DP-MIND: DIFFERENTIALLY PRIVATE MEMORY-GUIDED LONG-CONTEXT INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM-as-a-Service has become a dominant paradigm for accessing large language models without requiring local deployment. However, the process raises severe privacy concerns when user queries may include sensitive or confidential contexts. Differential privacy (DP) provides a principled approach to protecting such inputs, while existing DP inference methods are primarily designed for short- or moderate-context settings and struggle with million-token-scale inputs due to limited context capacity and severe utility degradation under DP perturbation.

In this paper, we propose DP-MIND, a memory-guided framework that scales differentially private inference to million-token contexts. DP-MIND enables private long-context inference by locally perturbing the input and processing it chunk by chunk with a compact memory that preserves task-relevant information across the sequence. To make this memory robust to DP noise, we optimize the memory construction policy directly on perturbed inputs using reinforcement learning, guided by both final answer correctness and memory consistency. We further recover perturbation-corrupted answer spans through a local denoising module without exposing the original context to the external model. We provide theoretical privacy analysis of DP-MIND and further conduct extensive empirical analysis across multiple long-context benchmarks with context lengths up to the million-token scale, demonstrating that DP-MIND consistently outperforms existing DP and long-context inference baselines in both scalability and utility.

1 INTRODUCTION

Large language models (LLMs) have rapidly become a general-purpose interface for knowledge-intensive tasks, including question answering, document analysis, code generation, personal assistance, and decision support (Zhao et al., 2026; Song et al., 2025). Rather than deploying these models locally, users increasingly access them through "LLM-as-a-Service" platforms (Brown et al., 2020; Ouyang et al., 2022; Chowdhery et al., 2022; Qwen Team, 2025), where prompts and contextual documents are submitted to remote model providers for inference. This service-oriented paradigm makes advanced LLMs broadly accessible, but it also raises a fundamental privacy concern: users often need to transmit sensitive context to remote model providers. Such context may contain private conversations, medical or financial records, source code, legal documents, or other confidential information (Chen et al., 2025; Bai et al., 2025).

Differential privacy (DP) provides a principled approach to protecting sensitive user contexts during remote LLM inference. Existing privacy-preserving inference methods primarily focus on improving the privacy–utility trade-off. DP-Forward perturbs intermediate representations (Du et al., 2023), while ObfusLM combines embedding obfuscation and shuffling against inversion attacks (Lin et al., 2025). Recent methods explore batch-level request sharing (Mai et al., 2026) and token-level DP that limits the influence of sensitive context tokens (Thareja et al., 2026). To recover utility after privacy protection, Split-and-Denoise and InferDPT employ denoising and response extraction, respectively (Mai et al., 2024; Tong et al., 2025). DP^3 further extends the investigation to multi-turn inference, addressing privacy-budget consumption and semantic consistency (Zheng et al., 2026).

Although prior methods have made substantial efforts to improve the "privacy–utility" trade-off, they are primarily designed for standard short- or moderate-context inference and do not address the key challenges posed by million-token-scale inputs: context lengths that exceed the model's

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

effective processing capacity, severe utility degradation caused by DP noise, and substantial inference overhead. Even when the input fits within the context window, LLMs may struggle to reliably retrieve and utilize relevant information from long contexts (Liu et al., 2024).

Driven by the limitations of standard transformer inference, recent non-private long context methods have explored several directions, including **context-window extension**, **sparse or linear attention** and **memory-augmented** inference. Context-window extension methods enlarge the usable context length through positional adaptation (Chen et al., 2023b; Peng et al., 2024; Ding et al., 2024; Yang et al., 2025), while sparse attention and memory-based methods improve long-context scalability through efficient attention patterns and compact memory states (Beltagy et al., 2020; Zaheer et al., 2020; Yu et al., 2026). These techniques demonstrate the feasibility of scaling non-private long-context inference and provide potential directions for DP long-context inference.

In this paper, we strive to bridge this gap by studying differentially private long-context inference through the lens of memory-augmented inference that are previously under-explored in DP-literature. We propose DP-MIND, a memory-guided framework for differentially private long-context inference. To address the challenges of limited context capacity, DP-MIND partitions the long context into locally perturbed chunks and processes them sequentially, while a compact memory propagates task-relevant information across chunks without exposing the clean context. The memory policy is further optimized directly under DP perturbation using reinforcement learning, guided jointly by final answer correctness and memory consistency with the corresponding clean reference memory. In addition, DP-MIND employs a trusted local denoising module that leverages the alignment between clean and perturbed contexts to restore corrupted answer spans, improving answer utility without exposing the clean context to the external LLM. Our main contributions are summarized as follows:

- To the best of our knowledge, we are the first to systematically study differentially private inference in million-token-scale long-context settings, where existing DP inference methods face fundamental challenges in scalability, utility, and efficiency.
- We propose DP-MIND, a memory-guided framework that combines chunk-wise perturbation with sequential memory propagation for DP long-context inference, and provide a comprehensive theoretical analysis of the privacy guarantee.
- We design a DP-aware reinforcement learning objective for memory construction, with a reward that jointly captures final answer correctness and memory consistency under DP perturbation. We further introduce a trusted local denoising module that restores corrupted answer spans while keeping the clean context entirely local.
- We conduct extensive experiments across multiple long-context benchmarks, privacy budgets, and context lengths up to the million-token scale. The results demonstrate that DP-MIND substantially outperforms existing DP and long-context inference baselines under DP perturbation.

2 Background and Related Work

2.1 Differentially Private Inference

Existing text privacy methods primarily protect sensitive inputs by perturbing tokens or intermediate representations. Early approaches inject calibrated noise into token embeddings and project the perturbed representations back to discrete tokens (Feyisetan et al., 2020; Xu et al., 2020). SanText and CusText further improve private text perturbation through natural token replacement and customized output sets, respectively (Yue et al., 2021; Chen et al., 2023a). For LLM inference, DP-Forward perturbs input representations during the forward pass (Du et al., 2023), while ObfusLM combines embedding obfuscation and shuffling to defend against embedding inversion attacks (Lin et al., 2025). Recent approaches further explore model-agnostic prompt protection through batch-level request sharing (Mai et al., 2026) and token-level differential privacy by bounding the influence of sensitive context tokens (Thareja et al., 2026). To improve inference utility, Split-and-Denoise and InferDPT introduce denoising and response extraction strategies, respectively (Mai et al., 2024; Tong et al., 2025), while DP^3 addresses privacy-budget consumption and semantic consistency in multi-turn inference (Zheng et al., 2026). Despite these advances, existing approaches do not systematically address the scalability and utility challenges of DP inference over million-token contexts.

