budgets (≈52 tokens QA / ≈963 tokens code). All methods share identical prompt templates and generation parameters (greedy, `max_new_tokens=50`). See Appendix A.

5 FINDINGS

5.1 QA: DUAL-CHANNEL MECHANISM AND CROSS-METHOD COMPARISON

Table 2 reports all conditions for LLaMA-3.1-8B. Table 3 extends the matched-compression comparison to all five QA architectures.

Table 2: LLaMA-3.1-8B QA results ($n = 108$, all conditions on identical samples). Horizontal rule separates moderate from extreme compression.

Method	Setting	F1	EM	TokRed
Baseline	full context	0.410	67.6%	0%
EHPC	4× (2,175 tok)	0.400	67.6%	74.5%
LongLLMLingua	4× (2,175 tok)	0.377	62.0%	74.8%
EHPC	10× (870 tok)	0.409	60.2%	89.4%
LongLLMLingua	10× (870 tok)	0.294	50.0%	88.6%
EHPC	2,048 tok	0.367	71.3%	49.1%
Q-only	no context	0.286	35.2%	99.4%
RSCE Vec	—	0.252	29.6%	99.4%
LongLLMLingua	Matched (52 tok)	0.209	32.4%	95.9%
EHPC	Matched (52 tok)	0.334	48.1%	98.0%
RSCE Vec+F	$O(1)$ amortized	**0.333**	29.6%	**99.4%**

Table 3: Cross-model matched-compression comparison ($n = 108$ all models). Retention = Vec+F F1 / Baseline F1. EHPC and LongLLMLingua TokRed ≈98% and ≈96% respectively; RSCE ≈99.4%. ⋆Qwen inverse fact-block effect (Vec+F < Q-only); see text. †No LongLLMLingua run for LLaMA-70B.

Model	Baseline	RSCE Vec+F	Retention	EHPC	LLMLingua
LLaMA-3.1 8B	0.410	0.333	81%	0.334	0.209
Qwen2.5 7B	0.153	0.094⋆	61%	0.145	0.078
DeepSeek-R1 14B	0.342	0.214	63%	0.214	0.172
Mistral 24B	0.548	0.353	64%	0.442	0.235
LLaMA-3.1 70B	0.604	0.365	60%	0.539	—†

Four findings emerge from these results.

Vec injection suppresses parametric memory in most architectures. On LLaMA-8B (0.252 vs. Q-only 0.286), Mistral-24B (0.230 vs. 0.243), and LLaMA-70B (0.278 vs. 0.302), Vec falls below Q-only. DeepSeek-R1-14B shows Vec tied with Q-only (0.165 = 0.165). Only Qwen shows Vec marginally above Q-only (0.126 vs. 0.111)—yet its fact block still fails (Vec+F = 0.094 < Q-only). This is consistent with dual-pathway interference (Shi et al., 2023; Mallen et al., 2023): the injected vector engages contextual retrieval circuitry in the model's MLP knowledge storage layers (Meng et al., 2022), suppressing parametric recall without providing sufficient token-level grounding to compensate.

The fact block is constitutive for four of five models. Vec+F > Vec for LLaMA-8B, DeepSeek-14B, Mistral-24B, and LLaMA-70B, and Vec+F > Q-only for all four—confirming the dual-channel design recovers quality that neither channel alone provides. The exception is Qwen-7B (Vec+F = 0.094 < Q-only = 0.111), where the `Facts:` prefix acts as an answer-space constraint rather than contextual grounding. Instruction-tuned models can exhibit large performance swings from prefix formatting (Sclar et al., 2024); Qwen's multi-objective RLHF alignment appears particularly sensitive to fact-list format in the 2WikiMultiHopQA task (where the effect is concentrated: Vec+F = 0.063 vs. Q-only = 0.088). Fact-block formatting should be instruction-template-aware in deployment.

At moderate compression, EHPC substantially outperforms RSCE. On LLaMA-8B, EHPC at 4× achieves F1 = 0.400, essentially matching the full-context baseline of 0.410, while RSCE reaches only

0.333. Token-selection methods retain natural-language coherence that the model can attend over; RSCE's distributed encoding discards sequential structure critical for multi-hop reasoning chains.

