InfiniteICL: Breaking the Limit of Context Window Size via Long Short-term Memory Transformation*

Bowen Cao◇ **Deng Cai**† **Wai Lam**◇
◇ The Chinese University of Hong Kong
bwcao@link.cuhk.edu.hk, thisisjcykcd@gmail.com, wlam@se.cuhk.edu.hk

Abstract

In-context learning (ICL) is critical for large language models (LLMs), but its effectiveness is constrained by finite context windows, particularly in ultra-long contexts. To overcome this, we introduce **InfiniteICL**, a framework that parallels context and parameters in LLMs with short- and long-term memory in human cognitive systems, focusing on transforming temporary context knowledge into permanent parameter updates. This approach significantly reduces memory usage, maintains robust performance across varying input lengths, and theoretically enables infinite context integration through the principles of context knowledge elicitation, selection, and consolidation. Evaluations demonstrate that our method reduces context length by 90% while achieving 103% average performance of full-context prompting across fact recall, grounded reasoning, and skill acquisition tasks. When conducting sequential multi-turn transformations on complex, real-world contexts (with length up to 2M tokens), our approach surpasses full-context prompting while using only 0.4% of the original contexts. These findings highlight InfiniteICL's potential to enhance the scalability and efficiency of LLMs by breaking the limitations of conventional context window sizes.

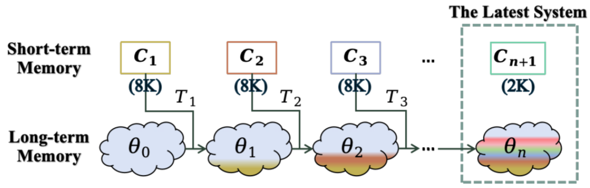

Figure 1: The core idea of our framework. The context window is refreshed after each transformation ($T_i : \theta_{i-1} + C_i \rightarrow \theta_i$), allowing infinite context input in a streaming fashion.

1 Introduction

In-context learning (ICL) has emerged as a cornerstone capability of large language models (LLMs), allowing training-free and user-friendly customization (Liu et al., 2024; Yang et al., 2024; Team et al., 2024). This ability is critical for real-world deployment in scenarios such as deep research integrating web information and lifelong learning (Zheng et al., 2025) through user interactions.

However, the effectiveness of ICL is constrained by the finite context windows of the Transformer architecture (Waswani et al., 2017), typically 8K-128K tokens. This limitation manifests through two compounding factors: (*i*) the quadratic complexity scaling of attention mechanism and the linear growth of KV cache memory (Liu et al., 2024), which impose prohibitive hardware infrastructure requirements for long-context deployment, and (*ii*) diminishing performance returns observed in tasks requiring extended contexts, such as many-shot learning (Agarwal et al., 2024) and cross-document reasoning (Bai et al., 2024). These constraints create a paradox: expanding context capacity inflates computational costs disproportionately while delivering marginal accuracy gains.

In response to these challenges, we propose a novel perspective that frames the dichotomy between context and parameters in LLMs as analogous to short-term and long-term memory in human cognitive systems (Cowan, 2008). Specifically, we posit that contexts function as short-term memory, capturing transient information relevant to the current input, while parameters serve as long-term memory, encoding accumulated knowledge over time. Building on this analogy, our approach focuses on transforming temporary context knowledge into permanent parameter updates, as depicted in Figure 1. This transformation draws inspiration from the human circadian rhythm, where daily ex-

*The work described in this paper is substantially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: 14200620).
†Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2025, pages 11402–11415
July 27 - August 1, 2025

periences are consolidated into long-term memories during sleep (Klinzing et al., 2019). It is further supported by recent studies demonstrating that ICL can be interpreted as meta-gradient updates (Dai et al., 2023; Von Oswald et al., 2023). Additionally, our method falls within the scope of test-time compute scaling (Wei et al., 2022; Wang et al., 2022; Akyürek et al., 2024), as it enhances model capabilities through strategic allocation of compute for test-time training.

Compared to conventional strategies of extending context windows, our approach offers three key advantages: (*i*) it reduces GPU memory usage by converting context into compact parameter updates; (*ii*) it potentially maintains robust performance across varied input lengths, mitigating the scaling issues of attention mechanisms; and (*iii*) it theoretically enables infinite context integration by continuously updating long-term memory, eliminating the need for ever-expanding context retention.

To implement this cognitive-inspired paradigm, we develop a framework with three principled design choices: (*i*) Context knowledge elicitation establishes a unified strategy for diverse scenarios through hybrid prompting - systematically steering the model to generate both task-specific interactions (*e.g.*, summarization, multi-step reasoning chains) and open-ended contextual expansions. This process constructs a transfer set $\mathcal{T}$ that comprehensively encodes contextual knowledge. (*ii*) Path selection optimizes $\mathcal{T}$ into $\mathcal{T}_k$ by retaining top-k interactions with maximum perplexity discrepancy between context-aware and context-free generations, prioritizing knowledge-critical pathways. (*iii*) Memory consolidation transforms these temporary contextual insights into permanent model parameter updates through knowledge distillation.

Our framework is rigorously evaluated through a systematic protocol designed to comprehensively assess its effectiveness across diverse scenarios. We first examine our approach in the single transformation setting, where the model undergoes a one-time conversion of context into long-term memory. When transforming 90% of the context into parameter updates, our method maintains 84% performance for fact recall on Natural Questions, 117% in grounded reasoning (Counterfact and Mquake), and 98% in 300-shot ICL (TREC and NLU), all compared to full-context prompting. These results collectively yield an average performance recovery of 103% relative to full-context prompting, consistently outperforming state-of-the-art context compression methods, with average scores of 70.4 versus 10.4 for LLMLingua-2 (Pan et al., 2024) and 7.3 for SnapKV (Li et al., 2024b).

We further validate our framework in sequential transformation scenarios, where multi-turn parameter updates are conducted with consecutive contexts. On LongBench v2 (Bai et al., 2024), our method achieves superior results to full-context prompting (up to $128K$) while utilizing merely 0.4% of the original contexts. Extended analysis reveals that existing context compression/distillation baselines exhibit progressive performance degradation as context lengths increase (e.g., relative performance declines range from 28% to 82% as context lengths grow from 8K to 64K tokens), with most baselines experiencing performance collapse at 2M tokens. In contrast, our method maintains largely stable performance throughout this length scaling trajectory. Additionally, we carry out systematic ablations to study the choices and effects of the three components (context knowledge elicitation, selection, and consolidation) in our framework.