2.2 LONG-CONTEXT INFERENCE

Existing approaches to long-context inference can be broadly categorized into three directions below. **Context window extension** methods enlarge the usable context length of pretrained LLMs through positional interpolation or rescaling, such as Position Interpolation, YaRN, and LongRoPE (Chen et al., 2023b; Peng et al., 2024; Ding et al., 2024). **Sparse attention** methods reduce the quadratic cost of full attention by using local, global, or random attention patterns, as in Longformer and BigBird (Beltagy et al., 2020; Zaheer et al., 2020). **Memory-augmented** inference offers another approach by processing long contexts sequentially while maintaining compact representations of previously observed information. MemAgent employs recurrent memory updates and reinforcement learning for long-context reasoning (Yu et al., 2026), while LycheeMemory combines chunk-wise compression, selective memory retrieval, and end-to-end reinforcement learning (Chen et al., 2026). Despite their advances in scalability, existing long-context methods are primarily designed for clean inputs and do not explicitly account for DP-induced information loss. Under DP perturbation, simply extending the context window or reducing attention complexity cannot address corrupted evidence, while compression and memory construction may further propagate errors from noisy inputs. These limitations highlight the need for scalable DP inference that can robustly preserve and propagate task-relevant information across long contexts.

3 PRELIMINARY

3.1 DIFFERENTIAL PRIVACY

Differential privacy (DP) (Dwork et al., 2006) provides a formal privacy guarantee, defined as follows:

Definition 3.1 ((ε, δ)-Differential Privacy)**.** *A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ satisfies (ε, δ)-differential privacy if for any two adjacent inputs $C \sim \widehat{C}$ and any measurable set $\mathcal{S} \subseteq \mathcal{R}$,*

$$\Pr[\mathcal{M}(C) \in \mathcal{S}] \leq e^{\varepsilon} \Pr[\mathcal{M}(\widehat{C}) \in \mathcal{S}] + \delta.$$

When $\delta = 0$, the mechanism satisfies pure ε-differential privacy.

We adopt metric differential privacy, where the privacy guarantee depends on a distance between inputs (**?**Feyisetan et al., 2020).

Definition 3.2 (d_χ-Privacy)**.** *Let $(\mathcal{X}, d_\chi)$ be a metric space where d_χ denotes a distance metric (e.g., Euclidean distance). A randomized mechanism $M : \mathcal{X} \to \mathcal{Y}$ satisfies ϵ-d_χ-privacy if for any $x, x' \in \mathcal{X}$ and any $S \subseteq \mathcal{Y}$:*

$$\Pr[M(x) \in S] \leq e^{\epsilon \cdot d_\chi(x, x')} \Pr[M(x') \in S]. \tag{1}$$

Given a tokenized query sequence $X = [x_1, \ldots, x_n]$, the embedding layer maps it to $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{n \times d}$ via an embedding matrix $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}| \times d}$, where $|\mathcal{V}|$ is the vocabulary size and d is the embedding dimension. Under the d_χ-privacy metric with parameter ϵ and L_2 distance, privacy is enforced by adding Laplacian noise $\boldsymbol{z}$ with density proportional to $\exp(-\epsilon\|z\|)$ (Wu et al., 2017). Specifically, the noise is sampled as $\boldsymbol{z} = lv$, where $l \sim \Gamma(d, 1/\epsilon)$ and v is uniformly drawn from the unit ball B^d, yielding the privatized embedding $M(\boldsymbol{x}_t) = \boldsymbol{x}_t + \boldsymbol{z}$.

3.2 POST-PROCESSING PROPERTY

A key property of differential privacy is that privacy cannot be weakened by post-processing. That is, once a mechanism produces a DP output, any further randomized or deterministic computation that only depends on this DP output does not incur additional privacy loss (Dwork & Roth, 2014).

Theorem 3.3 (Post-processing(Dwork & Roth, 2014))**.** *Let $\mathcal{M}$ be an (ε, δ)-DP mechanism, and let f be any randomized mapping that does not access the original private input except through $\mathcal{M}(C)$. Then the composed mechanism $f(\mathcal{M}(C))$ is also (ε, δ)-DP.*

The same principle also applies to metric DP: if $\mathcal{M}$ satisfies $\varepsilon d_{\mathcal{X}}$-privacy, then any post-processing of $\mathcal{M}(x)$ satisfies the same $\varepsilon d_{\mathcal{X}}$-privacy guarantee.

4 METHODOLOGY

In this section, we propose **DP-MIND**, a **D**ifferentially **P**rivate **M**emory-guided **IN**ference framework with local **D**enoising for privacy-preserving long-context question answering. Given a long context C and a query Q, DP-MIND aims to answer Q without exposing the clean context to the external LLM.

4.1 CHUNK-WISE DP MEMORY INFERENCE

As shown in Figure 1, the first stage of DP-MIND performs chunk-wise DP memory inference. Given a long context C and a query Q, DP-MIND first partitions C into N consecutive chunks,

$$C = (C_1, C_2, \ldots, C_N).$$

Instead of exposing these clean chunks to the external LLM, DP-MIND applies a DP perturbation mechanism to the whole context and obtains the corresponding DP-perturbed chunks:

$$(C'_1, C'_2, \ldots, C'_N) \leftarrow \mathcal{M}_{\mathrm{DP}}(C).$$

Here, C'_i denotes the noisy version of C_i after applying DP perturbation to the original context. Equivalently, the mechanism perturbs the tokens in the long context and then returns the noisy chunks according to the same chunk partition. The external LLM only receives the DP-perturbed chunks $\{C'_i\}_{i=1}^N$ and never observes the clean chunks $\{C_i\}_{i=1}^N$.

After DP perturbation, DP-MIND queries the external LLM sequentially, as shown in the lower part of Figure 1. The framework maintains a memory state M_i that summarizes task-relevant information from the processed noisy chunks. Starting from $M_0 = \emptyset$, the memory is updated as

$$M_i \leftarrow \mathsf{LLM}(M_{i-1}, C'_i, Q), \qquad i = 1, \ldots, N-1.$$

Thus, M_i serves as a compact task-oriented state carrying information from the first i DP-perturbed chunks. Since each memory state is computed only from DP-perturbed chunks and previous DP-derived memories, memory construction does not introduce direct access to the clean context.

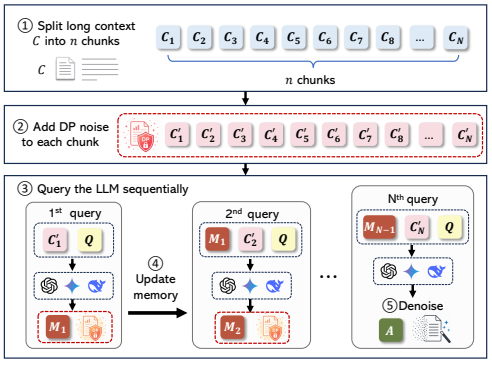


Figure 1: Workflow of DP-MIND.

At the final step, DP-MIND generates a DP answer from the last memory state, the final DP-perturbed chunk, and the query:

$$A_{\mathrm{DP}} \leftarrow \mathsf{LLM}(M_{N-1}, C'_N, Q).$$

Therefore, the entire external inference process only depends on the DP-perturbed context, the query, and the internal randomness of the external LLM. This preserves the privacy boundary of DP inference: the external LLM only accesses DP-perturbed chunks and DP-derived memories, while the clean context remains inside the trusted local environment.

4.2 PRIVACY ANALYSIS

We analyze the privacy guarantee of DP-MIND by showing that its entire external inference process is a post-processing of the underlying DP perturbation mechanism.

