Gated Differentiable Working Memory for Long-Context Language Modeling

Lingrui Mei[1,2,†] **Shenghua Liu**[1,2,†*], **Yiwei Wang**[3] **Yuyao Ge**[1,2,†] **Baolong Bi**[1,2,†]
Jiayu Yao[2,†] **Jun Wan**[4] **Ziling Yin**[1,2,†] **Jiafeng Guo**[1,2,†] **Xueqi Cheng**[1,2,†]
[1]State Key Laboratory of AI Safety
[2]Institute of Computing Technology, Chinese Academy of Sciences
[3]University of California, Merced [4]UBS AG
`{meilingrui25b, liushenghua, bibaolong23z, geyuyao24z, cxq}@ict.ac.cn`
`wangyw.evan@gmail.com` `jun.wan@ubs.com` `yzl204801@gmail.com`

Abstract

Long contexts break transformers: attention scores dilute across thousands of tokens, critical information gets lost in the middle, and the model cannot adapt to novel patterns at inference time. Recent work on test-time adaptation addresses this by maintaining a form of *working memory*—transient parameters updated on the current context—but existing approaches employ *uniform* write policies that waste computation on low-value regions and suffer from high gradient variance across semantically heterogeneous contexts. In this work, we reframe test-time adaptation as a budget-constrained memory consolidation problem, asking: *given limited computational budget, which parts of the context should be consolidated into working memory?* We propose GDWM (**G**ated **D**ifferentiable **W**orking **M**emory), a framework that introduces a Write Controller to gate the memory consolidation process. Our controller estimates *Contextual Utility*—an information-theoretic measure quantifying how much each region depends on long-range context—and allocates gradient steps accordingly, subject to a coverage constraint that ensures global representation. Experiments on ZeroSCROLLS and LongBench v2 benchmarks demonstrate that GDWM achieves comparable or superior performance with 4× fewer gradient steps compared to uniform baselines, establishing a new efficiency-performance Pareto frontier for test-time adaptation.

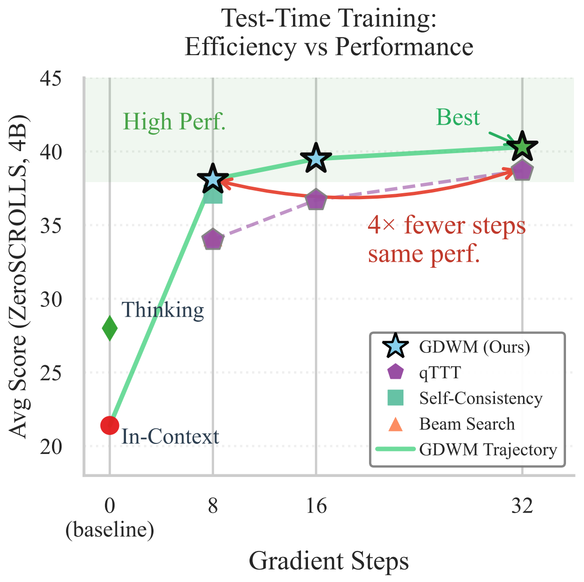

Figure 1: **Efficiency vs Performance on ZeroSCROLLS (QWEN3-4B).** GDWM achieves comparable performance to qTTT-32 with only 8 gradient steps (4× fewer), establishing a new Pareto frontier. Context-aware budget allocation enables faster convergence than uniform or sampling-based alternatives.

1 Introduction

Large Language Models (LLMs) struggle with long contexts: attention scores dilute, and models miss critical information in central positions (e.g., "Lost-in-the-Middle") (Liu et al., 2023a; Hsieh et al., 2024; Du et al., 2025b; Yao et al., 2025b). Recent work mitigates such long-context failures by equipping LLMs with *working-memory*—either as lightweight, differentiable *fast-weight* (Ba et al., 2016) states updated online via self-supervised gradients, or as explicit memory modules that incrementally read, consolidate, and overwrite salient information during inference (Yu et al., 2025; Bansal et al., 2025; Hong et al., 2025; Anonymous, 2025; Xu et al., 2025b; Ye et al., 2025).

However, current approaches predominantly rely on *uniform write policies*, stochastically sampling tokens across the entire context. This is suboptimal: information density is non-uniform, wasting budget on low-value regions, and global sampling exacerbates gradient variance by aggregating conflicting updates from semantically disparate regions.

We propose GDWM (**G**ated **D**ifferentiable **W**orking **M**emory), a framework that recasts test-time adaptation as budget-constrained memory con-

---

[*] Corresponding author.

[†] Also affiliated with University of Chinese Academy of Sciences.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 31885–31913
July 2-7, 2026

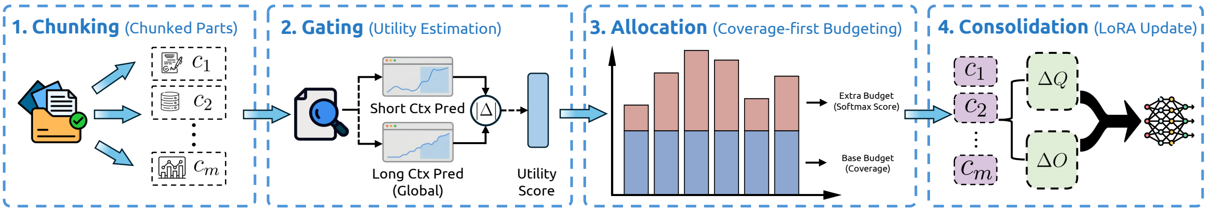

Figure 2: **High-level overview of GDWM.** The framework proceeds in four stages: **Chunk** the input into fixed-size units (approximating semantic segments), **Gate** each chunk via Contextual Utility (CPMI-based divergence), **Allocate** gradient budget proportionally subject to coverage constraints, and **Consolidate** into LoRA adapters.

solidation. The key question becomes: *given limited compute, which parts of the context should be consolidated into working memory?* GDWM answers this via a Write Controller that estimates *Contextual Utility*—the divergence between long-context and short-context predictions—and allocates gradient steps to regions where long-range dependencies are most critical. The framework is mechanism-agnostic: it provides a principled write policy that can be layered on top of any test-time adaptation architecture.

First, we formalize test-time adaptation as a budget-constrained resource allocation problem, cleanly separating the memory mechanism (how to update) from the write policy (where and how much to update). Second, we propose a chunk-wise, budget-aware algorithm driven by *Contextual Utility*—an information-theoretic measure grounded in Conditional Pointwise Mutual Information (CPMI) that identifies high-value long-range dependencies. Third, we prove that chunk-restricted sampling reduces gradient variance by eliminating inter-chunk interference via the Law of Total Variance. Extensive evaluation on ZeroSCROLLS and LongBench v2 across three model scales (1.7B, 4B, 8B) demonstrates that GDWM achieves comparable or superior performance with 4× fewer gradient steps, yielding significant gains on sparse-information tasks (up to +12.7% on Qasper, up to +11.2% on GovReport) while achieving 39% wall-clock speedup. Ablation studies confirm that CPMI-based selection, coverage constraints, and chunk-based processing are all essential components. We further show that GDWM outperforms alternative test-time scaling strategies (self-consistency, beam search), and provide a theoretical analysis linking optimal chunk size to task-specific evidence spans.

2 Related Work

Context Engineering Context engineering optimizes information payloads for LLMs through prompt design, in-context learning, and chain-of-thought prompting (Wei et al., 2022; Brown et al., 2020a). Retrieval-augmented generation (RAG) enhances knowledge access (Lewis et al., 2020; Karpukhin et al., 2020), while compression and hierarchical processing address quadratic scaling limitations (Li et al., 2023; Vaswani et al., 2017). Recent work tackles robustness via context-aware decoding and representation engineering (He et al., 2024a; Zhou et al., 2023).

LLM Memory Memory in LLMs addresses bottlenecks from the KV cache and model weights (Liu et al., 2023b; Dao et al., 2022), with solutions including paging (Kwon et al., 2023), compression (Zhang et al., 2024d), and dynamic eviction (Zhang et al., 2023; Yuan et al., 2025a). Agent memory research extends this by enabling persistent storage beyond context windows through hierarchical memory architectures (Fang et al., 2024; Zhong et al., 2023a) and structured memory types (Han et al., 2024; Terranova et al., 2025). Contemporary frameworks implement consolidation, updating, and forgetting operations (Park et al., 2023; Cao et al., 2025), though challenges remain in autonomous management and catastrophic forgetting (Xu et al., 2025c; Guo et al., 2023).

Test-Time Adaptation Recent work treats deployment as an optimization phase, updating model states at inference time via self-supervised objectives (Niu et al., 2022; Tang et al., 2023). Within LLMs, gradient-based test-time training adapts to task instances through neighbor retrieval, stream processing, or active selection (Hardt and Sun, 2024; Muhtar et al., 2024; Hübotter et al., 2025; Akyürek et al., 2025). Parallel efforts optimize test-time compute allocation through principled scaling and meta-learned control (Snell et al., 2024; Qu et al., 2025), while complementary strategies compress long contexts via selective augmentation and learned prompt compression (Xu et al., 2023; Jiang

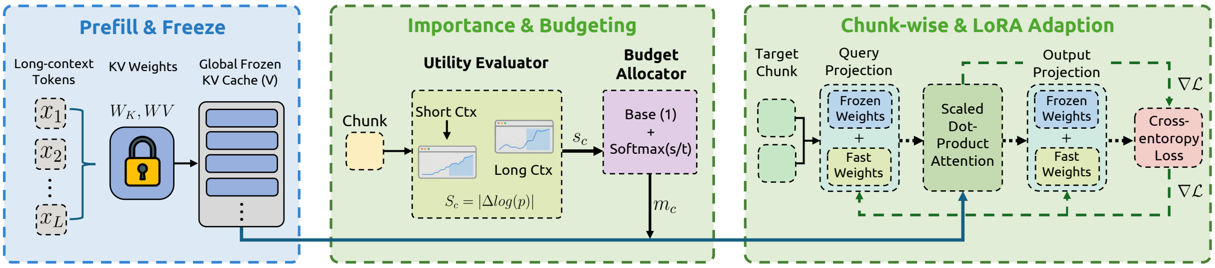

Figure 3: **Technical details of GDWM.** Left: Prefill-and-freeze KV cache enables efficient chunk-wise processing. Middle: Contextual Utility is computed as CPMI between global and local predictions, then converted to budget weights via softmax allocation. Right: LoRA adapters on W_Q/W_O projections are updated through chunk-wise next-token prediction loss.

et al., 2024).

3 Problem Formulation

We view the process of adapting an LLM to a long context $X = (x_1, x_2, \dots, x_L)$ as a Budget-Constrained Memory Consolidation problem.

Fast Weights. Let θ denote the parameters of a pre-trained LLM. We partition θ into frozen parameters θ_f (including the KV cache computation) and adaptable "fast weights" ϕ (e.g., LoRA adapters (Hu et al., 2021) on Query projections).

Budget-Constrained Optimization. We partition the input context X into M fixed-size chunks $\{C_1, C_2, \dots, C_M\}$. The goal is to determine an optimal integer allocation schedule $\mathbf{k} = (k_1, \dots, k_M)$, where k_c denotes the number of minibatch gradient updates performed on chunk C_c.

Since the downstream task loss $\mathcal{L}_{\text{task}}$ is unknown during inference, we employ the self-supervised next-token prediction loss as a surrogate objective. The optimization problem is formulated as:

$$\min_{\{k_c\}_{c=1}^{M}} \sum_{c=1}^{M} \mathbb{E}_{t\sim\mathcal{U}(\mathcal{I}_c)} \left[-\log P(x_t \mid x_{1:t-1}; \phi(\mathbf{k}), \theta_f)\right] \tag{1}$$

where $\mathcal{I}_c$ denotes the set of positions in chunk C_c, and $\mathbb{E}_{t\sim\mathcal{U}(\mathcal{I}_c)}$ denotes the expectation over positions t sampled uniformly from $\mathcal{I}_c$. The constraints are:

$$\sum_{c=1}^{M} k_c \leq K_{\text{total}} \quad \text{(Total Budget)} \tag{2}$$

$$k_c \geq k_{\min}, \ \forall c \in \{1, \dots, M\} \quad \text{(Coverage)} \tag{3}$$

Here, $\phi(\mathbf{k})$ is an *implicit function* of the allocation $\mathbf{k}$, defined by the gradient descent process: starting from $\phi^{(0)} = 0$, we sequentially perform k_c gradient updates on each chunk C_c, yielding $\phi(\mathbf{k}) = \phi^{(\sum_c k_c)}$. Since exactly solving this bi-level problem is intractable, we propose a heuristic approximation in Section 4 using Contextual Utility as a proxy for the gradient contribution of each chunk.

4 Method

Our method, GDWM, solves the optimization problem defined in Equation 1 via a four-stage process: (1) Chunking, (2) Gating (Importance Estimation), (3) Allocation, and (4) Consolidation. Figure 2 illustrates the high-level pipeline, and Figure 3 details the technical components.

4.1 Memory Interface

GDWM is mechanism-agnostic: the Write Controller operates as a policy layer orthogonal to the memory mechanism. While ϕ can be any differentiable parameters (linear, MLP, etc.), we focus on LoRA adapters for efficiency.

While GDWM is mechanism-agnostic and can be layered on top of any adaptation architecture, we instantiate it efficiently by attaching LoRA adapters to projection matrices while freezing the KV cache (see Section 5.1 for details). The fast weights ϕ are initialized to zero (identity mapping) at the start of each sequence.

4.2 Contextual Utility Estimation

To efficiently solve the allocation problem, we need a proxy for the gradient contribution of each chunk. We introduce *Contextual Utility*, an information-theoretic measure grounded in the cognitive notion of surprisal. The core intuition is simple: if a token x_t can be predicted easily using only local context, consolidating it into fast weights yields low utility. Conversely, if x_t is unpredictable locally but predictable globally, it represents a high-value long-range dependency.

Definition 1 (Contextual Utility)**.** *Let $\mathcal{I}_c$ denote the set of token positions in chunk C_c. The Contextual Utility is defined as the average surprisal divergence:*

$$U(C_c) = \frac{1}{|\mathcal{I}_c|} \sum_{t \in \mathcal{I}_c} \Delta_t$$

$$\Delta_t = \big|\log P(x_t \mid x_{1:t-1}) - \log P(x_t \mid x_{t-n:t-1})\big| \qquad (4)$$

where $P(x_t \mid x_{1:t-1})$ is the conditional probability using full context ("global prediction") and $P(x_t \mid x_{t-n:t-1})$ uses only the recent n tokens ("local prediction").

We take the absolute value to capture both directions: positions where long context *helps* ($P_{\text{full}} > P_{\text{local}}$) and where it *conflicts* ($P_{\text{full}} < P_{\text{local}}$). Both indicate regions requiring gradient-based calibration. Empirically, $|\Delta_t|$ outperforms $\max(0, \Delta_t)$ by 3-5% (see Appendix F for analysis).

Interpretation. The term inside the sum represents the *Surprisal Divergence*. When $P_{\text{full}} \approx P_{\text{local}}$, the long context provides no additional information (utility is zero). High utility indicates regions where global dependencies are critical for prediction.

Information-Theoretic Grounding. The quantity Δ_t has a precise information-theoretic interpretation: it equals the absolute value of *Conditional Pointwise Mutual Information* (CPMI) between the token x_t and the long-range prefix $x_{1:t-n}$, conditioned on the local context $x_{t-n:t-1}$. Formally, $\text{CPMI}(A; B \mid C) = \log P(A \mid B, C) - \log P(A \mid C)$ measures the information that B provides about A beyond what C already provides. High $|\Delta_t|$ indicates positions where the model's prediction strongly depends on (or conflicts with) long-range context—precisely where gradient-based memory consolidation is most valuable.

4.3 Budget-Aware Allocation Policy

Exact solution of the integer programming problem in Eq. (1) is intractable. We propose a heuristic approximation based on **Coverage-First Softmax Allocation**.

Step 1: Satisfy Coverage Constraint. Allocate $k_{\min}$ steps to every chunk:

$$k_c \leftarrow k_{\min}, \quad \forall c \in \{1, \ldots, M\} \qquad (5)$$

Handling Infeasible Budgets. If the total budget $K_{\text{total}} < M \cdot k_{\min}$, the coverage constraint cannot be strictly satisfied. In such cases, we relax the constraint by allocating $k_{\min}$ steps to the top-$\lfloor K_{\text{total}}/k_{\min} \rfloor$ chunks with the highest utility $U(C_c)$, and 0 to others. This fallback ensures that limited compute is invested in the most critical regions first.

