Risk-Aware Privacy Preservation for LLM Inference

Zhihuang Liu, Zhangdong Wang, Tongqing Zhou, Yonghao Tang, Yuchuan Luo, Zhiping Cai

***Abstract*—Large Language Model (LLM) inference services like ChatGPT are popular for enabling diverse tasks via prompts, yet they exacerbate privacy risks due to the potential exposure of sensitive data in user inputs. Existing local differential privacy (LDP)-based text sanitization mechanisms offer lightweight protection suitable for cloud-based LLM inference. Nevertheless, uniform privacy budget allocation and generalized sanitization mechanisms neglect the critical protection needs of sensitive user data, such as Personally Identifiable Information (PII). Empirical evidence of this work reveals that even with a strict privacy budget (ϵ=0.1), the sensitive information leakage rate can reach an alarmingly high 71.74%. To address these challenges, this paper proposes Rap-LI, a risk-aware privacy preservation framework for LLM inference, designed to be plug-and-play. Rap-LI performs risk identification and personalized labeling on user prompts, then develops a risk-aware LDP mechanism for text sanitization, formally proven to satisfy both token-level and sentence-level LDP guarantees. Extensive experimental results demonstrate Rap-LI's superior privacy-utility balance. It improves privacy protection against sensitive information leakage by an average of 51.68% compared to methods with comparable utility. Our code is available at https://github.com/Cristliu/RapLI.**

***Index Terms*—Large language models, private data leakage, personal identifiable information, differential privacy, text sanitization.**

I. Introduction

BUILDING on massive parameters tuned with tremendous data, Large Language Models (LLMs) demonstrate remarkable capability growth and market expansion. Nowadays, users increasingly access powerful LLM inference services through cloud-based APIs/interfaces [1], with ChatGPT serving over 300 million weekly users as a prime example [2]. During inference [3], users input a prompt (*see Table I for an example*) consisting of dynamic textual descriptions (*user prompt*) alongside an inherent *system prompt* that specifies response styles or task constraints [4]. LLMs then generate responses to the textual query and facilitate diverse applications [5] (e.g., information extraction and providing medical advice).

However, LLM inference introduces critical privacy concerns to the involved users [6]. To enhance task performance, users increasingly incorporate sensitive data into prompts [7], [8], such as medical histories or Personally Identifiable Information (PII) [9], [10]. Such prompts in LLM interactions in turn track user queries and profile them, which have been demonstrated to be exposed to LLM service providers, usually cloud-based, and utilized for training purposes [11]. Moreover, these prompts may unintentionally leak to other users through the model's responses [12] and are susceptible to various attacks [13], [14]. A well-known incident involves Samsung employees utilizing ChatGPT to debug source code and convert internal meeting notes into presentations. Thus, confidential corporate information, such as new program source code and meeting records, is exposed to ChatGPT's developer, OpenAI [15].

Due to cloud-based LLMs' internal inaccessibility or parameter opacity, typical privacy-preserving methods that require server collaboration, such as homomorphic encryption in cryptographic techniques [16] or fine-tuning techniques that involve retraining the model [17], [18], are generally impractical. Consequently, we note that privacy risks during LLM inference should be addressed locally on the user side to prevent sensitive data from being explicitly exposed to service providers or attackers. For this, a common practice is to anonymize or sanitize data before using LLM inference services [19]. However, simply masking sensitive data has proven insufficient for rigorous privacy protection purposes [12], as user privacy might still be inferred from other contextual information within the input prompts (see Table I) [13]. Another approach is to introduce local (large) language models, such as BERT, Llama, and DeepSeek [20], which are tasked with rewriting prompts to protect privacy [21], [22]. However, such intuitive rewriting either struggles to follow instructions and protect sensitive information [23], [24] or consumes local resources and increases response time [25], [26].

As the gold standard for bounding privacy risks, *differential privacy* (DP) [27] is well-known to provide provable protection for *text sanitization* [28]–[31]. In particular, *local differential privacy* [32] (LDP)-based text sanitization mechanisms are expected to offer lightweight solutions for protecting user prompts locally during LLM inference [3], [33]. By embedding dedicated noise, these mechanisms could perturb original sentences or tokens [34] (e.g., words, subwords, or characters) into sanitized versions. Wherein, sentence-level perturbations lack fine-grained control over each part of the sentence and often produce outputs that are difficult to map back to interpretable (or human-readable) sentence forms [19], [28]. Hence, token-level sanitization is generally favored in prompt privacy protection practice [3], [28]–[31], [33].

This work identifies that token-level LDP in LLM inference, favored for fine-grained control, uniformly applies the same privacy budget (ϵ) to all tokens. Such a "one-size-fits-all" strategy ignores varying sensitivity levels across textual components (e.g., diet v.s. illness). Consequently, it fails to

This work was supported in part by the National Natural Science Foundation of China under Grants 62472434, 62472431, and U22B2005; in part by the Key Program of Hunan Provincial Natural Science Foundation under Grant 2026JJ30028; and in part by the Science and Technology Innovation Program of Hunan Province under Grant 2023RC3027. (*Corresponding author: Yuchuan Luo; Zhiping Cai.*)

The authors are with the College of Computer Science and Technology, National University of Defense Technology, Changsha 410082, China (e-mail: lzhliu@nudt.edu.cn, wangzd@nudt.edu.cn, zhoutongqing@nudt.edu.cn, tangyh@nudt.edu.cn, luoyuchuan09@nudt.edu.cn, zpcai@nudt.edu.cn).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author. The material includes the Appendix referenced in the paper. Contact lzhliu@nudt.edu.cn for further questions about this work.

TABLE I: An example comparison between the uniform-budget-based prompt protection paradigm and our proposed method. Assume that the adversary has background knowledge (e.g., *knows that John, age 45, currently has a stomachache*). HighMask denotes simple masking of sensitive content such as PII. Information exposure is measured through both PII leakage and contextual privacy leakage, while utility preservation is evaluated based on the contextual relevance and medical accuracy of the LLM-generated response. Red rectangles (□, ◧, ■) indicate information exposure levels (lower is better for privacy), while blue rectangles (□, ◧, ■) indicate utility preservation (higher is better for utility).

<table>
<tr><td colspan="4"><b>System Prompt:</b> You are a helpful medical assistant. You need to answer users' questions and inquiries about healthcare. Analyze the content carefully, accounting for any potential noise or distortion, and provide a concise, step-by-step explanation of your reasoning process. Finally, provide a brief medical advice based on the information.</td></tr>
<tr><th>Method</th><th>User Prompt</th><th>Information Exposure ↓</th><th>Utility Preservation ↑</th></tr>
<tr><td>Original Text</td><td>Patient John, aged 45, was diagnosed with Type 2 diabetes and prescribed metformin. However, he began to have stomachache after taking the medication.</td><td>John, 45, Type 2 diabetes, metformin, stomachache ■■■■■</td><td>No perturbation, maximum utility preserved ■■■■■</td></tr>
<tr><td>HighMask</td><td>Patient [PERSON], aged [DATE], was diagnosed with Type 2 diabetes and prescribed metformin. However, he began to have stomachache after taking the medication.</td><td>Type 2 diabetes, metformin, stomachache ■■■■◧</td><td>Only name and age masked, good utility preserved ■■■■□</td></tr>
<tr><td>Existing Method (ϵ=0.1)</td><td>Geneient Bill, alcoholism 34, was incarcerated with Structure 17 obesity and harmful penicillin. likewise, he started to have nutritionalache after filling the pharmaceuticals.</td><td><i>None</i> □□□□□</td><td>Semantics distorted, key symptom "stomachache" changed to "nutritionalache", completely unrelated to the condition ◧□□□□</td></tr>
<tr><td>Existing Method (ϵ=1)</td><td>Pamient William, eroded 34, was incarcerated with Detachment 216 dementia and contemplated befriendedformin. conversely, he started to have gulpedache after giving the antibiotics.</td><td>-formin ◧□□□□</td><td>Semantics distorted, key symptom "stomachache" changed to "gulpedache", likely to cause misdiagnosis ■◧□□□</td></tr>
<tr><td>Existing Method (ϵ=8)</td><td>Betient James, age 34, was hospitalized with types 251 diabetes and recommends metformin. although, he started to have nauseaache after taken the medications.</td><td>diabetes, metformin ■■■◧□</td><td>Basic semantics preserved, symptom "stomachache" changed to "nauseaache", LLM can still provide relevant advice ■■■□□</td></tr>
<tr><td>Ours</td><td>Patient Donald, age 34, was diagnosed with Gender 38 physiology and instituted phenformin. Afterwards, he continued to have stomachache after gaining the treatments.</td><td>-formin, stomachache (that the user expects to keep) ■□□□□</td><td>Uses semantically similar substitutions, LLM can still provide satisfactory advice ■■■◧□</td></tr>
</table>

meet contextual privacy requirements in practical applications [35], [36], suffers from utility degradation due to semantic loss during sanitization, and struggles to resolve the inherent utility-privacy trade-off [19]. Table I presents an example that illustrates the above challenges in existing approaches. *Context-aware or personalized approaches remain notably absent in DP-based LLM inference protection and even natural language text sanitization*. These limitations also align with the broader trend of *privacy-preserving prompt engineering* summarized in the survey [19], which highlights the critical need for adaptive, context-sensitive protection approaches. More critically, we observe that the lack of risk awareness for LDP-based text sanitization in LLM inference opens the door for privacy leakage of high-risk information (e.g., PII), which is up to 71.74% even at ϵ=0.1 (*theoretical analysis in Sec. IV and empirically in Sec. VI-B*).

To mitigate this vulnerability, we propose a **R**isk-**a**ware **p**rivacy preservation framework for **L**LM **I**nference (Rap-LI). Compared to existing methods that apply a uniform privacy budget to all tokens requiring perturbation, Rap-LI embodies a *paradigm shift from uniform to risk-aware, fine-grained privacy protection*. This work first conceptualizes and implements *Risk-Aware* as a privacy protection principle that combines *context awareness* with *user customizability*, thereby aiming to adaptively protect sensitive information in user prompts based on actual needs. Yet, the construction of Rap-LI faces the following challenges: 1) **C1**: How to define the risk levels and privacy budgets of different tokens in the prompt, ensuring that sensitive information is emphasized while reflecting the user's actual personalized needs. 2) **C2**: How to satisfy the token-level LDP constraint when different token pairs (t, t') have varying privacy protection strengths? How to handle the ambiguity in managing the overall privacy budget of the prompt with different privacy budgets for different tokens? 3) **C3**: How to balance privacy and utility while enhancing the protection of high-risk information, especially when the loss of semantics leads to utility degradation?

To address C1, Rap-LI first conducts sensitivity detection on user prompts and assigns different risk labels to each token, which is then presented to the user for fine-grained adjustments based on specific use cases and personal privacy preferences (which also helps mitigate the balance issue in C3). The final labels are mapped to heterogeneous privacy budgets, enabling personalized protection. **For C2**, since standard uniform ϵ-LDP is inadequate under heterogeneous budgets, we present token-level $\epsilon_{(t,t')}$-LDP and sentence-level LDP to provide formal guarantees for risk-adaptive sanitization. **To relieve C3**, Rap-LI constructs a risk-adaptive, bounded candidate token space (selected by semantic similarity) and performs risk-aware sampling (via the exponential mechanism). In addition, we integrate task-aware prompt engineering to improve downstream robustness to perturbations, reducing utility degradation after sanitization.

The main contributions of this paper are summarized below:

- This work empirically examines the privacy protection of existing uniform-budget LDP sanitization in LLM inference, offering new insights that one-size-fits-all ϵ settings struggle to balance utility and privacy with significant risks of sensitive information leakage.
- This work proposes a novel framework (Rap-LI) to achieve risk-aware privacy preservation for LLM inference, mitigating the critical vulnerability of high-sensitivity data leakage in LDP-based prompt protection. Rap-LI is training-free and denoising-free, providing a generalizable and flexible solution for immediate deployment in cloud-based LLM inference.
- This work presents the formulation of token-level $\epsilon_{(t,t')}$-LDP and develops a token sanitization mechanism that enables a risk-aware implementation of such a LDP, along with maintaining provable sentence-level LDP guarantees.
- Extensive experiments across downstream tasks confirm Rap-LI's superior privacy-utility tradeoffs compared to existing methods, particularly in enhancing sensitive information privacy protection.
- We believe that this work could lay the foundation for exploring context-aware and personalized LDP in LLM inference, bridging the gap between theoretical privacy guarantees and practical usability.

II. Related Work

DP-based perturbation methods can be employed to provide provable privacy protection during LLM inference. Table II presents a comparative summary of the most relevant works.

TABLE II: Comparison of privacy-preserving prompt engineering methods for LLM inference and text processing. “Tasks” refer to downstream tasks completed by LM or LLM. NLU denotes Natural Language Understanding; NLG denotes Natural Language Generation.

