RSCE: Training-Free Residual Stream Encoding for Persistent Context Amortization

Adam Kamel*
University of Waterloo
atkamel@uwaterloo.ca

Eric Xu*
University of Waterloo
e67xu@uwaterloo.ca

Abstract

A central question in the knowledge lifecycle of language models is how externally injected signals interact with parametric memory accumulated during pretraining. We address this through Residual Stream Context Encoding (RSCE), a training-free method that encodes a context document ctx into a single vector $C \in \mathbb{R}^{d_M}$ via mean-pooling residual stream activations at a calibrated intermediate layer, then injects C as an additive shift at query time. This replaces $O(|T(\text{ctx})|)$ attention prefill with an $O(1)$ operation and reveals a previously undescribed *dual-pathway interference* effect: vector injection alone suppresses parametric recall *below* the question-only baseline across four of five tested architectures. This finding—absent in behavioral activation steering—provides mechanistic evidence that LLMs maintain separate contextual-retrieval and parametric-recall pathways that compete when externally injected signals are semantically rich but token-precision deficient. A dual-channel design pairing C with a compact explicit fact block F resolves this tension. We evaluate five decoder-only architectures (7B–70B) on multi-document QA (LongBench, $n = 108$) and six on cross-file code completion (RepoBench-C), comparing against LongLLMLingua and EHPC. 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. RSCE is the only method achieving 81% compression at 100% operational reliability on code.

1 Introduction

Large language models acquire substantial factual knowledge during pretraining, encoding it as distributed patterns in model weights (Meng et al., 2022; Dai et al., 2022). Retrieval-augmented generation (RAG) architectures inject *external* knowledge at inference time by prepending long context documents ctx to user queries. A foundational question in the knowledge lifecycle of such systems is: how does externally injected knowledge interact with the model's internal parametric memory? When the two sources agree, models can exploit both; when they conflict or the external signal is imprecise, interference can arise (Longpre et al., 2021; Mallen et al., 2023; Shi et al., 2023). Understanding this interaction has direct implications for RAG faithfulness and knowledge utilization reliability.

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. This yields an $O(1)$ amortized per-query cost for static contexts and simultaneously creates a controlled setting for probing how vector-encoded external knowledge interacts with parametric memory. A key mechanistic finding emerges: across four of five tested architectures, injecting C alone drives model performance *below* the no-context (question-only) baseline. This **dual-pathway interference effect**—absent in behavioral activation steering, where injected vectors consistently augment performance—provides evidence that LLMs maintain distinct contextual-retrieval and parametric-recall circuits that compete when the injected signal engages retrieval circuitry without sufficient token-level grounding. A dual-channel design pairing C with a minimal explicit fact block F resolves this tension by supplying precise named-entity anchors for attention heads to resolve against.

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

*Equal contribution.

Proceedings of the 4th Workshop on Towards Knowledgeable Foundation Models (KnowFM 2026), pages 138–146

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. Activation steering methods (Liu et al., 2024b; Todd et al., 2024) inject vectors to encode task demonstrations—but not factual document content. RSCE applies the same injection mechanism to a qualitatively different problem, uncovering a failure mode with no analog in behavioral steering.

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) **dual-pathway interference**, confirmed across five architectures, providing direct evidence for distinct contextual-retrieval and parametric-recall knowledge pathways in decoder-only transformers; (3) a **cross-model comparison** at matched extreme compression revealing a capacity-scaling effect where EHPC's advantage grows with model size; (4) a **compression-as-retrieval** explanation for LongLLMLingua's strong code performance (Liu et al., 2024a).

2 Related Work

2.1 Parametric vs. Contextual Knowledge Interaction

Longpre et al. (2021) and Mallen et al. (2023) show that LLMs often override correct parametric knowledge when provided with conflicting context, while Shi et al. (2023) demonstrates that irrelevant context systematically degrades reasoning. The mechanisms have been partially localized: Meng et al. (2022) and Dai et al. (2022) implicate feed-forward network layers as primary parametric memory storage, while attention heads read external context. RSCE's dual-pathway interference finding (Section 5) extends this picture: injecting a vector that engages contextual retrieval circuitry without token-level resolution suppresses parametric recall in a manner consistent with an attention-override mechanism.

2.2 Context Compression for RAG

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" locating important tokens at 0.88s latency—current state-of-the-art training-free hard-prompt compression. All require a per-query prefill (Li et al., 2025). Zhang et al. (2023), Xiao et al. (2024), and Li et al. (2024) evict KV entries within a single pass but produce no 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 and Artzi (2025) validates mean-pooling of hidden states as superior to alternative soft-compression architectures—but requires training, whereas RSCE is entirely training-free.