At extreme compression, the picture is model-dependent. On LLaMA-8B and DeepSeek-14B, RSCE Vec+F is essentially tied with EHPC (0.333/0.334 and 0.214/0.214). On Mistral-24B and LLaMA-70B, EHPC's advantage grows substantially (0.442 vs. 0.353 and 0.539 vs. 0.365). We attribute this to a capacity-scaling effect: larger models have stronger in-context reasoning that better leverages the sparse structural tokens EHPC retains. RSCE's distributed semantic vector does not benefit from this capacity scaling in the same way. RSCE retains its distinct advantages: zero per-query context prefill, 100% operational reliability, and strict $O(1)$ amortized cost. LongLLMLingua underperforms both methods at matched budgets across all tested models.

EM degradation under RSCE is structural. EM penalizes concise responses regardless of semantic correctness; compressed representations elicit shorter, targeted answers lacking the verbosity needed for substring containment. Token F1 is the appropriate primary metric.

5.2 Code Completion: Compression-as-Retrieval vs. Semantic Encoding

Tables 4 and 5 report RepoBench-C results.

Table 4: RepoBench-C RSCE results. †200 valid examples. DeepSeek-67B's baseline (0.147) is suppressed by its 4K context window; RSCE injection bypasses this constraint (+0.217).

		EditSim		ΔEditSim		
Model	Params	Vec	Vec+F	Vec	Vec+F	TokRed
LLaMA-3.1 8B	8B	0.317	0.319	−0.031	−0.029	81.2%
Qwen2.5 7B	7B	0.365	0.355	−0.028	−0.038	81.2%
DeepSeek-R1 14B	14B	0.322	0.330	−0.022	−0.014	81.2%
Mistral 24B	24B	0.381	0.377	−0.016	−0.020	81.1%
DeepSeek-LLM 67B	67B	0.352	0.364	+0.205	+0.217	81.2%
LLaMA-3.1 70B†	70B	0.348	0.352	−0.024	−0.021	80.3%
All 6 (avg)	—	0.348	0.350	+0.014	+0.017	81.0%
Excl. 67B (avg)	—	0.347	0.348	−0.024	−0.024	81.0%

Baseline EditSim: LLaMA-8B 0.348, Qwen 0.392, DeepSeek-14B 0.344, Mistral 0.397, DeepSeek-67B 0.147, LLaMA-70B 0.372.

Table 5: RepoBench-C baseline comparison. LongLLMLingua reported over 3 models (LLaMA-8B, Qwen-7B, DeepSeek-14B). EHPC reported over 4 models, *successful samples only* (53–65% success rate; see Appendix). RSCE is the only method achieving the target 81% compression at 100% reliability.

Method	Setting	Coverage	Avg EditSim	Actual TokRed
Baseline	full context	4 models	0.370*	0%
LongLLMLingua	6×	3 models, 199/200 OK	0.639	68.9%
LongLLMLingua	Matched	3 models, 199/200 OK	0.645	65.0%
EHPC	6×	4 models, ∼59% OK	0.425†	61%
EHPC	Matched	4 models, ∼57% OK	0.445†	43%
RSCE Vec+F	$O(1)$ amortized	4 models, **100%** OK	**0.347***	**81.2%**

*Excl. DeepSeek-67B. †Over successful samples only; actual per-instance numbers higher on easy subset.

LongLLMLingua substantially outperforms both RSCE and the full-context baseline on code (EditSim ≈0.64 vs. baseline ≈0.37 for the 3 tested models). For structured code contexts, perplexity-based compression acts as effective relevance filtering. Full-context baselines suffer from attention dilution across the ≈11,485-token average cross-file context (Liu et al., 2024a). LongLLMLingua's question-aware contrastive perplexity identifies syntactically surprising tokens—function signatures, type annotations, specific identifiers—precisely the tokens required for code completion. Performance

is highest when relevant information appears at the beginning or end of context (Liu et al., 2024a); compression and reordering place high-surprisal tokens in exactly those favored positions. This compression-as-retrieval mechanism has no analog in RSCE's distributed semantic encoding, which averages away the structural precision that code completion demands.