Method	**Description**	**DP-Based**	**Tasks**	**Risk/Context-aware**
SanText [29]	Token sampling from **complete** vocabulary space for text sanitization	✓(Local DP)	NLU	✗
CusText [30]	Token sampling from **constrained** vocabulary space for text sanitization	✓(Local DP)	NLU	✗
InferDPT [33]	Token sampling from **randomized** vocabulary space with local LLM denoising	✓(Local DP)	NLG	✗
SnD [3]	Text sanitization via **nearest** token selection from **randomized** vocabulary space	✓($d_\mathcal{X}$-privacy)	NLU	✗
DP-FUSION [37]	Blend output distributions to bound the influence of sensitive information	✓(Rényi DP)	NLG	✗
HaS [21]	Anonymize private entities via local models & restore in outputs	✗	NLU & NLG	✗
Kan’s [22]	Filter sensitive info via local LLM & restore in outputs	✗	NLU & NLG	✗
Ours	**Risk-aware** and **customizable token-level** LDP for text sanitization	✓(Local DP)	NLU & NLG	✓

A. DP-Based Privacy Preservation for LLM Inference

Researchers have investigated DP-based privacy protection for demonstration examples in prompts [38]–[41]. However, unlike safeguarding demonstration examples, directly protecting dynamic user prompts, which are closely tied to task completion, presents greater challenges. Most recently, Thareja et al. [37] proposed DP-FUSION, a token-level differentially private inference (DPI) method for LLMs that is particularly well-suited for document privatization scenarios and achieves strong utility-privacy trade-offs. DP-FUSION is primarily applicable to model-side (server) protection or scenarios involving white-box access (e.g., potentially open-source LLMs), where the provider utilizes privatized documents with the LLM to deliver services. On the other hand, methods based on LDP for sanitizing sensitive data on the user side have gained attention. For instance, Mai et al. [3] proposed the SnD framework, which privatizes token representation layers on the user side and trains a denoising module to enhance utility. Tong et al. [33] utilize LDP to generate perturbed prompts and leverage local LLMs for text extraction from perturbed outputs. Our Rap-LI falls under user-side (local) protection and is essential for users who must sanitize data before sending it to third-party, black-box cloud services.

B. DP-Based Text Sanitization in Language Models

DP-based text sanitization mechanisms anonymize data before analysis or input into language model (LM) servers by injecting noise into different levels of text representation. This work specifically focuses on token-level perturbations to explore fine-grained control. Feyisetan et al. [31] proposed a relaxation of LDP ($d_\mathcal{X}$-privacy), which adds noise to each token’s embedding and replaces it with the nearest token in the embedding space. However, this approach suffers from the “curse of dimensionality” due to the high dimensionality of token embeddings. Therefore, similarity-based methods have emerged, utilizing the exponential mechanism for private selection from a token output space, aiming to enhance utility. Yue et al. [29] propose SanText satisfying metric-LDP through distance measurement, yet it retains the entire vocabulary as candidate outputs, limiting performance gains. Chen et al. [30] introduced CusText, which reduces the size of the token output space and employs the exponential mechanism to sample outputs. Although it enhances utility, the fixed-size output space is vulnerable to embedding inversion attacks [42]. Both SanText and CusText are designed for privacy protection in LM training and testing datasets, requiring models to be trained on sanitized text. To perturb the instantly uploaded prompts in LLMs, Tong et al. [33] proposed InferDPT, which samples from dynamic token candidate spaces and denoises them using local LLMs, thus extending these principles to inference scenarios.

Unlike the aforementioned uniform-budget LDP methods, Rap-LI introduces a risk-aware paradigm that dynamically allocates privacy budgets based on token sensitivity and user preferences. By formalizing token-level $\epsilon_{(t,t')}$-LDP and sentence-level $d \cdot \epsilon_S$-LDP, Rap-LI provides rigorous privacy guarantees for heterogeneous budget allocation, thereby addressing the limitations of standard ϵ-LDP. Furthermore, its risk-aware sanitization mechanism—encompassing risk identification, risk-adaptive token space, and risk-aware similarity score computation, and risk-aware token sampling—highlights fundamental distinctions from related LDP approaches.

C. Non-LDP Privacy Preservation for LLM Inference

This work can be regarded as part of the emerging paradigm of *Privacy-Preserving Prompt Engineering*, systematically summarized in the survey by Edemacu and Wu [19], which also includes broader non-LDP protection methods. These methods typically employ heuristic rules or local models to sanitize text. For instance, Chen et al. [21] proposed the Hide and Seek (HaS) framework, which uses a local model to anonymize private entities in prompts and de-anonymize them in LLM responses. Kan et al. [22] introduced a text sanitization framework that filters sensitive information (using a local LLM) before transmission and restores it upon receiving the response. While these methods offer practical privacy protection, they often lack rigorous theoretical privacy guarantees compared to DP-based approaches, and their performance heavily relies on the capabilities of the local models used for sanitization and restoration. Other broader methods not discussed in this paper are inapplicable to our scenario, as they are either limited to open-source LLMs, focus on protecting data privacy during the training process, or aim to safeguard demonstration examples in In-Context Learning. Further details can be found in the survey [19].

III. Preliminary

A. LLM Inference

Let $\mathcal{LI}$ denote an LLM with an inference function $\mathcal{LI}: \mathcal{P} \rightarrow \mathcal{R}$, where $\mathcal{P}$ represents the prompt space and $\mathcal{R}$ represents the response space. Following standard prompt engineering practices, a prompt $P \in \mathcal{P}$ can be decomposed into $P = \langle P_{\text{sys}} \| P_{\text{usr}} \rangle$, where P_{sys} provides general instructions to guide the model's behavior, specifying task constraints or response formats, and P_{usr} contains dynamic user input, often including task-specific or sensitive information. For a given prompt input P, the output of LLM inference is represented as $R = \mathcal{LI}(P_{\text{sys}} \| P_{\text{usr}})$, where $R \in \mathcal{R}$ is the generated response.

B. Local Differential Privacy for Text Sanitization

Differential privacy (DP) [17], [27] serves as the *de facto* standard for sensitive data protection. DP provides provable privacy guarantees by adding noise to data. In practical applications, DP typically relies on a centralized trusted third party to perform data perturbation. In contrast, local differential privacy (LDP) [32] enables individual users to sanitize their data locally before sharing.

Definition III.1 (ϵ-Local Differential Privacy (ϵ-LDP)). Let $\mathcal{M}$: $\mathcal{X} \to \mathcal{Y}$ be a randomized mechanism mapping an input space $\mathcal{X}$ to an output space $\mathcal{Y}$. $\mathcal{M}$ satisfies *ϵ-local differential privacy* (ϵ-LDP) if, for any $x, x' \in \mathcal{X}$ and any $y \in \mathcal{Y}$, it holds that $\Pr[\mathcal{M}(x) = y] \leq e^{\epsilon}\Pr[\mathcal{M}(x') = y]$. x, x' are called *neighbors*, denoted as $x \sim x'$. The parameter $\epsilon \geq 0$ denotes the *privacy budget*, where smaller ϵ indicates stronger privacy protection, while larger ϵ provides weaker privacy guarantees.

LDP provides a strategy for text perturbation tasks, such as sanitizing LLM prompts, by allowing users to locally sanitize their data before sending it to the LLM server [19]. LDP ensures that from the same output y, it cannot be determined whether the input is x or x'. Recent studies employ the *exponential mechanism* for private token selection from the output space $\mathcal{Y}$, using similarity metrics to guide the replacement process.

Definition III.2 (Exponential Mechanism (EM)). Given input $x \in \mathcal{X}$ and output set $\mathcal{Y}$, the mechanism $\mathcal{M}^u_\epsilon$ preserves ϵ-LDP if the output $y \in \mathcal{Y}$ is selected according to:

$$\Pr\left[\mathcal{M}^u_\epsilon(x) = y\right] \propto \exp\left(\frac{\epsilon u(x,y)}{2\Delta(u)}\right), \tag{1}$$

where $\propto$ denotes the normalizing factor. The *scoring function* $u(x, y)$ determines the likelihood of selecting y given x, with larger $u(x, y)$ values indicating higher probabilities. The *sensitivity* $\Delta(u)$ of u is defined as $\Delta(u) = \max_{x \sim x', y \in \mathcal{Y}} |u(x, y) - u(x', y)|$, and $u(x, y)$ must satisfy a finite $\Delta(u)$.

Based on prior studies [29], [30], [33], token sanitization mechanisms can be formalized using the exponential mechanism as follows.

Definition III.3 (Token Sanitization via the Exponential Mechanism). Let t and τ represent the tokens x and y in Definitions III.1 and III.2, respectively, to specifically denote the token sanitization scenario. Given a token $t \in \mathcal{X}$, token sanitization aims to replace t with a sanitized token $\tau \in \mathcal{Y}$. Here, $\mathcal{Y}$ specifically refers to the set of tokens obtained from vocabulary tokenization, known as the tokenizer vocabulary [43]. This process first determines t's output space $\mathcal{Y}' \subseteq \mathcal{Y}$, consisting of tokens whose embeddings are close to the embedding of t based on a predefined similarity metric (including t itself). Then, according to Eq. 1, the new token τ is sampled from $\mathcal{Y}'$ to sanitize the input token t.

Lemma III.1. *The token sanitization via the exponential mechanism satisfies ϵ-LDP. Different variants of ϵ are possible for different scoring functions.*

The proof can be found in [29], [30], [33]. In the *SanText* [29] mechanism, $\mathcal{Y}'$ is the entire vocabulary, while in *CusText* [30], the size of $\mathcal{Y}'$ is limited to a predefined K to improve utility. In *RanText* [33], the size of $\mathcal{Y}'$ dynamically changes based on Laplace noise added to the token's embedding, offering resistance to embedding inversion attacks.

IV. Threat Model

Consider a set of adversaries $\mathcal{A} = \{\mathcal{A}_{\text{srv}}, \mathcal{A}_{\text{api}}\}$ that may threaten the privacy of user prompts. Adversary $\mathcal{A}_{\text{srv}}$, acting as the service provider for LLM inference, has full access to P_{usr} and R. $\mathcal{A}_{\text{api}}$ has access only to the LLM inference service and operates under limited permissions, observing only $\mathcal{R}$. However, $\mathcal{A}_{\text{api}}$ can exploit advanced attacks to infer information about P_{usr} and R. For instance, it can craft inputs P^*_{sys} and P^*_{usr} to obtain responses R^* that contain partial or complete tokens from P_{usr} and R.

In general, the sensitive information in R is derived from P_{usr}. Therefore, privacy protection should be applied at the source, P_{usr}, to resist attacks. Specifically, P_{usr} can be further represented as a sequence of tokens, i.e., $\langle t_1, t_2, \ldots, t_n \rangle$, which we denote as S for simplicity, where $t_i \in \mathcal{Y}$ and $\mathcal{Y}$ denotes the vocabulary of tokens. Even if the original input S is transformed into a perturbed sequence $\tilde{S} = \langle \tau_1, \tau_2, \ldots, \tau_n \rangle$ via noise-based protection mechanisms (with τ_i as perturbed t_i), $\mathcal{A}$ can still compromise the user's privacy through the following types of attacks. These attacks are visualized in Fig. 1 and summarized in Table III, along with the default implementation details adopted in this paper.

A. Potential Attacks on Prompts

Definition IV.1 (Mask Token Inference Attack (MaskInf)). The mask token inference attack [29], [30], [33] focuses on predicting masked tokens in a sanitized prompt $\tilde{S}$ by exploiting the language model's contextual understanding, and can be easily implemented using the BERT model. For each token $\tau_i \in \tilde{S}$, $\mathcal{A}$ constructs a masked version: $\tilde{S}^i_{\text{mask}} = \langle \tau_1, \ldots, [\text{MASK}], \ldots, \tau_n \rangle$, and predicts the masked token $t^*_i = \arg\max_{t \in \mathcal{Y}} \Pr[t | \tilde{S}^i_{\text{mask}}]$ using the model. The attack success rate on S is defined as: $r_{\text{MTIA}} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(t^*_i = t_i)$, where $\mathbb{I}(\cdot)$ is the indicator function, which takes the value 1 when $t^*_i = t_i$ and 0 otherwise. The relaxed version of this attack considers the attack successful if t^*_i is among the top m most probable candidate tokens.

Definition IV.2 (Embedding Inversion Attack (EmbInv)). The embedding inversion attack [3], [33], [42] attempts to reconstruct original tokens by analyzing the embeddings of sanitized tokens. Suppose $\mathcal{A}$ has access to the embedding space $\mathbb{R}^\theta$ used for token perturbations, where θ is the embedding dimension. Let $\mathbf{w}_{\tau_i} \in R^\theta$ denote the embedding of token τ_i. $\mathcal{A}$ can infer

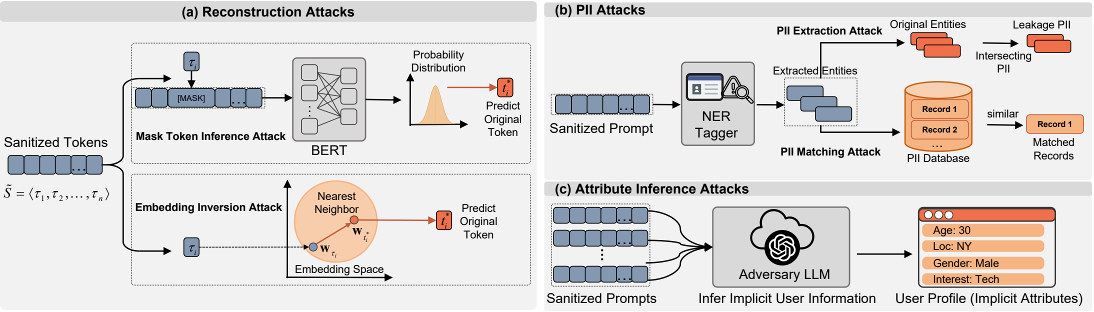


Fig. 1: Illustration of potential attacks on sanitized prompts.

the original token t_i's embedding $\mathbf{w}_{t_i}$ via a nearest neighbor search, i.e., $\mathbf{w}_{t_i^*} = \arg\min_{k \in \mathcal{Y}} d(\mathbf{w}_k, \mathbf{w}_{\tau_i})$ or $\mathbf{w}_{t_i^*} = \arg\max_{k \in \mathcal{Y}} \cos(\mathbf{w}_k, \mathbf{w}_{\tau_i})$, where $d(\cdot, \cdot)$ is the Euclidean distance and $\cos(\cdot, \cdot)$ is the cosine similarity. The attack success rate r_{EIA} is defined similarly to r_{MTIA}.