2 Related Work

Context Compression Some research focuses on reducing input context length to ensure it remains within a manageable computational budget. Xu et al. (2023); Lee et al. (2024) employ summarization models to generate concise scripts. Other approaches remove redundant tokens based on information entropy (Li et al., 2023; Jiang et al., 2023a,b; Pan et al., 2024) or downstream task performance (Jung and Kim, 2024; Huang et al., 2023), the latter requiring task-specific training. Some works focus on learning soft task prompts that encodes contexts into trainable vectors (Wingate et al., 2022; Chevalier et al., 2023; Ge et al., 2023; Kim et al., 2023; Mu et al., 2024). Additionally, there is a line of work that compresses context from the KV perspective, such as reducing KV activations along the length dimension (Li et al., 2024b; Zhang et al., 2023). Although these methods effectively shorten input text to enhance efficiency, they do so at the cost of diminished context integrity, with the loss becoming more pronounced as input texts lengthen.

Context Distillation Some existing works have explored injecting knowledge in prompt to model parameters. However, many of these focus on short contexts (Snell et al., 2022; Askell et al., 2021; Padmanabhan et al., 2024). Others tend to concentrate on specific types of context such as conversation history (Magister et al., 2024), sports documents (Mecklenburg et al., 2024), and task instructions (Choi et al., 2022), or they require additional parameters and depend on specialized pretraining (Muhtar et al., 2024; Chen et al., 2024), which limits their applicability. Additionally, Temp-LoRA (Wang et al., 2024) addresses long-form text generation by storing context information through training the model on the prompt and previously generated content. In contrast, we aim to present a general approach that is applicable to a wide range of scenarios. Our evaluation covers a variety of tasks, especially involving long contexts, to rigorously assess the effectiveness of context distillation.

3 Problem Definition

3.1 Overview

Human cognition processes information through two complementary memory systems: short-term memory for transient signal maintenance and long-term memory for persistent knowledge storage. Inspired by this neurocognitive architecture, we propose a novel perspective for language model memory system coordination, with a particular focus on transforming temporary context knowledge into permanent parameter updates, a process that emulates hippocampal-cortical memory consolidation mechanisms (Squire, 1992).

3.2 Short-term Memory

Input context in language models parallels with human short-term memory, exhibiting two neurocognitively grounded limitations: (*i*) Temporal decay: Mirroring the rapid fading of human temporal memory without rehearsal (Cowan, 2008), language models show severe performance degradation as sequence length increases. (*ii*) Capacity bottleneck: Memory constraints enforce strict token limits, analogous to Miller's "magical number seven" in human cognition (Miller, 1956).

3.3 Long-term Memory

Parameters in language models functions as a comprehensive knowledge repository, akin to the vast store of knowledge in human long-term memory. Unlike short-term memory, parameters remains unchanged unless deliberately updated and imposes no additional hardware burdens after fine-tuning.

3.4 Memory Transformation

The process of transforming context into parameters bears a striking resemblance to hippocampal-cortical consolidation observed in human memory systems (Squire, 1992). In this biological process, transient hippocampal memory traces are gradually integrated into neocortical networks through sleep-like replay mechanisms (Klinzing et al., 2019). During consolidation, neuronal connections undergo dynamic restructuring: new synapses form, while others are pruned. Concurrently, some memories are integrated into existing knowledge structures, others are reinforced, and some gradually attenuate. This parallel between neural network adaptation and biological memory consolidation offers intriguing insights into the nature of learning and memory in both artificial and natural systems.

This transformation offers three key practical advantages: (*i*) Efficiency: Converting context into parameters greatly reduces GPU memory usage for KV cache and computational costs for attention. (*ii*) Persistency: It potentially maintains performance across varying input lengths by avoiding the performance degradation of long-range attention. (*iii*) Scalability: It theoretically supports infinite context by continuously integrating context information into long-term memory.

Formally, let $P_\theta(y|c)$ denote the distribution generated by the teacher model conditioned on context c. We aim to learn parameters θ' such that the student model's unconditional distribution matches the teacher's conditional distribution:

$$P_{\theta'}(y) = P_\theta(y|c) \quad \forall y$$

This ensures that the student model internalizes the context c without requiring explicit attention to it during inference. Another perspective is to view the whole parameter set of a Transformer as the state of a meta recurrent neural network (RNN).

4 Our Framework

4.1 Overview

To tackle the above problem, our framework transforms short-term memory (context) into long-term memory (parameters) through three coordinated phases: (*i*) Context knowledge elicitation extracts task-specific knowledge while preserving general sequence completion capabilities via transfer set $\mathcal{T}$ construction. (*ii*) Path selection refines $\mathcal{T}$ into a high-quality subset $\mathcal{T}_k$ that maximize knowledge transfer potential through perplexity-guided filtering. (*iii*) Memory consolidation aligns the student model's distribution with the teacher's context-conditioned outputs over $\mathcal{T}_k$.

4.2 Context Knowledge Elicitation

Effective memory transformation necessitates comprehensive contextual understanding beyond superficial pattern extraction. Existing methods that rely on task-specific strategies (e.g., question generation (Mecklenburg et al., 2024; Magister et al., 2024) or knowledge propagation (Padmanabhan et al., 2024)) are confined to either surface-level patterns or partial facets of contextual knowledge, inherently failing to generalize across tasks and domains. To address this, we construct an elicitation framework designed to systematically excavate multidimensional contextual knowledge through dual objectives: (*i*) Diversity: Generate continuations that explore distinct facets of the context (*e.g.*, question answering, summarization). (*ii*) Skill specificity: Ensure continuations elicit targeted capabilities (*e.g.*, multi-hop inference, counterfactual reasoning).

Our approach begins by prompting the model to automatically infer potential queries based on the input context, followed by instructing it to produce context-grounded responsesses to these queries. This process leads to the development of a transfer set $\mathcal{T}$, defined as:

$$\mathcal{T} = \left\{ (c, \{q_i\}_{i=1}^n, \{r_i\}_{i=1}^n) \middle| \begin{matrix} q_i \sim M_\theta(\cdot|p(c)) \\ r_i \sim M_\theta(\cdot|c, q_i) \end{matrix} \right\}$$

where c represents the input context, $\{q_i\}_{i=1}^n$ is a set of generated queries, and $\{r_i\}_{i=1}^n$ contains the corresponding teacher responses. The structured prompt $p(c)$ (Appendix A.1) is meticulously crafted to include (i) a guideline directing the model to anticipate all potential real-user queries concerning the context, (ii) a hinting list of typical types of real-world user queries, (iii) specific examples for each query type, and (iv) output format requirements to facilitate parsing.