Theorem 4.1 (Privacy Guarantee of DP-MIND). *Let $\mathcal{M}_{\mathrm{DP}} : C \mapsto C'$ be the local context perturbation mechanism used by DP-MIND. Assume that $\mathcal{M}_{\mathrm{DP}}$ satisfies εd_χ-privacy, for a fixed query Q, define the DP-MIND mechanism*

$$\mathcal{M}_{\mathrm{DP-MIND}}(C, Q) := \mathcal{F}_{\mathrm{MIND}}\big(\mathcal{M}_{\mathrm{DP}}(C), Q\big),$$

where $\mathcal{F}_{\mathrm{MIND}}$ first splits $C' = \mathcal{M}_{\mathrm{DP}}(C)$ into $(C'_1, \ldots, C'_N)$, and then performs

$$M_i \leftarrow \mathsf{LLM}(M_{i-1}, C'_i, Q),\ i = 1, \ldots, N-1, \qquad A_{\mathrm{DP}} \leftarrow \mathsf{LLM}(M_{N-1}, C'_N, Q).$$

The output of $\mathcal{M}_{\mathrm{DP-MIND}}$ is the complete external view of this interaction, including the DP-perturbed chunks, generated memory states, and final DP answer. Then $\mathcal{M}_{\mathrm{DP-MIND}}$ also satisfies εd_χ-privacy.

Algorithm 1 $\mathcal{M}_{\mathrm{DP-MIND}}$: Chunk-wise DP Memory Inference with Local Denoising

Require: Long context $C = (w_1, \ldots, w_L)$, query Q, external LLM LLM, maximum restoration span length K

Ensure: Final answer A_{final}

1: Apply the DP perturbation mechanism $\mathcal{M}_{\mathrm{DP}}$ in Algorithm 3 to obtain
$$C' = (w'_1, \ldots, w'_L), \qquad (C'_1, \ldots, C'_N) \leftarrow \mathcal{M}_{\mathrm{DP}}(C).$$
2: Record local alignment $\Pi = \{(w'_t, w_t)\}_{t=1}^{L}$.
3: Initialize memory $M_0 \leftarrow \emptyset$.
4: **for** $i = 1$ to $N - 1$ **do**
5: Update memory $M_i \leftarrow \mathsf{LLM}(M_{i-1}, C'_i, Q)$.
6: **end for**
7: Generate DP answer $A_{\mathrm{DP}} \leftarrow \mathsf{LLM}(M_{N-1}, C'_N, Q)$.
8: Denoise answer $A_{\mathrm{final}} \leftarrow \mathsf{Denoise}(A_{\mathrm{DP}}, \Pi, K)$.
9: **return** A_{final}

Proof. Let $C' = \mathcal{M}_{\mathrm{DP}}(C)$. By assumption, $\mathcal{M}_{\mathrm{DP}}$ satisfies εd_χ-privacy. After obtaining C', DP-MIND only performs post-processing on the perturbed context: it splits C' into chunks, sequentially queries the LLM, updates the memory states, and generates the final DP answer. All these operations depend only on C', the fixed query Q, and the internal randomness of the LLM, without accessing the clean context C. Therefore,

$$\mathcal{M}_{\mathrm{DP-MIND}}(C, Q) = \mathcal{F}_{\mathrm{MIND}}\big(\mathcal{M}_{\mathrm{DP}}(C), Q\big)$$

is a post-processing of $\mathcal{M}_{\mathrm{DP}}(C)$. By the post-processing property of differential privacy (Dwork & Roth, 2014), post-processing does not increase the privacy loss. Thus, $\mathcal{M}_{\mathrm{DP-MIND}}$ satisfies the same εd_χ-privacy guarantee as $\mathcal{M}_{\mathrm{DP}}$. □

4.3 Local Answer Denoising

Task-relevant tokens, such as entities, numbers, and short answer spans, may be corrupted by DP perturbation, leading to utility loss in the generated answer. To recover such errors without revealing the clean context to the external LLM, DP-MIND performs local answer denoising after inference. During local DP perturbation, DP-MIND records the token-level alignment between the clean context $C = (w_1, \ldots, w_L)$ and the perturbed context $C' = (w'_1, \ldots, w'_L)$:

$$\Pi = \{(w'_t, w_t)\}_{t=1}^{L},$$

which is retained only in the trusted local environment. Using Π, we construct restoration rules for aligned spans of length up to K:

$$(w'_t, \ldots, w'_{t+k-1}) \rightarrow (w_t, \ldots, w_{t+k-1}), \qquad 1 \le k \le K.$$

Only perturbed spans with an unambiguous clean counterpart are included. Given the DP answer A_{DP}, the local module scans for matching noisy spans and replaces them with the corresponding clean spans, preferring the longest valid match. The denoised answer is therefore obtained as

$$A_{\mathrm{final}} \leftarrow \mathsf{Denoise}(A_{\mathrm{DP}}, \Pi, K).$$

Algorithm 2 summarizes this procedure. The denoising module only performs alignment-based span restoration rather than regenerating the answer. Since the clean context and alignment Π remain entirely within the trusted local environment, this step introduces no additional information leakage to the external LLM under our threat model.

4.4 DP-Aware Memory Policy Optimization

We optimize the memory construction policy directly under DP perturbation using reinforcement learning. The training objective targets two key challenges introduced by DP noise. An ***answer reward*** improves end-task utility by directly rewarding the correctness of the final denoised answer,

Algorithm 2 Local Answer Denoise in DP-MIND

Require: DP answer A_{DP}, local alignment $\Pi = \{(w'_t, w_t)\}_{t=1}^{L}$, max restoration span length K
Ensure: Final answer A_{final}
Build restoration map $\mathcal{R} \leftarrow \mathsf{BuildRestoreMap}(\Pi, K)$.
Tokenize A_{DP} as $(a_1, \ldots, a_m)$.
Initialize $A_{\mathrm{final}} \leftarrow \emptyset$ and $r \leftarrow 1$.
while $r \leq m$ **do**
  $(\bar{s}, k) \leftarrow \mathsf{LongestRestore}(a_r, \ldots, a_{\min(r+K-1,m)}, \mathcal{R})$.
  if $k > 0$ **then**
    Append $\bar{s}$ to A_{final} and set $r \leftarrow r + k$.
  else
    Append a_r to A_{final} and set $r \leftarrow r + 1$.
  end if
end while
return A_{final}

while a ***memory-consistency reward*** improves robustness by encouraging DP-derived memory states to remain close to their clean counterparts throughout sequential memory construction. Together, the two rewards jointly optimize final prediction quality and the stability of intermediate memory under DP perturbation. The overall reward is defined as

$$R_{\mathrm{final}} = R_{\mathrm{ans}} + \lambda R_{\mathrm{mem}},$$

where λ controls the contribution of the memory-consistency reward.