2.3 Activation Injection and the Residual Stream

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. RSCE shares this mechanism but targets factual document content, uncovering the parametric-memory interference effect described above.

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, producing a document-level representation analogous to a belief state (Shai et al., 2024).

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 $N^* \leq 1.1$ means RSCE is net-beneficial after a single additional query. Our code is available at `https://anonymous.4open.science/r/RSCE-2E4C/`.

3.2 Fact Block Construction

Mean-pooling preserves global semantic directions but irreversibly destroys token identities and sequential precision. 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. C provides the global semantic frame; 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: Mistral's sliding-window attention forces rapid shallow consolidation (25% depth), while larger full-attention models require deeper processing before residual states encode sufficient content (47–62%), consistent with observations that attention mechanism design governs semantic propagation through depth (Gromov et al., 2025).

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

Model	n_L	d_M	$f(M)$	Depth	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 (DeepSeek-LLM 67B lacks QA instruction-following calibration).

We sample $n = 108$ QA examples (HotpotQA: 17, 2WikiMultiHopQA: 91), filtering for 200–12,000-word contexts (Bai et al., 2024). Average baseline length ≈8,700 tokens vs. ≈52 for Vec+F (>99% reduction). Metrics: SQuAD-style Token F1 and Exact Match. For RepoBench-C, $n = 200$ samples per model from `tianyang/repobench_python_v1.1` `cross_file_first`, seed 42 (Liu et al., 2024c). Metric: character-level Edit Similarity.

LongLLMLingua (Jiang et al., 2024) uses `Llama-2-7b-hf` as a separate compressor (EMI). EHPC (Fei et al., 2025) is implemented in NMI mode with top-8 evaluator heads per model from a 50-probe NIAH pilot. Both 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 to all five QA architectures.

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

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

Table 3: Cross-model matched-compression ($n = 108$). Ret. = Vec+F F1 / Baseline F1. EHPC TokRed ≈98%; LLMLingua ≈96%; RSCE ≈99.4%. ⋆Qwen inverse fact-block effect; see text. †No LLMLingua run for LLaMA-70B.

Model	Base	Vec+F	Ret.	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. This result speaks directly to the interaction between externally injected knowledge representations and internally stored parametric knowledge. On LLaMA-8B (F1 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 ties (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 (Meng et al., 2022; Dai 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 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. Instruction-tuned models exhibit large performance swings from prefix formatting (Sclar et al., 2024); Qwen's RLHF alignment appears particularly sensitive to fact-list format in 2WikiMultiHopQA (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 (near the 0.410 baseline), while RSCE reaches only 0.333. Token-selection methods retain natural-language coherence; RSCE's distributed encoding discards sequential structure critical for multi-hop chains.

At extreme compression, results are model-dependent. On LLaMA-8B and DeepSeek-14B, RSCE Vec+F ties EHPC (0.333/0.334 and 0.214/0.214). On Mistral-24B and LLaMA-70B, EHPC's advantage grows (0.442 vs. 0.353 and 0.539 vs. 0.365). We attribute this to a capacity-scaling effect: larger models reason more effectively from sparse token signals (Brown et al., 2020); RSCE's fixed-quality vector does not benefit from increased reasoning capacity. RSCE retains distinct advantages: zero per-query context prefill, 100% reliability, and strict $O(1)$ amortized cost. LongLLMLingua underperforms both methods at matched budgets across all models.

EM degradation under RSCE is structural: compressed representations elicit shorter answers lacking the verbosity 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.

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. Compression and reordering place high-surprisal tokens in the favored beginning/end positions (Liu et al., 2024a), a compression-as-retrieval mechanism with no analog in RSCE's distributed semantic encoding.

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

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

		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%

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 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	$O(1)$	4 models, **100% OK**	**0.347***	**81.2%**

*Excl. DeepSeek-67B. †Over successful samples only.

turally constrained models.