4.3 Path Selection

Our path selection mechanism promotes effective knowledge transfer by advocating the distributional separation between teacher and student outputs. This approach preserves high-quality continuations that inherently contain richer transferable knowledge while filtering out unreliable samples caused by stochastic generation. Specifically, we implement a divergence-based selection strategy inspired by RHO-1 (Lin et al., 2024). Formally, for each query-response pair (q, r) generated under context c, we compute its perplexity discrepancy as:

$$\Delta_{\text{PPL}}(r) = \log P_\theta(r|c, q) - \log P_{\theta'}(r|q)$$

where $P_\theta(\cdot)$ and $P_{\theta'}(\cdot)$ denote the sequence probabilities from the teacher and student models, respectively. We then select the top-k pairs per context that maximize Δ_{PPL}, creating a refined subset $\mathcal{T}_k$.

4.4 Memory Consolidation

This process aims to integrate context knowledge into model parameter updates by reducing the inconsistency between the teacher's context-aware outputs and the student's context-agnostic predictions. Specifically, we optimize the student model $M_{\theta'}$ through knowledge distillation to align with the teacher's context-conditioned probability distribution over $\mathcal{T}_k$:

$$\begin{aligned} \mathcal{L} &= \mathbb{E}_{(c,Q,R)\sim\mathcal{T}_k} \mathbb{E}_{(q,r)\sim(Q,R)} \mathcal{D}(c, q, r) \\ \mathcal{D}(c, q, r) &= f\left(M_{\theta'}(r|q), M_\theta(r|c, q)\right) \end{aligned} \tag{1}$$

where $M(\cdot)$ denotes the model's outputs (*e.g.*, next-token distributions), while f measures the prediction mismatch between teacher and student. We explore three alignment approaches for implementing f: (*i*) Hidden-state proximity (*e.g.*, mean squared error on hidden states), (*ii*)) Logit-level distribution matching (*e.g.*, forward/reverse KL divergence), and (*iii*) Sequence-level imitation (*e.g.*, fine-tuning on teacher-generated sequences). Our comprehensive analysis in Section 6.3 reveals task-specific trade-offs between knowledge retention and generalization for these strategies.

5 Evaluation Setup

To validate the effectiveness of our approach, we perform controlled experiments under two circumstances: single transformation and sequential transformations. While the former isolates the efficacy of individual memory transformation, the latter assesses operational viability under real-world streaming contexts where inputs are gradually received and processed.

5.1 Single Transformation

To comprehensively evaluate memory transformation capabilities, we design four task types under this setting, covering diverse real-world scenarios. The context for all test cases is limited to within $8K$ to meet the context window size restrictions of most LLMs.

Document-based QA We use the Natural Questions (NQ) dataset (Kwiatkowski et al., 2019), which consists of real user queries paired with Wikipedia passages. This task evaluates the model's ability to identify and retain factual knowledge from long-form documents. We treat the Wikipedia passages as the contexts.

Many-shot In-context Learning To evaluate the model's skill acquisition capabilities, we construct many-shot test datasets from three classification tasks: TREC Coarse[1] (6-class question categorization), TREC Fine[1] (50-class question typing), and NLU[2] (68-class intent detection). To prevent information leakage from pre-training data, we replace original labels with randomly assigned numerical values. We randomly sample 300 examples from the training set of each dataset (i.e., 300-shots learning) as the contexts.

Knowledge Update. We evaluate the model's grounded reasoning capabilities using counterfactual knowledge updates. Two datasets are used: (*i*) CounterFact (Meng et al., 2022), which contains factual statements that contradict established knowledge, and (*ii*) MQuAKE (Zhong et al., 2023), which focuses on multi-hop question answering under sequential knowledge updates. We treat the factual statements in CounterFact and the concatenated sentences of updates in MQuAKE as the contexts.

Text Generation We evaluate text generation capabilities using the test set of PG19 book corpus (Rae et al., 2019). For each test case, we extract a 2048-token segment as context and the subsequent 256-token segment as the target continuation.

5.2 Sequential Transformations

In this setting, we aim to test memory transformation under realistic scenarios that involve: (*i*) Long-context processing: Inputs exceeding the context window, requiring multiple transformations. (*ii*) Real-world complexity: Diverse task types and input lengths, covering a range of difficulties. To meet these criteria, we adopt LongBench v2 (Bai et al., 2024), a comprehensive benchmark for long-context understanding. LongBench v2 spans diverse task types, including QA, ICL, dialogue understanding, code repository understanding, and structured data understanding, curated from real-world domains such as legal documents, scientific articles, and technical manuals. Context lengths range from 8K to 2M words, challenging models to handle extreme input sizes.

[1] https://huggingface.co/datasets/CogComp/trec
[2] https://huggingface.co/datasets/xingkunliuxtracta/nlu_evaluation_data

6 Experiments

Implementation Details We employ Llama3-8B-instruct (8K context window) as our base model (the results for Mistral-7B-instruct are provided in Appendix 6). The standard implementation of our method follows this protocol: (*i*) Generate 200 query-response pairs and 200 open-ended continuations for each context (Section 4.2), with each entry consisting of 512 tokens. (*ii*) Retain the top 200 entries as $\mathcal{T}_k$ (Section 4.3). (*iii*) Transform the context into parameter updates via forward KL divergence. To establish a standardized evaluation protocol across all context compression and distillation methods, we define retention ratio as $\rho = \frac{|c'|}{|c|}$, representing the proportion of retained content c' from the original context c. Unless otherwise specified, we set $\rho = 0.1$ in our experiments. For context compression baselines, c' is determined by the model, while for context distillation baselines and our method, c' comprises the last ρ proportion of the context. Further details are provided in Appendix A.1.

6.1 Baselines

Upper Bound (Full Context) Full-context prompting serves as the theoretical upper bound, ensuring complete contextual integrity. For the LongBench v2 data, where the context may exceed the context window size, we additionally use Llama3.1-8B-instruct (extended to 128K context) with middle-truncation Bai et al. (2024) as a practical upper bound.

Lower Bound (No Context) Context-agnostic performance (no context provided) serves as the universal lower bound across all tasks.