Answer reward R_{ans}. The answer reward is designed to directly improve end-task utility under DP perturbation. Given the DP-perturbed chunks $(C'_1, \ldots, C'_N)$, the policy sequentially constructs the memory trajectory

$$M_i^{\mathrm{DP}} \leftarrow \pi_\theta(M_{i-1}^{\mathrm{DP}}, C'_i, Q), \qquad i = 1, \ldots, N-1,$$

and generates the final DP answer, which is subsequently denoised in the trusted local environment:

$$A_{\mathrm{DP}} \leftarrow \pi_\theta(M_{N-1}^{\mathrm{DP}}, C'_N, Q), \qquad A_{\mathrm{final}} \leftarrow \mathsf{Denoise}(A_{\mathrm{DP}}, \Pi),$$

where Π denotes the locally maintained alignment between the clean and perturbed contexts. We then compute

$$R_{\mathrm{ans}} = \mathsf{Acc}(A_{\mathrm{final}}, Y),$$

where Y is the gold answer. By evaluating the final denoised answer, R_{ans} directly optimizes the utility of the complete DP-MIND inference pipeline.

Memory-consistency reward R_{mem}. The memory-consistency reward is designed to improve the robustness of sequential memory construction under DP noise. Since answer-level supervision only provides a terminal signal, it does not directly constrain how well intermediate memories preserve task-relevant information. We therefore construct a clean reference memory trajectory:

$$M_i^{\mathrm{clean}} \leftarrow \pi_{\mathrm{ref}}(M_{i-1}^{\mathrm{clean}}, C_i, Q), \qquad i = 1, \ldots, N-1,$$

and locally denoise the corresponding DP-derived memory:

$$\bar{M}_i^{\mathrm{DP}} \leftarrow \mathsf{Denoise}(M_i^{\mathrm{DP}}, \Pi).$$

We define the memory-consistency reward as the negative average embedding distance between the denoised DP-derived memory and its clean reference:

$$R_{\mathrm{mem}} = -\frac{1}{N-1} \sum_{i=1}^{N-1} \left\| \mathrm{Emb}(\bar{M}_i^{\mathrm{DP}}) - \mathrm{Emb}(M_i^{\mathrm{clean}}) \right\|_2,$$

where $\mathrm{Emb}(\cdot)$ maps a memory state to its embedding representation. A higher R_{mem} therefore corresponds to a smaller discrepancy from the clean reference trajectory, encouraging the policy to construct memory states that are more robust to DP perturbation.

Table 1: Sub-EM comparison on RULER-HQA under different privacy budgets and context lengths. Qwen2.5-7B denotes the standard baseline, Sparse Attention denotes Longformer-style local sliding-window attention (Beltagy et al., 2020), Long-Context Scaling denotes direct inference with Qwen2.5-7B-Instruct-1M (Yang et al., 2025), which is a variant designed for up to one million tokens of context, and Memory-Augmented denotes direct chunk-wise querying with recurrent memory propagation.

Privacy Budget	Method	7K	14K	28K	56K	112K	224K	448K	896K	1.8M
$\epsilon = 6$	Qwen2.5-7B (baseline)	36.72	37.50	25.78	–	–	–	–	–	–
	Sparse Attention	36.72	37.50	25.78	5.47	3.12	0.00	–	–	–
	Long-Context Scaling	40.62	40.62	35.16	32.03	22.66	18.75	8.56	0.00	–
	Memory-Augmented	38.28	32.03	33.59	28.91	25.00	17.97	8.59	6.25	0.00
	DP-MIND	**56.25**	**56.25**	**48.44**	**46.09**	**44.53**	**47.66**	**53.12**	**49.22**	**39.06**
$\epsilon = 7$	Qwen2.5-7B (baseline)	47.66	46.09	40.62	–	–	–	–	–	–
	Sparse Attention	47.66	46.09	40.62	6.25	4.69	0.00	–	–	–
	Long-Context Scaling	46.09	53.12	42.97	45.31	35.94	26.56	9.38	0.00	–
	Memory-Augmented	58.59	41.41	44.53	30.47	30.47	29.69	27.34	21.09	17.97
	DP-MIND	**75.78**	**74.22**	**71.09**	**73.44**	**68.56**	**70.31**	**71.88**	**64.81**	**59.38**
$\epsilon = 8$	Qwen2.5-7B (baseline)	55.47	49.22	50.00	–	–	–	–	–	–
	Sparse Attention	55.47	49.22	50.00	4.69	3.12	0.00	–	–	–
	Long-Context Scaling	55.47	51.56	50.78	55.47	45.31	37.50	9.38	0.00	–
	Memory-Augmented	59.38	44.53	42.19	39.06	42.19	39.84	29.69	32.81	22.66
	DP-MIND	**80.47**	75.00	**77.34**	**80.47**	**73.44**	**77.34**	**72.66**	**68.75**	**63.28**

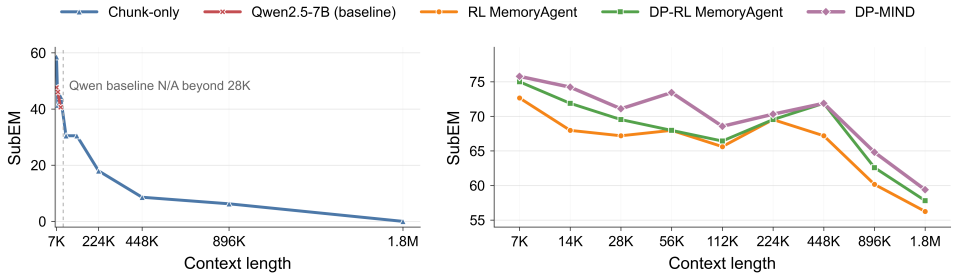


Figure 2: Results on RULER-HQA under $\epsilon = 7$. Chunk-only inference uses chunk-wise memory processing and can scale to long contexts, but its utility drops quickly without RL-based memory optimization. DP-MIND achieves the best overall performance.

5 Experiments

5.1 Experiment Setup

Models and Datasets. We evaluate DP-MIND using the Qwen2.5 family, including 7B and 14B models (Yang et al., 2024). For long-context scaling, we use the corresponding Qwen2.5-1M variants which supports approximately 1M tokens (Yang et al., 2025). We evaluate DP-MIND on RULER-HQA and NIAH. RULER-HQA is constructed from HotpotQA following the RULER protocol (Yang et al., 2018; Hsieh et al., 2024) and evaluates multi-hop QA over contexts from 7K to 1.8M tokens, with 128 examples per length. NIAH follows the RULER needle-in-a-haystack setting (Hsieh et al., 2024) and evaluates Single, Multi-Key, and Multi-Query retrieval at 32K, 128K, and 512K tokens.

5.2 Long-context Inference Task

DP-MIND is the only method that consistently supports ultra-long DP-perturbed contexts. Table 1 reports the SubEM results on RULER-HQA across different privacy budgets and context lengths. Here, "–" indicates that the corresponding method cannot process the input because the context length exceeds its supported context window or serving limit. The standard Qwen2.5-7B baseline only supports up to 28K tokens. Sparse Attention extends the usable length to 224K tokens, but still fails to handle 448K and longer contexts. Long-Context Scaling, implemented with Qwen2.5-7B-Instruct-1M, can process up to 1.8M tokens, but it relies on a model with an enlarged context window. In contrast, DP-MIND processes the context chunk by chunk and remains applicable across the full range from 7K to 1.8M tokens.