Existing studies include every inferred or reversed token in their evaluation of the two attacks. However, these results often contain numerous meaningless stop words or punctuation marks, such as *“is,“ “the,” “of,”* and *“-”*. In our implementation, such meaningless tokens are filtered out to obtain more meaningful statistical results.

Both attacks above treat each token uniformly, while existing privacy metrics overlook the measurement of sensitive information leakage. In practice, however, adversaries are more interested in inferring sensitive information. Therefore, **this work formalizes the *PII Extraction Attack*, *PII Matching Attack*, and *Personal Attribute Inference Attack* to further evaluate the effectiveness of prompt sanitization methods in protecting sensitive information**. The definitions are as follows.

Definition IV.3 (PII Extraction Attack (PII_Ext)). Define the PII extraction function as $\mathcal{E} : \mathcal{S} \to \mathcal{I}$. Given the original prompt S and the privacy-protected sanitized prompt $\tilde{S}$, $I_{\text{orig}} = \mathcal{E}(S)$ is the set of true PII tokens extracted from S, and $I_{\text{san}} = \mathcal{E}(\tilde{S})$ is the set of PII tokens extracted from $\tilde{S}$. The leaked PII set is given by the intersection $I_{\text{leak}} = I_{\text{orig}} \cap I_{\text{san}}$. The PII leakage rate is defined as $\lambda_{\text{PII}} = \frac{|I_{\text{leak}}|}{|I_{\text{orig}}|}$.

Consider an adversary with access to an external PII database $\mathcal{D}_{\text{PII}}$, collected from public sources (e.g., social media, data breaches). The adversary attempts to match the extracted perturbed PII I_{san} to entries in $\mathcal{D}_{\text{PII}}$, aiming to link it to a real-world entity or reveal the original user data S. The definition of the *PII Matching Attack* is provided below.

Definition IV.4 (PII Matching Attack (PII_Mat)). Let $I_{\text{san}} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_\eta\}$ be the set of perturbed PII tokens extracted from the perturbed user prompt, where each token is represented by an embedding vector $\mathbf{w}_k \in \mathbb{R}^\theta$. Let $\mathcal{D}_{\text{PII}} = \{I^{(1)}, I^{(2)}, \ldots, I^{(M)}\}$ be the PII database, where each record $I^{(j)} = \{\mathbf{x}_1^{(j)}, \mathbf{x}_2^{(j)}, \ldots, \mathbf{x}_{\mu_j}^{(j)}\}$ consists of token embeddings $\mathbf{x}_l^{(j)} \in \mathbb{R}^d$. For each record $I^{(j)}$, define the number of matched tokens as:

$$M^{(j)} = \left| \left\{ \mathbf{x}_l^{(j)} \mid \exists \mathbf{w}_k \in I_{\text{san}},\ sim(\mathbf{w}_k, \mathbf{x}_l^{(j)}) \geq \kappa \right\} \right|, \tag{2}$$

where $sim(\cdot, \cdot)$ is a similarity function (e.g., cosine similarity), and κ is the similarity threshold. A record $I^{(j)}$ is considered successfully matched if the proportion of matched tokens exceeds a predefined ratio threshold δ, i.e., $\frac{M^{(j)}}{|I^{(j)}|} \geq \delta$. The *PII matching attack* is considered successful if the adversary successfully matches the original PII set I_{orig} in the database, i.e., there exists $I^{(j)} = I_{\text{orig}}$ such that $\frac{M^{(j)}}{|I^{(j)}|} \geq \delta$.

Definition IV.5 (Personal Attribute Inference Attack (AttInf)). The personal attribute inference attack [13], [44] focuses on identifying personal attributes implicitly embedded in a user's text. Specifically, we assume a user interacts with an LLM through a set of sanitized texts $\{\tilde{S}_i\}_{i=1}^n$, which may implicitly reveal certain personal attributes such as `Age`, `Sex`, or `Location`. These attributes and their values are denoted as a set $\{(a_j, v_j)\}_{j=1}^\zeta$, where a_j represents the j-th attribute and v_j its corresponding value. An adversary $\mathcal{A}_{\text{api}}$, which accesses the LLM inference service $\mathcal{LI}$ via API, constructs a specialized prompt $P^*_{\mathcal{A}_{\text{api}}}$ to infer or extract the user's attribute–value pairs, thus producing $\{(\hat{a}_j, \hat{v}_j)\}_{j=1}^\zeta$. If $\mathcal{A}_{\text{api}}$ succeeds in matching $(\hat{a}_j, \hat{v}_j)$ to the ground-truth set (a_j, v_j) implied by $\{\tilde{S}_i\}_{i=1}^n$, the attack is valid.

Compared to attacks that only extract explicit symbolic information (e.g., names or ID numbers), personal attribute inference focuses on exploiting the language understanding and reasoning capabilities of powerful LLMs to unveil implicitly conveyed personal attributes. These attributes are often derived from writing style or contextual cues rather than explicit mentions. Hence, this attack achieves stealth and scalability without requiring explicit identifiers.

B. Limitations of Existing Countermeasures

W1: Why do existing LDP-based text sanitization methods inherently suffer from utility–privacy trade-offs?

Current LDP-based text sanitization applies **uniform perturbation to each token** without distinction, using **identical sampling operations**. Let $S = \langle t_1, t_2, \ldots, t_n \rangle$ represent a sentence (prompt) to be sanitized. For each token t_i, denote $\Pr[\mathcal{M}_\epsilon^u(t_i) \sim t_i]$ as Pr_{keep}, indicating the probability that the sanitized output is equivalent to t_i after sanitization according to

TABLE III: Attack background, objectives, and implementation details.

Attack	Knowledge → Goal	Implementation Details
MaskInf	Sanitized tokens → Reconstruct original tokens by masking each token and predicting the masked position.	The top-m most probable candidate tokens are predicted; m=1 and m=5 are used in experiments. All methods show higher attack success rates at m=5; we report only m=1 results for brevity.
EmbInv	Embedding space for perturbation & Sanitized tokens → Retrieve the original tokens via nearest-neighbour search in the embedding space.	For every sanitized token, the m closest embeddings ($m \in \{1, 5\}$) are queried and ranked by cosine similarity. All methods show higher attack success rates at m=5; we report only m=1 results for brevity.
PII_Ext	Sanitized tokens → Extract any explicit PII that remains in the sequence.	No additional parameters or background knowledge are required.
PII_Mat	Sanitized tokens & External PII database (e.g., all user profiles or records) → Link leaked PII to a real-world profile or recover the original user data.	Similarity threshold for embedding vectors is set to $\kappa = 0.7$; a match is accepted when more than $\delta = 0.6$ of tokens exceed this threshold.
AttInf	Sanitized tokens → Infer latent personal attributes (e.g., location, age).	Follows the official SynthPAI implementation [44], using GPT-4-Turbo for attribute inference and replacing the original decider with "model" during evaluation.

Definition III.3, implying weaker privacy but higher utility. The privacy objective of token sanitization is to sample an equivalent token with the lowest possible probability, i.e., $Pr_{\text{keep}} \leq F_{pri}$, where F_{pri} favors low values. Conversely, the utility objective is to maximize Pr_{keep}, i.e., $Pr_{\text{keep}} \geq F_{uti}$, where F_{uti} favors high values. According to Eq. 1, Pr_{keep} monotonically increases with ϵ, resulting theoretically in $F_{uti} \leq F_{pri}$. However, practical applications often demand higher F_{uti} (for better utility) and lower F_{pri} (for stronger privacy), creating an inherent contradiction. Hence, the utility–privacy trade-off challenge (**W1**) arises: Increasing ϵ elevates Pr_{keep}, thus improving utility but weakening privacy. Conversely, decreasing ϵ strengthens privacy but degrades utility. *When identical ϵ values are uniformly applied across all tokens, this challenge becomes inherently difficult to alleviate.*

W2: Why can they not always mitigate sensitive information leakage in LLM inference?

To justify **W2**, assume that the sentence S contains h sensitive tokens. The probability of at least one sensitive word leaking is $Pr_{\text{anyleak}} = 1 - \left(1 - Pr_{\text{keep}}\right)^h$. Since each token shares the same ϵ, all tokens have identical Pr_{keep}. Even if Pr_{keep} is reduced to 1%, the sensitive information leakage probability approaches 10%. However, in practical applications, Pr_{keep} is often increased to preserve the semantics of the entire sentence and thus enhance utility. For instance, Pr_{keep} will be increased to 5%, 10% or even 50%, which corresponds to leakage probabilities of sensitive information reaching 40%, 65% or 99.9%. *Fundamentally, this issue arises because every token is sanitized with the same probability, regardless of its sensitivity.*

Simply put, in existing solutions, if one aims to ensure sensitive information is "rarely retained", ϵ must be set very small. However, this also means common words are "rarely retained", thereby reducing utility; and vice versa. The aforementioned analysis can be empirically verified through various metrics observed in Fig. 3-6, which reflect changes with varying ϵ. In addition, although SanText+ [29] and CusText+ [30] suggest that low-frequency or stop words can be exempt from LDP processing, this approach primarily aims at utility improvement and only performs coarse-grained filtering of low-risk tokens. Consequently, potential privacy issues remain unresolved.

V. The Proposed Rap-LI

In this section, we present Rap-LI, a risk-aware privacy preservation framework for LLM inference (Fig. 2). The *risk-aware* mechanism is implemented through the collaboration of automatic risk identification and personalized adjustment. After automatically detecting privacy risks, it allows users to refine the default annotations according to their preferences, thereby providing plug-and-play yet customizable privacy protection.

A. Risk Identification and Personalized Labeling

To achieve targeted protection for sensitive information in user input S, we first perform risk identification and classification. Risk identification is typically based on Named Entity Recognition (NER) [45], which can be implemented using popular tools like Flair, Presidio, and transformer-based language models that adaptable across diverse domains and languages. Using NER taggers, high-risk information (hereafter defaulting to PII) is extracted as $I = \mathcal{E}(S)$.

Next, we assign a risk level to each token in $S = (t_1, t_2, \ldots, t_n)$. For a token t_i: If t_i belongs to the extracted PII set I (e.g., `[name]`), it is labeled as high risk and added to the set T_{hs}. If t_i is a stop word, special word (e.g., "the", "of", "`[UNK]`"), subwords with suffixes like "##in", or punctuation, it is labeled as low or no risk and added to T_{ls}. Otherwise, t_i is classified as medium risk (e.g., "Team," "Human") and added to T_{ms}. In practical deployments, users can adjust token-level privacy risks locally to meet compliance requirements or personal preferences (see Sec. VI-D). For instance, they may elevate certain business-related terms to high-risk status or relax protection for some common nouns. This risk-level adjustment offers an intuitive way to express privacy needs and therefore proves more user-friendly than asking users to set LDP budget parameters [35].

We then map the privacy risk level of each token t_i to a corresponding privacy budget $\hat{\epsilon}_{t_i}$: when $t_i \in T_{\text{hs}}$ or T_{ms}, $\hat{\epsilon}_{t_i} = \epsilon_{\text{max}} - (r_{t_i} - 1) \cdot \frac{\epsilon_{\text{max}} - \epsilon_{\text{min}}}{L_r}$; when $t_i \in T_{\text{ls}}$, $\hat{\epsilon}_{t_i} = \infty$. Here ϵ_{max} and ϵ_{min} denote maximum/minimum budgets, r_{t_i} represents the risk rank, and L_r denotes the total number of risk levels. A smaller $\hat{\epsilon}_{t_i}$ indicates a stronger privacy preference.

Further, we consider the overall privacy budget ϵ_S for the sentence $S = (t_1, t_2, \ldots, t_n)$, which is defined as $\epsilon_S = \frac{1}{n} \sum_{i=1}^{n} \hat{\epsilon}_{t_i}$. The final privacy budget for each token t_i in Rap-LI is determined as $\epsilon_{t_i} = \min(\hat{\epsilon}_{t_i}, \epsilon_S)$.

After these steps (appear in Algorithm 1 of Appendix A), each token is assigned a privacy budget. To balance privacy and utility, T_{hs} and T_{ms} tokens undergo token-level adaptive LDP protection (Sec. V-B), while tokens in T_{ls} remain unchanged.