Context Compression ReadAgent (Lee et al., 2024) segments and summarizes input, while Selective Context (Li et al., 2023), LLMLingua (Jiang et al., 2023a), and LLMLingua-2 (Pan et al., 2024) directly eliminate redundant tokens. SnapKV (Li

Method	Doc-based QA NQ	Many-shot ICL trec_fine	trec_coarse	nlu	Knowledge Update counterfact	mquake	Avg.	Text Generation PG19
Upper Bound								
Full Context	53.6(1)	61.2(1)	79.2(1)	78.4(1)	46.8(1)	92.4(1)	68.6(1)	14.6(1)
Context Compression								
Local Context	31.4(.31)	36.6(.60)	65.2(.82)	33.0(.42)	17.5(.37)	24.8(.18)	34.8(.47)	19.9(.34)
Selective Context	20.2(-.04)	0.4(.00)	0.0(0)	3.0(.04)	6.8(.14)	10.8(.01)	6.9(.03)	326.3(-38.36)
LLMLingua	19.6(-.06)	2.8(.04)	23.0(.29)	15.7(.20)	47.1(1.01)	92.0(1.00)	33.4(.44)	70.7(-6.08)
LLMLingua-2	29.6(.25)	0.4(.00)	0.4(.01)	0.3(.00)	6.6(.14)	25.0(.19)	10.4(.08)	57.7(-4.44)
SnapKV	0.4(-.65)	0.8(.01)	20.4(.26)	0.2(.00)	1.2(.02)	21.0(.14)	7.3(.03)	625.9(-76.19)
ReadAgent	43.0(.67)	17.2(.28)	16.0(.20)	5.5(.07)	0.2(-.00)	11.2(.02)	15.5(.16)	20.8(.22)
Context Distillation								
Temp-LoRA	37.2(.49)	12.4(.20)	20.8(.26)	4.2(.05)	39.6(.85)	1.6(-.10)	19.3(.22)	19.9(.34)
Ours	**45.2(.74)**	**59.0(.96)**	**83.4(1.05)**	**72.7(.93)**	**66.9(1.43)**	**95.4(1.04)**	**70.4(1.03)**	**17.9(.59)**
Lower Bound								
No Context	21.4(0)	0.2(0)	0.0(0)	0.1(0)	0.3(0)	9.6(0)	5.3(0)	22.6(0)

Table 1: Results for single transformation tasks: PPL is reported for text generation and EM for other tasks. Each entry $x(y)$ indicates x as the model's performance ($\times 100\%$) and y as the recovery rate relative to the bounds. Underlined values indicate unsurpassed full-context prompting, and bold indicates the best among other methods.

et al., 2024b) retains the most important key-value pairs within the context's KV cache. We also incorporate a practical baseline called Local Context, which retains the last ρ proportion of the context, allowing us to isolate the benefits of knowledge transformation from context retention effects.

Context Distillation We employ Temp-LoRA (Wang et al., 2024), a method that directly fine-tunes the model on the context.

6.2 Metrics

We evaluate performance using task-specific metrics: exact match (EM) for document-based QA, many-shot ICL, and knowledge update tasks, perplexity (PPL) for text generation, and accuracy for Longbench v2. We exclude text generation when reporting average single transformation performance due to differing metric scales. To quantify the effectiveness of memory transformation, we introduce the recovery rate, defined as:

$$R = \frac{M - L}{U - L}$$

where M, U, and L denote method performance, upper bound, and lower bound respectively. This unitless metric enables cross-task comparison of context utilization efficiency.

6.3 Single Transformation Tasks

We systematically evaluate memory transformation efficacy and assess individual component impacts

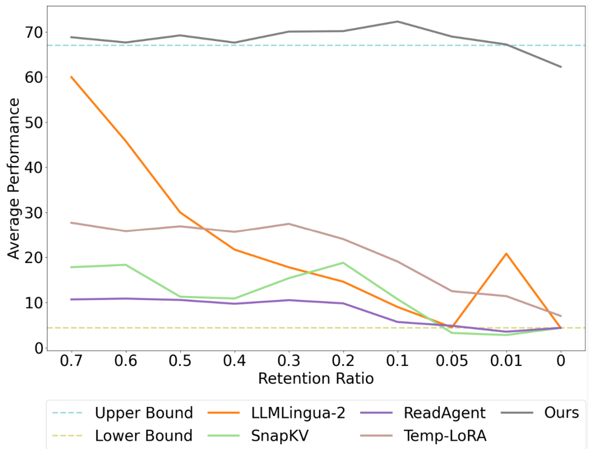

Figure 2: Average model performance on single transformation tasks across different retention ratios. To maintain the readability of the figure, we only show the results of some representative baselines.

within the controlled single-transformation setting. The ablation study of the context knowledge elicitation method is provided in Appendix A.5.

Results The results in Table 1 reveal three key findings: (*i*) First, while full-context prompting establishes an upper-bound performance benchmark by enabling full-context attention, our method demonstrates unexpected superiority in specific tasks, achieving an average performance recovery rate of 103%. Notably, in reasoning-intensive scenarios such as many-shot ICL (98% average recovery) and knowledge updates (123% average recov-

Method	N=50	N=100	N=200
Random	68.54	69.83	69.91
KL-based	66.54	68.12	70.60
PPL-based	**69.61**	**70.47**	**72.27**

Table 2: Ablation of path selection methods, with models trained on N continuations retained from 400.

ery), our approach shows remarkable effectiveness. (*ii*) Second, comparison with the Local Context baseline confirms that the performance gains of our method arise from permanent parameter updates rather than simply leveraging recent context segments. This distinction is important, as several alternative baselines perform poorly, with some even failing to surpass the lower bound. (*iii*) The third finding highlights domain-specific limitations: in document-based QA (NQ) focused on fact recall, while outperforming all baselines, our method achieves only 74% performance recovery. This aligns with human cognitive processing, where converting context into knowledge prioritizes comprehension over rote memorization, especially when queries are underspecified.

Performance vs. Retention Ratio Building on initial validation, we further analyze how retention ratio affects model performance to identify operational boundaries. We vary the ratio from 0.7 to 0, representing the gradual compression/parameterization of the context. Figure 2 shows our method maintains stable performance across ratios (68.8 $\rightarrow$ 62.3; peak 72.3 at a 0.1 ratio), while all baseline methods degrade and collapse at extreme ratios. This also evidences our approach's capacity to leverage transformed long-term memory instead of relying on residual context fragments.