Step 2: Distribute Remaining Budget. The remaining budget $K_{\text{rem}} = \max(0, K_{\text{total}} - M \cdot k_{\min})$ is distributed based on normalized utility:

$$w_c = \frac{\exp(U(C_c)/\tau)}{\sum_{j=1}^{M} \exp(U(C_j)/\tau)} \qquad (6)$$

$$k_c \leftarrow k_c + \lfloor K_{\text{rem}} \cdot w_c \rfloor \qquad (7)$$

To ensure the full budget is utilized, we apply the *Largest Remainder Method* (see Appendix I) to distribute residual steps, guaranteeing $\sum_c k_c = K_{\text{total}}$ exactly. The temperature τ controls allocation sharpness. When $\tau \to 0$, all extra budget concentrates on the highest-utility chunk; when $\tau \to \infty$, allocation becomes uniform (ignoring utility); and $\tau = 1.0$ provides balanced allocation that proves empirically optimal. This policy directs the optimizer's focus to high-utility regions while maintaining global awareness through the coverage constraint—hence the term "gated" in our framework name.

4.4 Structured Memory Consolidation

Finally, we execute updates via chunk-restricted sampling. For each chunk C_c, we perform k_c gradient descent steps. In each step, we uniformly sample a minibatch of positions $\mathcal{I} \subset \mathcal{I}_c$ and compute the loss:

$$\mathcal{L} = -\frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \log P(x_i \mid x_{1:i-1}; \phi, \theta_f) \qquad (8)$$

The KV cache can be precomputed and reused across all k_c steps for efficiency (see Section 5.1).

4.5 Why Does This Work?

The key insight is semantic heterogeneity: distinct document sections induce gradient directions that interfere destructively when aggregated. Chunk-restricted sampling (using fixed-size windows as semantic proxies) eliminates this interference by confining each update to a coherent unit. Formally, by the Law of Total Variance, this eliminates the inter-chunk variance component from gradient estimates (Theorem 1). The coverage constraint prevents

METHOD	STEPS	SUMMARIZATION		QUESTION ANSWERING		COMPREHENSION	REASONING	*Avg.*
		GovReport	QMSum	Qasper	NarrativeQA	Quality	Musique	
				QWEN3-1.7B				
In-context	–	22.2	6.4	25.8	14.9	48.1	11.8	21.5
Thinking	–	21.5	7.6	22.3	9.2	61.8	22.4	24.1
qTTT	8	23.9	8.3	27.3	9.4	72.1	19.9	26.8
qTTT	16	25.1	9.3	29.5	11.2	74.1	23.1	28.7
qTTT	32	26.8	9.7	31.5	12.4	76.5	26.4	30.6
GDWM (Ours)	8	28.5	9.6	33.8	12.2	76.5	27.0	31.3
GDWM (Ours)	16	29.1	9.9	34.6	12.6	77.0	27.9	31.8
GDWM (Ours)	32	**29.8**	**10.2**	**35.5**	**13.0**	**77.5**	**28.8**	**32.5**
Δ vs Best		+11.2%	+5.2%	+12.7%	+4.8%	+1.3%	+9.1%	+6.2%
				QWEN3-4B				
In-context	–	24.8	11.2	23.5	10.9	40.8	17.0	21.4
Thinking	–	20.8	7.5	25.2	29.8	76.1	8.3	28.0
qTTT	8	29.2	8.3	29.5	32.3	81.3	23.7	34.0
qTTT	16	31.9	8.6	32.1	35.3	84.8	27.5	36.7
qTTT	32	33.2	8.9	33.8	38.4	87.2	30.8	38.7
GDWM (Ours)	8	34.2	8.4	35.2	37.5	82.2	31.2	38.1
GDWM (Ours)	16	35.0	8.8	35.8	38.6	86.8	32.0	39.5
GDWM (Ours)	32	**35.8**	**9.2**	**36.5**	**39.8**	**87.5**	**32.8**	**40.3**
Δ vs Best		+7.8%	+3.4%	+8.0%	+3.6%	+0.3%	+6.5%	+4.1%
				QWEN3-8B				
In-context	–	22.5	8.8	20.1	35.4	90.5	22.9	33.4
Thinking	–	18.2	9.8	21.5	19.6	71.8	43.8	30.8
qTTT	8	25.3	8.5	22.8	38.5	91.2	42.0	38.1
qTTT	16	27.9	8.7	24.5	40.6	93.1	46.2	40.2
qTTT	32	29.8	9.0	27.0	42.4	94.9	49.6	42.2
GDWM (Ours)	8	30.2	8.2	27.5	42.0	93.8	49.5	41.9
GDWM (Ours)	16	30.8	8.6	28.1	43.1	94.3	50.2	42.5
GDWM (Ours)	32	**31.5**	**9.0**	**28.8**	**44.2**	**94.8**	**51.0**	**43.2**
Δ vs Best		+5.7%	0.0%	+6.7%	+4.2%	-0.1%	+2.8%	+2.4%

Table 1: Main results on ZeroSCROLLS. GDWM-32 achieves consistent improvements over qTTT-32 across all tasks (+2.4–6.2% average). GDWM-8 demonstrates Pareto-optimal efficiency: 4× fewer gradient steps with comparable performance to qTTT-32.

mode collapse, ensuring global representation. See Appendix C for full analysis and Appendix A for the complete algorithm.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate on ZeroSCROLLS (Shaham et al., 2023) (6 tasks: GovReport, QMSum, Qasper, NarrativeQA, Quality, MuSiQue) and LongBench v2 (Bai et al., 2025), covering summarization, QA, multi-hop reasoning, and code understanding across sparse-to-dense information distributions.

Baselines. We compare against: (1) **In-context** (Brown et al., 2020a)—standard inference; (2) **Thinking** (Yang et al., 2025)—inference with chain-of-thought; (3) **qTTT** (Bansal et al., 2025)—with uniform sampling (8-32 steps).

Implementation. We use Qwen3 models (1.7B/4B/8B) as base LLMs with LoRA adapters (rank=16, α=32) applied to query and output projection matrices (W_Q, W_O). For test-time adaptation, we employ AdamW optimizer with learning rate 1×10^{-4} and no weight decay. Default hyperparameters: chunk size $S = 1024$ tokens, temperature $\tau = 1.0$, minimum coverage $k_{\text{min}} = 1$, total gradient steps $K_{\text{total}} = 8$ (for GDWM-8) or $K_{\text{total}} = 32$ (for GDWM-32), and maximum context length 32K tokens. CPMI estimation uses a sliding window of 512 tokens for local context. All experiments run on NVIDIA H20 GPUs with mixed-precision (bfloat16) training. For efficiency, we implement gradient

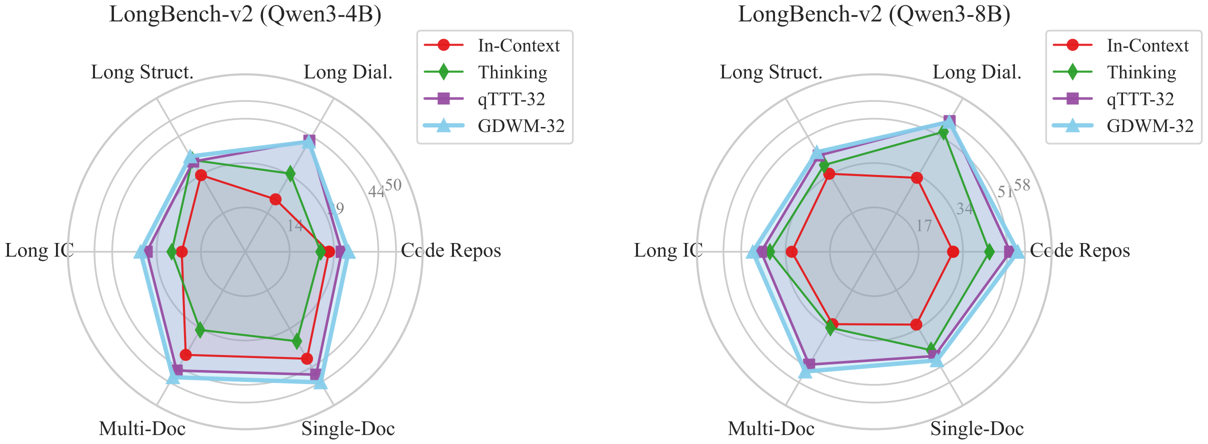

Figure 4: **Task-wise Performance on LongBench v2.** Radar charts comparing GDWM-32 (light blue) against baselines on 4B (left) and 8B (right) models. GDWM achieves consistent improvements on Code Repositories and Multi-Doc QA where information is sparse and localized, while showing competitive performance on Long Dialogue where global coverage is required.

checkpointing and flash attention v2 to reduce memory overhead during test-time adaptation.

5.2 Main Results

Table 1 presents our main results. GDWM-32 achieves consistent improvements over qTTT-32 across all tasks (+2.4–6.2% average). The largest gains appear on sparse-information tasks: GovReport (+11.2% on 1.7B, +7.8% on 4B) and Qasper (+12.7% on 1.7B, +6.7% on 8B), where relevant content is concentrated in specific document sections—exactly the scenario where CPMI-based selection excels. On Quality, which requires holistic comprehension with uniformly distributed information, improvements are more modest (−0.1%–+1.3%), revealing an expected characteristic of selective allocation.

As shown in Figure 1, GDWM-8 demonstrates Pareto-optimal efficiency: with 4× fewer gradient steps (8 vs 32), it achieves performance comparable to qTTT-32 (within 1 point margin on 4B, within 0.5 points on 8B average), while GDWM-32 pushes the frontier further with consistent 2.4–6.2% gains across model scales. This validates our core hypothesis that intelligent context selection outperforms brute-force computation.

To further validate GDWM's robustness across diverse task types, we evaluate on LongBench v2 (Bai et al., 2025) with results visualized in Figure 4. GDWM-32 achieves competitive or superior performance across model sizes and task categories. On Code Repositories (+5.5% on 8B) and Multi-Document QA (+6.2% on 8B)—tasks requiring precise retrieval from sparse, localized information—GDWM achieves consistent gains. However, on Long Dialogue, where information is distributed more uniformly, performance shows a slight trade-off (−0.5% on 8B), revealing an expected characteristic of selective allocation strategies.

CONFIGURATION	GOVRPT	QASPER	MUSIQUE	Δ
GDWM (full)	**28.5**	**33.8**	**27.0**	–
w/o CPMI (uniform)	23.9	27.3	19.9	−20.4%
w/o coverage	27.8	31.1	15.0	−17.3%
w/o chunking	24.7	27.8	10.0	−30.0%

Table 2: Ablation study on Qwen3-1.7B. Chunking is most critical (−30.0%); CPMI selection and coverage constraint both essential.

5.3 Ablation Studies

We conduct ablation experiments on Qwen3-1.7B to validate each component of GDWM, with results shown in Table 2.

Replacing CPMI-based selection with uniform sampling results in a 20.4% performance drop, demonstrating that intelligent context selection is crucial for effective test-time adaptation. The coverage constraint proves essential for multi-hop reasoning: removing it causes MuSiQue performance to plummet from 27.0 to 15.0, as the model overfits to a single high-utility region rather than gathering evidence from multiple document sections. Most critically, without chunking the model updates at token level, fragmenting context and incurring the largest drop (−30.0%), supporting our variance-reduction analysis.

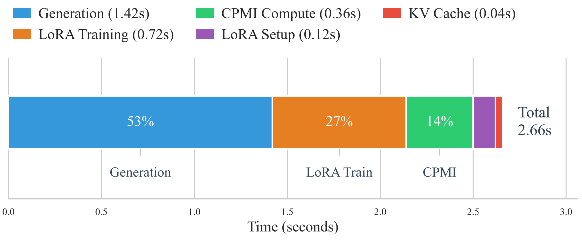

Figure 5: **Time Breakdown per Sample (Qwen3-4B).** Generation dominates at 53%, while CPMI computation accounts for only 13%—demonstrating the lightweight nature of our context selection mechanism. The 4× reduction in gradient steps (32→8) yields 39% net wall-clock speedup.

5.4 Scaling at Test Time

Table 3 compares GDWM against other test-time scaling approaches on Qwen3-4B under equal computational budget. Self-Consistency (SC-8) performs exceptionally well on multiple-choice tasks, marginally outperforming GDWM on Quality (82.7 vs 82.2) by aggregating diverse reasoning paths. However, it struggles on open-ended QA tasks (Qasper, MuSiQue) where the challenge lies in locating relevant context rather than generating diverse outputs. Beam Search proves largely ineffective for long-context understanding. GDWM maintains a 7.3-point lead on average (45.7 vs 38.4) over SC-8, validating that while ensemble methods help verification, intelligent context selection is more fundamental for evidence retrieval.

5.5 Efficiency Analysis

The primary overhead of GDWM stems from CPMI computation, which requires two forward passes per chunk (global and local predictions). As shown in Figure 5, with our default configuration ($S = 1024$, $K = 8$), the CPMI computation adds only 0.36s (~13% of total time), while the 4× reduction in gradient steps (from 32 to 8) saves 1.79s. The net result is a **39% wall-clock time reduction** compared to the 32-step baseline, demonstrating that intelligent context selection is not only more effective but also more efficient than brute-force uniform sampling.

Chunk Size Selection. Table 4 analyzes the trade-off between chunk granularity, computational overhead, and task performance. We adopt $S = 1024$ as the default configuration, which achieves the best *cross-task robustness*: avoiding catastrophic failures while maintaining competitive performance across all task types.

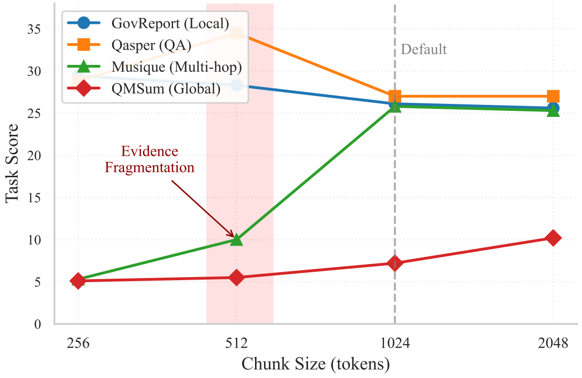

Figure 6: **Chunk Size Sensitivity.** Different task types exhibit distinct optimal chunk sizes. Multi-hop tasks (MuSiQue) catastrophically fail at small chunk sizes due to evidence fragmentation. $S = 1024$ provides robust cross-task performance.

Theoretical Analysis. The observed chunk size sensitivity has a principled explanation rooted in task-specific *evidence span*—the typical token distance over which task-relevant information is distributed. Let $E_{\mathcal{T}}$ denote this characteristic span for task $\mathcal{T}$. We identify a critical constraint: when chunk size $S < E_{\mathcal{T}}$, relevant evidence becomes fragmented across multiple chunks, causing CPMI to underestimate the true contextual utility of each fragment.

This framework explains the empirical patterns in Table 4. For **local reasoning tasks** such as GovReport, evidence is highly concentrated, so smaller chunks ($S = 256$) maximize selection precision. However, for **extractive QA tasks** like Qasper, performance peaks at intermediate granularity ($S = 512$), suggesting that while locality is important, overly small chunks ($S = 256$) may begin to fragment answer-relevant paragraphs. In contrast, **multi-hop reasoning tasks** like MuSiQue require evidence chains that span multiple paragraphs ($E_{\mathcal{T}} \approx 1000$+ tokens); when $S = 512 < E_{\mathcal{T}}$, the reasoning chain is severed—each fragment appears low-utility in isolation, leading to catastrophic failure (10.0 vs 25.8). Meanwhile, **summarization tasks** such as QMSum require near-uniform coverage ($E_{\mathcal{T}} \approx$ full document), where larger chunks ($S = 2048$) better preserve global structure.

The choice of $S = 1024$ represents the *minimum chunk size that avoids evidence fragmentation* for multi-hop tasks while retaining sufficient granularity for local reasoning—a principled middle ground validated by our cross-task robustness results. See Appendix J for formal analysis.