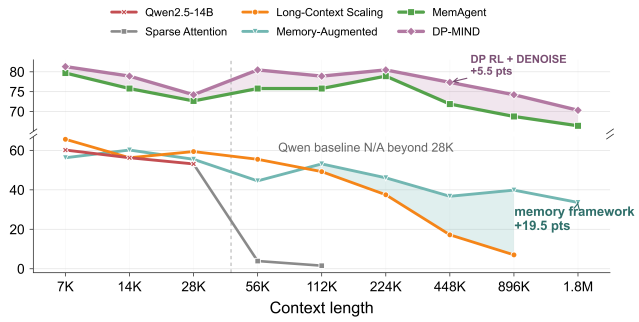


Figure 3: Scaling performance on RULER-HQA with Qwen2.5-14B under $\epsilon = 7$. DP-MIND consistently achieves the best overall performance, while the advantage of memory-based inference becomes more pronounced as the context length increases.

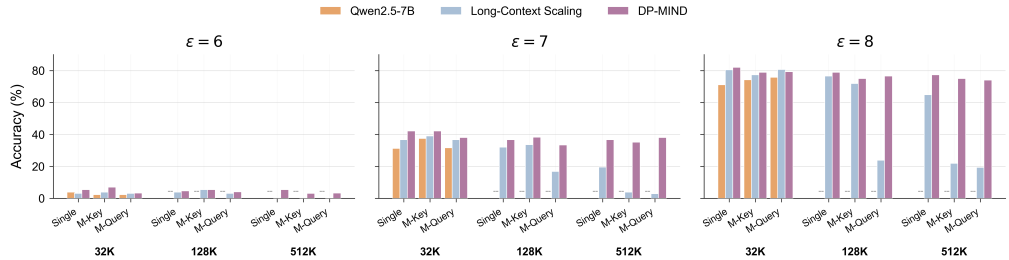


Figure 4: NIAH results under different privacy budgets and context lengths. Results are reported on Single, Multi-Key (M-Key), and Multi-Query (M-Query) with length ranging from 32k to 512k.

DP-MIND achieves much higher utility than scalable baselines. Although Long-Context Scaling can process ultra-long inputs, its utility degrades rapidly as the context grows, while Sparse Attention deteriorates even earlier. Under $\epsilon = 7$, Long-Context Scaling drops from 45.31 SubEM at 56K to 26.56 at 224K and 0.00 at 896K and 1.8M, whereas DP-MIND maintains 73.44, 70.31, 64.81, and 59.38, respectively. These results demonstrate that simply extending the context window or adopting sparse attention is insufficient under DP perturbation, while memory-guided inference preserves task-relevant information more effectively.

DP-MIND consistently generalizes to the larger 14B backbone. We further evaluate DP-MIND with Qwen2.5-14B to examine whether its effectiveness generalizes to a larger model scale. As shown in Figure 3, DP-MIND achieves the best overall performance across all evaluated context lengths. Compared with MemAgent, DP-MIND consistently improves performance at every context length, showing that DP-aware RL training and local answer denoising remain effective when scaling from 7B to 14B. Moreover, the benefit of the memory-based inference framework becomes increasingly pronounced as the context grows. Compared with direct long-context scaling, Memory-Augmented inference provides relatively modest gains in the short-context regime, but the gap widens substantially at longer contexts, reaching approximately 19.5 SubEM points at 1.8M tokens. These results further demonstrate that the advantage of DP-MIND is not specific to the 7B backbone and that memory-guided inference becomes increasingly valuable in the ultra-long-context regime.

5.3 Long-context Retrieval Task

We further extend the evaluation to NIAH to examine long-context retrieval under DP perturbation. As shown in Figure 4, DP-MIND consistently outperforms the baselines across different context lengths and privacy budgets. **Across privacy budgets, DP-MIND remains effective under different perturbation strengths.** When $\epsilon = 6$, severe DP noise makes retrieval challenging for all methods; at $\epsilon = 7$, DP-MIND maintains relatively stable performance as the context length increases, while the baselines degrade more noticeably; and at $\epsilon = 8$, DP-MIND achieves the largest overall gains. **The advantage of DP-MIND also becomes more pronounced as the retrieval context grows.** The performance gap between DP-MIND and the baselines generally widens at longer context lengths, showing that memory-guided inference is particularly effective for preserving and retrieving sparse target information from ultra-long DP-perturbed contexts.

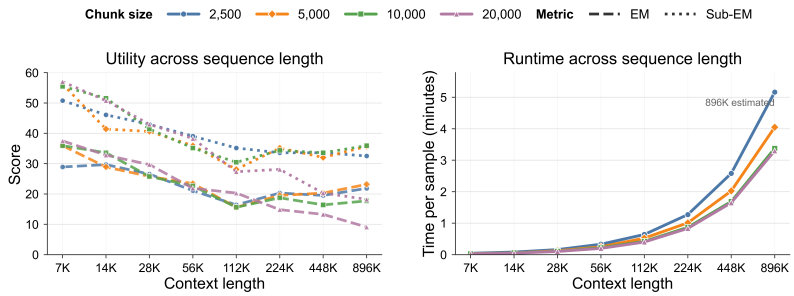


Figure 5: Impact of chunk size on utility and runtime. Larger chunks reduce inference time, while moderate chunk sizes generally maintain stronger EM and SubEM across long contexts.

5.4 Ablation Study

Chunk size provides a trade-off between utility and inference efficiency. As shown in Figure 5, DP-MIND remains relatively stable across a wide range of chunk sizes, indicating that its performance is not overly sensitive to this hyperparameter. For utility, chunk sizes of 5K–10K generally achieve competitive EM and SubEM across different context lengths, while an excessively large chunk size of 20K tends to degrade performance in the ultra-long regime. In contrast, runtime is more sensitive to chunk size: larger chunks reduce the number of recurrent inference steps and therefore substantially lower the time per sample, with the efficiency gap becoming increasingly pronounced as the context length grows. Overall, a moderate chunk size provides a favorable balance between retrieval utility and inference efficiency.

Memory propagation, DP-aware RL, and local denoising progressively improve long-context utility. Table 3 shows that Chunk-only inference degrades rapidly as the context grows, while introducing memory propagation substantially improves long-context robustness. Training directly on DP-perturbed inputs provides further gains, and local answer denoising enables DP-MIND to achieve the best overall performance. We further evaluate DP-MIND on the larger Qwen2.5-14B backbone. As shown in Figure 3, DP-MIND consistently performs best, with the advantage of memory-based inference becoming more pronounced at longer context lengths.

Table 2: Robustness of DP-MIND under different DP perturbation mechanisms.

Perturbation	Length	Qwen-1M	DP-MIND
d_χ-privacy (Feyisetan et al., 2020)	56K	45.31	73.44 (+ 28.13)
	112K	35.94	68.56 (+ 32.62)
	224K	26.56	70.31 (+ 43.75)
	448K	9.38	71.88 (+ 62.50)
SanText (Yue et al., 2021)	56K	52.34	82.03 (+ 29.69)
	112K	50.78	78.91 (+ 28.13)
	224K	37.50	78.12 (+ 40.62)
	448K	6.25	75.00 (+ 68.75)
InferDPT (Tong et al., 2025)	56K	57.81	80.47 (+ 22.66)
	112K	50.78	78.91 (+ 28.13)
	224K	34.38	78.12 (+ 43.74)
	448K	7.81	71.88 (+ 64.07)

DP-MIND remains effective across different DP perturbation mechanisms. As shown in Table 2, to evaluate its robustness to different perturbation mechanism, we further adopt the perturbation mechanisms from SanText (Yue et al., 2021) and InferDPT (Tong et al., 2025). Across all three settings and all evaluated context lengths, DP-MIND consistently outperforms Qwen-1M. The gains become particularly pronounced at 448K, reaching 62.50, 68.75, and 64.07 SubEM points under the three perturbation mechanisms, respectively. These results demonstrate that DP-MIND remains effective across different DP perturbation mechanism.