Effect of path selection strategies To show the effect of our PPL-based path selection strategy, we perform an ablation study using 10% of the data from each task. Results in Table 2 highlight our method's superiority over both random selection and KL divergence-based choices in memory transformation. Our PPL-differential strategy consistently outperforms (peak 72.27 at $N = 200$), surpassing both random selection (+2.36 points) and the KL-based method (+1.67 points) at $N = 200$. Notably, the PPL-based selection exhibits a progressive quality improvement (69.61 $\rightarrow$ 72.27), suggesting its effectiveness in identifying high-information-density continuations. We also investigate the impact of retaining different numbers and lengths of continuations on model performance and find that selecting top-200 continuations, each 512 tokens long, yields the best results. The results are summarized in Appendix A.4.

Method	QA	ICL	KU	Avg.	Text Gen
Upper Bound					
Full Context	<u>54.0</u>	72.9	64.5	67.0	18.2
Logit-level Alignment					
Forward KL	46.0	**73.5**	**83.5**	**72.3**	22.7
Reverse KL	44.0	72.5	80.0	70.3	23.5
Adaptive KL	44.0	72.7	83.0	71.4	23.0
DPKD	38.0	58.9	64.0	57.1	342.3
Hidden-state-level Alignment					
MSE	26.0	63.3	77.5	61.8	23.8
Sequence-level Imitation					
SeqKD	**48.0**	68.9	35.5	54.3	**5.5**
Lower Bound					
No Context	20.0	0.1	3	4.4	27.6

Table 3: Ablation of transformation loss. We report the average performance for each single transformation task across its datasets, with task names abbreviated.

Impact of the transformation loss We examine six transformation loss variants for our framework, using the same aforementioned 10% of data from each task for experimentation. Detailed descriptions of these loss functions are provided in Appendix A.3, with the results summarized in Table 3. Our key findings reveal task-dependent trade-offs: (*i*) Hidden-state-level alignment is inadequate for fact recall: Results on NQ demonstrate that logit-level distribution matching and sequence-level imitation better preserve factual relationships than hidden-state proximity, *e.g.*, FKL outperforms MSE by 20%. (*ii*) Sequence-Level imitation fails in reasoning: SeqKD collapses catastrophically in multi-hop reasoning (6.0 vs. FKL's 96.0 on Mquake), validating that per-token alignment better preserves causal dependencies. (*iii*) PPL's narrow scope: While SeqKD excels in text generation (5.5 PPL vs. FKL's 22.7 on PG19), this metric fails to capture holistic performance, especially in compositional reasoning. Notably, FKL and MSE display minimal differences on PG19 (22.7 vs. 23.8), yet diverge significantly in other tasks, underscoring the need for multi-dimensional evaluation. These results establish FKL as the most balanced choice.

Method	Overall	Difficulty		Length (<32K; 32K-128K; >128K)		
		Easy	Hard	Short	Medium	Long
Upper Bound						
128K Context♣	30.0(1)	30.7(1)	29.6(1)	35.0(1)	27.9(1)	25.9(1)
Context Compression						
8K Context	18.5(-.57)	21.9(.00)	16.4(-1.05)	17.2(-1.47)	20.0(0)	17.6(-.28)
Local Context	15.5(-.97)	15.6(-.71)	15.4(-1.20)	22.2(-.77)	10.7(-1.18)	13.9(-.86)
Selective Context	16.3(-.87)	16.2(-.65)	16.4(-1.05)	21.1(-.92)	13.5(-.82)	13.9(-.86)
LLMLingua	22.7(0)	25.0(.35)	21.2(-.30)	26.7(-.15)	21.9(.24)	17.6(-.29)
LLMLingua-2	22.3(-.05)	21.9(0)	22.5(-.10)	28.9(.15)	16.7(-.41)	22.2(.43)
SnapKV	1.4(-2.90)	1.7(-2.86)	0.0(-3.09)	1.0(-2.95)	1.9(-2.83)	1.6(-2.87)
ReadAgent	16.1(-.89)	16.2(-.65)	16.1(-1.10)	21.7(-.85)	14.9(-.65)	9.3(-1.58)
Context Distillation						
Temp-LoRA	32.4(1.33)	34.4(1.42)	31.2(1.25)	33.3(.77)	27.0(.88)	41.7(3.44)
Ours	**44.7(3.01)**	**49.5(3.13)**	**41.8(2.89)**	**41.7(1.92)**	**46.1(3.30)**	**47.2(4.30)**
Lower Bound						
No Context	22.7(0)	21.9(0)	23.2(0)	27.8(0)	20.0(0)	19.4(0)

Table 4: Results for sequential transformation tasks: We report accuracy, as all data is organized in a multiple-choice QA format. Each entry $x(y)$ indicates x as the model's performance ($\times 100\%$) and y as the recovery rate relative to the bounds. ♣: results from Bai et al. (2024).

6.4 Sequential Transformations Tasks

We extend our evaluation to complex, real-world scenarios to gain insights into practical contexts.

Results As shown in Table 4, our method consistently outperforms all other approaches across various categories, including overall performance, difficulty levels, and text lengths. Notably, our method demonstrates remarkable efficacy in handling longer contexts, as evidenced by the substantial performance gains in the "Medium" and "Long" categories. This suggests a robust capacity for managing extended sequences, traditionally challenging due to memory constraints. Furthermore, the substantial improvements observed in both the "Easy" and "Hard" difficulty categories indicate the model's adaptability and effectiveness across varying task complexities. In comparison to 128K Context and context compression methods, the significant performance advantages achieved by our method underscore the efficacy of memory transformation in enhancing contextual understanding and application.