Method	Budget	GovReport	Qasper	Quality	Musique	*Avg.*
Thinking	1×	20.1	24.5	76.2	7.5	32.1
Self-Consistency (SC-8)	8×	24.5	28.4	82.7	18.1	38.4
Beam Search (k=8)	8×	23.0	26.2	77.8	14.2	35.3
qTTT (8 steps)	8×	29.2	29.5	81.3	23.7	40.9
GDWM (Ours)	8×	**34.2**	**35.2**	82.2	**31.2**	**45.7**

Table 3: Test-time scaling comparison on Qwen3-4B under equal compute budget (8× base). Intelligent context selection (GDWM) outperforms brute-force sampling (SC-8) and search (Beam) by significant margins, particularly on sparse-information tasks.

Chunk	CPMI%	Overhead	Avg	Worst
512	25%	+12%	19.4	10.0[†]
1024	**13%**	**+6%**	**21.7**	**25.8**
2048	9%	−10%	22.3	25.3

Table 4: Chunk size trade-offs on representative tasks (GovReport, Qasper, Musique, QMSum). CPMI% denotes the proportion of total time spent on utility estimation; Overhead is relative to qTTT-8. $S = 1024$ achieves the best *cross-task robustness*: the only configuration without catastrophic failure (Worst ≥ 25). [†] Small chunks fragment multi-hop evidence chains, causing CPMI to underestimate chunk utility (see Appendix J).

5.6 Discussion

Our experiments reveal a clear pattern regarding when GDWM excels. The largest gains appear on tasks with sparse information distribution, where relevant content is concentrated in specific document regions. GovReport (+5.7% on 8B) and Qasper (+6.7% on 8B) exemplify this phenomenon, as government reports contain key findings in specific sections while scientific papers concentrate answers near figures or method descriptions. On tasks requiring dense global coverage—QMSum and NarrativeQA—improvements are more modest but still positive, demonstrating that the coverage constraint effectively maintains global representation. Quality shows minimal improvement (0.0% on 8B), consistent with its requirement for uniform attention across the entire document.

Scaling Behavior. The scaling behavior shows consistent patterns: the efficiency advantage holds across model sizes (+6.2% for 1.7B, +4.1% for 4B, +2.4% for 8B average improvement for GDWM-32 vs qTTT-32). Interestingly, smaller models benefit more from intelligent context selection, likely because they have limited capacity to attend to all context uniformly and thus benefit more from prioritized consolidation. Gains on sparse tasks remain substantial across scales (Qasper: +12.7% on 1.7B, +8.0% on 4B, +6.7% on 8B; GovReport: +11.2% on 1.7B, +7.8% on 4B, +5.7% on 8B).

Chunk Size Trade-offs. Figure 6 visualizes the task-specific chunk size preferences. The dramatic performance collapse of MuSiQue at $S = 512$ (dropping to 10.0 from 25.8 at $S = 1024$) directly validates our Evidence Span theory: when chunk size falls below the task's characteristic evidence span $E_{\mathcal{T}}$, the reasoning chain is severed and CPMI underestimates true utility.

Three limitations merit future investigation: (i) fixed-size chunking can split coherent regions in irregularly structured documents, potentially harming selection reliability; (ii) the optimal chunk size is task-dependent (Table 4), motivating adaptive or structure-aware chunking; and (iii) for dense-coverage tasks (e.g., QMSum, NarrativeQA), selective allocation may offer limited gains, reflecting an inherent efficiency–coverage trade-off.

6 Conclusion

We presented GDWM (Gated Differentiable Working Memory), a framework that recasts test-time adaptation as a budget-constrained memory consolidation problem. By introducing a Write Controller that estimates Contextual Utility—an information-theoretic measure of long-range dependency—and allocates gradient budget via a coverage-first strategy, GDWM reduces gradient steps by 4× compared to uniform baselines while achieving comparable or superior performance on sparse-information tasks, with 39% wall-clock speedup. Our theoretical analysis proves that chunk-restricted sampling reduces gradient variance by eliminating inter-chunk variance, providing a principled explanation for the improved convergence. Future directions include semantic-aware dynamic chunking, task-adaptive temperature scheduling, and extending GDWM to multimodal contexts.

Limitations

While GDWM establishes a new Pareto frontier for efficient test-time adaptation, we identify three limitations inherent to our budget-constrained design. First, our current implementation employs fixed-size chunking ($S = 1024$). Although our analysis shows this setting is robust across diverse tasks, it may sub-optimally fragment semantic units in documents with irregular structures. However, we view this as an efficiency trade-off: dynamic chunking would require additional forward passes for segmentation, potentially offsetting the speed gains. Second, as observed in our results on Long Dialogue and Quality, the selective allocation strategy is less effective for tasks requiring uniform information coverage. This is a structural property of any compressive memory system rather than a flaw of GDWM specifically; users must weigh the $4\times$ efficiency gain against the marginal performance trade-off on dense-coverage tasks. Third, GDWM introduces hyperparameters (e.g., temperature τ, min coverage $k_{\min}$). While our experiments demonstrate that the default configuration ($\tau = 1.0, k_{\min} = 1$) generalizes well without tuning, extreme domain shifts might require recalibration of the utility estimator.

Ethical Considerations

Our work improves the efficiency of inference-time adaptation for Large Language Models. By reducing the gradient steps required for effective context adaptation by $4\times$, GDWM can lower the computational cost of long-context deployment and thereby supports the broader goal of Green AI.

GDWM is a general-purpose optimization layer and does not introduce new task capabilities or new access to information beyond what the underlying base model and the provided context already permit. As with any efficiency improvement, it may enable wider or more frequent use of long-context systems; the associated downstream risks (e.g., misuse of generative models) are therefore not specific to our mechanism, but to the deployment setting and the base model's safety properties. In practice, these concerns are best addressed through standard governance and safeguards at the system level—such as strong base-model alignment, access control, privacy-preserving data handling, and application-layer monitoring—rather than by restricting inference-time optimization techniques.

Acknowledgements

This work is supported in part by the National Key R&D Program of China under Grant Nos. 2023YFA1011602, Beijing Natural Science Foundation No. 4262033 and the National Natural Science Foundation of China under Grant Nos. U25B2076, 62441229, and 62377043.

References

Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. 2025. The surprising effectiveness of test-time training for few-shot learning. *Preprint*, arXiv:2411.07279.

J. Allingham, Jie Ren, Michael W. Dusenberry, J. Liu, Xiuye Gu, Yin Cui, Dustin Tran, and Balaji Lakshminarayanan. 2023. A simple zero-shot prompt weighting technique to improve prompt ensembling in text-image models. In *International Conference on Machine Learning*.

Petr Anokhin, Nikita Semenov, Artyom Y. Sorokin, Dmitry Evseev, M. Burtsev, and Evgeny Burnaev. 2024. Arigraph: Learning knowledge graph world models with episodic memory for llm agents. In *International Joint Conference on Artificial Intelligence*.

Anonymous. 2025. In-place test-time training. In *ICLR 2026 Conference Submission*. Under review at ICLR 2026; Available on OpenReview.

Jimmy Ba, Geoffrey Hinton, Volodymyr Mnih, Joel Z. Leibo, and Catalin Ionescu. 2016. Using fast weights to attend to the recent past. *Preprint*, arXiv:1610.06258.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2025. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *Preprint*, arXiv:2412.15204.

S. Banasik. 2025. Memory access characterization of large language models in cpu environment and its potential impacts. *arXiv.org*.

Rachit Bansal, Aston Zhang, Rishabh Tiwari, Lovish Madaan, Sai Surya Duvvuri, Devvrit Khatri, David Brandfonbrener, David Alvarez-Melis, Prajjwal Bhargava, Mihir Sanjay Kale, and Samy Jelassi. 2025. Let's (not) just put things in context: Test-time training for long-context llms. *Preprint*, arXiv:2512.13898.

Saikat Barua. 2024. Exploring autonomous agents through the lens of large language models: A review. *arXiv.org*.

Baolong Bi, Shaohan Huang, Yiwei Wang, Tianchi Yang, Zihan Zhang, Haizhen Huang, Lingrui Mei, Junfeng Fang, Zehao Li, Furu Wei, and 1 others. 2024a. Context-dpo: Aligning language models for context-faithfulness. *ACL 2025*.

Baolong Bi, Shenghua Liu, Lingrui Mei, Yiwei Wang, Junfeng Fang, Pengliang Ji, and Xueqi Cheng. 2025a. Decoding by contrasting knowledge: Enhancing large language model confidence on edited facts. In *ACL 2025*, pages 17198–17208.

Baolong Bi, Shenghua Liu, Xingzhang Ren, Dayiheng Liu, Junyang Lin, Yiwei Wang, Lingrui Mei, Junfeng Fang, Jiafeng Guo, and Xueqi Cheng. 2025b. Refinex: Learning to refine pre-training data at scale from expert-guided programs. *arXiv preprint arXiv:2507.03253*.

Baolong Bi, Shenghua Liu, Yiwei Wang, Lingrui Mei, and Xueqi Cheng. 2024b. Lpnl: Scalable link prediction with large language models. In *ACL 2024*, pages 3615–3625.

Baolong Bi, Shenghua Liu, Yiwei Wang, Lingrui Mei, Junfeng Fang, Hongcheng Gao, Shiyu Ni, and Xueqi Cheng. 2024c. Is factuality enhancement a free lunch for llms? better factuality can lead to worse context-faithfulness. *ICLR 2025*.

Baolong Bi, Shenghua Liu, Yiwei Wang, Lingrui Mei, Hongcheng Gao, Junfeng Fang, and Xueqi Cheng. 2024d. Struedit: Structured outputs enable the fast and accurate knowledge editing for large language models. *arXiv preprint arXiv:2409.10132*.

Baolong Bi, Shenghua Liu, Yiwei Wang, Lingrui Mei, Hongcheng Gao, Yilong Xu, and Xueqi Cheng. 2024e. Adaptive token biaser: Knowledge editing via biasing key entities. In *EMNLP 2024*, pages 11071–11083.

Baolong Bi, Shenghua Liu, Yiwei Wang, Siqian Tong, Lingrui Mei, Yuyao Ge, Yilong Xu, Jiafeng Guo, and Xueqi Cheng. 2025c. Reward and guidance through rubrics: Promoting exploration to improve multi-domain reasoning. *arXiv preprint arXiv:2511.12344*.

Baolong Bi, Shenghua Liu, Yiwei Wang, Yilong Xu, Junfeng Fang, Lingrui Mei, and Xueqi Cheng. 2025d. Parameters vs. context: Fine-grained control of knowledge reliance in language models. *arXiv preprint arXiv:2503.15888*.

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

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

Zhihang Cai, Xingjun Zhang, Zhendong Tan, and Zheng Wei. 2025. Nqkv: A kv cache quantization scheme based on normal distribution characteristics. *arXiv.org*.

Zouying Cao, Jiaji Deng, Li Yu, Weikang Zhou, Zhaoyang Liu, Bolin Ding, and Hai Zhao. 2025. Remember me, refine me: A dynamic procedural memory framework for experience-driven agent evolution.

Debashish Chakraborty, Eugene Yang, Daniel Khashabi, Dawn J. Lawrie, and Kevin Duh. 2025. Principled context engineering for rag: Statistical guarantees via conformal prediction.

Vivek Chari, Guanghui Qin, and Benjamin Van Durme. 2025. Kv-distill: Nearly lossless learnable context compression for llms. *Preprint*, arXiv:2503.10337.

Jianing Chen, Zehao Li, Yujun Cai, Hao Jiang, Shuqin Gao, Honglong Zhao, Tianlu Mao, and Yucheng Zhang. 2025a. From tokens to nodes: Semantic-guided motion control for dynamic 3d gaussian splatting. *Preprint*, arXiv:2510.02732.

Jianing Chen, Zehao Li, Yujun Cai, Hao Jiang, Chengxuan Qian, Juyuan Kang, Shuqin Gao, Honglong Zhao, Tianlu Mao, and Yucheng Zhang. 2025b. Haif-gs: Hierarchical and induced flow-guided gaussian splatting for dynamic scene. In *NeurIPS 2025*.

Juan Chen, Baolong Bi, Wei Zhang, Jingyan Sui, Xiaofei Zhu, Yuanzhuo Wang, Lingrui Mei, and Shenghua Liu. 2025c. Rethinking all evidence: Enhancing trustworthy retrieval-augmented generation via conflict-driven summarization. *arXiv preprint arXiv:2507.01281*.

Lizhe Chen, Yan Hu, Yu Zhang, Yuyao Ge, Haoyu Zhang, and Xingquan Cai. 2024. Frequency-importance gaussian splatting for real-time lightweight radiance field rendering. *Multimedia Tools and Applications*, 83(35):83377–83401.

Lizhe Chen, Binjia Zhou, Yuyao Ge, Jiayi Chen, and Shiguang Ni. 2025d. Pis: Linking importance sampling and attention mechanisms for efficient prompt compression. *arXiv preprint arXiv:2504.16574*.

Shaoshen Chen, Yangning Li, Zishan Xu, Yinghui Li, Xin Su, Zifei Shan, and Hai tao Zheng. 2025e. Dast: Context-aware compression in llms via dynamic allocation of soft tokens. *Preprint*, arXiv:2502.11493.

Yuxin Chen, Peng Tang, Weidong Qiu, and Shujun Li. 2025f. Using llms for automated privacy policy analysis: Prompt engineering, fine-tuning and explainability. *arXiv.org*.

Feng Cheng, Cong Guo, Chiyue Wei, Junyao Zhang, Changchun Zhou, Edward Hanson, Jiaqi Zhang, Xiaoxiao Liu, H. Li, and Yiran Chen. 2025. Ecco: Improving memory bandwidth and capacity for llms via entropy-aware cache compression. In *International Symposium on Computer Architecture*.

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. *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*, pages 3829–3846, Singapore. Association for Computational Linguistics.

Krishna Teja Chitty-Venkata, Jie Ye, Xian-He Sun, Anthony Kougkas, M. Emani, Venkat Vishwanath, and Bogdan Nicolae. 2025. Pagedeviction: Structured block-wise kv cache pruning for efficient large language model inference. *arXiv.org*.

Jincheng Dai, Zhuowei Huang, Haiyun Jiang, Chen Chen, Deng Cai, Wei Bi, and Shuming Shi. 2024. Corm: Cache optimization with recent message for large language model inference.

Tri Dao, Daniel Y. Fu, Stefano Ermon, A. Rudra, and Christopher R'e. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Neural Information Processing Systems*.

Jingcheng Deng, Liang Pang, Zihao Wei, Shichen Xu, Zenghao Duan, Kun Xu, Yang Song, Huawei Shen, and Xueqi Cheng. 2025. Latent reasoning in llms as a vocabulary-space superposition. *arXiv preprint arXiv:2510.15522*.

Yiming Du, Wenyu Huang, Danna Zheng, Zhaowei Wang, Sébastien Montella, Mirella Lapata, Kam-Fai Wong, and Jeff Z. Pan. 2025a. Rethinking memory in llm based agents: Representations, operations, and emerging topics.

Yufeng Du, Minyang Tian, Srikanth Ronanki, Subendhu Rongali, Sravan Bodapati, Aram Galstyan, Azton Wells, Roy Schwartz, Eliu A Huerta, and Hao Peng. 2025b. Context length alone hurts llm performance despite perfect retrieval. *Preprint*, arXiv:2510.05381.

Yuchen Duan, Zhe Chen, Yusong Hu, Weiyun Wang, Shenglong Ye, Botian Shi, Lewei Lu, Qibin Hou, Tong Lu, Hongsheng Li, Jifeng Dai, and Wenhai Wang. 2025a. Docopilot: Improving multimodal models for document-level understanding. *Computer Vision and Pattern Recognition*.

Zenghao Duan, Wenbin Duan, Zhiyi Yin, Yinghan Shen, Shaoling Jing, Jie Zhang, Huawei Shen, and Xueqi Cheng. 2025b. Related knowledge perturbation matters: Rethinking multiple pieces of knowledge editing in same-subject. *arXiv preprint arXiv:2502.06868*.

Zenghao Duan, Liang Pang, Zihao Wei, Wenbin Duan, Yuxin Tian, Shicheng Xu, Jingcheng Deng, Zhiyi Yin, and Xueqi Cheng. 2026a. Circular reasoning: Understanding self-reinforcing loops in large reasoning models. *arXiv preprint arXiv:2601.05693*.