EHPC's 40–50% compression failure rate on RepoBench-C (attention memory budget exceeded for long code sequences) makes its EditSim over successful samples unrepresentative of the full distribution. RSCE achieves the 81% compression target at 100% reliability—the only method to do so—with a scale-invariant 2–4 EditSim percentage-point overhead (Sec. 5.3).

DeepSeek-LLM 67B inverts the pattern: its 4K context window truncates the average 11,485-token cross-file context, suppressing the baseline to 0.147. RSCE bypasses the window constraint entirely, yielding Vec+F = 0.364 (+0.217). Activation injection can thus extend effective context for architecturally constrained models.

5.3 Scale Invariance on Code

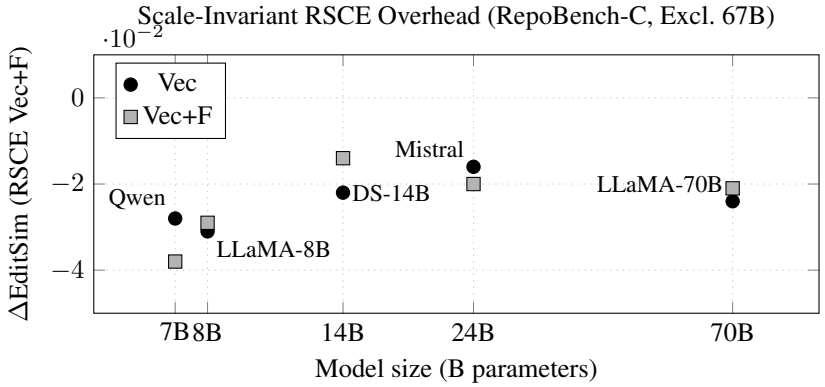

Figure 1: RSCE Vec+F overhead on RepoBench-C (excl. DeepSeek-67B context outlier) forms a flat band of -0.038 to -0.014 ΔEditSim across an order of magnitude in parameter count. The overhead is architecture-agnostic, enabling principled deployment planning independent of model size.

Figure 1 shows that RSCE code overhead is flat across 7B–70B parameters. This scale invariance—which does not hold for QA F1 retention (60–81%, driven by instruction-format sensitivity and model-specific parametric knowledge quality)—indicates that code compression fidelity is governed by layer geometry and attention structure captured in the calibration sweep, not raw parameter count.

6 Discussion

Dual-pathway interference. The Vec $<$ Q-only or Vec $=$ Q-only finding across 4 of 5 architectures provides mechanistic insight into how LLMs process injected context signals. We consider two hypotheses. Under the attention override hypothesis (Longpre et al., 2021; Mallen et al., 2023): injecting C biases representations toward contextual retrieval mode, activating the attention heads responsible for reading external context while suppressing MLP-based parametric recall circuits (Meng et al., 2022; Dai et al., 2022). The model anticipates finding the answer by attending to context, but mean-pooling has destroyed token-level resolution, so the retrieval attempt fails. The fact block F resolves the tension by providing exact token anchors. Under the distributional shift hypothesis: the additive vector pushes residual stream norms outside the training distribution, causing feed-forward network misfires on factual recall neurons (Geva et al., 2021). The consistency of Vec $\leq$ Q-only across architectures with different attention mechanisms (SWA, GQA, full attention) and training objectives (SFT, DPO, CoT distillation) suggests the attention override mechanism is more likely—distributional shift would vary more by architecture.

This interference is absent in behavioral steering (Liu et al., 2024b; Todd et al., 2024), where injected vectors augment task performance monotonically. The difference is structural: behavioral vectors

encode procedural abstractions that operate alongside default processing; factual content vectors engage a separate context reading pathway that competes with parametric recall.

Capacity-scaling of EHPC vs. RSCE. The growing EHPC advantage on larger models (from $+0.001$ on LLaMA-8B to $+0.174$ on LLaMA-70B at matched compression) reflects an asymmetry in how model scale interacts with the two compression paradigms. Larger models reason more effectively from sparse token signals (Brown et al., 2020): given 52 carefully selected tokens, a 70B model can infer relational chains and bridge multi-hop questions far more effectively than an 8B model. RSCE's distributed semantic vector provides a fixed-quality activation shift that does not benefit from increased reasoning capacity—the model cannot read more from a vector than is encoded in it. RSCE's advantage over EHPC is thus most pronounced on smaller architectures where token-level reasoning is more limited.