B. Risk-Aware LDP for Text Sanitization

Definition V.1 (Token-Level $\epsilon_{(t,t')}$-LDP). Let a set $T \subseteq \mathcal{X}$ contain tokens that share the same sanitized token set $\Gamma \subseteq \mathcal{Y}$. For any pair of neighbors $t \sim t' \in T$ and for any $\tau \in \Gamma$,

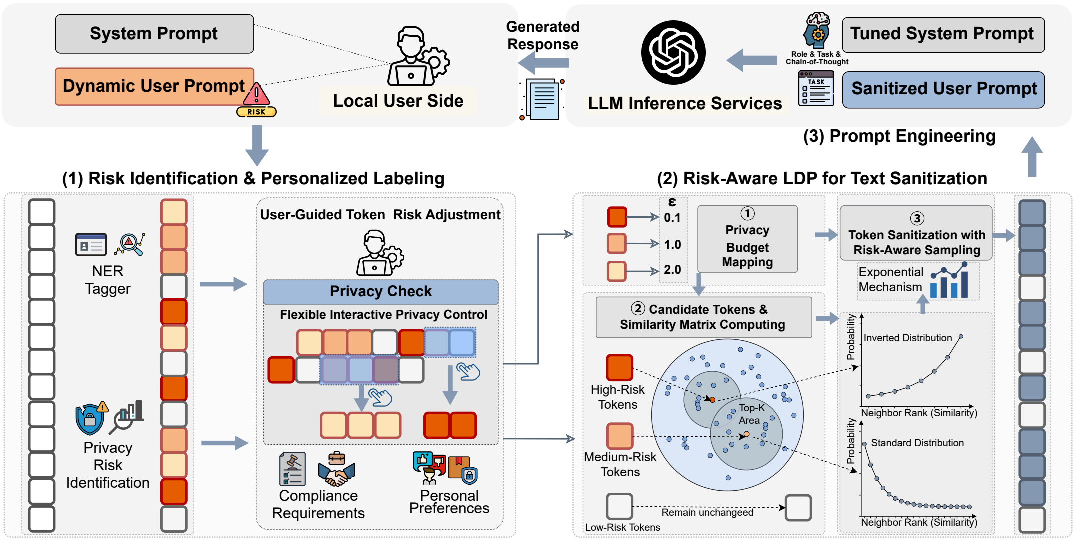


Fig. 2: Overview of Rap-LI. It first performs risk identification and personalized labeling on dynamic user prompts, then applies risk-aware LDP for text sanitization, followed by prompt engineering to preserve LLM inference utility.

a sampling mechanism $\mathcal{M}$ satisfies token-level $\epsilon_{(t,t')}$-LDP if $\Pr[\mathcal{M}^u_\epsilon(t) = \tau] \le e^{\epsilon_{(t,t')}} \Pr[\mathcal{M}^u_\epsilon(t') = \tau]$. Here, ϵ_t and $\epsilon_{t'}$ are the privacy budgets of t and t', respectively, with $\epsilon_t, \epsilon_{t'} \ge 0$. $e^{\epsilon_{(t,t')}}$ is a function of ϵ_t and $\epsilon_{t'}$.

Recalling Definition III.3, token sanitization via the exponential mechanism involves two steps: (1) determining the output set for each input token from the tokenizer vocabulary based on similarity metrics, and (2) sampling a sanitized token from the corresponding candidate set for each token.

*1) **Privacy-Adaptive Token Space** K:* Rap-LI incorporates privacy risk into the calculation of K. Intuitively, when the privacy budget is low, a larger token candidate space K is required to provide stronger masking for sensitive content. To flexibly control the decay rate of K, we adopt an exponential decay strategy [46]. Given a base value K_{base}, $K(\epsilon_i)$ is calculated as $K(\epsilon_i) = K_{\text{base}} + \lfloor A/\epsilon_i^p \rceil$, where A and p control the initial size of K and the decay rate. For $\epsilon_i < 1$, K is large, while for $\epsilon_i > 1$, K rapidly converges, naturally aligning with the privacy implications of ϵ-LDP [27], [47]. This privacy-adaptive token space construction applies universally to all tokens requiring protection, including both high-risk and medium-risk tokens. Consequently, each token $t_i \in T_{\text{hs}} \cup T_{\text{ms}}$ is assigned a corresponding candidate token set of size $K(\epsilon_i)$, which adapts to its privacy budget ϵ_i.

Nevertheless, when the output space $\mathcal{Y}'$ includes tokens equivalent to the original token t (e.g., tokens differing only in case or plurality), the exponential mechanism tends to select such equivalent tokens with high probability. This issue persists when K is excessively large, as shown in Sec. VI-B. Consequently, high-risk PII tokens are still likely to be retained, which increases their vulnerability to extraction or matching attacks. *This limitation is a common drawback of existing text sanitization methods.* To address this issue, Rap-LI further obfuscates high-risk tokens using risk-aware token sampling.

*2) **Risk-Aware Token Sampling**:* To ensure token-level $\epsilon_{(t,t')}$-LDP under the exponential mechanism (Definition III.2), a scoring function $u(t, \tau)$ is defined to reflect both the text sanitization task and the bounded sensitivity $\Delta(u)$. For an input token $t \in T$ (i.e., $t_i \in T_{\text{hs}} \cup T_{\text{ms}}$), similarity metrics (e.g., Euclidean distance $d(\mathbf{w}_t, \mathbf{w}_\tau)$ or cosine similarity $\cos(\mathbf{w}_t, \mathbf{w}_\tau)$) are used to calculate scores for each token $\tau \in \Gamma$ in the output set. Then,

$$u(t, \tau) \propto \cos(\mathbf{w}_t, \mathbf{w}_\tau) \text{ or } u(t, \tau) \propto -d(\mathbf{w}_t, \mathbf{w}_\tau), \quad (3)$$

where $\mathbf{w}_t$ and $\mathbf{w}_\tau$ are the embeddings of t and τ, respectively. A larger cosine similarity or a smaller Euclidean distance indicates stronger relevance between tokens. Scores can be normalized using min-max normalization, such as:

$$u(t, \tau) = -\frac{d(\mathbf{w}_t, \mathbf{w}_\tau) - \min_{\tau' \in \Gamma} d(\mathbf{w}_t, \mathbf{w}_{\tau'})}{\max_{\tau' \in \Gamma} d(\mathbf{w}_t, \mathbf{w}_{\tau'}) - \min_{\tau' \in \Gamma} d(\mathbf{w}_t, \mathbf{w}_{\tau'})}. \quad (4)$$

Thus, $\Delta(u) = \max_{t \sim t', \tau \in \Gamma} |u(t, \tau) - u(t', \tau)|$ has an upper bound of 1. Let $\mathbf{U} \in \mathbb{R}^{|T| \times |\Gamma|}$ represent the score matrix, where $u(t_i, \tau_j^{(i)})$ denotes the score between input token t_i and candidate token $\tau_j^{(i)}$. Each row is sorted in descending order of scores. For high-risk tokens $t_i \in T_{\text{hs}}$, the similarity-based scoring introduces bias, as the highest-scoring tokens often include equivalent tokens of t_i (i.e., tokens with minor transformations, even including t_i itself). Worse still, due to score normalization, even when a lower ϵ_{t_i} is personalized for t_i, the probability of sampling equivalent tokens via Eq. 1 remains highest, thereby increasing the exposure risk of t_i. To mitigate this, we apply a score-reverse operation to the rows of $\mathbf{U}$ corresponding to high-risk tokens. For each $j \in [|\Gamma|]$, the j-th column score is swapped with the $(|\Gamma| - j + 1)$-th column score. The adjusted score matrix $\mathbf{U}^* \in \mathbb{R}^{|T| \times |\Gamma|}$ is defined as:

$$u^*(t_i, \tau_j^{(i)}) = \begin{cases} u(t_i, \tau_{|\Gamma|-j+1}^{(i)}) & \text{if } t_i \in T_{\text{hs}} \\ u(t_i, \tau_j^{(i)}) & \text{otherwise} \end{cases}. \quad (5)$$

After inversion, the scores for equivalent tokens are significantly reduced for $t_i \in T_{\text{hs}}$. A smaller K (e.g., K_{base}) can then effectively conceal high-risk tokens without sacrificing utility. Notably, unlike high-risk tokens, medium-risk tokens $t_i \in T_{\text{ms}}$ do not undergo the score-reverse operation. This design rests on two rationales: (1) medium-risk tokens typically do not contain PII information; therefore, preserving semantic similarity through higher selection probabilities for semantically close tokens maintains better utility; (2) the privacy-adaptive token space $K(\epsilon_i)$ ensures adequate protection by expanding the candidate pool according to the assigned privacy budget. Furthermore, for semantic-sensitive tasks, since not all medium-risk tokens (belonging to non-PII tokens) require perturbation in practice, selectively perturbing medium-risk tokens with probability $pr < 1$ (otherwise they would be simply preserved like low-risk tokens) can improve utility. The ablation study in Appendix F empirically validates the effectiveness of the above design.

Lemma V.1. *The sensitivity of the adjusted scoring function is 1, i.e.,* $\Delta(u^*) = 1$.

Proof. By Eq. 5, for $\forall t \sim t' \in T$ and $\forall \tau \in \Gamma$, $u^*(t,\tau) = u(t,\tau')$, where $\tau' \in \Gamma$ and $u(t,\tau')$ is one of the elements in the i-th row of matrix $\mathbf{U}$. Thus,

$$\begin{aligned}\Delta(u^*) &= \max_{t\sim t', \tau\in\Gamma} |u^*(t,\tau) - u^*(t',\tau)| \\ &= \max_{t\sim t', \tau', \tau''\in\Gamma} |u(t,\tau') - u(t',\tau'')|, \end{aligned} \tag{6}$$

where $\tau', \tau'' \in \Gamma$. Although τ' and τ'' may differ, both $u(t,\tau')$ and $u(t',\tau'')$ are elements of matrix $\mathbf{U}$, which has already been normalized. Therefore, $\Delta(u^*) = 1$. Intuitively, reordering the scores within certain rows of $\mathbf{U}$ does not alter the overall range of the scores. Hence, $\Delta(u^*)$ has the same upper bound of 1 as $\Delta(u)$. □

Finally, the mechanism $\mathcal{M}^{u^*}_{\epsilon_t}$ samples τ for $t \in T$ from Γ:

$$\Pr\big[\mathcal{M}^{u^*}_{\epsilon_t}(t) = \tau\big] \propto \exp\left(\frac{\epsilon_t u^*(t,\tau)}{2\Delta(u^*)}\right), \tag{7}$$

where the normalizing factor is $1/\sum_{\tau'\in\Gamma} \exp\left(\frac{\epsilon_t u^*(t,\tau')}{2\Delta(u^*)}\right)$.

The above token sanitization via the exponential mechanism pseudocode appears in Algorithm 2 of Appendix A.

Theorem V.1. *Given tokens* $t, t' \in T$ *and any sanitized output* $\tau \in \Gamma$, *the token-level risk-aware sanitization mechanism* $\mathcal{M}^{u^*}_{\epsilon_t}$ *satisfies Token-Level* $\epsilon_{(t,t')}$*-LDP, where* $\epsilon_{(t,t')} = (\epsilon_t + \epsilon_{t'})/2$.

Proof. To demonstrate that the token-level adaptive sanitization mechanism $\mathcal{M}$ satisfies token-level $\epsilon_{t,t'}$-LDP, let $\forall t \sim t' \in T, \forall \tau \in \Gamma$. By the definition of $\mathcal{M}^{u^*}_{\epsilon_t}$, we have:

$$\begin{aligned}\Pr\left[\mathcal{M}^{u^*}_{\epsilon_t}(t) = \tau\right] &= \frac{\exp\left(\frac{\epsilon_t u^*(t,\tau)}{2\Delta(u^*)}\right)}{\sum_{\tau'\in\Gamma}\exp\left(\frac{\epsilon_t u^*(t,\tau')}{2\Delta(u^*)}\right)}, \\ \Pr\left[\mathcal{M}^{u^*}_{\epsilon'_t}(t') = \tau\right] &= \frac{\exp\left(\frac{\epsilon'_t u^*(t',\tau)}{2\Delta(u^*)}\right)}{\sum_{\tau'\in\Gamma}\exp\left(\frac{\epsilon'_t u^*(t',\tau')}{2\Delta(u^*)}\right)}. \end{aligned} \tag{8}$$

Then, the privacy ratio can be written as:

$$\frac{\Pr\left[\mathcal{M}^{u^*}_{\epsilon_t}(t) = \tau\right]}{\Pr\left[\mathcal{M}^{u^*}_{\epsilon'_t}(t') = \tau\right]} = A \cdot B, \tag{9}$$

where

$$A = \frac{\exp\left(\frac{\epsilon_t u^*(t,\tau)}{2\Delta(u^*)}\right)}{\exp\left(\frac{\epsilon'_t u^*(t',\tau)}{2\Delta(u^*)}\right)} \text{ and } B = \frac{\sum_{\tau'\in\Gamma}\exp\left(\frac{\epsilon'_t u^*(t',\tau')}{2\Delta(u^*)}\right)}{\sum_{\tau'\in\Gamma}\exp\left(\frac{\epsilon_t u^*(t,\tau')}{2\Delta(u^*)}\right)}. \tag{10}$$

Note that $u^*(t,\tau)$ and $u^*(t',\tau)$ are normalized to $[0,1]$, $\Delta(u^*) = 1$, and $\epsilon_t, \epsilon_{t'} \geq 0$. Consequently, multiplying ϵ_t by a fraction $u^*(t,\tau)$ (which lies between 0 and 1) and subtracting a positive constant results in a value smaller than ϵ_t. Hence, we have:

$$A = \exp\left(\frac{\epsilon_t u^*(t,\tau) - \epsilon_{t'} u^*(t',\tau)}{2\Delta(u^*)}\right) \leq \exp\left(\frac{\epsilon_t}{2}\right). \tag{11}$$

To analyze the upper bound of B, we let the denominator take its minimum value and the numerator take its maximum value:

$$\begin{aligned} B &= \frac{\sum_{\tau'\in\Gamma}\exp\left(\frac{\epsilon_{t'} u^*(t',\tau')}{2}\right)}{\sum_{\tau'\in\Gamma}\exp\left(\frac{\epsilon_t u^*(t,\tau')}{2}\right)} \leq \frac{\sum_{\tau'\in\Gamma}\exp\left(\frac{\epsilon_{t'}\cdot 1}{2}\right)}{\sum_{\tau'\in\Gamma}\exp\left(\frac{\epsilon_t\cdot 0}{2}\right)} \\ &= \frac{|\Gamma|\exp\left(\frac{\epsilon_{t'}}{2}\right)}{|\Gamma|} = \exp\left(\frac{\epsilon_{t'}}{2}\right). \end{aligned} \tag{12}$$

Therefore,

$$\frac{\Pr[\mathcal{M}^{u^*}_{\epsilon_t}(t) = \tau]}{\Pr[\mathcal{M}^{u^*}_{\epsilon_{t'}}(t') = \tau]} = A \cdot B \leq \exp\left(\frac{\epsilon_{t'} + \epsilon_t}{2}\right). \tag{13}$$

This completes the proof. □

*3) **Sentence-Level Privacy Guarantee**:* To further strengthen privacy guarantees beyond the token level, we extend our analysis to the sentence level, ensuring that the entire prompt enjoys rigorous global privacy protection.