6 Conclusion

In this work, we study differentially private long-context inference at the million-token scale and propose **DP-MIND**, a memory-guided framework that sequentially processes DP-perturbed chunks while propagating compact memory states across the context. DP-MIND preserves the privacy guarantee of the underlying perturbation mechanism and improves utility through DP-aware reinforcement learning for robust memory construction and trusted local denoising. Experiments on RULER-HQA and NIAH across different privacy budgets, perturbation mechanisms, and model scales show that DP-MIND maintains strong utility in the million-token regime, demonstrating the effectiveness of memory-guided inference for scalable private LLM inference.

AI USE STATEMENT

We used generative AI tools to assist with language editing, improving the clarity and organization of the manuscript, and refining LaTeX formatting and visualization code. All AI-assisted content was carefully reviewed and revised by the authors. We independently verified the technical claims, mathematical proofs, experimental results, and references, and checked the correctness of the final figures and tables. We take full responsibility for the final content of this work, including any text or artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

We provide implementation details, training configurations, and evaluation protocols in the main paper and appendix to facilitate reproducibility. The appendix includes the complete proofs of our theoretical results, hyperparameter settings, and additional experimental results. All experiments are conducted using publicly available models, datasets, and evaluation benchmarks. Our implementation and scripts for reproducing the experiments are available at `https://anonymous.4open.science/r/DP-MIND-2CEC`.

REFERENCES

Tianyu Bai, Yunhe Feng, and Song Fu. Safeguarding user data privacy in online large language model services. *Journal of Systems Architecture*, pp. 103555, 2025.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Tom B. Brown, Benjamin Mann, Nick Ryder, et al. Language models are few-shot learners. *NeurIPS*, 2020.

Huimin Chen, Fengran Mo, Yanhao Wang, Cen Chen, Jian-Yun Nie, Chengyu Wang, and Jamie Cui. A customized text sanitization mechanism with differential privacy. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 5747–5758. Association for Computational Linguistics, 2023a.

Kang Chen, Xiuze Zhou, Yuanguo Lin, Shibo Feng, Li Shen, and Pengcheng Wu. A survey on privacy risks and protection in large language models. *Journal of King Saud University Computer and Information Sciences*, 37(7):163, 2025.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023b.

Zhuoen Chen, Dongfang Li, Meishan Zhang, Baotian Hu, and Min Zhang. Dynamic long context reasoning over compressed memory via end-to-end reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8064–8083, 2026.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. LongRoPE: Extending LLM context window beyond 2 million tokens. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Minxin Du, Xiang Yue, Sherman S. M. Chow, Tianhao Wang, Chenyu Huang, and Huan Sun. DP-Forward: Fine-tuning and inference on language models with differential privacy in forward pass. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2665–2679. Association for Computing Machinery, 2023.

Cynthia Dwork and Aaron Roth. *The Algorithmic Foundations of Differential Privacy*, volume 9 of *Foundations and Trends in Theoretical Computer Science*. Now Publishers Inc., 2014.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology-EUROCRYPT 2006*, pp. 486–503. Springer, 2006.

Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. Privacy- and utility-preserving textual analysis via calibrated multivariate perturbations. In *Proceedings of the Thirteenth ACM International Conference on Web Search and Data Mining*, pp. 178–186, 2020.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? In *Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2404.06654.

Yu Lin, Ruining Yang, Yunlong Mao, Qizhi Zhang, Jue Hong, Quanwei Cai, Ye Wu, Huiqi Liu, Zhiyu Chen, Bing Duan, et al. Obfuslm: Privacy-preserving language model service against embedding inversion attacks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1160–1174, 2025.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

Peihua Mai, Ran Yan, Zhe Huang, Youjia Yang, and Yan Pang. Split-and-denoise: Protect large language model inference with local differential privacy. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34281–34302. PMLR, 2024. URL https://proceedings.mlr.press/v235/mai24a.html.

Peihua Mai, Xuanrong Gao, Youlong Ding, Xianglong Du, Wei Liu, and Yan Pang. Sharedrequest: Privacy-preserving model-agnostic inference for large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7129–7150, 2026.

Long Ouyang, Jeffrey Wu, Xu Jiang, et al. Training language models to follow instructions with human feedback. *NeurIPS*, 2022.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. In *International Conference on Learning Representations*, 2024.

Alibaba Cloud Qwen Team. Qwen2 technical report. https://qwen2.org/paper/, 2025.

Zirui Song, Bin Yan, Yuhan Liu, Miao Fang, Mingzhe Li, Rui Yan, and Xiuying Chen. Injecting domain-specific knowledge into large language models: a comprehensive survey. *arXiv preprint arXiv:2502.10708*, 2025.

Rushil Thareja, Preslav Nakov, Praneeth Vepakomma, and Nils Lukas. Dp-fusion: Token-level differentially private inference for large language models. In *International Conference on Learning Representations*, volume 2026, pp. 151088–151120, 2026.

Meng Tong, Kejiang Chen, Jie Zhang, Yuang Qi, Weiming Zhang, Nenghai Yu, Tianwei Zhang, and Zhikun Zhang. Inferdpt: Privacy-preserving inference for closed-box large language models. *IEEE Transactions on Dependable and Secure Computing*, 22(5):4625–4640, 2025.

Xi Wu, Fengan Li, Arun Kumar, Kamalika Chaudhuri, Somesh Jha, and Jeffrey Naughton. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In *Proceedings of the 2017 ACM international conference on management of data*, pp. 1307–1322, 2017.

Zekun Xu, Abhinav Aggarwal, Oluwaseyi Feyisetan, and Nathanael Teissier. A differentially private text perturbation method using a regularized mahalanobis metric. In *Proceedings of the Second Workshop on Privacy in NLP*, pp. 7–17. Association for Computational Linguistics, 2020.

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

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, et al. Qwen2.5-1m technical report. *arXiv preprint arXiv:2501.15383*, 2025. URL https://arxiv.org/abs/2501.15383.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, et al. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. In *International Conference on Learning Representations*, volume 2026, pp. 39458–39486, 2026.

Xiang Yue, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman S. M. Chow. Differential privacy for text analytics via natural text sanitization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3853–3866. Association for Computational Linguistics, 2021.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17283–17297, 2020.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Zican Dong, Yupeng Hou, Beichen Zhang, Yingqian Min, Junjie Zhang, Peiyu Liu, et al. A survey of large language models. *Frontiers of Computer Science*, 20(12):2012627, 2026.

Lele Zheng, Chao Zhang, Feiyang Yuan, Ke Cheng, Tao Zhang, Anxiao Song, and Yulong Shen. Dp3: Differentially private prompt perturbation for multi-turn llm inference. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 18538–18552, 2026.