Domain-dependent optimality. Code completion requires structural precision: correct function signatures, exact type annotations, specific identifier names—low-redundancy facts distributed across the cross-file context. Perplexity-based token selection identifies these structurally surprising tokens automatically. QA at extreme compression requires semantic framing: the model must know it is answering about a particular entity, time period, or relationship chain, even if precise token-level detail is unavailable. RSCE's distributed vector encoding is better suited to preserving this semantic frame across the full 8,700-token context. A practical architecture combining both approaches—RSCE for the persistent static context frame, LongLLMLingua or EHPC for dynamically selected snippets—would likely outperform either alone.

Deployment guarantees. Regardless of quality comparisons, RSCE offers guarantees that token-selection methods cannot: zero per-query context prefill, 100% compression reliability, and strictly $O(1)$ amortized per-query cost. For production systems serving many queries over static contexts, these guarantees dominate quality differences at moderate scale. The amortization break-even of $N^* \leq 1.1$ means RSCE is net-beneficial from the second query onward.

7 Conclusion

We have presented RSCE, a training-free, $O(1)$ amortized context encoding method with zero per-query context forward pass and 100% operational reliability. Across five decoder-only architectures, vector injection alone suppresses parametric recall below the no-context baseline—a dual-pathway interference effect absent in behavioral steering—while the paired fact block recovers 60–81% of full-context F1 at $\sim$99% token reduction. At extreme compression, RSCE is competitive with EHPC on smaller architectures (LLaMA-8B, DeepSeek-14B) while a capacity-scaling effect gives EHPC a growing quality advantage on larger models. On RepoBench-C, LongLLMLingua substantially outperforms both methods via compression-as-retrieval; RSCE uniquely offers 81% compression at 100% reliability. The scale-invariant 2–4 point code overhead from 7B to 70B enables architecture-agnostic deployment planning. These results establish RSCE as suited for high-throughput multi-query deployments over static contexts where the $O(1)$ amortization guarantee and operational reliability outweigh EHPC's quality advantage, particularly for architectures below $\sim$24B parameters.

Limitations

QA evaluations use the same $n = 108$ sample set across all five models; however, the small HotpotQA subset ($n = 17$) remains insufficient for per-task statistical confidence. We use mean-pooling and $\alpha = 1.0$ without systematic ablation of pooling strategy, scale factor, or positional targeting. Qwen's inverse fact-block effect requires instruction-format-aware fact-block construction not yet implemented. The dual-pathway interference mechanism is hypothesized but not directly probed via residual norm measurement or attention pattern analysis. Mean-pooling destroys sequential structure, limiting RSCE to contexts where ordering is not the primary inference target. EHPC's RepoBench-C failure rate warrants investigation as a separate research question.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Li, Zhiyuan Liu, and Jie Tang. LongBench: A bilingual,

multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 2020.

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. xRAG: Extreme context compression for retrieval-augmented generation with one token. In *Advances in Neural Information Processing Systems*, 2024.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. 2022.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

Weizhi Fei, Xueyan Niu, Guoqing Xie, Yingqing Liu, Bo Bai, and Wei Han. Efficient prompt compression with evaluator heads for long-context transformer inference. In *Advances in Neural Information Processing Systems*, 2025.

Yair Feldman and Yoav Artzi. Simple context compression: Mean-pooling and multi-ratio training. *arXiv preprint arXiv:2510.20797*, 2025.

Tao Ge, Jing Hu, Lei Guo, Guanghao Yu, Shuming Nie, Hao Li, Xia Dong, and Furu Huang. In-context autoencoder for context compression in a large language model. In *International Conference on Learning Representations*, 2024.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. The unreasonable ineffectiveness of the deeper layers. *arXiv preprint arXiv:2403.17887*, 2024.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, 2023a.

Yucheng Li, Bo Dong, Chenghua Lin, and Frank Guerin. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023b.

Yuhong Li, Yingbing Han, Ziyue Zhao, and Dacheng Du. SnapKV: LLM knows what you are looking for before generation. In *Advances in Neural Information Processing Systems*, 2024.

Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. Prompt compression for large language models: A survey. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics*, 2025.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 2024a.