5.3 Scale Invariance on Code

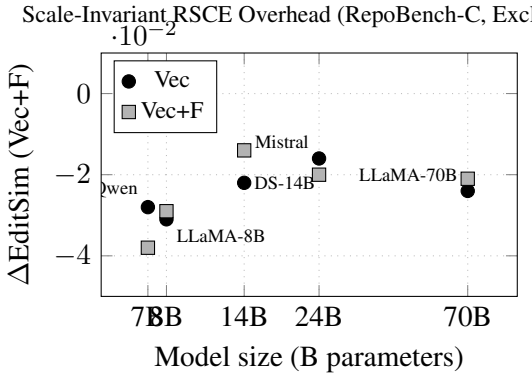

Figure 1: RSCE Vec+F overhead on RepoBench-C (excl. DeepSeek-67B) forms a flat band of -0.038 to -0.014 ΔEditSim across an order of magnitude in parameter count, enabling architecture-agnostic deployment planning.

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 and knowledge routing. The Vec $\leq$ Q-only finding across 4 of 5 architectures provides mechanistic insight into how LLMs route externally injected knowledge signals relative to internally stored parametric knowledge. Under the *attention override hypothesis*: injecting C biases representations toward contextual retrieval mode, activating context-reading attention heads while suppressing MLP-based parametric recall (Meng et al., 2022; Dai et al., 2022). The model anticipates finding the answer in context, but mean-pooling has destroyed token-level resolution so the retrieval attempt fails. The fact block F resolves this by providing exact token anchors. Under the *distributional shift hypothesis*: the additive vector pushes residual stream norms outside the training distribution, causing feed-forward 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) favors the attention override hy-

pothesis, as 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 performance monotonically. Behavioral vectors encode procedural abstractions alongside default processing; factual content vectors engage a separate context-reading pathway that competes with parametric recall. This has direct implications for knowledge utilization faithfulness in RAG systems.

Implications for knowledge injection design. Our results reveal a failure mode in knowledge injection via activation space: even when a vector faithfully encodes document-level semantics, its injection can suppress the model's own parametric knowledge if the signal format mismatches the model's context-reading circuitry expectations. This suggests a design principle: *external knowledge representations must provide sufficient token-level resolution to satisfy context-reading circuitry, or they will actively degrade reliance on parametric knowledge rather than complement it.* The dual-channel RSCE design operationalises this principle in a training-free setting.

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 paradigms. Larger models reason more effectively from sparse token signals (Brown et al., 2020); RSCE's fixed-quality activation shift does not benefit from increased reasoning capacity. RSCE's advantage over EHPC is therefore most pronounced on architectures below $\sim$24B parameters.

Domain-dependent optimality. Code completion requires structural precision (correct function signatures, exact identifiers) that perplexity-based token selection identifies automatically. QA at extreme compression requires semantic framing that RSCE's distributed vector preserves better. A hybrid combining RSCE for persistent static context framing with LongLLMLingua or EHPC for dynamic snippets would likely outperform either alone.

Deployment guarantees. RSCE offers guarantees that token-selection methods cannot: zero per-query context prefill, 100% compression reliability, and strictly $O(1)$ amortized cost. The 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 while a capacity-scaling effect gives EHPC a growing quality advantage on larger models. On RepoBench-C, LongLLMLingua substantially outperforms both via compression-as-retrieval; RSCE uniquely offers 81% compression at 100% reliability with scale-invariant 2–4 point overhead from 7B to 70B.

Beyond its practical contributions, the dual-pathway interference finding constitutes a concrete, reproducible probe into the interaction between externally injected knowledge and parametric memory in decoder-only transformers. The result that semantically rich but token-precision-deficient signals actively suppress parametric recall—rather than degrading gracefully—illuminates a previously undescribed mode of knowledge conflict in LLMs, with direct implications for the design of faithful RAG and knowledge augmentation systems.

Limitations

QA evaluations use the same $n = 108$ sample set across all five models; the small HotpotQA subset ($n = 17$) is 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.

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

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*.

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

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

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

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, and 6 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*.

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

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

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. 2024. In-context autoencoder for context compression in a large language model. In *International Conference on Learning Representations*.

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

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. 2025. The unreasonable ineffectiveness of the deeper layers. In *International Conference on Learning Representations*.

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

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. 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*.

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

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

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

Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. 2025. 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*.

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

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

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

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

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. 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*.

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

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

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

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. 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*.

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

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

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

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

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

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. H_2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*.

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. 2023. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*.

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 in NMI mode (same model for compression and inference): 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.

B Per-Task QA Breakdown

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

Table 6: RSCE per-task QA breakdown for all five models. HotpotQA $n = 17$; 2WikiMultiHopQA $n = 91$. *Vec+F exceeds baseline on HotpotQA due to Qwen's low baseline (0.226).

Model	Task	Base F1	Q-only	Vec	Vec+F	Ret.
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%

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.