Zenghao Duan, Zhiyi Yin, Zhichao Shi, Liang Pang, Shaoling Jing, Zihe Huang, Jiayi Wu, Yu Yan, Jingcheng Deng, Huawei Shen, and 1 others. 2026b. Projecting out the malice: A global subspace approach to llm detoxification. *arXiv preprint arXiv:2601.06226*.

Junjie Fang, Likai Tang, Hongzhe Bi, Yujia Qin, Si Sun, Zhenyu Li, Haolun Li, Yongjian Li, Xin Cong, Yukun Yan, Xiaodong Shi, Sen Song, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. Unimem: Towards a unified view of long-context large language models. *arXiv.org*.

Runnan Fang, Yuan Liang, Xiaobin Wang, Jialong Wu, Shuofei Qiao, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. 2025. Memp: Exploring agent procedural memory. *arXiv.org*.

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. *Preprint*, arXiv:2501.12959.

Honghao Fu, Yuan Ouyang, Kai-Wei Chang, Yiwei Wang, and Yujun Cai. 2025a. Contextnav: Towards agentic multimodal in-context learning. *arXiv preprint arXiv:2510.04560*.

Honghao Fu, Junlong Ren, Qi Chai, Deheng Ye, Yujun Cai, and Hao Wang. 2025b. Vistawise: Building cost-effective agent with cross-modal knowledge graph for minecraft. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21895–21909.

Honghao Fu, Hao Wang, Jing Jih Chin, and Zhiqi Shen. 2025c. Brainvis: Exploring the bridge between brain and visual signals via image reconstruction. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Honghao Fu, Yufei Wang, Wenhan Yang, Alex C Kot, and Bihan Wen. 2024a. Dp-iqa: Utilizing diffusion prior for blind image quality assessment in the wild. *arXiv preprint arXiv:2405.19996*.

Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hanna Hajishirzi, Yoon Kim, and Hao Peng. 2024b. Data engineering for scaling language models to 128k context. In *International Conference on Machine Learning*.

Pin Gao, Lingfan Yu, Yongwei Wu, and Jinyang Li. 2018. Low latency rnn inference with cellular batching. In *European Conference on Computer Systems*.

Haonan Ge, Yiwei Wang, Kai-Wei Chang, Hang Wu, and Yujun Cai. 2025a. Framemind: Frame-interleaved video reasoning via reinforcement learning. *arXiv preprint arXiv:2509.24008*.

Yuyao Ge, Shenghua Liu, Baolong Bi, Yiwei Wang, Lingrui Mei, Wenjie Feng, Lizhe Chen, and Xueqi Cheng. 2024. Can graph descriptive order affect solving graph problems with llms? *ACL 2025*.

Yuyao Ge, Shenghua Liu, Yiwei Wang, Lingrui Mei, Baolong Bi, Xuanshan Zhou, Jiayu Yao, Jiafeng Guo, and Xueqi Cheng. 2025b. Focusing by contrastive attention: Enhancing vlms' visual reasoning. *arXiv preprint arXiv:2509.06461*.

Yuyao Ge, Shenghua Liu, Yiwei Wang, Lingrui Mei, Lizhe Chen, Baolong Bi, and Xueqi Cheng. 2025c. Innate reasoning is not enough: In-context learning enhances reasoning large language models with less overthinking. *arXiv preprint arXiv:2503.19602*.

Yuyao Ge, Lingrui Mei, Zenghao Duan, Tianhao Li, Yujia Zheng, Yiwei Wang, Lexin Wang, Jiayu Yao, Tianyu Liu, Yujun Cai, and 1 others. 2025d. A survey of vibe coding with large language models. *arXiv preprint arXiv:2510.12399*.

Yuyao Ge, Zhongguo Yang, Lizhe Chen, Yiming Wang, and Chengyang Li. 2023. Attack based on data: a novel perspective to attack sensitive points directly. *Cybersecurity*, 6(1):43.

Sanjay Govindan, M. Pagnucco, and Yang Song. 2025. Temporal alignment of time sensitive facts with activation engineering. In *Conference on Empirical Methods in Natural Language Processing*.

Jing Guo, Nan Li, J. Qi, Hang Yang, Ruiqiao Li, Yuzhen Feng, Si Zhang, and Ming Xu. 2023. Empowering working memory for large language model agents. *arXiv.org*.

Md. Asif Haider, Ayesha Binte Mostofa, Sk Md Mosaddek Hossain, Anindya Iqbal, and Toufique Ahmed. 2024. Prompting and fine-tuning large language models for automated code review comment generation. *arXiv.org*.

Shanshan Han, Qifan Zhang, Yuhang Yao, Weizhao Jin, Zhaozhuo Xu, and Chaoyang He. 2024. Llm multi-agent systems: Challenges and open problems. *arXiv.org*.

Moritz Hardt and Yu Sun. 2024. Test-time training on nearest neighbors for large language models. *Preprint*, arXiv:2305.18466.

Jerry Zhi-Yang He, Sashrika Pandey, Mariah L. Schrum, and Anca Dragan. 2024a. Context steering: Controllable personalization at inference time.

Zhiyuan He, Huiqiang Jiang, Zilong Wang, Yuqing Yang, Luna K. Qiu, and Lili Qiu. 2024b. Position engineering: Boosting large language models through positional information manipulation. In *Conference on Empirical Methods in Natural Language Processing*.

Noah Hollmann, Samuel G. Müller, and F. Hutter. 2023. Large language models for automated data science: Introducing caafe for context-aware automated feature engineering. *Neural Information Processing Systems*.

Kelly Hong, Anton Troynikov, and Jeff Huber. 2025. Context rot: How increasing input tokens impacts llm performance. Technical report, Chroma.

Haowen Hou, Fei Ma, Binwen Bai, Xinxin Zhu, and F. Yu. 2024a. Enhancing and accelerating large language models via instruction-aware contextual compression. *arXiv.org*.

Yuki Hou, Haruki Tamoto, Qinghua Zhao, and Homei Miyashita. 2024b. Synapticrag: Enhancing temporal memory retrieval in large language models through synaptic mechanisms. *Annual Meeting of the Association for Computational Linguistics*.

N. Houlsby, A. Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and S. Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? *Preprint*, arXiv:2404.06654.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander J. Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *Annual Meeting of the Association for Computational Linguistics*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. 2024a. Hiagent: Hierarchical working memory management for solving long-horizon agent tasks with large language model. *Annual Meeting of the Association for Computational Linguistics*.

Sihao Hu, Tiansheng Huang, Fatih Ilhan, S. Tekin, Gaowen Liu, R. Kompella, and Ling Liu. 2024b. A survey on large language model-based game agents. *arXiv.org*.

Yan Hu, Lizhe Chen, Hanna Xie, Yuyao Ge, Shun Zhou, and Xingquan Cai. 2024c. Real-time non-photorealistic rendering method for black and white comic style in games and animation. *Journal of System Simulation*, 36(7):1699–1712.

Yang Hu, Xingyu Zhang, Xueji Fang, Zhiyang Chen, Xiao Wang, Huatian Zhang, and Guojun Qi. 2025a. Slot: Sample-specific language model optimization at test-time. *Preprint*, arXiv:2505.12392.

Yuanzhe Hu, Yu Wang, and Julian McAuley. 2025b. Evaluating memory in llm agents via incremental multi-turn interactions. *arXiv.org*.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, Senjie Jin, Jiejun Tan, Yanbin Yin, Jiongnan Liu, Zeyu Zhang, Zhongxiang Sun, Yutao Zhu, Hao Sun, Boci Peng, and 28 others. 2026. Memory in the age of ai agents. *Preprint*, arXiv:2512.13564.

Qishuo Hua, Lyumanshan Ye, Dayuan Fu, Yang Xiao, Xiaojie Cai, Yunze Wu, Jifan Lin, Junfei Wang, and Pengfei Liu. 2025. Context engineering 2.0: The context of context engineering. *arXiv.org*.

Hai Huang. 2025. Directed information γ-covering: An information-theoretic framework for context engineering. *arXiv.org*.

Le Huang, Hengzhi Lan, Zijun Sun, Chuan Shi, and Ting Bai. 2024. Emotional rag: Enhancing role-playing agents through emotional retrieval. In *2024 IEEE International Conference on Knowledge Graph (ICKG)*.

Zhengjun Huang, Zhoujin Tian, Qintian Guo, Fangyuan Zhang, Yingli Zhou, Di Jiang, and Xiaofang Zhou. 2025. Licomemory: Lightweight and cognitive agentic memory for efficient long-term reasoning. *arXiv.org*.

Jonas Hübotter, Sascha Bongni, Ido Hakimi, and Andreas Krause. 2024. Efficiently learning at test-time: Active fine-tuning of llms. *ArXiv*, abs/2410.08020.

Jonas Hübotter, Sascha Bongni, Ido Hakimi, and Andreas Krause. 2025. Efficiently learning at test-time: Active fine-tuning of llms. *Preprint*, arXiv:2410.08020.

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. *Preprint*, arXiv:2310.06839.

Josip Juki'c, Martin Tutek, and Jan Snajder. 2025. Context parametrization with compositional adapters. *arXiv.org*.

Seungyeon Jwa, Daechul Ahn, Reokyoung Kim, Dongyeop Kang, and Jonghyun Choi. 2025. Becoming experienced judges: Selective test-time learning for evaluators. In *unknown*.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. 2025a. Memory os of ai agent. *arXiv.org*.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. 2025b. Memory os of ai agent. *arXiv.org*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Yu Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Conference on Empirical Methods in Natural Language Processing*.

Richard Katrix, Quentin Carroway, Rowan Hawkesbury, and Matthias Heathfield. 2025. Context-aware semantic recomposition mechanism for large language models. *arXiv.org*.

Anant Khandelwal, Manish Gupta, and Puneet Agrawal. 2025. Cocoa: Confidence and context-aware adaptive decoding for resolving knowledge conflicts in large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

Nikoletta Koilia and C. Kachris. 2024. Hardware acceleration of llms: A comprehensive survey and comparison. *arXiv.org*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Haotong Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Symposium on Operating Systems Principles*.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, F. Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, M. Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Neural Information Processing Systems*.

Dongfang Li, Zetian Sun, Xinshuo Hu, Baotian Hu, and Min Zhang. 2024a. Cmt: A memory compression method for continual knowledge learning of large language models. *arXiv.org*.

Mufei Li, Dongqi Fu, Limei Wang, Si Zhang, Hanqing Zeng, Kaan Sancak, Ruizhong Qiu, H. Wang, Xiaoxin He, Xavier Bresson, Yinglong Xia, Chonglin Sun, and Pan Li. 2025a. Haystack engineering: Context engineering for heterogeneous and agentic long-context evaluation. *arXiv.org*.

Sifan Li, Yujun Cai, Bryan Hooi, Nanyun Peng, and Yiwei Wang. 2025b. Do "new snow tablets" contain snow? large language models over-rely on names to identify ingredients of chinese drugs. *Preprint*, arXiv:2504.03786.

Sifan Li, Yujun Cai, and Yiwei Wang. 2025c. SemVink: Advancing VLMs' semantic understanding of optical illusions via visual global thinking. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27167–27177, Suzhou, China. Association for Computational Linguistics.

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

Zhecheng Li, Guoxian Song, Yujun Cai, Zhen Xiong, Junsong Yuan, and Yiwei Wang. 2025d. Texture or semantics? vision-language models get lost in font recognition. In *Conference on Language Modeling COLM, 2025.*

Zhecheng Li, Guoxian Song, Yiwei Wang, Zhen Xiong, Junsong Yuan, and Yujun Cai. 2025e. A^2R^2: Advancing img2latex conversion via visual reasoning with attention-guided refinement. *arXiv preprint arXiv:2507.20890.*

Zhecheng Li, Guoxian Song, Yiwei Wang, Zhen Xiong, Junsong Yuan, and Yujun Cai. 2025f. Generalist scanner meets specialist locator: A synergistic coarse-to-fine framework for robust gui grounding. *arXiv preprint arXiv:2509.24133.*

Zhecheng Li, Yiwei Wang, Bryan Hooi, Yujun Cai, Naifan Cheung, Nanyun Peng, and Kai-Wei Chang. 2024b. Think carefully and check again! meta-generation unlocking llms for low-resource cross-lingual summarization. *arXiv preprint arXiv:2410.20021.*

Zhecheng Li, Yiwei Wang, Bryan Hooi, Yujun Cai, Nanyun Peng, and Kai-Wei Chang. 2024c. Drs: Deep question reformulation with structured output. In *Association for Computational Linguistics ACL, 2025.*

Zhecheng Li, Yiwei Wang, Bryan Hooi, Yujun Cai, Zhen Xiong, Nanyun Peng, and Kai-Wei Chang. 2024d. Vulnerability of llms to vertically aligned text manipulations. In *Association for Computational Linguistics ACL, 2025.*

Zhiyu Li, Shichao Song, Hanyu Wang, Simin Niu, Ding Chen, Jiawei Yang, Chenyang Xi, Huayi Lai, Jihao Zhao, Yezhaohui Wang, Junpeng Ren, Zehao Lin, Jiahao Huo, Tianyi Chen, Kai Chen, Ke-Rong Li, Zhiqiang Yin, Qingchen Yu, Bo Tang, and 3 others. 2025g. Memos: An operating system for memory-augmented generation (mag) in large language models. *arXiv.org*.

Zhiyu Li, Shichao Song, Chenyang Xi, Hanyu Wang, Chen Tang, Simin Niu, Ding Chen, Jiawei Yang, Chunyu Li, Qingchen Yu, Jihao Zhao, Yezhaohui Wang, Peng Liu, Zehao Lin, Pengyuan Wang, Jiahao Huo, Tianyi Chen, Kai Chen, Ke-Rong Li, and 20 others. 2025h. Memos: A memory os for ai system. *arXiv.org*.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, and 1 others. 2025i. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419.*

Chang Liu, Hongkai Chen, Yujun Cai, Hang Wu, Qingwen Ye, Ming-Hsuan Yang, and Yiwei Wang. 2025a. Structured attention matters to multimodal llms in document understanding. *arXiv preprint arXiv:2506.21600.*

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023a. Lost in the middle: How language models use long contexts. *Preprint*, arXiv:2307.03172.

Quancai Liu, Haihui Fan, Jinchao Zhang, Xiangfang Li, Chuanrong Li, and Bo Li. 2025b. DisComp: A two-stage prompt optimization framework combining task-agnostic and task-aware compression. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1033–1044, Albuquerque, New Mexico. Association for Computational Linguistics.

Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. 2023b. Scissorhands: Exploiting the persistence of importance hypothesis for llm kv cache compression at test time. *Neural Information Processing Systems*.

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

Linyue Ma, Yilong Xu, Xiang Long, and Zhi Zheng. 2025. An efficient rubric-based generative verifier for search-augmented llms. *arXiv preprint arXiv:2510.14660.*

Yansheng Mao, Jiaqi Li, Fanxu Meng, Jing Xiong, Zilong Zheng, and Muhan Zhang. 2024. Lift: Improving long context understanding through long input fine-tuning. *arXiv.org*.

Lingrui Mei, Shenghua Liu, Yiwei Wang, Baolong Bi, and Xueqi Cheng. 2024a. Slang: New concept comprehension of large language models. *EMNLP 2024.*

Lingrui Mei, Shenghua Liu, Yiwei Wang, Baolong Bi, Yuyao Ge, Jun Wan, Yurong Wu, and Xueqi Cheng. 2025a. a1: Steep test-time scaling law via environment augmented generation. *arXiv preprint arXiv:2504.14597.*

Lingrui Mei, Shenghua Liu, Yiwei Wang, Baolong Bi, Jiayi Mao, and Xueqi Cheng. 2024b. "not aligned" is not "malicious": Being careful about hallucinations of large language models' jailbreak. *COLING 2025.*

Lingrui Mei, Shenghua Liu, Yiwei Wang, Baolong Bi, Ruibin Yuan, and Xueqi Cheng. 2024c. Hiddenguard: Fine-grained safe generation with specialized representation router. *arXiv preprint arXiv:2410.02684.*

Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Baolong Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, Chenlin Zhou, Jiayi Mao, Tianze Xia, Jiafeng Guo, and Shenghua Liu. 2025b. A survey of context engineering for large language models. *arXiv.org*.

Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Baolong Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, and 1 others. 2025c. A survey of context engineering for large language models. *arXiv preprint arXiv:2507.13334*.

Adilet Metinov, Gulida M. Kudakeeva, Bolotbek uulu Nursultan, and G. Kabaeva. 2025. Adaptive soft rolling kv freeze with entropy-guided recovery: Sublinear memory growth for efficient llm inference.