Sheng Liu, Haotian Ye, Lei Xing, and James Zou. In-context vectors: Making in-context learning more effective and controllable through latent space steering. In *International Conference on Machine Learning*, 2024b.

Tianyang Liu, Canwen Xu, and Julian McAuley. RepoBench: Benchmarking repository-level code auto-completion systems. In *International Conference on Learning Representations*, 2024c.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, 2022.

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

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *International Conference on Machine Learning*, 2024.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *International Conference on Learning Representations*, 2024.

Adam S. Shai, Sarah E. Marzen, Lucas Teixeira, Alexander Gietelink Oldenziel, and Paul M. Riechers. Transformers represent belief state geometry in their residual stream. In *Advances in Neural Information Processing Systems*, 2024.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, 2023.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron Wallace, and David Bau. Function vectors in large language models. In *International Conference on Learning Representations*, 2024.

Alexander Turner, Lisa Thiergart, Gavin Udell, David Purves, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*, 2023.

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.

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. H_2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, 2023.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Hua, Josephine Li, Amanda Askell, Anna Jones, Nat DasSarma, Ethan Perez, Saurabh Ghaisas, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A Baseline Implementation Details

Both baselines share infrastructure with RSCE: identical sample indices, prompt templates (`Context: / Question: / Answer:` for QA; `# Cross-file context: / # Current file: / # Complete the next line:` for code), and generation parameters (greedy, `max_new_tokens=50`, bfloat16, H100-80GB). Failed compression attempts record uncompressed token counts, preserving benchmark integrity.

LongLLMLingua (Jiang et al., 2024) uses `NousResearch/Llama-2-7b-hf` as a separate compressor (EMI); QA uses `condition_compare=True`, `rank_method=longllmlingua`, `reorder_context=sort`; code disables question-aware mode. Three progressively relaxed parameter bundles are attempted on split-document and merged-context inputs.

EHPC (Fei et al., 2025) is implemented from the paper specification in NMI mode (same model for both compression and inference, vs. the paper's EMI using GPT-3.5-Turbo; our scores are expected to be modestly lower): a 50-probe NIAH pilot per model selects top-8 evaluator heads; targeted forward hooks capture only the evaluator layer (8 GiB budget); observation-window rows are summed and smoothed with a 1D pooling kernel (size 3); prompts reconstruct from retained token IDs with character-offset accounting to prevent BPE artifacts. EHPC's RepoBench-C failures (40–50%) stem from the attention memory budget being exceeded on long code sequences; all reported EditSim values are computed over successful samples only and are noted explicitly throughout the paper.

B Per-Task QA Breakdown

Table 6: RSCE per-task QA breakdown for all five models. HotpotQA $n = 17$; 2WikiMultiHopQA $n = 91$.

Model	Task	Base F1	Q-only F1	Vec F1	Vec+F F1	Retention
LLaMA-8B	HotpotQA	0.688	0.378	—	0.493	72%
LLaMA-8B	2WikiMQA	0.358	0.269	—	0.303	85%
Mistral-24B	HotpotQA	0.622	0.425	0.455	0.554	89%
Mistral-24B	2WikiMQA	0.534	0.208	0.188	0.316	59%
DeepSeek-14B	HotpotQA	0.495	0.109	0.118	0.328	66%
DeepSeek-14B	2WikiMQA	0.313	0.175	0.174	0.193	62%
LLaMA-70B	HotpotQA	0.627	0.521	0.521	0.583	93%
LLaMA-70B	2WikiMQA	0.600	0.261	0.232	0.325	54%
Qwen-7B	HotpotQA	0.226	0.231	0.223	0.259	115%*
Qwen-7B	2WikiMQA	0.139	0.088	0.108	0.063	45%

*Vec+F exceeds baseline on HotpotQA due to Qwen's low baseline (0.226).

HotpotQA consistently shows higher RSCE retention than 2WikiMultiHopQA. HotpotQA requires bridging two supporting facts—a structure the residual encoding's global semantic frame can partially represent. 2WikiMultiHopQA requires longer multi-hop chains where sequential token precision matters more. The LLaMA-70B HotpotQA result (93% retention) demonstrates that at large scale, RSCE can approach full-context performance on simpler multi-hop tasks.