Definition V.2 (Sentence-Level $d \cdot \epsilon_S$-LDP). Let $\mathcal{O}$ denote a set of sentences represented as token sequences, where all sentences share the same sanitized sentence set Θ. For any pair of neighboring sentences $S, S' \in \mathcal{O}$, where at most d tokens differ in corresponding positions *excluding low-risk tokens*, and for any sanitized output $\tilde{S} \in \Theta$, the sanitization mechanism $\mathcal{M}$ satisfies Sentence-Level $d \cdot \epsilon_S$-LDP if $\Pr[\mathcal{M}(S) = \tilde{S}] \leq e^{d\cdot\epsilon_S}\Pr[\mathcal{M}(S') = \tilde{S}]$, where $|S| = |S'| = |\tilde{S}|$ and $\epsilon_S \geq 0$ is the overall sentence-level privacy budget. Typically, when S and S' differ in exactly one position, $\mathcal{M}$ provides ϵ_S-LDP.

Each sentence $S = \{t_1, t_2, \ldots, t_n\} \in \mathcal{O}$ is decomposed into n tokens. For each $t_i \in (T_{\text{hs}} \cup T_{\text{ms}})$, the token-level sanitization mechanism $\mathcal{M}^{u^*}_{\epsilon_t}$ in Rap-LI samples a sanitized token τ_i from the candidate set Γ. For low-risk tokens $t_i \in T_{\text{ls}}$, we retain $\tau_i = t_i$. The sanitized sentence is then constructed as $\tilde{S} = \{\tau_1, \tau_2, \ldots, \tau_n\} \in \Theta$. Two token-sequence sentences $S = \{t_1, t_2, \ldots, t_n\}$ and $S' = \{t'_1, t'_2, \ldots, t'_n\}$ are considered neighboring if they differ by at most $|T_{\text{hs}}| + |T_{\text{ms}}|$ tokens, where

TABLE IV: Task and dataset details.

Task	Dataset	Set	Samples	Average PII
Topic Classification	AGNews[1]	Test	7600	8.76
PII Document Classification	PIIDocs[2]	Train	4119	7.37
Chinese Spam Detection	SpamEmail[3]	Test	1000	7.18
Multi-turn Dialogue Summary	SAMSum[4]	Val+Test	1637	6.91
Personal Attribute Inference	SynthPAI[5]	Train	300 (7785)	8.96 (0.35)

the differing tokens belong to either high-risk (T_{hs}) or medium-risk (T_{ms}) categories. Note that low-risk tokens are not replaced by the sanitization mechanism and remain unchanged across neighboring sentences. Therefore, they are excluded from the definition of adjacency and do not affect privacy guarantees.

Theorem V.2. *Given sentences $S = \{t_1, t_2, \ldots, t_n\}$, $S' = \{t'_1, t'_2, \ldots, t'_n\}$ in $\mathcal{O}$ (all low-risk tokens are identical, i.e., $t_i = t'_i$ for all $t_i \in T_{ls}$), and any sanitized output $\tilde{S} = \{\tau_1, \tau_2, \ldots, \tau_n\} \in \Theta$, the sanitization mechanism in Rap-LI provides sentence-level $d \cdot \epsilon_S$-LDP, where d is the number of differing positions between S and S'.*

Proof. See Appendix B. □

C. Prompt Engineering for LLM Inference

Rap-LI is designed as a training-free and denoising-free framework to enhance privacy protection during real-time inference. Existing studies [48] reveal the limited robustness of LLMs to perturbed prompts. To address this limitation, prompt engineering is employed to improve the utility of LLMs on sanitized user prompts without relying on additional denoising processes. Specifically, the prompt engineering strategy includes clearly defining the LLM's role to provide task-specific context, explicitly describing the task while accounting for potential input perturbations, and applying Chain-of-Thought (CoT) prompting techniques [49] to facilitate step-by-step reasoning and improve utility. The specific prompts used in this paper are detailed in Appendix C. The prompt engineering relies on fundamental reasoning capabilities shared by modern instruction-tuned LLMs [50], [51]. The strategies employed in our prompt engineering, such as Role-Play Instructions and CoT, are widely adopted across LLMs with diverse capabilities [52], [53]. Therefore, the proposed prompt engineering strategies are robust across various LLM capabilities.

VI. Experiment and Evaluation

In this section, we empirically evaluate the utility and privacy capabilities of the LDP-based text sanitization mechanism and the proposed Rap-LI framework for LLM inference.

A. Setup

1) Datasets: We evaluate Rap-LI on datasets for Natural Language Understanding (NLU) tasks, including *topic classification*, *PII document classification*, and *Chinese spam detection*, as well as Natural Language Generation (NLG) tasks, specifically *multi-turn dialogue summarization*. The corresponding datasets are *AGNews*, *PIIDocs*, *SpamEmail*, and *SAMSum*, respectively. Moreover, we assess the robustness of our approach against advanced personal attribute inference (*AttInf*) attacks using the *SynthPAI* dataset [44]. Task and dataset details are provided in Table IV, with further descriptions and data samples provided in Appendix D.

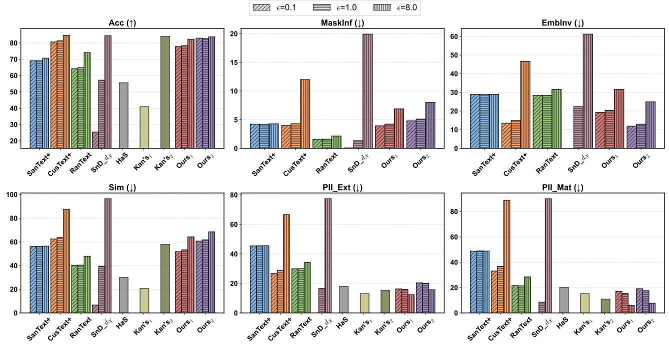


Fig. 3: Performance evaluation on *AGNews*.

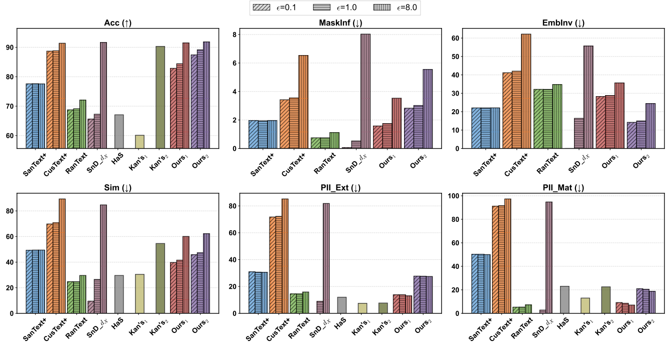


Fig. 4: Performance evaluation on *PIIDocs*.

2) Baselines: SanText+ [29], CusText+ [30], RanText from InferDPT [33], and $d_{\mathcal{X}}$ from SnD [3] are used as baselines to compare utility and privacy. Additionally, we examine the denoising capability described in InferDPT [33] by testing whether performance improves when applying local LLM denoising. Furthermore, non-LDP methods, including HaS [21] and Kan's approach [22] (utilizing `Llama2-7B` for Kan's$_1$ and `Llama3-8B` for Kan's$_2$), are also used for comparison. For the Chinese *SpamEmail* dataset, the above DP-based baselines are not applicable because their vocabularies, tokenizers, and similarity computation are primarily designed for English, leading to incompatible or unreliable Chinese processing. Therefore, we compare *SpamEmail* results only with the non-LDP baselines (HaS and Kan's), which are based on local (large) LM rewriting and support Chinese prompts. To evaluate the impact of prompt engineering in Rap-LI, we adopt prompt styles from [39], [40], [44], [54] for comparison.

3) Evaluation Metrics: Evaluation involves the following metrics: *a) Utility on downstream tasks:* For NLU tasks, utility is assessed using Accuracy, following related work [3], [29], [30]. For NLG tasks, utility is measured by ROUGE-1,

[1] https://huggingface.co/datasets/fancyzhx/ag_news
[2] https://huggingface.co/datasets/gretelai/gretel-pii-masking-en-v1
[3] https://www.jizhi-dataset.top/index/category/detail/26
[4] https://huggingface.co/datasets/Samsung/samsum
[5] https://huggingface.co/datasets/RobinSta/SynthPAI

ROUGE-2, and ROUGE-L [55]–[58]. *b) Privacy protection capability for prompts:* Privacy is evaluated through attack success rates as defined in Sec. IV (sentence-rewriting-based methods HaS and Kan's are excluded from MaskInf and EmbInv evaluations, which target token-level reconstruction). Additionally, we measure the similarity (*Sim*) between the original and sanitized prompts. Lower values on these metrics indicate better privacy protection. *c) Intermediate Metrics:* To provide complementary insights into text quality after sanitization, we measure Perplexity (PPL), BLEU, ROUGE-L, and Token Error Rate (TER). These metrics bridge utility and privacy interpretation: from a utility perspective, lower PPL and TER with higher BLEU and ROUGE-L indicate better semantic preservation; from a privacy perspective, however, higher similarity (high BLEU and ROUGE-L, low TER) suggests increased vulnerability to attacks. *d) Overall Performance Score:* We compute a harmonic mean weighted score combining utility and privacy: $1/(\frac{w}{\text{Utility}} + \frac{1-w}{\text{Privacy}})$, where w is the utility weight (e.g., 0.5 for equal weighting). The harmonic mean penalizes methods with imbalanced performance, thus ensuring both dimensions contribute meaningfully. Particularly, since the ROUGE upper bound for unsanitized texts in SAMSum is not 100, we normalize the ROUGE scores by dividing each method's ROUGE score by that of the unsanitized texts, using this normalized value as the utility score when calculating the overall score. For the privacy component, we subtract attack success rates or similarity values from 1, thereby ensuring higher scores reflect improved privacy. Weights within each category are equally distributed across metrics.

4) Implementations: Privacy risk levels are divided into five categories: tokens in T_{hs} (e.g., PII) are assigned the highest risk, non-sensitive T_{ls} tokens remain unperturbed. The remaining tokens in T_{ms} are *randomly assigned among the other levels to simulate varying user preferences.* We test $\epsilon_{\text{min}} \in \{0.1, 1, 8\}$ and $\epsilon_{\text{max}} = 8$, with sentence-level averages of 3.64, 4.13, and 8. Baseline configurations and hyperparameters are aligned with their original descriptions or experimental settings. Rap-LI's input $\mathcal{X}$ and output $\mathcal{Y}$ sets are derived from either: (1) Ours$_1$: Subword vocabulary from pre-trained models (e.g., `DistilBERT`), where embeddings and tokenization are inherited, and out-of-vocabulary (OOV) tokens are passed through the model to obtain embeddings; (2) Ours$_2$: `GloVe` vocabulary [59] with spaCy tokenization [29], where OOV tokens are retained as raw text [30] (hence unsuitable for *SpamEmail*). For the summarization-oriented NLG task (*SAMSum*), to reduce unnecessary semantic distortion, we perturb medium-risk tokens with probability $pr = 0.3$ rather than perturbing all of them. An ablation study of pr on utility, privacy, and overall score can be found in Appendix F. *Additional implementation details can be found in Appendix E.*

To evaluate LLMs' inference performance on sanitized prompts for downstream tasks, we employ four GPT variants, `DeepSeek-R1` [20] (which claims inference performance comparable to `OpenAI-o1`). When testing local LLM denoising, we use `Llama3-8B`. Prompts are optimized via iterative testing (Appendix C). Implementation details for the various attacks used in privacy evaluations are presented in Table III.

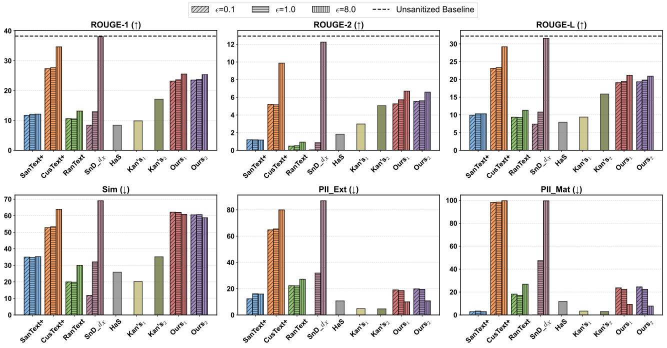


Fig. 5: Performance evaluation on *SAMSum*.

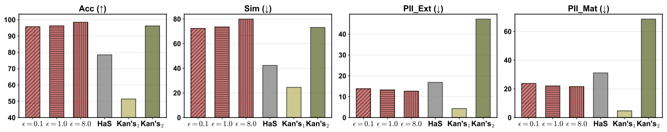


Fig. 6: Performance evaluation on *SpamEmail*.