Ali Modarressi, Ayyoob Imani, Mohsen Fayyaz, and Hinrich Schütze. 2023. Ret-llm: Towards a general read-write memory for large language models. *arXiv.org*.

Dr.Farheen Mohammed. 2025. Towards standardized evaluation of large language model-based agents. *INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*.

Mohammad Mahdi Moradi, Hossam Amer, Sudhir Mudur, Weiwei Zhang, Yang Liu, and Walid Ahmed. 2025. Continuous self-improvement of large language models by test-time training with verifier-driven sample selection. *ArXiv*, abs/2505.19475.

Dilxat Muhtar, Yelong Shen, Yaming Yang, Xiaodong Liu, Yadong Lu, Jianfeng Liu, Yuefeng Zhan, Hao Sun, Weiwei Deng, Feng Sun, Xueliang Zhang, Jianfeng Gao, Weizhu Chen, and Qi Zhang. 2024. Streamadapter: Efficient test time adaptation from contextual streams. *Preprint*, arXiv:2411.09289.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, S. Zheng, P. Zhao, and Mingkui Tan. 2022. Efficient test-time model adaptation without forgetting. In *International Conference on Machine Learning*.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv.org*.

N. C. Ohalete, Kevin B. Gittner, and Lauren M. Matheny. 2025. Costar-a: A prompting framework for enhancing large language model performance on point-of-view questions. *arXiv.org*.

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

Liang Pang, Kangxi Wu, Sunhao Dai, Zihao Wei, Zenghao Duan, Jia Gu, Xiang Li, Zhiyi Yin, Jun Xu, Huawei Shen, and 1 others. 2025. Large language model sourcing: A survey. *arXiv preprint arXiv:2510.10161*.

Daehee Park, Jaeseok Jeong, Sung-Hoon Yoon, Jaewoo Jeong, and Kuk-Jin Yoon. 2024. T4p: Test-time training of trajectory prediction via masked autoencoder and actor-specific token memory. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15065–15076.

Haewon Park, Gyubin Choi, Minjun Kim, and Yohan Jo. 2025a. Context-robust knowledge editing for language models. *Annual Meeting of the Association for Computational Linguistics*.

J. Park, Joseph C. O'Brien, Carrie J. Cai, M. Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *ACM Symposium on User Interface Software and Technology*.

Joon Park, Kyohei Atarashi, Koh Takeuchi, and Hisashi Kashima. 2025b. Emulating retrieval augmented generation via prompt engineering for enhanced long context comprehension in llms. *arXiv.org*.

Mathis Pink, Qinyuan Wu, Vy A. Vo, Javier S. Turek, Jianing Mu, Alexander Huth, and Mariya Toneva. 2025. Position: Episodic memory is the missing piece for long-term llm agents. *arXiv.org*.

Patricia Porretta, Sylvester Pakenham, Huxley Ainsworth, Gregory Chatten, Godfrey Allerton, Simon Hollingsworth, and Vance Periwinkle. 2025. Latent convergence modulation in large language models: A novel approach to iterative contextual realignment. *arXiv.org*.

Yuxiao Qu, Matthew Y. R. Yang, Amrith Rajagopal Setlur, Lewis Tunstall, Edward Beeching, Ruslan Salakhutdinov, and Aviral Kumar. 2025. Optimizing test-time compute via meta reinforcement fine-tuning. *ArXiv*, abs/2503.07572.

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Inbal Magar, Omri Abend, Ehud Karpas, A. Shashua, Kevin Leyton-Brown, and Y. Shoham. 2022. Parallel context windows for large language models. *Annual Meeting of the Association for Computational Linguistics*.

Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shuangyang Yang, Minjia Zhang, Dong Li, and Yuxiong He. 2021. Zero-offload: Democratizing billion-scale model training. In *USENIX Annual Technical Conference*.

Alireza Rezazadeh, Zichao Li, Wei Wei, and Yujia Bao. 2024. From isolated conversations to hierarchical schemas: Dynamic tree memory representation for llms. In *International Conference on Learning Representations*.

Sourjya Roy, Shrihari Sridharan, Surya Selvam, and A. Raghunathan. 2025. Kv-car: Kv cache compression using autoencoders and kv reuse in large language models.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, S. Mondal, and Aman Chadha. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv.org*.

Rana Salama, Jason Cai, Michelle Yuan, Anna Currey, Monica Sunkara, Yi Zhang, and Yassine Benajiba. 2025. Meminsight: Autonomous memory augmentation for llm agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. 2023. Zeroscrolls: A zero-shot benchmark for long text understanding. *Preprint*, arXiv:2305.14196.

Lianlei Shan, Shixian Luo, Zezhou Zhu, Yu Yuan, and Yong Wu. 2025. Cognitive memory in large language models. *arXiv.org*.

Weizhou Shen, Chenliang Li, Fanqi Wan, Shengyi Liao, Shaopeng Lai, Bo Zhang, Yingcheng Shi, Yuning Wu, Gang Fu, Zhansheng Li, Bin Yang, Ji Zhang, Fei Huang, Jingren Zhou, and Ming Yan. 2025. Qwenlong-cprs: Towards ∞-llms with dynamic context optimization. *arXiv.org*.

Ying Sheng, Lianmin Zheng, Binhang Yuan, Zhuohan Li, Max Ryabinin, Daniel Y. Fu, Zhiqiang Xie, Beidi Chen, Clark W. Barrett, Joseph Gonzalez, Percy Liang, Christopher Ré, Ion Stoica, and Ce Zhang. 2023. High-throughput generative inference of large language models with a single gpu. In *International Conference on Machine Learning*.

Kaize Shi, Xueyao Sun, Xiaohui Tao, Lin Li, Qika Lin, and Guandong Xu. 2025. Concept than document: Context compression via amr-based conceptual entropy.

Weijia Shi, Xiaochuang Han, M. Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and S. Yih. 2023. Trusting your evidence: Hallucinate less with context-aware decoding. *North American Chapter of the Association for Computational Linguistics*.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Eliciting knowledge from language models using automatically generated prompts. In *Conference on Empirical Methods in Natural Language Processing*.

Noah Shinn, Federico Cassano, Beck Labash, A. Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. *Neural Information Processing Systems*.

H. Shinwari and Muhammad Usama. 2025. Memory-augmented architecture for long-term context handling in large language models. *arXiv.org*.

Alina Shutova, Vladimir Malinovskii, Vage Egiazarian, Denis Kuznedelev, D. Mazur, Nikita Surkov, Ivan Ermakov, and Dan Alistarh. 2025. Cache me if you must: Adaptive key-value quantization for large language models. In *International Conference on Machine Learning*.

Sonish Sivarajkumar, Mark Kelley, Alyssa Samolyk-Mazzanti, Shyam Visweswaran, and Yanshan Wang. 2024. An empirical evaluation of prompting strategies for large language models in zero-shot clinical natural language processing: Algorithm development and validation study. *JMIR Medical Informatics*.

C. Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *ArXiv*, abs/2408.03314.

Woomin Song, Seunghyuk Oh, Sangwoo Mo, Jaehyung Kim, Sukmin Yun, Jung-Woo Ha, and Jinwoo Shin. 2024. Hierarchical context merging: Better long context understanding for pre-trained llms. In *International Conference on Learning Representations*.

Sruthi Sridhar, Abdulrahman Khamaj, and M. Asthana. 2023. Cognitive neuroscience perspective on memory: overview and summary. *Frontiers in Human Neuroscience*.

Sakhinana Sagar Srinivas and Venkataramana Runkana. 2025. Scaling test-time inference with policy-optimized, dynamic retrieval-augmented generation via kv caching and decoding. *arXiv.org*.

Huashan Sun, Shengyi Liao, Yansen Han, Yu Bai, Yang Gao, Cheng Fu, Weizhou Shen, Fanqi Wan, Mingshi Yan, Ji Zhang, and Fei Huang. 2025. Solopo: Unlocking long-context capabilities in llms via short-to-long preference optimization. *arXiv.org*.

Sijun Tan, Xiuyu Li, Shishir G. Patil, Ziyang Wu, Tianjun Zhang, Kurt Keutzer, Joseph E. Gonzalez, and Raluca A. Popa. 2024. Lloco: Learning long contexts offline. In *Conference on Empirical Methods in Natural Language Processing*.

Yushun Tang, Ce Zhang, Heng Xu, Shuoshuo Chen, Jie Cheng, Luziwei Leng, Qinghai Guo, and Zhihai He. 2023. Neuro-modulated hebbian learning for fully test-time adaptation. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3728–3738.

Alessandra Terranova, Bjorn Ross, and Alexandra Birch. 2025. Evaluating long-term memory for long-context question answering. *arXiv.org*.

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

Akash Vishwakarma, Hojin Lee, Mohith Suresh, Priyam Shankar Sharma, Rahul Vishwakarma, Sparsh Gupta, and Yuvraj Anupam Chauhan. 2025. Cognitive weave: Synthesizing abstracted knowledge with a spatio-temporal resonance graph. *arXiv.org*.

Jun Wan and Lingrui Mei. 2025. Large language models as computable approximations to solomonoff induction. *arXiv preprint arXiv:2505.15784*.

Junhao Wang, Daoguang Zan, Shulin Xin, Siyao Liu, Yurong Wu, and Kai Shen. 2025a. Swe-mirror: Scaling issue-resolving datasets by mirroring issues across repositories. *Preprint*, arXiv:2509.08724.

Lei Wang, Chengbang Ma, Xueyang Feng, Zeyu Zhang, Hao ran Yang, Jingsen Zhang, Zhi-Yang Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Ji rong Wen. 2023. A survey on large language model based autonomous agents. *Frontiers of Computer Science*.

Renxi Wang, Xudong Han, Lei Ji, Shu Wang, Timothy Baldwin, and Haonan Li. 2024a. Toolgen: Unified tool retrieval and calling via generation. In *International Conference on Learning Representations*.

Rushi Wang, Jiateng Liu, Cheng Qian, Yifan Shen, Yanzhou Pan, Zhaozhuo Xu, Ahmed Abbasi, Heng Ji, and Denghui Zhang. 2025b. Context engineering for trustworthiness: Rescorla wagner steering under mixed and inappropriate contexts. *arXiv.org*.

Xiangfeng Wang, Zaiyi Chen, Zheyong Xie, Tong Xu, Yongyi He, and Enhong Chen. 2024b. In-context former: Lightning-fast compressing context for large language model. In *Conference on Empirical Methods in Natural Language Processing*.

Yu Wang and Xi Chen. 2025. Mirix: Multi-agent memory system for llm-based agents. *arXiv.org*.

Yu Wang, Ryuichi Takanobu, Zhiqi Liang, Yuzhen Mao, Yuanzhe Hu, Julian McAuley, and Xiaojian Wu. 2025c. Mem-α: Learning memory construction via reinforcement learning. *arXiv.org*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *Neural Information Processing Systems*.

Zihao Wei, Liang Pang, Jiahao Liu, Jingcheng Deng, Shicheng Xu, Zenghao Duan, Jingang Wang, Fei Sun, Xunliang Cai, Huawei Shen, and 1 others. 2025. Stop spinning wheels: Mitigating llm overthinking via mining patterns for early reasoning exit. *arXiv preprint arXiv:2508.17627*.

Benjue Weng. 2024. Navigating the landscape of large language models: A comprehensive review and analysis of paradigms and fine-tuning strategies. *arXiv.org*.

Hang Wu, Yujun Cai, Haonan Ge, Hongkai Chen, Ming-Hsuan Yang, and Yiwei Wang. 2025a. Refineshot: Rethinking cinematography understanding with foundational skill evaluation. *arXiv preprint arXiv:2510.02423*.

Hang Wu, Hongkai Chen, Yujun Cai, Chang Liu, Qingwen Ye, Ming-Hsuan Yang, and Yiwei Wang. 2025b. Dimo-gui: Advancing test-time scaling in gui grounding via modality-aware visual reasoning. In *EMNLP 2025*.

Haoyi Wu and Kewei Tu. 2024. Layer-condensed kv cache for efficient inference of large language models. *Annual Meeting of the Association for Computational Linguistics*.

Tianyu Wu, Lingrui Mei, Ruibin Yuan, Lujun Li, Wei Xue, and Yike Guo. 2024. You know what i'm saying: Jailbreak attack via implicit reference. *arXiv preprint arXiv:2410.03857*.

Yangjian Wu and Gang Hu. 2023. Exploring prompt engineering with gpt language models for document-level machine translation: Insights and findings. In *Conference on Machine Translation*.

Yaxiong Wu, Sheng Liang, Chen Zhang, Yichao Wang, Yongyue Zhang, Huifeng Guo, Ruiming Tang, and Yong Liu. 2025c. From human memory to ai memory: A survey on memory mechanisms in the era of llms. *arXiv.org*.

Yurong Wu, Fangwen Mu, Qiuhong Zhang, Jinjing Zhao, Xinrun Xu, Lingrui Mei, Yang Wu, Lin Shi, Junjie Wang, Zhiming Ding, and 1 others. 2025d. Vulnerability of text-to-image models to prompt template stealing: A differential evolution approach. *ACL 2025*.

Zijun Wu, Yongchang Hao, and Lili Mou. 2025e. Tokmem: Tokenized procedural memory for large language models. *arXiv.org*.

Rui Xie, Asad Ul Haq, Linsen Ma, Yunhua Fang, Zirak Burzin Engineer, Liu Liu, and Tong Zhang. 2025. Reimagining memory access for llm inference: Compression-aware memory controller design. *arXiv.org*.

Zhen Xiong, Yujun Cai, Bryan Hooi, Nanyun Peng, Zhecheng Li, and Yiwei Wang. 2025a. Enhancing llm character-level manipulation via divide and conquer. *arXiv preprint arXiv:2502.08180*.

Zhen Xiong, Yujun Cai, Zhecheng Li, and Yiwei Wang. 2025b. Mapping the minds of llms: A graph-based analysis of reasoning llm. *arXiv preprint arXiv:2505.13890*.

Zhen Xiong, Yujun Cai, Zhecheng Li, and Yiwei Wang. 2025c. Unveiling the potential of diffusion large language model in controllable generation. *arXiv preprint arXiv:2507.04504*.

Zhen Xiong, Yujun Cai, Zhecheng Li, Junsong Yuan, and Yiwei Wang. 2025d. Thinking with sound: Audio chain-of-thought enables multimodal reasoning in large audio-language models. *arXiv preprint arXiv:2509.21749*.

Zidi Xiong, Yuping Lin, Wenya Xie, Pengfei He, Jiliang Tang, Himabindu Lakkaraju, and Zhen Xiang. 2025e. How memory management impacts llm agents: An empirical study of experience-following behavior. *arXiv.org*.

Chunxue Xu, Yiwei Wang, Bryan Hooi, Yujun Cai, and Songze Li. 2025a. How does watermarking affect visual language models in document understanding? *Preprint*, arXiv:2504.01048.

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

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025b. A-mem: Agentic memory for llm agents. *Preprint*, arXiv:2502.12110.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025c. A-mem: Agentic memory for llm agents. *arXiv.org*.

Yilong Xu, Jinhua Gao, Xiaoming Yu, Baolong Bi, Huawei Shen, and Xueqi Cheng. 2024. Aliice: Evaluating positional fine-grained citation generation. In *NAACL 2025 Main Conference*.

Yilong Xu, Jinhua Gao, Xiaoming Yu, Yuanhai Xue, Baolong Bi, Huawei Shen, and Xueqi Cheng. 2025d. Training a utility-based retriever through shared context attribution for retrieval-augmented language models. *arXiv preprint arXiv:2504.00573*.

Yilong Xu, Xiang Long, Zhi Zheng, and Jinhua Gao. 2025e. Ravine: Reality-aligned evaluation for agentic search. *arXiv preprint arXiv:2507.16725*.

Tengfei Xue, Xuefeng Li, Roman Smirnov, Tahir Azim, Arash Sadrieh, and Babak Pahlavan. 2024. Ninjallm: Fast, scalable and cost-effective rag using amazon sagemaker and aws trainium and inferentia2. *arXiv.org*.

Yu Yan, Sheng Sun, Zenghao Duan, Teli Liu, Min Liu, Zhiyi Yin, LeiJingyu LeiJingyu, and Qi Li. 2025a. from benign import toxic: Jailbreaking the language model via adversarial metaphors. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4785–4817.