Performance vs. Context Length Given that contexts of varying lengths can present different challenges, we analyze the stability and scalability of all methods in processing these variations. As shown in Figure 3, where LongBench v2 results are reorganized by context length, baseline methods exhibit severe performance degradation (average decline: $15.4\% \pm 12.8\%$) within the 8K–64K range, while our method demonstrates markedly greater stability, even as contexts extend to 2M. Notably, a paradoxical performance spike occurs in certain baselines at the 512K length. We attribute this anomaly to pretrained priors rather than genuine context utilization capabilities, as evidenced by analogous improvements in no-context performance at the same length. When contexts further scale to 1M–2M, baselines suffer catastrophic failure, whereas our method sustains robust perfor-

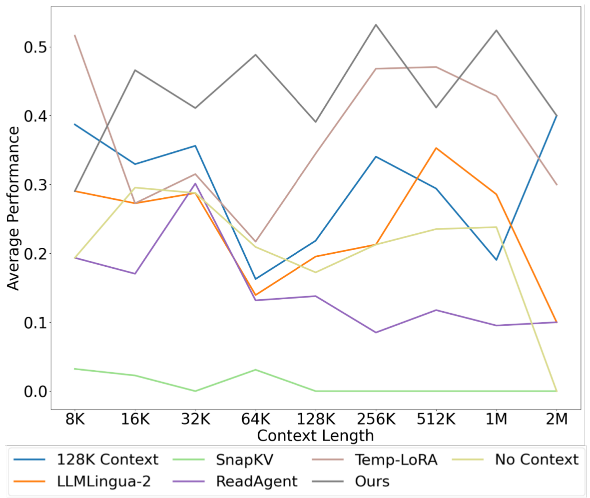

Figure 3: Performance comparison across different context lengths in sequential transformation tasks. To maintain the readability of the figure, we only show the results of some representative baselines.

Length per Turn	Avg Turn	Overall Performance
1k	277	45.54
2k	138	43.75
3k	93	44.64
4k	69	46.43
5k	53	49.11
6k	46	46.43

Table 5: Average transformation turns and performance by context length per turn.

mance. These findings underscore our method's architectural advances in persistently handling contexts far exceeding conventional window sizes.

Impact of Context Length per Transformation Turn To analyze how context segmentation granularity affects sequential transformation efficacy, we uniformly sample 100 contexts (112 associated questions) from Longbench v2. As shown in Table 5, performance peaks at 49.11 when processing 5K-token chunks, requiring fewer transformation turns (Avg= 53). Shorter chunks degrade performance due to fragmented knowledge integration, while longer chunks (6K) reduce performance by 2.68 points, likely due to information overload affecting the model's capacity to consolidate knowledge. These results suggest an optimal range of around 5K tokens for an 8K context window model, effectively balancing semantic coherence and manageable information load.

7 Conclusion

This work parallels LLMs' context and parameters with human memory systems, establishing a framework to transform temporary context knowledge into permanent parameter updates through elicitation, path selection, and memory consolidation. In single transformation tasks, our method remarkably surpasses full-context prompting performance, achieving an average 103% recovery rate. Further analysis reveals that baselines exhibit brittle performance under high compression ratios, whereas our method maintains robust performance, even maintaining 62.25 performance at extreme ratios where most baselines collapse to no-context performance. Our framework's generalizability is further evidenced by its consistent performance across varying context lengths and task difficulties in sequential transformation tasks, highlighting its applicability in real-world scenarios.

Limitations

While our framework demonstrates significant advantages in memory transformation, several limitations warrant discussion:

Model Diversity Our experiments are conducted primarily using Llama-3-8B-instruct, which may limit the generalizability of our findings. A broader range of models could provide additional insights.

Compute Cost Our framework currently encounters limitations regarding computational efficiency during gradient-based fine-tuning. Future work may consider to use hypernetwork for much efficient transformation.

References

Rishabh Agarwal, Avi Singh, Lei M Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, et al. 2024. Many-shot in-context learning. *arXiv preprint arXiv:2404.11018*.

Ekin Akyürek, Mehul Damani, Linlu Qiu, Han Guo, Yoon Kim, and Jacob Andreas. 2024. The surprising effectiveness of test-time training for abstract reasoning. *arXiv preprint arXiv:2411.07279*.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, et al. 2024. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*.

Tong Chen, Hao Fang, Patrick Xia, Xiaodong Liu, Benjamin Van Durme, Luke Zettlemoyer, Jianfeng Gao, and Hao Cheng. 2024. Generative adapter: Contextualizing language models in parameters with a single forward pass. *arXiv preprint arXiv:2411.05877*.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting language models to compress contexts. *arXiv preprint arXiv:2305.14788*.

Eunbi Choi, Yongrae Jo, Joel Jang, and Minjoon Seo. 2022. Prompt injection: Parameterization of fixed inputs. *arXiv preprint arXiv:2206.11349*.

Nelson Cowan. 2008. What are the differences between long-term, short-term, and working memory? *Progress in brain research*, 169:323–338.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can gpt learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4005–4019.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. 2023. In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945*.

Xijie Huang, Li Lyna Zhang, Kwang-Ting Cheng, M Yang, and Mao Yang. 2023. Fewer is more: Boosting llm reasoning with reinforced context pruning. *arXiv preprint arXiv:2312.08901*.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023a. Llmlingua: Compressing prompts for accelerated inference of large language models. *arXiv preprint arXiv:2310.05736*.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023b. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. *arXiv preprint arXiv:2310.06839*.

Hoyoun Jung and Kyung-Joong Kim. 2024. Discrete prompt compression with reinforcement learning. *IEEE Access*.

Jang-Hyun Kim, Junyoung Yeom, Sangdoo Yun, and Hyun Oh Song. 2023. Compressed context memory for online language model interaction. *arXiv preprint arXiv:2312.03414*.

Jens G Klinzing, Niels Niethard, and Jan Born. 2019. Mechanisms of systems memory consolidation during sleep. *Nature neuroscience*, 22(10):1598–1610.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. 2024. A human-inspired reading agent with gist memory of very long contexts. *arXiv preprint arXiv:2402.09727*.

Yixing Li, Yuxian Gu, Li Dong, Dequan Wang, Yu Cheng, and Furu Wei. 2024a. Direct preference knowledge distillation for large language models. *arXiv preprint arXiv:2406.19774*.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6342–6353.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024b. Snapkv: Llm knows what you are looking for before generation. *arXiv preprint arXiv:2404.14469*.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, Weizhu Chen, et al. 2024. Not all tokens are what you need for pretraining. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Lucie Charlotte Magister, Katherine Metcalf, Yizhe Zhang, and Maartje ter Hoeve. 2024. On the way to llm personalization: Learning to remember user conversations. *arXiv preprint arXiv:2411.13405*.

Nick Mecklenburg, Yiyou Lin, Xiaoxiao Li, Daniel Holstein, Leonardo Nunes, Sara Malvar, Bruno Silva, Ranveer Chandra, Vijay Aski, Pavan Kumar Reddy Yannam, et al. 2024. Injecting new knowledge into large language models via supervised fine-tuning. *arXiv preprint arXiv:2404.00213*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

George A Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81.