B. Performance Comparison

The main evaluation results on *AGNews*, *PIIDocs*, *SpamEmail*, and *SAMSum* datasets (with `GPT-3.5-Turbo` by default for LLM inference) , are presented in Figs. 3-6 and Tables V-IX. Key findings are summarized as follows:

Rap-LI achieves a balance between utility and privacy. As shown in experimental results, CusText+ achieves better utility across downstream tasks but exhibits poor privacy performance, especially as ϵ increases. RanText and SnD_$d_\mathcal{X}$ provide strong privacy but at the cost of significantly lower utility. SanText+ shows minimal variation in both utility and privacy with changes in ϵ, as it selects sanitized tokens from the entire vocabulary. HaS tends to achieve stronger privacy protection but incurs noticeable utility loss because it anonymizes entities via an uncontrollable black-box model. Kan's methods demonstrate that the performance of such non-LDP approaches is highly sensitive to local model capacity and post-processing stability (with Kan's$_2$ generally outperforming Kan's$_1$). Particularly, regarding the Chinese *SpamEmail* dataset, Kan's$_1$ frequently fails to complete tasks, whereas Kan's$_2$ remains constrained by the capabilities of the employed local LLMs. Unlike these imbalanced baselines, Rap-LI consistently provides comparable and stable performance across downstream tasks.

Table VII presents text quality metrics that provide complementary insights into the sanitization process. Rap-LI achieves balanced intermediate metrics: moderate PPL (lower than SanText+ and RanText), competitive BLEU and ROUGE-L (preserving semantic content better than high-privacy baselines), and controlled TER. It is worth noting that these intermediate metrics exhibit dataset-dependent scales (e.g., shorter texts in AGNews yield higher PPL variance), but consistent trends emerge across datasets.

Rap-LI improves resistance against sensitive information leakage while maintaining utility. Compared with CusText+, which achieves comparable and acceptable utility, Rap-LI enhances average PII privacy across all datasets by 51.68%

TABLE V: Overall performance scores across different Utility:Privacy weight distributions. The best and second-best results are highlighted in **bold** and underlined, respectively. *Italicized* values for non-LDP methods indicate comparable performance in that scenario.

		AGNews					PIIDocs					SAMSum				
Method	ϵ	0.5:0.5	0.6:0.4	0.7:0.3	0.8:0.2	0.9:0.1	0.5:0.5	0.6:0.4	0.7:0.3	0.8:0.2	0.9:0.1	0.5:0.5	0.6:0.4	0.7:0.3	0.8:0.2	0.9:0.1
SanText+	0.1	66.06	66.64	67.24	67.85	68.46	73.10	73.96	74.83	75.73	76.65	41.23	37.45	34.30	31.64	29.37
	1.0	66.05	66.63	67.23	67.84	68.46	73.15	74.01	74.88	75.78	76.70	42.02	38.29	35.17	32.52	30.24
	8.0	66.72	67.48	68.25	69.04	69.85	73.15	73.99	74.85	75.74	76.64	42.07	38.34	35.21	32.56	30.28
CusText+	0.1	76.07	76.93	77.81	78.72	79.64	59.30	63.51	68.36	74.02	80.69	39.54	43.06	47.25	52.35	**58.69**
	1.0	75.39	76.52	77.67	78.86	80.09	58.83	63.09	68.02	73.78	80.60	39.17	42.76	47.06	52.32	58.92
	8.0	53.99	58.21	63.15	68.99	76.04	47.28	52.34	58.59	66.55	77.02	31.09	35.73	41.98	50.89	64.60
RanText	0.1	69.47	68.36	67.29	66.24	65.23	75.82	74.29	72.82	71.41	70.06	37.53	33.93	30.97	28.47	26.36
	1.0	69.85	68.80	67.78	66.79	65.82	76.06	74.57	73.13	71.75	70.42	37.33	33.72	30.74	28.25	26.14
	8.0	72.59	72.88	73.18	73.48	73.78	76.82	75.82	74.85	73.89	72.97	42.81	39.60	36.83	34.43	32.32
SnD_$d_{\mathcal{X}}$	0.1	40.38	36.12	32.67	29.82	27.43	78.63	75.63	72.85	70.27	67.87	31.72	27.98	25.02	22.63	20.66
	1.0	67.50	65.16	62.97	60.92	59.00	76.61	74.54	72.58	70.72	68.95	40.16	37.46	35.09	33.01	31.16
	8.0	45.32	49.95	55.63	62.76	72.00	50.67	55.64	61.70	69.24	78.89	25.77	30.23	36.56	46.23	62.86
HaS	-	64.64	62.60	60.69	58.88	57.18	72.31	71.20	70.12	69.07	68.05	34.55	30.92	27.97	25.54	23.50
Kan's$_1$	-	54.97	51.44	48.34	45.59	43.13	69.74	67.57	65.54	63.62	61.82	41.18	37.13	33.81	31.03	28.68
Kan's$_2$	-	77.54	78.76	80.03	81.33	*82.68*	79.96	81.84	83.80	85.87	88.03	59.54	56.11	53.05	50.31	47.84
Ours$_1$	0.1	78.08	78.03	77.97	77.91	77.86	82.16	82.30	82.44	82.58	82.72	60.77	59.98	59.21	58.46	57.73
	1.0	78.26	78.28	78.29	78.31	78.32	82.72	83.06	83.39	83.73	84.07	61.88	61.17	60.49	59.81	59.15
	8.0	78.90	79.56	80.23	80.91	81.60	**83.11**	**84.66**	**86.27**	**87.95**	**89.69**	68.37	**67.46**	**66.57**	**65.71**	**64.86**
Ours$_2$	0.1	**79.65**	**80.29**	**80.93**	**81.59**	**82.25**	**82.26**	**83.25**	**84.25**	**85.28**	**86.34**	**61.37**	**60.68**	**60.00**	**59.34**	**58.69**
	1.0	**79.50**	**80.12**	**80.75**	**81.39**	**82.05**	**82.80**	**84.00**	**85.23**	**86.50**	**87.80**	**62.20**	**61.51**	**60.85**	**60.19**	**59.56**
	8.0	**79.13**	**80.01**	**80.91**	**81.82**	**82.76**	80.92	82.89	84.96	87.14	89.43	**68.39**	67.33	66.30	65.30	64.33

TABLE VI: Performance metrics comparison for the *SpamEmail*.

		Text Quality				Weighted Harmonic Mean					Time Cost	
Method	ϵ	PPL	BLEU	ROUGE	TER	0.5:0.5	0.6:0.4	0.7:0.3	0.8:0.2	0.9:0.1	User-side	Inference
HaS	-	132.42	10.26	9.08	89.72	73.94	74.81	75.69	76.60	77.53	0.52	3.30
Kan's$_1$	-	57.45	8.01	8.61	89.05	65.12	61.82	58.84	56.13	53.66	1.39	3.08
Kan's$_2$	-	11.77	24.07	29.22	63.36	53.46	58.68	65.02	72.90	82.96	1.38	3.33
Ours	0.1	154.19	17.55	10.74	72.48	**76.24**	**79.49**	**83.02**	**86.88**	**91.12**	0.015+*1e-05*	3.28
	1.0	126.03	18.52	10.41	49.89	**76.67**	**79.93**	**83.47**	**87.35**	**91.60**	0.014+*1e-05*	3.80
	8.0	22.43	30.74	10.66	41.67	**76.04**	**79.68**	**83.67**	**88.09**	**93.00**	0.014+*1e-05*	3.25

(specifically, 31.57% on AGNews, 67.47% on PIIDocs, and 56.01% on SAMSum). Notably, on the PIIDocs dataset, CusText+ shows high text quality yet remains significantly vulnerable to PII attacks (e.g., 71.74% PII_Ext even under a strict privacy budget of $\epsilon = 0.1$). CusText+ exhibits high privacy risks due to two main factors: (1) whitespace-based tokenization fails to separate special characters from PII (e.g., "\n\nJane"), causing these sequences to be identified as out-of-vocabulary (OOV) tokens; and (2) OOV tokens are directly retained during sanitization. Consequently, CusText+ faces potential privacy exposure. Note that Rap-LI mitigates high-risk PII exposure by reversing the score matrix for PII-related tokens. Consequently, as ϵ increases, the probability of selecting PII tokens decreases (Eq. 7). When ϵ is small, the smoothing effect of the scoring function u^* increases the selection probability for low-probability PII tokens. However, this does not significantly impact privacy, as demonstrated by the robust privacy performance of Rap-LI at $\epsilon = 0.1$.

Rap-LI demonstrates better overall performance. The overall scores are shown in Table V-VI. Utility is given higher weight than privacy, since poor utility significantly undermines practical usability. Our methods achieve superior overall performance across various ratios and privacy budgets (ϵ). Specifically, by averaging over all Utility:Privacy ratios and $\epsilon \in \{0.1, 1, 8\}$, our method consistently demonstrates superiority in average gain: Ours$_1$ achieves gains of 14.45 (*AGNews*), 9.28 (*PIIDocs*), 21.18 (*SAMSum*), and 16.34 (*SpamEmail*), while achieves Ours$_2$ gains of 16.50 (*AGNews*), 10.09 (*PIIDocs*), and 21.47 (*SAMSum*). On the NLG task (*SAMSum*), the overall score is relatively lower for several reasons. First, text generation tasks (e.g., summarization) are more sensitive to token-level perturbations, as key information slots (e.g., who/what/when) must be consistently preserved. Second, ROUGE scores—dependent on n-gram overlap—drop significantly due to lexical mismatches, even if semantics remain similar (e.g., changing "John met Mary" to "He met her"). Nevertheless, the interpretation of the overall score should be distinguished from conventional metrics like accuracy. For example, although CusText+ yields a seemingly low overall score of 64.60 (Utility:Privacy = 0.9:0.1), its utility remains viable: the ROUGE-1 score (34.64) reaches 90.59% of the unsanitized baseline (38.24). Consequently, the lower overall score primarily reflects compromised privacy rather than unusable utility. In practice, method selection should comprehensively weigh utility, privacy, and overall scores against specific application requirements. We mitigate these NLG challenges via ROUGE normalization, probabilistic perturbation of non-PII tokens (Appendix F), and optional local post-processing (Fig. 7(d)).

TABLE VII: Text quality after sanitization (intermediate metrics).

		AGNews				PIIDocs				SAMSum			
Method	ϵ	PPL	BLEU	ROUGE-L	TER	PPL	BLEU	ROUGE-L	TER	PPL	BLEU	ROUGE-L	TER
SanText+	0.1	>1000	20.52	50.40	51.02	658.11	8.71	32.54	81.72	>1000	13.68	42.11	74.77
	1.0	>1000	20.53	50.40	50.97	668.09	8.71	32.59	81.64	>1000	13.69	42.10	74.64
	8.0	>1000	20.61	50.58	50.53	698.98	8.76	32.75	81.22	>1000	13.81	42.15	73.44
CusText+	0.1	544.75	16.39	46.21	53.31	38.69	70.01	87.85	12.16	165.55	39.18	62.56	32.57
	1.0	504.36	17.17	47.67	51.99	44.96	51.40	69.31	26.15	157.53	40.52	63.18	31.83
	8.0	43.26	56.42	78.51	22.06	9.67	76.81	87.14	10.95	34.20	69.76	83.97	14.91
RanText	0.1	>1000	5.62	23.47	82.51	>1000	10.20	11.62	86.83	>1000	5.04	14.57	77.35
	1.0	>1000	5.62	23.17	82.51	>1000	10.20	11.62	86.83	>1000	5.05	14.56	77.35
	8.0	>1000	7.23	28.85	71.17	>1000	11.18	15.09	85.37	>1000	7.18	20.58	72.79
SnD_$d_{\mathcal{X}}$	0.1	>1000	0.06	0.10	99.71	>1000	0.02	0.10	99.88	>1000	0.01	0.22	99.95
	1.0	>1000	3.79	29.01	79.94	>1000	3.94	19.66	92.61	>1000	3.99	28.30	76.98
	8.0	11.49	85.80	100.00	10.67	12.69	63.64	99.99	34.72	9.07	85.14	99.81	9.78
HaS	-	49.36	14.66	26.51	86.04	48.72	11.82	19.58	87.67	41.96	8.81	18.99	90.27
Kan's$_1$	-	67.98	7.96	18.30	86.88	35.24	9.54	15.80	86.89	85.34	0.52	8.28	94.58
Kan's$_2$	-	44.08	6.06	26.50	84.07	19.36	15.70	28.65	64.21	64.35	0.82	10.35	93.01
Ours$_1$	0.1	808.96	10.54	44.29	59.02	209.19	17.71	29.27	64.75	50.89	45.90	73.38	28.75
	1.0	714.03	11.60	45.87	57.86	194.25	18.61	30.75	63.95	47.31	46.30	73.82	28.49
	8.0	181.45	27.78	61.28	44.42	82.14	32.54	47.41	53.42	23.47	52.14	78.20	25.24
Ours$_2$	0.1	507.25	14.67	46.79	53.80	120.61	23.23	37.98	57.02	119.65	53.38	72.03	21.81
	1.0	464.99	15.69	48.15	52.77	112.55	24.16	39.40	56.17	117.69	53.80	72.35	21.59
	8.0	154.64	29.54	61.75	41.06	56.08	38.60	51.96	45.67	73.74	59.22	75.93	18.84

Time Overhead Analysis. To demonstrate the feasibility of Rap-LI for large-scale cloud-based LLM inference, we analyze the time complexity of its key components The theoretical time complexity is summarized in Table VIII. The user-side/local operations (risk detection, privacy budget mapping, and sanitization) exhibit a combined complexity of $O(N^2 \cdot d + S \cdot V \cdot d)$, where the similarity search constitutes the primary computational bottleneck. Yet, GPU acceleration ensures practical efficiency. Empirical time costs in Table VI and IX show that the total user-side overhead of Rap-LI averages ∼0.1s, demonstrating superior or comparable efficiency compared to existing baselines. This overhead is negligible compared to LLM inference time (≈ 2-3s). Nevertheless, non-LDP methods (HaS and Kan's) incur higher overhead due to reliance on local language models for anonymization (0.53s-2.01s) and recovery (0.84s-1.45s). Additionally, Appendix F presents performance and time cost comparisons of HaS and Kan's under settings with and without the local Seek/Recovery module.