Yu Yan, Sheng Sun, Zhe Wang, Yijun Lin, Zenghao Duan, Min Liu, Jianping Zhang, and 1 others. 2025b. Confusion is the final barrier: Rethinking jailbreak evaluation and investigating the real misuse threat of llms. *arXiv preprint arXiv:2508.16347*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Hanmei Yang, Jin Zhou, Yao Fu, Xiaoqun Wang, Ramine Roane, Hui Guan, and Tongping Liu. 2024. Protrain: Efficient llm training via memory-aware techniques.

Linyi Yang, Shuibai Zhang, Zhuohao Yu, Guangsheng Bao, Yidong Wang, Jindong Wang, Ruochen Xu, Weirong Ye, Xing Xie, Weizhu Chen, and Yue Zhang. 2023. Supervised knowledge makes large language models better in-context learners. In *International Conference on Learning Representations*.

Jiayu Yao, Shenghua Liu, Yiwei Wang, Rundong Cheng, Lingrui Mei, Baolong Bi, Zhen Xiong, and Xueqi Cheng. 2025a. Not in sync: Unveiling temporal bias in audio chat models. *arXiv preprint arXiv:2510.12185*.

Jiayu Yao, Shenghua Liu, Yiwei Wang, Lingrui Mei, Baolong Bi, Yuyao Ge, Zhecheng Li, and Xueqi Cheng. 2025b. Who is in the spotlight: The hidden bias undermining multimodal retrieval-augmented generation. *Preprint*, arXiv:2506.11063.

Jiayu Yao, Shenghua Liu, Yiwei Wang, Lingrui Mei, Baolong Bi, Yuyao Ge, Zhecheng Li, and Xueqi Cheng. 2025c. Who is in the spotlight: The hidden bias undermining multimodal retrieval-augmented generation. *EMNLP 2025*.

Rui Ye, Zhongwang Zhang, Kuan Li, Huifeng Yin, Zhengwei Tao, Yida Zhao, Liangcai Su, Liwen Zhang, Zile Qiao, Xinyu Wang, Pengjun Xie, Fei Huang, Siheng Chen, Jingren Zhou, and Yong Jiang. 2025. Agentfold: Long-horizon web agents with proactive context management. *Preprint*, arXiv:2510.24699.

Zhonghua Yi, Ge Niu, Lei Wang, Weijing Tang, and Liqiu Zhang. 2024. A method for building large language models with predefined kv cache capacity. *arXiv.org*.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. 2025. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. *Preprint*, arXiv:2507.02259.

Jian Yuan, Ziwei He, Haoli Bai, Jingwen Leng, and Bo Jiang. 2025a. Weightedkv: Attention scores weighted key-value cache merging for large language models. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*.

Ruibin Yuan, Hanfeng Lin, Shuyue Guo, Ge Zhang, Jiahao Pan, Yongyi Zang, Haohe Liu, Yiming Liang, Wenye Ma, Xingjian Du, and 1 others. 2025b. Yue: Scaling open foundation models for long-form music generation. *arXiv preprint arXiv:2503.08638*.

Ge Yuyao, Cheng Yiting, Wang Jia, Zhou Hanlin, and Chen Lizhe. 2022. Vision transformer based on knowledge distillation in tcm image classification. In *2022 IEEE 5th International Conference on Computer and Communication Engineering Technology (CCET)*, pages 120–125. IEEE.

Hui Zeng, Daming Zhao, Pengfei Yang, Wenxuan Hou, Tianyang Zheng, Hui Li, Weiye Ji, and Jidong Zhai.

2025. Lethe: Layer- and time-adaptive kv cache pruning for reasoning-intensive llm serving.

Ruihong Zeng, Jinyuan Fang, Siwei Liu, and Zaiqiao Meng. 2024. On the structural memory of llm agents. *arXiv.org*.

Chen Zhang, Kuntai Du, Shu Liu, Woosuk Kwon, Xiangxi Mo, Yufeng Wang, Xiaoxuan Liu, Kaichao You, Zhuohan Li, Mingsheng Long, Jidong Zhai, Joseph Gonzalez, and Ion Stoica. 2025a. Jenga: Effective memory management for serving llm with heterogeneity. In *Symposium on Operating Systems Principles*.

Dianxing Zhang, Wendong Li, Kani Song, Jiaye Lu, Gang Li, Liuchun Yang, and Sheng Li. 2025b. Memory in large language models: Mechanisms, evaluation and evolution. *arXiv.org*.

Guangzi Zhang, Lizhe Chen, Yu Zhang, Yan Liu, Yuyao Ge, and Xingquan Cai. 2024a. Translating words to worlds: Zero-shot synthesis of 3d terrain from textual descriptions using large language models. *Applied Sciences*, 14(8):3257.

Gui-Min Zhang, Muxin Fu, and Shuicheng Yan. 2025c. Memgen: Weaving generative latent memory for self-evolving agents. *arXiv.org*.

Guibin Zhang, Haotian Ren, Chong Zhan, Zhenhong Zhou, Junhao Wang, He Zhu, Wangchunshu Zhou, and Shuicheng Yan. 2025d. Memevolve: Meta-evolution of agent memory systems. *Preprint*, arXiv:2512.18746.

Jiaxin Zhang, Yiqi Wang, Xihong Yang, Siwei Wang, Yu Feng, Yu Shi, Ruichao Ren, En Zhu, and Xinwang Liu. 2024b. Test-time training on graphs with large language models (llms). *Proceedings of the 32nd ACM International Conference on Multimedia*.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. 2024c. Long context compression with activation beacon. *Preprint*, arXiv:2401.03462.

Rongzhi Zhang, Kuan Wang, Liyuan Liu, Shuohang Wang, Hao Cheng, Chao Zhang, and Yelong Shen. 2024d. Lorc: Low-rank compression for llms kv cache with a progressive compression strategy. *arXiv.org*.

Yanqi Zhang, Yuwei Hu, Runyuan Zhao, John C. S. Lui, and Haibo Chen. 2024e. Diffkv: Differentiated memory management for large language models with parallel kv compaction. In *Symposium on Operating Systems Principles*.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2024f. A survey on the memory mechanism of large language model-based agents. *ACM Trans. Inf. Syst.*

Zeyu Zhang, Quanyu Dai, Rui Li, Xiaohe Bo, Xu Chen, and Zhenhua Dong. 2025e. Learn to memorize: Optimizing llm-based agents with adaptive memory framework. *arXiv.org*.

Zhenyu (Allen) Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark W. Barrett, Zhangyang Wang, and Beidi Chen. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Neural Information Processing Systems*.

Zhisong Zhang, Yan Wang, Xinting Huang, Tianqing Fang, Hongming Zhang, Chenlong Deng, Shuaiyi Li, and Dong Yu. 2024g. Attention entropy is a key factor: An analysis of parallel context encoding with full-attention-based pre-trained language models. *Annual Meeting of the Association for Computational Linguistics*.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Y. Liu, and Gao Huang. 2023. Expel: Llm agents are experiential learners. In *AAAI Conference on Artificial Intelligence*.

Yu Zhao, Alessio Devoto, Giwon Hong, Xiaotang Du, Aryo Pradipta Gema, Hongru Wang, Kam-Fai Wong, and Pasquale Minervini. 2024a. Steering knowledge selection behaviours in llms via sae-based representation engineering. *North American Chapter of the Association for Computational Linguistics*.

Zheng Zhao, Emilio Monti, Jens Lehmann, and H. Assem. 2024b. Enhancing contextual understanding in large language models through contrastive decoding. *North American Chapter of the Association for Computational Linguistics*.

Jiani Zheng, Lu Wang, Fangkai Yang, Chaoyun Zhang, Lingrui Mei, Wenjie Yin, Qingwei Lin, Dongmei Zhang, Saravan Rajmohan, and Qi Zhang. 2025. Vem: Environment-free exploration for training gui agent with value environment model. *arXiv preprint arXiv:2502.18906*.

Xiaoji Zheng, Lixiu Wu, Zhijie Yan, Yuanrong Tang, Hao Zhao, Chen Zhong, Bokui Chen, and Jiangtao Gong. 2024. Large language models powered context-aware motion prediction in autonomous driving. In *IEEE/RJS International Conference on Intelligent RObots and Systems*.

Wanjun Zhong, Lianghong Guo, Qi-Fei Gao, He Ye, and Yanlin Wang. 2023a. Memorybank: Enhancing large language models with long-term memory. In *AAAI Conference on Artificial Intelligence*.

Wanjun Zhong, Lianghong Guo, Qi-Fei Gao, He Ye, and Yanlin Wang. 2023b. Memorybank: Enhancing large language models with long-term memory. In *AAAI Conference on Artificial Intelligence*.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023. Context-faithful prompting for large language models. In *Conference on Empirical Methods in Natural Language Processing*.

Zihan Zhou, Chong Li, Xinyi Chen, Shuo Wang, Yu Chao, Zhili Li, Haoyu Wang, Qi Shi, Zhixing Tan, Xu Han, Xiaodong Shi, Zhiyuan Liu, and Maosong Sun. 2024. Llm×mapreduce: Simplified long-sequence processing using large language models. *Annual Meeting of the Association for Computational Linguistics*.

Runchuan Zhu, Bowen Jiang, Lingrui Mei, Fangkai Yang, Lu Wang, Haoxiang Gao, Fengshuo Bai, Pu Zhao, Qingwei Lin, Saravan Rajmohan, and 1 others. 2025a. Adaptflow: Adaptive workflow optimization via meta-learning. *EMNLP 2025*.

Tongyao Zhu, Qian Liu, Haonan Wang, Shiqi Chen, Xiangming Gu, Tianyu Pang, and Min-Yen Kan. 2025b. Skyladder: Better and faster pretraining via context window scheduling. *arXiv.org*.

Yuxin Zuo, Kaiyan Zhang, Shang Qu, Li Sheng, Xuekai Zhu, Yuchen Zhang, Biqing Qi, Youbang Sun, Ganqu Cui, Ning Ding, and Bowen Zhou. 2025. Ttrl: Test-time reinforcement learning. *ArXiv*, abs/2504.16084.

A Algorithm

Algorithm 1 presents the complete GDWM procedure. The algorithm takes as input a context sequence X and a total gradient budget K_{total}, and outputs adapted fast weights ϕ. The procedure consists of three phases: (1) **Prefill** the KV cache for the entire context (Line 1); (2) **Estimate** Contextual Utility for each chunk via CPMI (Lines 3–6) and allocate budget accordingly (Lines 7–8); (3) **Consolidate** by performing chunk-restricted gradient updates (Lines 9–13). The total number of gradient steps across all chunks equals $K_{\text{total}} = \sum_{c=1}^{M} k_c$, which corresponds to the "Gradient Steps" reported in our experiments (e.g., GDWM-8 uses $K_{\text{total}} = 8$, GDWM-32 uses $K_{\text{total}} = 32$).

B Notation

We summarize the key notation used throughout this paper and clarify potential ambiguities.

Input and Chunking.

- $X = (x_1, x_2, \ldots, x_L)$: The input context sequence of length L.
- C_c: The c-th chunk, a contiguous subsequence of X.
- $M = \lceil L/S \rceil$: Total number of chunks.
- S: Chunk size (number of tokens per chunk); default $S = 1024$.

Algorithm 1 GDWM: Gated Differentiable Working Memory

```
Require: Context $X = x_{1:L}$, total steps $K_{\text{total}}$, chunk size $S$, local window $n$, temperature $\tau$, min coverage $k_{\min}$
Ensure: Adapted fast weights $\phi$
 1: $\mathbf{K}, \mathbf{V} \leftarrow$ PrefillAndFreeze($X$) {Compute frozen KV cache}
 2: Partition $X$ into $\{C_1, \ldots, C_M\}$ with $M = \lceil L/S \rceil$
 3: for each chunk $c = 1, \ldots, M$ do
 4:     $U(C_c) \leftarrow \frac{1}{|\mathcal{I}_c|} \sum_{t \in \mathcal{I}_c} |\log P_{\text{full}}(x_t) - \log P_{\text{local}}(x_t)|$
 5: end for
 6: Compute weights $w_c$ via Eq. (6)
 7: Allocate $\{k_c\}$ via Coverage-First + Softmax (Eq. 7)
 8: for each chunk $c = 1, \ldots, M$ do
 9:     for $j = 1, \ldots, k_c$ do
10:         $\mathcal{I} \leftarrow$ UniformSample($C_c, B$) {$B$: batch size}
11:         $\mathcal{L} \leftarrow -\frac{1}{B} \sum_{i \in \mathcal{I}} \log P(x_i \mid x_{1:i-1}; \phi, \theta_f)$
12:         $\phi \leftarrow \phi - \eta \nabla_\phi \mathcal{L}$
13:     end for
14: end for
15: return $\phi$
```

Model Parameters.

- θ: Full parameters of the pre-trained LLM.
- θ_f: Frozen parameters, including weights used to compute the KV cache.
- ϕ: Adaptable "fast weights" (e.g., LoRA adapters on W_Q, W_O).
- ϕ_0: Initial state of fast weights (typically zero for LoRA).
- $\phi(\mathbf{k})$: Final state of ϕ after applying the allocation schedule $\mathbf{k}$.

Budget Allocation.

- $\mathbf{k} = (k_1, \ldots, k_M)$: Allocation vector; k_c is the number of gradient updates on chunk C_c.
- K_{total}: Total gradient budget across all chunks.
- $k_{\min}$: Minimum coverage constraint per chunk; default $k_{\min} = 1$.
- τ: Temperature for softmax allocation; default $\tau = 1.0$.

Contextual Utility.

- $U(C_c)$: Contextual Utility of chunk C_c, measuring long-range dependency.
- $\Delta_t = |\log P(x_t \mid x_{1:t-1}) - \log P(x_t \mid x_{t-n:t-1})|$: Surprisal divergence at position t.
- n: Local window size for computing P_{local}; default $n = 512$.

Sampling and Distributions.

- $\mathcal{I}_c$: Set of token positions in chunk C_c.
- $\mathcal{U}(\mathcal{I}_c)$: Uniform distribution over positions in chunk C_c.
- $x_t \sim \mathcal{U}(\mathcal{I}_c)$: Position t sampled uniformly from chunk C_c.
- $\mathcal{I} \subset \mathcal{I}_c$: Minibatch of positions for gradient computation.

KV Cache and Global Context.

- $\mathbf{K}, \mathbf{V}$: Frozen key-value cache for the entire input X, computed via PREFILLANDFREEZE.
- $x_{1:t-1}$: **Full document prefix** from position 1 to $t-1$, *not* chunk-local context.

Key Clarification: Global vs. Local Context. A potential source of confusion is the conditioning context in $P(x_t \mid x_{1:t-1})$. Throughout this paper, $x_{1:t-1}$ refers to the *entire* prefix of the document, accessible via the frozen KV cache $(\mathbf{K}, \mathbf{V})$. The chunking strategy determines *where* gradient updates are computed (i.e., which positions t contribute to the loss), but does *not* restrict the context the model can attend to. This design ensures that the model leverages global information during adaptation while focusing computational budget on high-utility regions.

Sampling Strategy. For each gradient step $j \in \{1, \dots, k_c\}$ on chunk C_c, we sample a minibatch $\mathcal{I}_j$ of B positions *independently* from the uniform distribution $\mathcal{U}(\mathcal{I}_c)$. Formally:

$$\mathcal{I}_j \overset{\text{i.i.d.}}{\sim} \mathcal{U}(\mathcal{I}_c)^B, \quad \forall j \in \{1, \dots, k_c\} \tag{9}$$

This means:

- **Across steps**: Each minibatch $\mathcal{I}_j$ is sampled independently; the same position may appear in multiple steps.
- **Within step**: Positions are sampled i.i.d.; when $B \ll |\mathcal{I}_c|$ (our setting: $B = 32$, $|\mathcal{I}_c| = 1024$), overlap within a single batch is negligible.

This is the standard stochastic gradient descent (SGD) sampling scheme. The independence across steps ensures unbiased gradient estimates, while chunk-restriction (sampling only from $\mathcal{I}_c$, not the full document) eliminates inter-chunk variance as shown in Theorem 1.

C Theoretical Analysis

A key advantage of GDWM over standard TTT methods (which sample uniformly across the full context) is stability. We analyze this through the lens of gradient variance.