Jesse Mu, Xiang Li, and Noah Goodman. 2024. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36.

Dilxat Muhtar, Yelong Shen, Yaming Yang, Xiaodong Liu, Yadong Lu, Jianfeng Liu, Yuefeng Zhan, Hao Sun, Weiwei Deng, Feng Sun, et al. 2024. Streamadapter: Efficient test time adaptation from contextual streams. *arXiv preprint arXiv:2411.09289*.

Shankar Padmanabhan, Yasumasa Onoe, Michael Zhang, Greg Durrett, and Eunsol Choi. 2024. Propagating knowledge updates to lms through distillation. *Advances in Neural Information Processing Systems*, 36.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, et al. 2024. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. *arXiv preprint arXiv:2403.12968*.

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, and Timothy P Lillicrap. 2019. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*.

Charlie Snell, Dan Klein, and Ruiqi Zhong. 2022. Learning by distilling context. *arXiv preprint arXiv:2209.15189*.

Larry R Squire. 1992. Memory and the hippocampus: a synthesis from findings with rats, monkeys, and humans. *Psychological review*, 99(2):195.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Y Wang, D Ma, and D Cai. 2024. With greater text comes greater necessity: Inference-time training helps long text generation. *arXiv preprint arXiv:2401.11504*.

A Waswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, A Gomez, L Kaiser, and I Polosukhin. 2017. Attention is all you need. In *NIPS*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

David Wingate, Mohammad Shoeybi, and Taylor Sorensen. 2022. Prompt compression and contrastive conditioning for controllability and toxicity reduction in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5621–5634.

Taiqiang Wu, Chaofan Tao, Jiahao Wang, Runming Yang, Zhe Zhao, and Ngai Wong. 2024. Rethinking kullback-leibler divergence in knowledge distillation for large language models. *arXiv preprint arXiv:2404.02657*.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *arXiv preprint arXiv:2310.04408*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710.

Junhao Zheng, Chengming Shi, Xidi Cai, Qiuke Li, Duzhen Zhang, Chenxing Li, Dong Yu, and Qianli Ma. 2025. Lifelong learning of large language model based agents: A roadmap. *arXiv preprint arXiv:2501.07278*.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. 2023. Mquake: Assessing knowledge editing in language models via multi-hop questions. *arXiv preprint arXiv:2305.14795*.

A Appendix

A.1 Implementation Details

During the context knowledge elicitation phase, we set the temperature to 1 for sampling queries and responses, and employ the following hybrid prompt to guide the model in generating high-quality queries:

> Please prepare to analyze the text provided below. As you read, simulate real-world user queries about the content, such as summarizing, detailing, or inferring knowledge. For example, consider the following potential user queries if applicable to the provided text:
> 1. Ask for a concise summary that captures the main points and essential details of the text. Anticipate user requests such as, "What are the central arguments?" or "Can you summarize the main events of the story?"
> 2. Formulate questions about key details or themes within the text, such as "What achievements did Anthony Joshua achieve in boxing?" or "What was the date and venue of event?"
> 3. Identify and explore patterns or examples, create similar formatted examples or pose questions, such as "What is the labeling criteria for these examples?"
> 4. Integrate and reflect on new knowledge from the text, asking for the implications and applications of the new knowledge.
> 5. Request repetition of specific sentences or paragraphs from the text.
> 6. ...
>
> Here is the text for analysis:
> ======= Text Begins Here =======
> {context}
> ======= Text Ends Here =======
> Let's review the potential user queries to see if they apply to the provided text.
> 1. Ask for a concise summary that captures the main points and essential details of the text. Anticipate user requests such as, "What are the central arguments?" or "Can you summarize the main events of the story?"
> 2. Formulate questions about key details or themes within the text, such as "What achievements did Anthony Joshua achieve in boxing?" or "What was the date and venue of event?"
> 3. Identify and explore patterns or examples, create similar formatted examples or pose questions, such as "What is the labeling criteria for these examples?"
> 4. Integrate and reflect on new knowledge from the text, asking for the implications and applications of the new knowledge.
> 5. Request repetition of specific sentences or paragraphs from the text.
> 6. ...
>
> Determine suitable query types for the text and generate a comprehensive set of queries that thoroughly cover the content.
> Make the subject of the query clear and avoid using pronouns like "it," "he," or "she" to prevent ambiguity.
> Output 20 queries directly, each on a separate line, numbered from "1." to "20." Conclude with "|||||" as the end symbol.

The memory consolidation process involves LoRA fine-tuning on 200 query-response pairs (160 for training and 40 for validation), with the loss calculated only on responses of up to 512 tokens. This approach results in minimal computational overhead. For instance, fine-tuning LLama-3-8B-instruct on a 7K-token context with these pairs takes an average of 2.3 minutes on 8×A100 GPUs. The parameters for LoRA fine-tuning are as follows: lora_rank=8, lora_alpha=16, and lora_dropout=0.05, with a learning rate of $1e^{-4}$. Training is stopped if the development set loss does not decrease after two consecutive evaluations. In the sequential transformation setting, each transformation follows the same protocol. For testing, we set the temperature to 0 (i.e., greedy decoding) to ensure reproducibility.

Importantly, the model is trained only once and can subsequently be used for inference without the transformed context, thereby eliminating compute associated with attention and reducing memory consumption for the KV cache. Consequently, the training compute becomes insignificant in inference-intensive and extremely long context scenarios. For instance, on trec_coarse (300-shots ICL), post-transformation, the generation speed is 3.5x faster.