C. Diagnostic Experiments

Fig. 7 summarizes the optional utility-improvement strategies (with default ϵ=1.0, AGNews): upgrading the black-box inference LLM, prompt tuning, and local post-processing/denoising with `Llama3-8B`.

1) Comparison across different LLM inference models: **Enhanced LLMs mitigate utility loss after sanitization.** As

TABLE VIII: Time complexity analysis of Rap-LI components.

Module	Component	Complexity
1. Risk Detection	Presidio (Regex + SpaCy) Flair (Transformer NER)	$O(N)$ $O(N^2 \cdot d)$
2. Privacy Mapping	Privacy Budget Mapping	$O(N)$
3. Sanitization	Similarity Computation Candidate Token Sampling	$O(S \cdot V \cdot d)$ $O(S \cdot K)$
Total (User-side)	**Stages 1-3**	$O(N^2 \cdot d + S \cdot V \cdot d)$

N: Input sequence length; d: Embedding dimension; S: Number of sensitive tokens; V: Vocabulary size; K: Number of candidate tokens ($K \ll V$).

TABLE IX: Time (s) comparison across different methods. The upright entries in the user-side phase denote the sanitization cost (e.g., similarity computation and token sampling), while the *italicized* entries indicate our additional time overhead, including risk detection and privacy budget mapping. The *italicized* entries in the inference phase represent the supplementary recovery overhead.

Method	ϵ	**AGNews**		**PIIDocs**		**SAMSum**	
		User-side	**Inference**	**User-side**	**Inference**	**User-side**	**Inference**
Unsanitized	-	-	2.24	-	1.87	-	2.28
SanText+	0.1	0.03	2.81	0.02	2.27	0.05	2.45
	1.0	0.03	3.00	0.02	2.25	0.04	2.43
	8.0	0.03	2.79	0.02	2.26	0.04	2.45
CusText+	0.1	0.04	2.89	0.02	2.12	0.07	2.44
	1.0	0.04	2.79	0.02	2.15	0.06	2.41
	8.0	0.03	2.39	0.02	2.05	0.06	2.37
RanText	0.1	0.08	2.77	0.09	2.10	0.23	2.41
	1.0	0.08	2.79	0.09	2.05	0.23	2.37
	8.0	0.04	2.65	0.04	2.14	0.11	2.41
SnD_$d_{\mathcal{X}}$	0.1	0.24	2.24	0.31	1.82	0.65	2.30
	1.0	0.24	2.75	0.31	2.01	0.86	2.40
	8.0	0.24	2.23	0.31	1.87	1.07	2.29
HaS	-	0.73	2.84	0.56	2.92	0.86	2.81+*0.84*
Kan$_1$	-	1.49	2.49	1.66	2.81	1.82	2.29+*1.45*
Kan$_2$	-	1.08	2.96	1.06	2.85	2.01	2.47+*1.03*
Ours$_1$	0.1	0.08+*0.02*	2.71	0.09+*0.03*	2.12	0.16+*0.03*	2.59
	1.0	0.08+*0.02*	2.73	0.09+*0.04*	2.12	0.17+*0.03*	2.55
	8.0	0.08+*0.02*	2.51	0.09+*0.04*	2.08	0.16+*0.03*	2.43
Ours$_2$	0.1	0.01+*0.02*	2.33	0.01+*0.02*	2.01	0.02+*0.03*	2.60
	1.0	0.01+*0.02*	2.33	0.01+*0.02*	1.98	0.02+*0.03*	2.79
	8.0	0.01+*0.02*	2.29	0.01+*0.02*	1.94	0.02+*0.03*	2.64

shown in Fig. 7(a), using enhanced LLMs such as `GPT-4o`, `GPT-4-Turbo`, and `DeepSeek-R1` (listed in decreasing order of utility) improves utility compared to `GPT-3.5-Turbo` for privacy-preserving inference, although this also increases inference time. Also, the gap between the utility of sanitized and original text decreases. Conversely, `GPT-4o-mini` leads to greater utility degradation and widens the gap. Therefore, employing superior models further reduces the need for additional denoising or training.

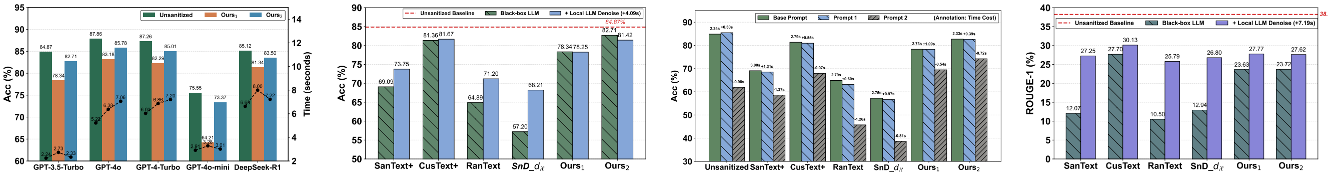
2) Advantages over the "Training and Denoising" Setting: a) Our method demonstrates a clear advantage over training-based approaches such as SnD [3] in terms of time cost. Training SnD on GPT2-XL (1.5B parameters) takes 30.3 hours on a single A6000 GPU. As reported in [3], training time increases substantially with model size; therefore, training on models such as GPT-3.5 (175B) becomes prohibitively expensive. In contrast, our method completes execution within seconds, demonstrating a significant efficiency gain. b) Local LLM denoising improves low-utility methods, with negligible impact on high-utility methods. Following [33]'s idea, we apply local LLM-based denoising using `Llama3-8B` (a stronger model than used in [33]) across all methods. The denoising is guided by the GPT inference process and its results; the specific prompt is provided in Appendix C. Fig. 7(b) shows that local LLM denoising improves utility for low-performing methods but has little or no impact on high-performing methods such as CusText+ and Rap-LI. In some cases, accuracy even decreases. Furthermore, the denoising process introduces an additional latency of approximately 4 seconds. This further validates that our framework requires no denoising and no additional cost to enhance the utility.

3) Impact of Prompt Engineering: **Prompt engineering enhances utility in LLM-based inference tasks.** We compare our tuned system prompt (*Base*) with two alternative prompt styles: *Prompt1*, inspired by the investigator-style, guess-based, and step-by-step reasoning approaches from [44], [54], and *Prompt2*, adopting a concise style as in [39], [40]. The differences among these prompts are detailed in Appendix C. As shown in Fig. 7(c), prompts that incorporate role-setting and chain-of-thought (CoT) reasoning achieve improved performance, thereby demonstrating the effectiveness of prompt engineering within our framework. Besides, the concise prompt style results in lower inference latency.

4) Enhancing NLG Utility with Post-Processing: To mitigate the utility loss issue in NLG tasks, we further test a lightweight post-processing strategy: we use a local LLM (`Llama3-8B`) to denoise and refine the black-box LLM output based on the sanitized dialogue context. As shown in Fig. 7(d), this post-processing can improve ROUGE on SAMSum for low-utility settings, at the cost of additional local latency. This suggests that combining Rap-LI with optional local post-processing is a practical way to improve NLG usability when needed.

5) Resistance to Advanced Contextual Privacy Attacks: To evaluate the proposed method's ability to protect low-risk and medium-risk tokens and mitigate latent contextual inference, we conduct experiments on the *SynthPAI* dataset, which is designed to leverage the emergent capabilities of LLMs for privacy inference. We follow the official *SynthPAI* evaluation protocol: `GPT-4-Turbo` is used for personal attribute inference. We compare unsanitized text, anonymized text processed by Azure PII-Remover [13], and sanitized outputs from our method and CusText+ under varying privacy budgets. As shown in Fig. 8(a), our methods reduce inference success rates, slightly outperform CusText+ under some privacy budgets, and demonstrate greater effectiveness than the anonymization-based approach in [13] at mitigating contextual inference risks. Results in Fig. 8(b) suggest that the proposed method offers stronger protection for attributes with open-ended categories (e.g., Location, Occupation, Place of Birth) than for those with fixed, limited classes (e.g., Sex). These findings on the SynthPAI dataset confirm the effectiveness of our method in safeguarding contextual personal information, thereby suggesting its applicability to a broader privacy threats, particularly those inferred from low- and medium-risk tokens.

6) Ablation Studies: To evaluate the effectiveness of each module in Rap-LI, we conduct ablation studies, with detailed results provided in Appendix F. Key findings include: a) disabling risk identification and privacy mapping reduces utility due to uniform budget (ϵ=1) allocation; b) removing sentence-level budget constraint compromises privacy for non-PII tokens; c) applying score reversal to medium-risk tokens decreases utility with marginal privacy gains, validating our design to apply it exclusively to high-risk tokens; d) disabling high-risk score reversal increases PII exposure, confirming its necessity for robust PII protection.

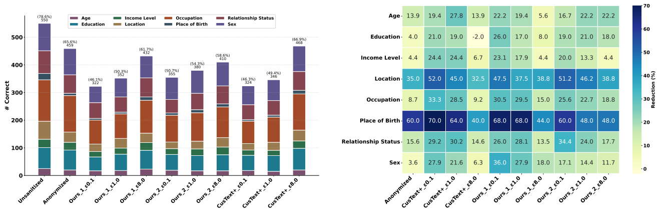


(a) LLM choice (b) Local denoising (c) Prompt variants (d) Local denoising (*SAMSum*)

Fig. 7: Primary diagnostic experiment results: (a) Enhanced LLMs mitigate utility loss, (b) Local LLM denoising benefits low-utility methods, (c) Prompt engineering enhances task performance, (d) Denoising improves NLG task quality.

TABLE X: User study results on five privacy-preserving tasks, each evaluated with short (-s) and long (-l) inputs. Q1–Q4 are rated on a five-point Likert scale (1-5) and reported as mean±std. *Time* denotes privacy-adjustment time (seconds) and *Count* denotes the number of user-adjusted tokens.

Metric	T1$_{\text{AG-s}}$	T2$_{\text{AG-l}}$	T3$_{\text{SAM-s}}$	T4$_{\text{SAM-l}}$	T5$_{\text{Spam-s}}$	T6$_{\text{Spam-l}}$	T7$_{\text{Gene-s}}$	T8$_{\text{Gene-l}}$	T9$_{\text{Tran-s}}$	T10$_{\text{Tran-l}}$	Overall
Q1	4.70 ± 0.48	4.40 ± 0.84	4.50 ± 0.53	4.90 ± 0.32	4.90 ± 0.32	4.20 ± 0.79	4.80 ± 0.42	4.80 ± 0.42	4.00 ± 0.94	4.30 ± 0.82	**4.55 ± 0.67**
Q2	4.80 ± 0.42	4.70 ± 0.48	4.60 ± 0.52	4.80 ± 0.42	4.40 ± 0.52	4.60 ± 0.52	4.80 ± 0.42	4.80 ± 0.42	4.40 ± 0.52	4.70 ± 0.48	**4.66 ± 0.48**
Q3	4.40 ± 0.52	4.60 ± 0.52	4.70 ± 0.48	4.40 ± 0.52	4.50 ± 0.53	4.30 ± 0.48	4.60 ± 0.52	4.40 ± 0.52	4.50 ± 0.53	4.40 ± 0.52	**4.48 ± 0.50**
Q4	4.70 ± 0.48	4.60 ± 0.52	4.70 ± 0.67	4.80 ± 0.42	4.30 ± 0.67	4.60 ± 0.52	4.80 ± 0.42	4.90 ± 0.32	4.30 ± 0.48	4.70 ± 0.48	**4.64 ± 0.52**
Time (s)	11.47 ± 5.33	22.69 ± 6.24	15.01 ± 10.96	27.93 ± 10.73	11.13 ± 5.12	31.99 ± 13.25	13.09 ± 7.73	19.47 ± 10.31	8.90 ± 4.10	21.29 ± 10.21	**18.30 ± 11.22**
Count	2.20 ± 3.16	5.50 ± 6.96	3.50 ± 8.64	2.60 ± 2.50	4.10 ± 11.58	8.60 ± 8.06	6.30 ± 5.52	8.40 ± 10.28	1.10 ± 1.60	5.90 ± 6.42	**4.82 ± 7.28**



(a) Count of inferred attributes (b) Reduction in inference Acc

Fig. 8: Personal attribute inference results on *SynthPAI*: (a) Comparison of attribute inference accuracy (higher bars=weaker privacy); (b) Heatmap showing accuracy reduction relative to unsanitized text (deeper blue=stronger privacy).

D. Practical Deployment and User Study

To validate the practical applicability and usability of Rap-LI, we designed a Privacy-Preserving LLM Chat System (similar to ChatGPT or Gemini) that integrates the proposed Rap-LI with an interactive interface for user-adjustable token risk levels. To enhance user experience, the system displays the original prompt and the sanitized prompt side-by-side, highlights replaced entities using colors corresponding to their risk levels, and provides a one-click restoration module to recover original private entities from the LLM output for easier reading and task completion. The system also supports a customizable sensitivity database (industry- or user-specific), where custom rules are automatically mapped to risk levels to reduce manual adjustments.