C.1 Variance Reduction via Chunk-Restricted Sampling

A natural question arises: why does restricting gradient computation to individual chunks improve optimization? Long documents exhibit *semantic heterogeneity*—different sections address distinct topics and induce gradient directions that may interfere destructively when aggregated. Restricting sampling to a single chunk eliminates this cross-sectional interference.

Theorem 1 (Variance Reduction)**.** *Let g be the gradient estimator for a single update step. Assume the document consists of M chunks, where chunk c has gradient mean μ_c and variance σ_c^2. Under global uniform sampling:*

$$\operatorname{Var}(g_{\text{global}}) = \underbrace{\mathbb{E}_c[\sigma_c^2]}_{\textit{intra-chunk}} + \underbrace{\operatorname{Var}_c(\mu_c)}_{\textit{inter-chunk}} \tag{10}$$

Under chunk-restricted sampling (conditioned on chunk c), we have $\operatorname{Var}(g \mid c) = \sigma_c^2$, eliminating the inter-chunk variance term. Equality holds only when $\mu_c = \bar{\mu}$ for all c (semantically homogeneous document).

Consequence. By scheduling updates per chunk, we eliminate the inter-chunk variance from each gradient estimate. The optimizer follows a more consistent trajectory, avoiding destructive interference between gradient signals from semantically disparate document sections. This provides a principled explanation for the empirical observation that GDWM achieves lower perplexity with fewer optimization steps.

C.2 Role of the Coverage Constraint

For tasks requiring global understanding (e.g., summarization), the coverage constraint ($k_c \geq k_{\min}$) acts as a regularizer against *mode collapse*.

Intuition. Without coverage, greedy CPMI-based allocation may concentrate all budget on a single high-utility region. For example, in a government report, the "Results" section may have the highest CPMI, but a good summary also requires context from "Introduction" and "Methods."

Claim (Informal). If the optimal output requires information from all M sections but the model adapts only to a subset $\mathcal{S} \subsetneq \{1, \dots, M\}$, the coverage gap translates to an $O(|\mathcal{S}|/M)$ recall bound.

The constraint $k_c \geq k_{\min}$ ensures minimum representation of every section, preventing pathological allocation while allowing CPMI to modulate *relative* budget.

D Proofs

D.1 Proof of Theorem 1 (Variance Reduction)

Let $C \in \{1, \dots, M\}$ be a random variable indicating which chunk a position is sampled from. Let g denote the gradient at that position.

By the **Law of Total Variance**:

$$\mathrm{Var}(g) = \mathbb{E}[\mathrm{Var}(g \mid C)] + \mathrm{Var}(\mathbb{E}[g \mid C]) \quad (11)$$

Let $\mu_c = \mathbb{E}[g \mid C = c]$ denote the mean gradient in chunk c, and $\sigma_c^2 = \mathrm{Var}(g \mid C = c)$ denote the within-chunk variance. Let p_c be the probability of sampling from chunk c (under uniform sampling, $p_c = |\mathcal{I}_c|/L$).

Expanding each term:

$$\mathbb{E}[\mathrm{Var}(g \mid C)] = \sum_{c=1}^{M} p_c \sigma_c^2 \quad (12)$$

$$\mathrm{Var}(\mathbb{E}[g \mid C]) = \sum_{c=1}^{M} p_c (\mu_c - \bar{\mu})^2 \quad (13)$$

where $\bar{\mu} = \sum_c p_c \mu_c$ is the global mean.

Therefore:

$$\mathrm{Var}(g_{\text{global}}) = \underbrace{\sum_{c=1}^{M} p_c \sigma_c^2}_{\text{intra-chunk variance}} + \underbrace{\sum_{c=1}^{M} p_c (\mu_c - \bar{\mu})^2}_{\text{inter-chunk variance}} \quad (14)$$

Under **chunk-restricted sampling**, when we sample from a specific chunk c, the gradient variance is exactly σ_c^2 (the intra-chunk variance of that chunk). Since the inter-chunk term $\sum_c p_c (\mu_c - \bar{\mu})^2 \geq 0$, we have:

$$\mathrm{Var}(g \mid C = c) = \sigma_c^2 \leq \mathrm{Var}(g_{\text{global}}) \quad (15)$$

Equality holds if and only if the inter-chunk variance is zero, i.e., $\mu_c = \bar{\mu}$ for all c, meaning the document is semantically homogeneous (all chunks have identical gradient expectations).

□

E Coverage Constraint Analysis

The constraint $k_c \geq k_{\min}$ in Eq. (3) prevents the model from overfitting to a single high-utility region. This is crucial for tasks requiring global understanding.

Example: GovReport Summarization. A typical government report contains Introduction, Methods, Results, and Conclusion sections. Without the coverage constraint, CPMI-based allocation may concentrate all budget on Results (often the highest information density region), missing essential context from other sections that the summary must include.

Failure Mode. If the optimal summary requires information from all M sections but the model adapts only to a subset $\mathcal{S} \subsetneq \{1, \dots, M\}$, the coverage gap directly translates to recall loss proportional to $|\mathcal{S}|/M$.

The constraint ensures that every section receives at least minimal gradient exposure, preventing pathological allocation while still allowing CPMI to modulate the *relative* budget across sections.

F Justification for Absolute Value in Δ_t

The absolute value in $\Delta_t = |\log P(x_t \mid x_{1:t-1}) - \log P(x_t \mid x_{t-n:t-1})|$ captures both positive and negative divergence.

Information-Theoretic Perspective (Pointwise KL). Mathematically, the quantity Δ_t corresponds to the magnitude of the *Pointwise Kullback-Leibler Divergence* contribution at token x_t. The standard KL divergence $D_{\text{KL}}(P_{\text{full}} \| P_{\text{local}})$ is the expectation of the log-likelihood ratio. Our utility metric $U(C_c)$ essentially estimates the *L1 norm* of this pointwise divergence over the chunk. We prefer the absolute value (L1-like) over the raw difference (which would sum to the standard KL) because gradient updates are driven by the *magnitude* of the error signal.

- **Positive divergence** ($P_{\text{full}} > P_{\text{local}}$): Long-range context reduces surprisal (adds information).
- **Negative divergence** ($P_{\text{full}} < P_{\text{local}}$): Long-range context increases surprisal (introduces conflict).

Both cases represent significant deviations between the global and local models, identifying regions where the fast weights ϕ must adapt to reconcile these discrepancies.

Empirical Validation. Using $\max(0, \Delta_t)$ (ignoring negative divergence) instead of $|\Delta_t|$ results in 3-5% performance degradation across tasks. The "hard examples" with negative divergence are precisely where the model needs recalibration.

G Hyperparameter Sensitivity

We briefly analyze the sensitivity of GDWM to its two key hyperparameters: the temperature τ and the minimum coverage $k_{\min}$.

Temperature τ. The temperature controls the sharpness of the softmax allocation (Eq. 6).

- $\tau \to 0$ **(Greedy):** The budget concentrates entirely on the single chunk with the highest utility. This risks overfitting to one region and failing on multi-hop tasks.
- $\tau \to \infty$ **(Uniform):** The policy degenerates into uniform sampling (equivalent to qTTT), losing the efficiency benefits of selection.
- $\tau \approx 1.0$ **(Balanced):** Empirically, values in $[0.5, 1.5]$ perform robustly. We adopt $\tau = 1.0$ as a tuning-free default.

Minimum Coverage $k_{\min}$. We set $k_{\min} = 1$ to ensure no chunk is entirely starved of gradients. Increasing $k_{\min}$ reduces the budget available for utility-based redistribution, pushing the behavior closer to uniform sampling. $k_{\min} = 1$ represents the minimal constraint necessary to prevent mode collapse while maximizing the freedom of the allocation policy.

H Testable Predictions from Theorem 1

Theorem 1 yields three empirically verifiable predictions:

1. **Gradient Norm Variance**: The sequence $\{\|g_t\|\}$ under **GDWM** should exhibit lower variance than under global uniform sampling.
2. **Within-Chunk Cosine Similarity**: The cosine similarity $\cos(g_i, g_j)$ for positions i, j within the same chunk should be higher than for positions in different chunks.
3. **Loss Curve Monotonicity**: The training loss curve should converge more monotonically with fewer oscillations under chunk-restricted sampling.

These predictions are consistent with our empirical observations and provide a testable framework for validating the theoretical analysis.

I Largest Remainder Allocation Method

After the initial floor allocation in Eq. (7), there may be residual steps due to rounding. We distribute these using the *Largest Remainder Method*:

1. Compute fractional remainders: $r_c = K_{\text{rem}} \cdot w_c - \lfloor K_{\text{rem}} \cdot w_c \rfloor$
2. Sort chunks by r_c in descending order
3. Assign one additional step to each of the top-R chunks, where $R = K_{\text{total}} - \sum_c \lfloor K_{\text{rem}} \cdot w_c \rfloor - M \cdot k_{\min}$

This guarantees that the total allocation exactly equals K_{total} while respecting the utility-based priority ordering.

J Evidence Span Analysis

We formalize the relationship between chunk size and task performance through the concept of *evidence span*.

Definition 2 (Evidence Span). *For a task $\mathcal{T}$ with query q and context X, the evidence span $E_{\mathcal{T}}$ is the minimum contiguous token range required to contain all information necessary for correct response generation:*

$$E_{\mathcal{T}} = \min_{[i,j]} \{ j - i : X_{[i,j]} \textit{ suffices for } \mathcal{T} \} \quad (16)$$

Evidence Fragmentation Problem. When chunk size $S < E_{\mathcal{T}}$, the evidence is partitioned across multiple chunks. Let the evidence span $[a, b]$ be split into $k = \lceil (b - a)/S \rceil$ chunks. For each fragment C_i, the CPMI estimate becomes:

$$\hat{U}(C_i) = \text{CPMI}(C_i) < \text{CPMI}([a, b]) = U_{\text{true}} \quad (17)$$

This underestimation occurs because each fragment, viewed in isolation, appears to have low contextual dependency—the long-range signal is diluted across fragments.

Optimal Chunk Size Selection. The optimal chunk size satisfies:

$$S^* = \min\{S : S \geq E_{\mathcal{T}},\ \forall \mathcal{T} \in \text{target tasks}\} \tag{18}$$

For a diverse benchmark like ZeroSCROLLS containing both local and multi-hop tasks, $S = 1024$ emerges as the robust choice: it satisfies $S \geq E_{\mathcal{T}}$ for most multi-hop instances while maintaining reasonable granularity for local tasks.

Empirical Validation. Table 4 in the main text validates this analysis:

- At $S = 512$: MuSiQue collapses to 10.0 (evidence fragmentation)
- At $S = 1024$: MuSiQue recovers to 25.8 ($S \geq E_{\mathcal{T}}$)
- At $S = 2048$: Marginal improvement (25.3) with reduced local precision

This provides a principled explanation for why $S = 1024$ achieves the best cross-task robustness: it is the minimum chunk size that avoids catastrophic evidence fragmentation across the task distribution. For tasks with intermediate evidence spans like Qasper, performance naturally peaks at the corresponding granularity ($S = 512$), confirming that $S \approx E_{\mathcal{T}}$ is the theoretical optimum.

J.1 Information-Theoretic Lower Bound on Fragmented Utility

The empirical observation that fragmented evidence leads to utility underestimation admits a formal information-theoretic explanation. We show that chunking an evidence span introduces a *systematic negative bias* in utility estimation due to the loss of *synergistic information*.

Definition 3 (Contextual Utility as Mutual Information)**.** *Let G denote the global context (long-range prefix) and L denote the local context (sliding window). For a token x_t, the Contextual Utility can be interpreted as the absolute mutual information gain:*

$$\begin{aligned} U(x_t) &= |I(x_t; G) - I(x_t; L)| \\ &\approx |I(x_t; G \setminus L \mid L)| \end{aligned} \tag{19}$$

where $G \setminus L$ represents the distant context beyond the local window. For a chunk C, the aggregate utility is $U(C) = \sum_{x_t \in C} U(x_t)$.

Proposition 1 (Utility Underestimation Under Fragmentation)**.** *Let $E = C_1 \cup C_2$ be an evidence span partitioned into two adjacent chunks. Then the sum of individual chunk utilities is bounded above by the utility of the unified span:*

$$U(C_1) + U(C_2) \leq U(E) + \epsilon \tag{20}$$

where $\epsilon \leq 0$ when the chunks exhibit information synergy *(i.e., positive interaction information). Equality holds if and only if C_1 and C_2 are informationally independent given the global context.*

Proof. We leverage the chain rule of mutual information. Let G denote the global context. The joint utility of the unified evidence span $E = C_1 \cup C_2$ satisfies:

$$I(E; G) = I(C_1; G) + I(C_2; G \mid C_1) \tag{21}$$

The fragmented utility estimation treats C_1 and C_2 independently:

$$\hat{U}(E) = I(C_1; G) + I(C_2; G) \tag{22}$$

The *fragmentation gap* is therefore:

$$\begin{aligned} \Delta_{\text{frag}} &= I(E; G) - \hat{U}(E) \\ &= I(C2; G \mid C_1) - I(C2; G) \\ &= -I(C_1; C_2; G) \end{aligned} \tag{23}$$

where $I(C_1; C_2; G)$ is the *interaction information* (also known as co-information or multivariate mutual information), defined as:

$$I(C_1; C_2; G) = I(C_2; G) - I(C_2; G \mid C_1) \tag{24}$$

Interpretation. The interaction information $I(C_1; C_2; G)$ measures the degree to which C_1 and C_2 *synergistically* inform G:

- **Positive interaction** ($I > 0$): C_1 and C_2 are *redundant*—knowing one reduces the information gain from the other. Fragmentation causes *overestimation* (rare).
- **Negative interaction** ($I < 0$): C_1 and C_2 are *synergistic*—together they provide more than the sum of parts. This is characteristic of **reasoning chains**. Fragmentation causes *underestimation*.

For multi-hop reasoning tasks like MuSiQue, where the answer requires synthesizing facts from multiple locations, the evidence exhibits strong negative interaction information. Thus:

$$I(C_1; C_2; G) < 0 \implies \Delta_{\text{frag}} > 0 \\ \implies U(E) > \hat{U}(E) \tag{25}$$

This proves that fragmented utility estimation is a *systematic lower bound* on the true utility when evidence is synergistic. □

Consequence for Chunk Size Selection. Proposition 1 rigorously justifies the catastrophic performance collapse at small chunk sizes on multi-hop tasks. The fragmentation gap Δ_{frag} is not merely noise but a *structured bias* scaling with inter-chunk synergy. The optimal S^* must satisfy $S \geq E_{\mathcal{T}}$ to ensure synergistic evidence is not partitioned.

Connection to Cognitive Science. This parallels the *binding problem* in cognitive neuroscience: a reasoning chain stored in working memory must be represented as a unified chunk to preserve logical coherence. Fragmenting it destroys emergent meaning, analogous to how fragmenting a sentence into words loses compositional semantics.

K Theoretical Justification for CPMI vs. Surprisal

A natural alternative to our CPMI-based utility is a simpler, non-uniform gating policy based solely on *Surprisal* (or Perplexity), i.e., prioritizing chunks with high $\mathcal{L}_{\text{local}}(x) = -\log P(x \mid x_{\text{local}})$. While intuitively appealing (allocating compute to "hard" regions), this approach is mathematically suboptimal for long-context adaptation.

Proposition. Surprisal measures *difficulty*, whereas CPMI measures *dependency*.

Let the information content of a token x_t be decomposed as:

$$I(x_t \mid x_{\text{global}}) = I(x_t \mid x_{\text{local}}) \\ - \underbrace{I(x_t; x_{\text{distant}} \mid x_{\text{local}})}_{\text{CPMI}} \tag{26}$$

High surprisal ($I(x_t \mid x_{\text{local}})$ is large) can arise from two sources:

1. **Aleatoric Uncertainty:** The token is inherently unpredictable (e.g., a random name or number), regardless of context.

2. **Epistemic Uncertainty (Contextual):** The token is predictable given long-range context but unpredictable locally.

Gradient adaptation on Type 1 tokens is wasteful—it forces the model to memorize noise. Adaptation on Type 2 tokens is high-value—it retrieves recoverable information. CPMI ($|\log P_{\text{full}} - \log P_{\text{local}}|$) specifically isolates Type 2 uncertainty by cancelling out the intrinsic difficulty. A surprisal-based baseline would confuse noise with signal, allocating budget to intrinsically hard but context-irrelevant tokens. Thus, CPMI is the theoretically correct objective for *context-dependent* memory consolidation.