A Implementation Details

A.1 DP Perturbation

Before sending the context to the external LLM, DP-MIND locally applies a token-level DP perturbation mechanism to protect the original context. Specifically, each token is mapped to its embedding representation, perturbed with calibrated noise, and then projected back to the nearest token in the vocabulary. The resulting DP-perturbed context is subsequently partitioned into sequential chunks and used as the input to the memory-guided inference process. Algorithm 3 summarizes the complete procedure.

Algorithm 3 Long-Context DP Perturbation Mechanism $\mathcal{M}_{\mathrm{DP}}$

Require: Long context $C = (w_1, \ldots, w_L)$, number of chunks N, embedding map $\phi : \mathcal{W} \to \mathbb{R}^d$, vocabulary $\mathcal{W}$, privacy parameter $\varepsilon > 0$
Ensure: DP-perturbed chunks $(C'_1, \ldots, C'_N)$
1: **for** $t = 1$ to L **do**
2: Compute embedding $\phi_t \leftarrow \phi(w_t)$.
3: Sample noise Z_t with density $p_Z(z) \propto \exp(-\varepsilon \|z\|_2)$.
4: Perturb embedding $\widehat{\phi}_t \leftarrow \phi_t + Z_t$.
5: Project back to token $w'_t \leftarrow \mathrm{argmin}_{u \in \mathcal{W}} \|\phi(u) - \widehat{\phi}_t\|_2$.
6: **end for**
7: Form the perturbed context $C' = (w'_1, \ldots, w'_L)$ and split it into N chunks $(C'_1, \ldots, C'_N)$.
8: **return** $(C'_1, \ldots, C'_N)$

A.2 Memory-based Inference Details

For the chunk-based recurrent baseline, we first tokenize each input context using the Qwen2.5-7B-Instruct tokenizer and partition it into sequential, non-overlapping chunks of 5,000 tokens, with the last chunk allowed to contain fewer tokens. The chunks are processed in their original order. For each chunk, the model receives the question, the current chunk, and the memory accumulated from previous chunks, and updates the memory while preserving question-relevant information. The memory is initialized as "No previous memory," and each memory update is constrained to at most 1,024 generated tokens. After all chunks have been processed, the final answer is generated using the question and the final accumulated memory. We use a temperature of 0.7 and top-p of 0.95, and evaluate 128 examples under each experimental setting.

A.3 Prompts

We use fixed prompt templates for memory updating and final answer generation across all experiments. The memory update prompt incrementally integrates task-relevant information from each chunk, while the final answer prompt generates the prediction using only the accumulated differentially private memory. The complete templates are provided below.

Memory Update Prompt

You are given a problem, a section of an article that may contain information relevant to the problem, and a previous memory. Read the section carefully and update the memory with any new information that helps answer the problem. Preserve all relevant details from the previous memory while adding useful information from the current section.

```
<problem>
{question}
</problem>

<memory>
{previous_memory}
</memory>

<section>
{current_chunk}
</section>
```

Updated memory:

Final Answer Prompt

You are given a problem and a differentially private memory. The memory may contain noise or omit some details due to privacy-preserving perturbation. Answer the problem using only the memory, without attempting to reconstruct any original sensitive context.

Output format:
`"Therefore, the answer is (insert answer here)".`

```
<problem>
{question}
</problem>

<memory>
{final_memory}
</memory>
```

Your answer:

B Evaluation Details

B.1 Benchmarks

RULER-HQA. RULER-HQA is constructed from HotpotQA (Yang et al., 2018) following the RULER-style synthetic construction (Hsieh et al., 2024). Each example contains supporting evidence mixed with a large number of distractor documents, enabling controlled evaluation of long-context extrapolation. We synthesize 128 examples for each context scale. The evaluated contexts range from approximately **7K** to **1.8M** tokens.

NIAH. NIAH evaluates whether the model can retrieve a short target span from a long haystack. We use three variants from RULER (Hsieh et al., 2024): repetitive haystack with a numeric needle, essay haystack with a numeric needle, and essay haystack with a UUID needle. We evaluate 128 examples at each context length, ranging from **32K** to **512K** tokens.

B.2 Evaluation Metrics

We normalize both predictions and gold answers by lowercasing, removing punctuation and articles, and fixing whitespace. EM measures exact string match after normalization. F1 measures token-level overlap between the prediction and the gold answer. SubEM counts a prediction as correct if it contains the complete normalized gold answer span. When multiple gold answers are available, we report the maximum score over all gold answers. For DP-MIND, local span restoration is applied before evaluation on answer-based tasks, and EM, F1, and SubEM are computed on the restored answer.

C Additional Experiment Results

This section provides additional experimental results on RULER-HQA to further support the effectiveness of DP-MIND. We first present an ablation study under $\epsilon = 7$ to isolate the contribution of each component, including DP-perturbed training and local answer denoising. We then report complete comparisons across multiple privacy budgets and context lengths using F1, Exact Match (EM), and Sub-Exact Match (SubEM). Overall, the results show that DP-MIND consistently improves over sparse-attention, long-context scaling, and memory-augmented baselines, with especially clear gains in ultra-long contexts.

Table 3: Ablation study of DP-MIND on RULER-HQA under $\varepsilon = 7$, measured by SubEM. Chunk-only splits the context into chunks and queries them without memory propagation. RL MemoryAgent denotes the original memory-based inference method (Yu et al., 2026). DP-RL MemoryAgent is trained on DP-perturbed data. DP-MIND is our full method with local answer denoising. Numbers in parentheses denote improvements over RL MemoryAgent.

Length	Chunk-only	RL MemoryAgent	DP-RL MemoryAgent	DP-MIND
7K	58.59	72.66	75.00 (+2.34)	**75.78** (+3.12)
14K	41.41	67.97	71.88 (+3.91)	**74.22** (+6.25)
28K	44.53	67.19	69.53 (+2.34)	**71.09** (+3.90)
56K	30.47	67.97	67.97 (+0.00)	**73.44** (+5.47)
112K	30.47	65.62	66.43 (+0.81)	**68.56** (+2.13)
224K	17.97	69.53	69.53 (+0.00)	**70.31** (+0.78)
448K	8.59	67.19	71.88 (+4.69)	**71.88** (+4.69)
896K	6.25	60.16	62.59 (+2.43)	**64.81** (+2.22)
1.8M	0.00	56.25	57.81 (+1.56)	**59.38** (+3.13)

Table 4: F1 comparison on RULER-HQA under different privacy budgets and context lengths. All values are percentages.