Method	Doc-based QA NQ	Many-shot ICL trec_fine	trec_coarse	nlu	Knowledge Update counterfact	mquake	Avg.	Text Generation PG19
Upper Bound								
Full Context	47.6(1)	47.2(1)	78.8(1)	78.5(1)	50.3(1)	86.6(1)	64.8(1)	9.5(1)
Context Compression								
Local Context	31.6(.42)	37.0(.78)	60.8(.77)	32.0(.41)	14.3(.28)	9.4(.03)	30.9(.44)	12.4(.50)
Selective Context	23.2(.12)	0(-.00)	0(0)	4.5(.06)	6.6(.12)	8.0(.01)	7.1(.04)	140.7(-21.62)
LLMLingua	25.0(.18)	9.0(.19)	17.8(.23)	15.6(.20)	50.2(1.00)	**87.4(1.01)**	34.2(.49)	48.5(-5.73)
LLMLingua-2	36.0(.58)	0.6(.01)	0(0)	0(0)	5.4(.10)	9.4(.03)	8.6(.06)	20.5(-.90)
SnapKV	35.6(.57)	2.6(.05)	18.8(.24)	0.7(.01)	9.2(.18)	32.0(.31)	16.5(.20)	13.57(.30)
ReadAgent	17.6(-0.09)	1.6(.03)	5.2(.07)	0.5(.01)	0.5(.00)	7.8(.01)	5.5(.01)	**5.4(1.70)**
Context Distillation								
Temp-LoRA	**47.0(.98)**	13.4(.28)	21.6(.27)	4.7(.06)	66.3(1.32)	57.2(.63)	35.0(.50)	30.8(-2.68)
Ours	43.4(.85)	**48.6(1.03)**	**85.6(1.09)**	**71.9(.92)**	**67.8(1.35)**	81.2(.93)	**66.4(1.03)**	11.3(.69)
Lower Bound								
No Context	20.0(0)	0.2(0)	0(0)	0(0)	0.4(0)	7.4(0)	4.7(0)	15.3(0)

Table 6: Single transformation task results for Mistral-7B-instruct-v0.3: PPL is reported for text generation and EM for other tasks. Each entry $x(y)$ indicates x as the model's performance ($\times 100\%$) and y as the recovery rate relative to the bounds. Underlined values indicate unsurpassed full-context prompting, and bold indicates the best among other methods.

A.2 Mistral Results

In Table 6, we observe similar findings to those with Llama: (*i*) First, our method demonstrates unexpected superiority in specific tasks compared to full-context prompting, achieving an average performance recovery rate of 103%. Notably, in reasoning-intensive scenarios such as many-shot ICL (101% average recovery) and knowledge updates (114% average recovery), our approach shows remarkable effectiveness. (*ii*) Second, comparison with the Local Context baseline confirms that the performance gains of our method arise from permanent parameter updates rather than simply leveraging recent context segments. (*iii*) Overall, our method achieves the best performance among all baselines.

A.3 loss variants

(*i*) Forward KL (FKL) minimizes KL$(M_\theta(y|c,x) \parallel M_{\theta'}(y|x))$ per token, enforcing precise distribution matching. (*ii*) Reverse KL (RKL) optimizes KL$(M_{\theta'}(y|x) \parallel M_\theta(y|c,x))$, prioritizing mode coverage over exact alignment. (*iii*) Adaptive KL (AKL) (Wu et al., 2024) dynamically blends FKL/RKL weights based on teacher-student distribution divergence. (*iv*) DPKD (Li et al., 2024a) incorporates preference-driven RKL with length normalization. (*v*) MSE aligns hidden states between models, bypassing distributional objectives. (*vi*) SeqKD applies sequence-level teacher forcing over sampled continuations.

N \ L	128	256	384	512
100	64.13	65.37	67.00	70.47
200	66.47	66.30	69.67	72.27
300	66.30	67.10	68.17	69.53
400	66.05	67.35	68.95	69.40

Table 7: Ablation: Continuation Count and Length. Each set of N continuations comprises an equal mix of query-response pairs and open-ended continuations.

A.4 Impact of the number and length of the continuations

We assess the impact of $\mathcal{T}_k$'s scale on memory consolidation. To ensure that extending count/length settings provides additional information, each setup includes all content from the smaller or shorter settings. The results in Table 7 reveal a non-linear relationship between continuation configurations and model performance. Performance generally improves with longer continuations (L), peaking at 72.27 for $N = 200$ and $L = 512$. However, a quality-quantity trade-off emerges: the performance inversion between $N = 200, L = 512$ (72.27) and $N = 300, L = 512$ (69.53) suggests excessive continuation counts degrade information density, emphasizing that high-quality continuations outweigh sheer quantity for effective knowledge consolidation.

Method	Doc-based QA NQ	Many-shot ICL trec_fine	trec_coarse	nlu	Knowledge Update counterfact	mquake	Avg.	Text Generation PG19
Upper Bound								
Full Context	<u>54.0(1)</u>	61.2(1)	79.2(1)	<u>78.4(1)</u>	47.0(1)	82.0(1)	67.0(1)	<u>18.2(1)</u>
Our Method								
OE	36.0(.47)	61.2(1)	80.0(1.01)	**75.3(.96)**	59.0(1.26)	88.0(1.08)	66.6(.99)	**22.6(.53)**
QR	42.0(.65)	61.2(1)	**85.0(1.07)**	53.6(.68)	68.0(1.45)	88.0(1.08)	66.3(.99)	29.7(-.22)
Mix (Ours)	**46.0(.76)**	**63.2(1.03)**	**85.0(1.07)**	72.4(.92)	**71.0(1.51)**	**96.0(1.18)**	**72.3(1.08)**	22.7(.52)
Lower Bound								
No Context	20.0(0)	0.2(0)	0.0(0)	0.1(0)	0.0(0)	6.0(0)	4.4(0)	27.6(0)

Table 8: Results for single transformation tasks using 10% of the data from each task. PPL is reported for text generation and EM for other tasks. Each entry $x(y)$ indicates x as the model's performance ($\times 100\%$) and y as the recovery rate relative to the bounds. Underlined values indicate unsurpassed full-context prompting, and bold indicates the best among other methods. OE stands for open-ended continuation generation, and QR denotes query-response pair generation. We generate 400 sequences for OE and QR, and 200 OE + 200 QR for Mix, selecting 200 based on our path selection method for memory consolidation.

A.5 Ablation of context knowledge elicitation strategy

We further analyze the effectiveness of our context knowledge elicitation method through ablation experiments. In practice, we first have the model generate 200 continuations by following the query "Extension:" for a given context. Then, we obtain 200 model-generated queries using our prompt design, followed by context-grounded response generation. These 400 outputs are then filtered through path selection to retain 200 samples for training. We compare this approach with directly generating 400 open-ended continuations or 400 query-response pairs, followed by identical selection of 200 samples for training. As shown in Table 8, our findings reveal that Open-Ended generation (OE) performs better for challenging many-shot tasks (nlu) and text generation tasks, while Query-Response generation (QR) demonstrates superior performance on fact recall (NQ) and grounded reasoning (Counterfact). Our hybrid method (Mix) combines their strengths, achieving even better performance than either individual approach on tasks such as NQ. This indicates that the combination of OE and QR can more comprehensively elicit contextual knowledge.