L Detailed Related Work

Context Engineering Context engineering optimizes information payloads for large language models, spanning techniques from foundational prompt design to advanced management strategies (Mei et al., 2025b,c; Huang, 2025; Hua et al., 2025; Sahoo et al., 2024; Haider et al., 2024; Pang et al., 2025). Core methodologies include in-context learning and chain-of-thought prompting for adaptive reasoning (Weng, 2024; Allingham et al., 2023; Brown et al., 2020a; Chen et al., 2025f; Wei et al., 2022; Ohalete et al., 2025; Ge et al., 2025c), with recent extensions to multimodal agentic in-context learning (Fu et al., 2025a). Recent approaches focus on retrieval-augmented generation (RAG) and automated prompt optimization (Lewis et al., 2020; Nogueira and Cho, 2019; Shin et al., 2020; Li et al., 2025a; Karpukhin et al., 2020; Chen et al., 2025c), with advanced retrieval systems incorporating utility-based training through shared context attribution (Xu et al., 2025d) and fine-grained citation generation evaluation (Xu et al., 2024). Efficient context management techniques address quadratic scaling limitations through window extension, compression, and hierarchical processing (Li et al., 2023; Vaswani et al., 2017; Mao et al., 2024; Fu et al., 2024b; Duan et al., 2025a; Zhu et al., 2025b; Tan et al., 2024; Wang et al., 2024b; Song et al., 2024; Zhou et al., 2024; Hou et al., 2024a; Ratner et al., 2022; Zhang et al., 2024g; Sun et al., 2025; Wu et al., 2025e; Wang et al., 2024a; Liu et al., 2025b). Parallel efforts tackle robustness and knowledge conflicts via context-aware decoding, representation engineering, and activation alignment (He et al., 2024b,a; Govindan et al., 2025; Park et al., 2025a; Wang et al., 2025b; Zhao et al., 2024b; Longpre et al., 2021; Khandelwal et al., 2025; Shi et al., 2023; Zhou et al., 2023; Zhao et al., 2024a; Shen et al., 2025; Porretta et al., 2025; Katrix et al., 2025; Juki'c et al., 2025; Houlsby et al., 2019; Park et al., 2025b; Bi et al., 2024a,c, 2025d; Ge et al., 2023). Recent work also reveals that LLMs exhibit systematic biases, such as over-relying on surface-level naming patterns when identifying drug ingredients (Li et al., 2025b), and evaluates the impact of watermarking on visual language models in document understanding tasks (Xu et al., 2025a). Safety-critical research addresses fine-grained safe generation via specialized representation routers (Mei et al., 2024c), distinguishes misalignment from maliciousness in jailbreak scenarios (Mei et al., 2024b), investigates jailbreak attacks through implicit references (Wu et al., 2024), adversarial metaphors (Yan et al., 2025a), and prompt template stealing vulnerabilities in text-to-image models (Wu et al., 2025d), while also rethinking jailbreak evaluation to investigate real misuse threats (Yan et al., 2025b) and exploring global subspace approaches for LLM detoxification (Duan et al., 2026b). Additionally, structured output approaches improve question reformulation and cross-lingual summarization through meta-generation (Li et al., 2024c,b). Research on LLM reasoning capabilities includes new concept comprehension through slang understanding (Mei et al., 2024a), graph descriptive order effects on graph problem solving (Ge et al., 2024), and scalable link prediction with LLMs (Bi et al., 2024b). Domain-specific applications demonstrate these methods in tabular analysis, translation, clinical NLP, and traffic prediction (Hollmann et al., 2023; Wu and Hu, 2023; Sivarajkumar et al., 2024; Zheng et al., 2024; Yang et al., 2023), often utilizing filtering mechanisms to enhance retrieval precision (Shi et al., 2025; Cheng et al., 2024; Chakraborty et al., 2025), with emerging work on learning to refine pre-training data at scale (Bi et al., 2025b).

Memory Management Memory management in large language models has evolved to address the critical bottlenecks imposed by model weights and, increasingly, the ephemeral activation memory required for inference. While parametric memory stores implicit knowledge in model weights (Li et al., 2025g; Zhang et al., 2025b; Hsieh et al., 2023), the key-value (KV) cache has emerged as the dominant constraint, often consuming significantly more memory than parameters and scaling linearly with sequence length (Liu et al., 2023b; Zhang et al., 2024e; Kwon et al., 2023; Dai et al., 2024; Cai et al., 2025; Roy et al., 2025; Shutova et al., 2025; Barua, 2024; Modarressi et al., 2023; Li et al., 2025h; Cheng et al., 2025; Brown et al., 2020b; Gao et al., 2018; Dao et al., 2022). To mitigate these limitations, research has introduced virtual memory and paging techniques akin to operating systems (Kwon et al., 2023; Xue et al., 2024; Koilia and Kachris, 2024; Zhang et al., 2025a; Chitty-Venkata et al., 2025), as well as advanced compression and quantization methods to reduce footprints without quality loss (Xie et al., 2025; Srinivas and Runkana, 2025; Zhang et al., 2024d). Dynamic eviction strategies, including heavy-hitter identification and attention-based pruning, selectively discard less critical tokens to maintain fixed budgets (Yuan et al., 2025a; Zhang et al., 2023; Zeng et al., 2025; Shinwari and Usama, 2025; Yi et al., 2024). Parallelization and offloading frameworks further optimize resource utilization across devices (Yang et al., 2024; Ren et al., 2021; Sheng et al., 2023; Wu and Tu, 2024; Banasik, 2025), while hierarchical and biologically inspired architectures offer structured approaches to long-term memory and personalization (Fang et al., 2024; Rezazadeh et al., 2024; Kang et al., 2025a; Hou et al., 2024b; Li et al., 2024a; Metinov et al., 2025; Zhong et al., 2023a; Huang et al., 2024). Complementary work on knowledge editing addresses how to efficiently update stored knowledge in model parameters, with methods ranging from adaptive token biasing to structured output acceleration (Bi et al., 2024e,d, 2025a), while recent studies reveal that related knowledge perturbation matters when editing multiple pieces of knowledge for the same subject (Duan et al., 2025b).

Agent Memory Research on agent memory in large language models addresses the constraints of limited context windows and static training data by enabling persistent information storage and retrieval (Hu et al., 2026). Existing approaches categorize memory into distinct types, differentiating between short-term working memory and long-term storage (Han et al., 2024; Wang and Chen, 2025; Sridhar et al., 2023), as well as distinguishing parametric memory in model weights from explicit contextual memory (Du et al., 2025a; Shan et al., 2025). More granular taxonomies identify specialized forms such as episodic, consensus, semantic, and procedural memory (Han et al., 2024; Terranova et al., 2025). Structurally, methods range from knowledge-organization and retrieval-oriented mechanisms to architecture-driven hierarchies (Kang et al., 2025b; Zhong et al., 2023b). While early work relied heavily on Retrieval Augmented Generation (RAG) and static vector databases (Lewis et al., 2020; Xu et al., 2025c; Vishwakarma et al., 2025), recent advancements explore graph-based systems (Anokhin et al., 2024; Vishwakarma et al., 2025) and diverse formats like natural language or embeddings (Wang et al., 2023; Shinn et al., 2023). Contemporary frameworks implement core operations—consolidation, updating, indexing, and forgetting—to manage memory dynamics, often incorporating biologically inspired mechanisms like forgetting curves (Xiong et al., 2025e; Park et al., 2023; Zhao et al., 2023; Cao et al., 2025). Despite these advances, challenges remain in autonomous memory management, with many systems relying on manual predefinitions (Zhang et al., 2025e; Wang et al., 2025c; Zhang et al., 2024f) or suffering from structural rigidity and catastrophic forgetting (Xu et al., 2025c; Zhang et al., 2025c; Guo et al., 2023), motivating meta-evolutionary approaches to dynamically optimize agent memory systems (Zhang et al., 2025d). Applications span gaming, dialogue, and procedural tasks (Hu et al., 2024b; Zeng et al., 2024; Hu et al., 2024a; Fang et al., 2025; Mohammed, 2025), with emerging work on environment-free exploration for GUI agents (Zheng et al., 2025), adaptive workflow optimization via meta-learning (Zhu et al., 2025a), cross-modal knowledge graphs for cost-effective game agents (Fu et al., 2025b), reality-aligned evaluation for agentic search (Xu et al., 2025e), coarse-to-fine frameworks for robust GUI grounding (Li et al., 2025f), and scaling issue-resolving capabilities across repositories (Wang et al., 2025a). Future research directions emphasize unified theoretical frameworks (Wu et al., 2025c), episodic memory for single-shot learning (Pink et al., 2025), and scalable, autonomous systems capable of self-evolution (Salama et al., 2025; Zhang et al., 2024f; Hu et al., 2025b; Huang et al., 2025).

Test-Time Adaptation. A growing line of work treats deployment as an optimization phase, improving robustness and adaptation by updating model states at inference time under shift or novelty, often via self-supervised objectives and safeguards against forgetting or noisy updates (Niu et al., 2022; Tang et al., 2023; Park et al., 2024; Zhang et al., 2024b). Within LLMs, inference-time optimization spans gradient-based test-time training on task instances or auxiliary data, including adaptation driven by retrieved neighbors, contextual streams, active/verification-guided sample selection, selective test-time learning for evaluation models, efficient rubric-based generative verification for search-augmented systems, and sample-specific language model optimization (Hardt and Sun, 2024; Muhtar et al., 2024; Hübotter et al., 2024; Hübotter et al., 2025; Moradi et al., 2025; Jwa et al., 2025; Akyürek et al., 2025; Ma et al., 2025; Hu et al., 2025a). In parallel, test-time compute can be optimized at the policy level—allocating computation across candidate solutions or update actions—through principled test-time scaling, meta-learned compute control, reinforcement learning at inference, or environment-augmented generation that achieves steep scaling laws (Snell et al., 2024; Qu et al., 2025; Zuo et al., 2025; Mei et al., 2025a). Recent work on reward and rubric-based guidance promotes exploration to improve multi-domain reasoning (Bi et al., 2025c). Recent theoretical work has also drawn connections between LLMs and Solomonoff induction, providing foundations for understanding test-time computation (Wan and Mei, 2025), investigates latent reasoning in LLMs as vocabulary-space superposition (Deng et al., 2025), and analyzes circular reasoning patterns that cause self-reinforcing loops in large reasoning models (Duan et al., 2026a). Systematic surveys categorize reasoning LLMs from System 1 to System 2 paradigms (Li et al., 2025i). Graph-based analysis provides insights into reasoning patterns of LLMs (Xiong et al., 2025b), divide-and-conquer strategies enhance character-level manipulation (Xiong et al., 2025a), and pattern mining approaches mitigate LLM overthinking via early reasoning exit (Wei et al., 2025). Audio chain-of-thought reasoning extends these paradigms to multimodal settings (Xiong et al., 2025d). Complementary inference-time strategies reduce effective context cost by compressing or distilling long inputs and intermediate representations, ranging from selective augmentation/compression in retrieval pipelines to learned and training-free prompt compression, dynamic allocation of soft tokens, activation-based beacons, near-lossless KV compression, and importance sampling-based prompt compression (Xu et al., 2023; Chevalier et al., 2023; Jiang et al., 2024; Pan et al., 2024; Fei et al., 2025; Chen et al., 2025e; Zhang et al., 2024c; Chari et al., 2025; Chen et al., 2025d).

Extended Applications and Broader Techniques The core principles underlying context engineering and test-time adaptation extend naturally to diverse application domains. Retrieval-augmented systems face challenges including hidden biases that can undermine performance across modalities (Yao et al., 2025c). Research on model robustness reveals vulnerabilities to various input manipulations such as adversarial text arrangements (Li et al., 2024d) and disambiguation challenges in recognition tasks (Li et al., 2025d), with studies examining how models handle semantic understanding requiring global reasoning (Li et al., 2025c). Attention-based refinement mechanisms have been applied to structured output generation such as LaTeX conversion (Li et al., 2025e), controllable generation via diffusion models (Xiong et al., 2025c), and contrastive attention for enhanced reasoning (Ge et al., 2025b). Interface understanding benefits from test-time scaling approaches (Wu et al., 2025b) and structured attention mechanisms for document processing (Liu et al., 2025a). Temporal reasoning advances through frame-interleaved approaches via reinforcement learning (Ge et al., 2025a), while domain-specific understanding is enhanced through foundational skill evaluation (Wu et al., 2025a). Temporal bias analysis reveals unique challenges in sequential data processing (Yao et al., 2025a). Dynamic scene modeling benefits from semantic-guided control (Chen et al., 2025a) and hierarchical flow-guided representations (Chen et al., 2025b). Efficient rendering techniques include frequency-importance methods for real-time processing (Chen et al., 2024), non-photorealistic rendering for stylized outputs (Hu et al., 2024c), and zero-shot synthesis from textual descriptions (Zhang et al., 2024a). Knowledge distillation enables efficient model deployment for domain-specific tasks (Yuyao et al., 2022). Signal processing techniques bridge different data modalities through reconstruction (Fu et al., 2025c), while generative priors enable quality assessment (Fu et al., 2024a). Long-form content generation extends to creative domains with scaled foundation models (Yuan et al., 2025b), and emerging surveys explore new programming paradigms with LLMs (Ge et al., 2025d).

M Datasets

We evaluate GDWM on two complementary long-context benchmarks: **ZeroSCROLLS**—a *zero-shot* suite adapted from SCROLLS with reliable, task-specific automatic metrics—and **LongBench v2**—a *realistic* long-context benchmark using multiple-choice questions for robust evaluation.

Task	Type	Metric	Avg #Words
GovReport	Summarization	ROUGE	7,273
QMSum	QB-Summ	ROUGE	10,839
Qasper	QA	F1	3,531
NarrativeQA	QA	F1	49,384
QuALITY (Quality)	MC-QA	Accuracy	4,248
MuSiQue	QA (Multi-hop)	F1	1,749

Table 5: ZeroSCROLLS tasks used in this work and their official metrics/statistics (Shaham et al., 2023).

M.1 ZeroSCROLLS

Benchmark overview. ZeroSCROLLS is a zero-shot benchmark for long-text understanding that contains *no training split* and only small validation sets, with each task capped at 500 examples to keep evaluation affordable. It extends SCROLLS by adapting six long-document tasks and adding four new tasks, covering summarization, question answering, and information aggregation.

Tasks used in this work. Following prior long-context evaluation practice, we select six representative ZeroSCROLLS tasks spanning both *sparse-evidence* and *dense-coverage* regimes: (i) **GovReport** and **QMSum** for (query-based) summarization, (ii) **Qasper** and **NarrativeQA** for long-context QA, (iii) **QuALITY** (denoted as **Quality** in our paper) for multiple-choice comprehension, and (iv) **MuSiQue** for multi-hop QA.

Evaluation metrics. ZeroSCROLLS uses task-aligned automatic metrics: ROUGE for summarization, F1 for extractive/abstractive QA-style tasks, and Accuracy for multiple-choice QA. We report the official metrics with the benchmark's evaluation scripts.

M.2 LongBench v2

Benchmark overview. LongBench v2 is a realistic long-context benchmark designed to test *deep understanding and reasoning* over long inputs. It contains **503** challenging **multiple-choice** questions with contexts ranging from **8k to 2M words**, organized into **6** major categories and **20** subtasks. All examples are in English, and each instance includes a long context, a question, four options, a ground-truth answer, and annotated evidence for verification. The benchmark emphasizes evaluation

Category	#Questions	Median #Words
Single-Document QA	175	51k
Multi-Document QA	125	34k
Long In-context Learning	81	71k
Long-dialogue History Understanding	39	25k
Code Repository Understanding	50	167k
Long Structured Data Understanding	33	49k

Table 6: LongBench v2 category-level statistics reported in Bai et al. (2025).

reliability by using accuracy-based scoring (rather than free-form generation metrics).

Task categories and statistics. Table 6 summarizes the six categories and their median context lengths. Notably, LongBench v2 includes long-dialogue history understanding and code-repository understanding, which directly stress *memory* and *retrieval under long context*.

Evaluation protocol. We follow the official LongBench v2 evaluation protocol and report **accuracy** (overall and category-wise). For consistency across settings, all methods are evaluated under the same context-length budget of the underlying base model; when an instance exceeds the model limit, we follow the benchmark-recommended preprocessing/truncation behavior (Bai et al., 2025).