Privacy Budget	Method	7K	14K	28K	56K	112K	224K	448K	896K	1.8M
$\epsilon = 6$	Qwen2.5-7B (baseline)	36.12	37.84	27.98	–	–	–	–	–	–
	Sparse Attention	36.12	37.84	27.98	4.81	2.56	0.00	–	–	–
	Long-Context Scaling	43.94	42.29	36.80	34.10	27.23	21.69	12.42	0.00	0.00
	Memory-Augmented	39.98	33.56	34.38	27.88	27.52	22.88	26.51	20.09	20.94
	DP-MIND	**53.30**	**54.93**	**53.57**	**48.11**	**48.36**	**48.23**	**52.43**	**47.62**	**42.41**
$\epsilon = 7$	Qwen2.5-7B (baseline)	41.70	45.15	40.90	–	–	–	–	–	–
	Sparse Attention	41.70	45.15	40.90	5.27	3.18	–	–	–	–
	Long-Context Scaling	50.38	51.08	42.78	45.42	35.97	28.17	8.88	4.69	0.00
	Memory-Augmented	51.76	43.30	40.55	39.46	29.64	31.73	31.77	30.74	21.69
	DP-MIND	**71.49**	**71.92**	**69.46**	**66.16**	**63.62**	**71.28**	**68.68**	**58.11**	**58.11**
$\epsilon = 8$	Qwen2.5-7B (baseline)	43.19	40.90	44.53	–	–	–	–	–	–
	Sparse Attention	43.19	40.90	44.53	4.25	3.01	0.48	–	–	–
	Long-Context Scaling	55.77	53.23	51.14	54.53	44.60	37.32	10.95	8.88	4.81
	Memory-Augmented	53.17	41.29	45.09	38.85	37.47	40.06	30.74	34.14	27.73
	DP-MIND	**77.26**	**72.64**	**74.21**	**74.45**	**69.78**	**72.48**	**69.86**	**65.92**	**62.35**

Table 5: Exact Match (EM) comparison on RULER-HQA under different privacy budgets and context lengths. All values are percentages.

Privacy Budget	Method	7K	14K	28K	56K	112K	224K	448K	896K	1.8M
$\epsilon = 6$	Qwen2.5-7B (baseline)	21.09	22.66	16.41	–	–	–	–	–	–
	Sparse Attention	21.09	22.66	16.41	0.78	0.00	0.00	–	–	–
	Long-Context Scaling	26.56	25.00	21.88	17.97	14.84	14.06	7.03	2.56	0.00
	Memory-Augmented	24.22	19.53	19.53	10.16	14.06	10.94	12.50	8.59	10.16
	DP-MIND	**33.59**	**39.84**	**35.94**	**32.03**	**28.91**	**31.25**	**36.72**	**32.81**	**28.12**
$\epsilon = 7$	Qwen2.5-7B (baseline)	24.22	29.69	29.69	–	–	–	–	–	–
	Sparse Attention	24.22	29.69	29.69	0.78	0.00	–	–	–	–
	Long-Context Scaling	35.16	32.03	28.91	31.25	21.88	16.41	4.69	4.81	0.00
	Memory-Augmented	35.94	28.91	25.78	23.44	15.62	19.53	20.31	12.50	14.84
	DP-MIND	**55.47**	**55.47**	**54.69**	**47.66**	**48.44**	**57.81**	**50.00**	**44.53**	**45.31**
$\epsilon = 8$	Qwen2.5-7B (baseline)	28.12	26.56	32.81	–	–	–	–	–	–
	Sparse Attention	28.12	26.56	32.81	0.00	0.00	0.00	–	–	–
	Long-Context Scaling	37.50	35.16	35.16	39.84	29.69	24.22	15.47	16.41	14.06
	Memory-Augmented	35.94	25.00	28.91	23.44	20.31	25.78	19.53	18.75	17.19
	DP-MIND	**60.94**	**55.47**	**60.16**	**58.59**	**56.25**	**60.16**	**53.12**	**51.56**	**50.00**

Table 6: F1 comparison on RULER-HQA for Qwen2.5-14B series under $\epsilon = 7$.

Method	7K	14K	28K	56K	112K	224K	448K	896K	1.8M
Qwen2.5-14B (baseline)	23.31	22.77	19.64	–	–	–	–	–	–
Sparse Attention	23.31†	22.77†	19.64†	3.57	3.64	0.06	–	–	–
Long-Context Scaling	48.52	37.13	34.96	30.40	27.23	21.21	15.87	10.25	–
Memory-Augmented	53.71	55.05	53.40	40.97	44.64	44.27	36.40	36.35	32.80
DP-MIND	**75.77**	**72.97**	**67.56**	**71.11**	**71.43**	**72.69**	**66.00**	**65.13**	**62.23**

Table 7: Exact Match (EM) comparison on RULER-HQA for Qwen2.5-14B series under $\epsilon = 7$.

Method	7K	14K	28K	56K	112K	224K	448K	896K	1.8M
Qwen2.5-14B (baseline)	11.72	9.38	8.59	–	–	–	–	–	–
Sparse Attention	11.72	9.38	8.59	3.12	3.91	0.00	–	–	–
Long-Context Scaling	32.81	21.09	24.22	18.75	17.19	12.50	6.25	2.34	–
Memory-Augmented	37.50	36.72	34.38	25.78	28.91	31.25	22.66	23.44	19.53
DP-MIND	**58.59**	**53.12**	**48.44**	**50.78**	**51.56**	**54.69**	**49.22**	**48.44**	**46.40**

Table 8: Sub-Exact Match (Sub-EM) comparison on RULER-HQA for Qwen2.5-14B series under $\epsilon = 7$.

Method	7K	14K	28K	56K	112K	224K	448K	896K	1.8M
Qwen2.5-14B (baseline)	60.16	56.25	53.12	–	–	–	–	–	–
Sparse Attention	60.16	56.25	53.12	3.91	1.56	0.00	–	–	–
Long-Context Scaling	65.62	56.25	59.38	55.47	49.22	37.50	17.19	7.03	–
Memory-Augmented	56.25	60.16	55.47	44.53	53.12	46.09	36.72	39.84	33.59
DP-MIND	**81.31**	**78.91**	**74.22**	**80.47**	**78.91**	**80.47**	**77.34**	**74.22**	**70.31**

Table 9: NIAH results under different privacy budgets and context lengths. Results are reported on Single, Multi-Key (M-Key), and Multi-Query (M-Query) settings, where ↑ indicates higher is better.

Privacy Budget	Method	32K			128K			512K		
		Single↑	M-Key↑	M-Query↑	Single↑	M-Key↑	M-Query↑	Single↑	M-Key↑	M-Query↑
$\epsilon = 6$	Qwen2.5-7B	3.91	2.34	2.34	–	–	–	–	–	–
	Qwen2.5-7B-1M	3.12	3.91	3.12	3.91	5.47	3.12	0.00	0.00	0.39
	DP-MIND	**5.47**	**7.03**	**3.32**	4.69	5.47	4.10	5.47	3.12	3.32
$\epsilon = 7$	Qwen2.5-7B	31.25	37.50	31.64	–	–	–	–	–	–
	Qwen2.5-7B-1M	36.72	39.06	36.72	32.03	33.59	16.80	19.53	3.91	2.93
	DP-MIND	**42.19**	**42.19**	**38.09**	36.72	38.28	33.40	36.72	35.16	38.09
$\epsilon = 8$	Qwen2.5-7B	71.09	74.22	75.78	–	–	–	–	–	–
	Qwen2.5-7B-1M	80.47	77.34	**80.66**	76.56	71.88	23.83	64.84	21.88	19.34
	DP-MIND	**82.03**	**78.91**	79.30	78.91	75.00	76.56	77.34	75.00	74.02