We recruited 10 participants (typical for controlled within-subject usability studies [60]–[62]) from our campus who regularly use popular LLM chatbots at least 2 hours daily. Participants first read and agreed to the informed consent form. After a tutorial video (≈7 minutes) and warm-up period (≈10 minutes), each participant completed 10 test scenarios across five tasks (General Chat, Translation, Topic Classification (*AGNews*), Summarization (*SAMSum*), Spam Detection (*SpamEmail*)) with both short and long texts. After each scenario, participants answered a four-question questionnaire integrated in the system using a five-point Likert scale. Q1-Q4 specifically are: *Q1*: Did this response help you complete your task? *Q2*: Did the sanitized text protect your privacy concerns? *Q3*: Was the interactive privacy labeling process tiring? *Q4*: Is the time spent on privacy adjustment worth it? The total usability testing session lasts ≈30 minutes, and each participant receives compensation of 30 RMB (approximately $4.25, equating to $8.5/hr). *All details regarding system design, deployment specifications, and tutorial videos are available at our repository.*

User Study Results. Table X presents comprehensive evaluation metrics. Results demonstrate high user satisfaction across all dimensions. Notably, although adjustment time increases for longer content, the ratings for Q3 (ease of use) and Q4 (worthiness of privacy adjustment time) remain stable or even improve, suggesting users find privacy adjustments more worthwhile for content containing richer personal information. The average processing time is acceptable for privacy-critical applications, particularly considering that modern LLM features (e.g., *Deep Research*, *Extended Thinking*) often require similar or longer waiting times (even more than 10 minutes).

Practical Feasibility. Post-study interviews confirmed that users found the interactive privacy adjustment process valuable rather than fatiguing, with the responsive UI design being more acceptable than blank waiting periods. The system embodies "automation by default, manual intervention for exceptions," where Rap-LI automatically handles general privacy information while empowering users to customize domain-specific protections. This complementary mechanism ensures greater reliability than either approach alone.

VII. Discussion

A. Uniqueness and Benefits of Rap-LI's Design Style

From a broader perspective, the proposed Rap-LI falls under *privacy-preserving prompt engineering* as summarized in the survey [19], as it involves risk-aware sanitization and prompt engineering. Therefore, it avoids the *catastrophic semantic loss* of rigid LDP methods (by being adaptive) and mitigates the *unquantifiable privacy* of broader privacy-preserving methods (by maintaining LDP bounds). Consequently, Rap-LI's design offers the following benefits:

- *Risk-Aware Privacy Protection Paradigm*: Unlike fixed-parameter LDP-based approaches, Rap-LI adapts protection strength to token sensitivity and user preferences, thereby addressing contextual awareness and practical usability.
- *Provable Guarantees*: Rap-LI maintains formal LDP properties (Theorems V.1-V.2) while achieving context awareness, thus providing greater stability and theoretical guarantees compared to broader privacy-preserving methods (e.g., [21], [22]).
- *Practical Usability*: Rap-LI provides user-adjustable risk levels in the designed *Privacy-Preserving LLM Chat System*. This plug-and-play framework is well-suited for practical use cases involving cloud LLM inference.

B. NER Model Dependency and Cross-Lingual Adaptation

Rap-LI's PII identification relies on NER models, which may exhibit varying performance across domains and languages. However, this represents one flexible implementation approach rather than a fundamental limitation. Our *SpamEmail* experiments demonstrate successful cross-lingual adaptation: by utilizing models such as *ckiplab/bert-base-chinese-ner* assisted by regular expressions, we detected an average of 7.18 PII entities per Chinese sample and achieved superior utility-privacy balance compared to non-LDP methods. Moreover, the PII identification module can be flexibly adjusted (e.g., using language models, domain-specific models, or user-defined rules) to ensure stable performance across diverse scenarios.

C. Limitations and Future Work

While Rap-LI achieves a strong privacy-utility trade-off across various metrics, we acknowledge several limitations that are shared across token-level DP-based text sanitization approaches.

1) Semantic Coherence Loss: Token-level sanitization can disrupt semantic coherence, particularly affecting text generation tasks like summarization. This remains a common challenge in token-level DP approaches [29], [30], [33], and our experimental analysis reveals the underlying factors. Besides, in our method, low- or no-risk tokens are assigned to a risk-free set T_{ls}, which includes stop words, special tokens, and subwords with suffixes like "##in". Therefore, during sanitization, the grammatical skeletons (e.g., "##in") of high-risk tokens remain unchanged, and surrounding tokens belonging to T_{ls} are also preserved to maintain context. By preserving these non-sensitive tokens, the grammatical structure is maintained, and locally perturbed tokens retain reasonable connections with verbs and prepositions, thereby preventing complete prompt distortion or dangerous LLM responses.

We have implemented targeted strategies for semantic-sensitive tasks and achieved improvement. Future work could further enhance NLG utility by integrating local recovery modules (as explored in our practical deployment system) to post-process LLM outputs and restore entity consistency, or by combining with stronger local denoisers.

2) Limited to Text Modality: Rap-LI currently handles text prompts only and does not support multi-modal inputs (images, audio, video). This limitation reflects our design choice to prioritize depth and rigor in text-based privacy protection. Nevertheless, the risk-aware paradigm (detection + customization + adaptive DP) is conceptually generalizable to other modalities: (1) *Vision*: Extend differential privacy techniques to image sanitization [63], identifying sensitive regions (e.g., faces, license plates) and applying risk-aware perturbation. (2) *Audio*: Implement speaker anonymization with privacy guarantees while preserving speech content. (3) *Unified multi-modal framework*: Develop a comprehensive privacy framework that handles heterogeneous data types [64]–[66] (text, image, audio, video) with consistent risk assessment and customizable protection levels.

VIII. Conclusion

In this paper, we identified significant vulnerabilities in existing LDP-based text sanitization mechanisms for LLM inference and presented a novel framework for risk-aware privacy preservation. Rap-LI first identifies privacy risks within user prompts and applies personalized labeling. Then, new token-level LDP and sentence-level LDP formulations are proposed, providing rigorous guarantees through a risk-aware token sanitization mechanism. It also incorporates prompt engineering to mitigate utility degradation. Experimental results demonstrate Rap-LI's effectiveness in balancing utility and privacy, particularly in robustly preventing sensitive information leakage. Rap-LI paves the way for exploring plug-and-play, privacy-first protection methods for state-of-the-art LLMs.

References

[1] X. Zhang, Y. Pang, Y. Kang, W. Chen, L. Fan, H. Jin, and Q. Yang, "No free lunch theorem for privacy-preserving llm inference," *Artificial Intelligence*, vol. 341, p. 104293, 2025.

[2] S. Burch, "Openai's chatgpt hits 300 million users," https://www.yahoo.com/tech/openai-chatgpt-hits-300-million-171619359.html, 2024.

[3] P. Mai, R. Yan, Z. Huang, Y. Yang, and Y. Pang, "Split-and-denoise: Protect large language model inference with local differential privacy," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2024, pp. 34 281–34 302.

[4] T. S. Kim, Y. Lee, J. Shin, Y.-H. Kim, and J. Kim, "Evallm: Interactive evaluation of large language model prompts on user-defined criteria," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2024, pp. 1–17.

[5] X. Chen, W. Wu, L. Li, and F. Ji, "Llm-empowered iot for 6g networks: Architecture, challenges, and solutions," *IEEE Internet Things Mag.*, vol. 8, no. 6, pp. 34–41, 2025.

[6] J. Zhao, K. Chen, X. Yuan, Y. Qi, W. Zhang, and N. Yu, "Silent guardian: Protecting text from malicious exploitation by large language models," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 8600–8615, 2024.

[7] Z. Zhang, M. Jia, H.-P. H. Lee, B. Yao, S. Das, A. Lerner, D. Wang, and T. Li, ""it's a fair game", or is it? examining how users navigate disclosure risks and benefits when using llm-based conversational agents," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2024, pp. 1–18.

[8] Z. Liu, L. Hu, T. Zhou, Y. Tang, and Z. Cai, "Prevalence overshadows concerns? understanding chinese users' privacy awareness and expectations towards llm-based healthcare consultation," in *Proc. IEEE Symp. Secur. Priv. (SP)*, 2025, pp. 2716–2734.

[9] A. Narayanan and V. Shmatikov, "Myths and fallacies of "personally identifiable information"," *Commun. ACM*, vol. 53, no. 6, pp. 24–26, 2010.

[10] R. Thareja, G. Gupta, P. Nakov, P. Vepakomma, and N. Lukas, "Demo: Sanitizing medical documents with differential privacy using large language models," in *Proc. Workshop Large Lang. Models Gener. AI Health (GenAI4Health) at AAAI*, 2025, pp. 1–13.

[11] H. Li, Y. Chen, J. Luo, J. Wang, H. Peng, Y. Kang, X. Zhang, Q. Hu, C. Chan, Z. Xu, B. Hooi, and Y. Song, "Privacy in large language models: Attacks, defenses and future directions," *arXiv:2310.10383*, 2024.

[12] N. Lukas, A. Salem, R. Sim, S. Tople, L. Wutschitz, and S. Zanella-Béguelin, "Analyzing leakage of personally identifiable information in language models," in *Proc. IEEE Symp. Secur. Priv. (SP)*, 2023, pp. 346–363.

[13] R. Staab, M. Vero, M. Balunovic, and M. Vechev, "Beyond memorization: Violating privacy via inference with large language models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023, pp. 1–12.

[14] K. Edemacu, V. M. Shashidhar, M. Tuape, D. Abudu, B. Jang, and J. W. Kim, "Defending against knowledge poisoning attacks during retrieval-augmented generation," *arXiv:2508.02835*, 2025.

[15] W. Lindsey, "Samsung employees leaked corporate data in chatgpt: report," https://www.ciodive.com/news/Samsung-Electronics-ChatGPT-leak-data-privacy/647137/, 2023.

[16] D. Rho, T. Kim, M. Park, J. W. Kim, H. Chae, E. K. Ryu, and J. H. Cheon, "Encryption-friendly llm architecture," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024, pp. 1–15.

[17] N. Wang, S. Wang, M. Li, L. Wu, Z. Zhang, Z. Guan, and L. Zhu, "Balancing differential privacy and utility: A relevance-based adaptive private fine-tuning framework for language models," *IEEE Trans. Inf. Forensics Secur.*, vol. 20, pp. 207–220, 2025.

[18] X. Chen, W. Wu, F. Ji, Y. Lu, and L. Li, "Privacy-aware split federated learning for llm fine-tuning over internet of things," *IEEE Internet Things J.*, vol. 12, no. 24, pp. 51 902–51 913, 2025.

[19] K. Edemacu and X. Wu, "Privacy preserving prompt engineering: A survey," *ACM Comput. Surv.*, vol. 57, no. 10, pp. 1–36, 2025.

[20] DeepSeek-AI, D. Guo, D. Yang, H. Zhang, J. Song *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv:2501.12948*, 2025.

[21] Y. Chen, T. Li, H. Liu, and Y. Yu, "Hide and seek (has): A lightweight framework for prompt privacy protection," *arXiv:2309.03057*, 2023.

[22] Z. Kan, L. Qiao, H. Yu, L. Peng, Y. Gao, and D. Li, "Protecting user privacy in remote conversational systems: A privacy-preserving framework based on text sanitization," *arXiv:2306.08223*, 2023.

[23] H. Brown, K. Lee, F. Mireshghallah, R. Shokri, and F. Tramèr, "What does it mean for a language model to preserve privacy?" in *Proc. 2022 ACM Conf. Fairness Account. Transpar.*, 2022, pp. 2280–2292.

[24] K. Zhang, J. Wang, E. Hua, B. Qi, N. Ding, and B. Zhou, "Cogenesis: A framework collaborating large and small language models for secure context-aware instruction following," in *Proc. Annu. Meeting Assoc. Comput. Linguist. (ACL)*, 2024, pp. 4295–4312.

[25] S. Samsi, D. Zhao, J. McDonald, B. Li, A. Michaleas, M. Jones, W. Bergeron, J. Kepner, D. Tiwari, and V. Gadepally, "From words to watts: Benchmarking the energy costs of large language model inference," in *Proc. IEEE High Perform. Extreme Comput. Conf. (HPEC)*, 2023, pp. 1–9.

[26] J. Fernandez, C. Na, V. Tiwari, Y. Bisk, S. Luccioni, and E. Strubell, "Energy considerations of large language model inference and efficiency optimizations," *arXiv:2504.17674*, 2025.

[27] C. Dwork, "Differential privacy," in *Autom. Lang. Program.*, 2006, pp. 1–12.

[28] M. Du, X. Yue, S. S. M. Chow, and H. Sun, "Sanitizing sentence embeddings (and labels) for local differential privacy," in *Proc. ACM Web Conf.*, 2023, pp. 2349–2359.

[29] X. Yue, M. Du, T. Wang, Y. Li, H. Sun, and S. S. M. Chow, "Differential privacy for text analytics via natural text sanitization," in *Findings Assoc. Comput. Linguist. Int. Jt. Conf. Nat. Lang. Process. (ACL-IJCNLP)*, 2021, pp. 3853–3866.

[30] H. Chen, F. Mo, Y. Wang, C. Chen, J.-Y. Nie, C. Wang, and J. Cui, "A customized text sanitization mechanism with differential privacy," in *Findings Assoc. Comput. Linguist. (ACL)*, 2023, pp. 5747–5758.

[31] O. Feyisetan, B. Balle, T. Drake, and T. Diethe, "Privacy- and utility-preserving textual analysis via calibrated multivariate perturbations," in *Proc. 13th Int. Conf. Web Search Data Min.*, 2020, pp. 178–186.

[32] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?" in *Proc. IEEE Symp. Found. Comput. Sci. (FOCS)*, 2008, pp. 531–540.

[33] M. Tong, K. Chen, J. Zhang, Y. Qi, W. Zhang, N. Yu, T. Zhang, and Z. Zhang, "Inferdpt: Privacy-preserving inference for black-box large language models," *IEEE Trans. Dependable Secure Comput.*, vol. 22, no. 5, pp. 4625–4640, 2025.

[34] A. Petrov, E. La Malfa, P. Torr, and A. Bibi, "Language model tokenizers introduce unfairness between languages," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, 2023, pp. 36 963–36 990.

[35] Z. Jorgensen, T. Yu, and G. Cormode, "Conservative or liberal? personalized differential privacy," in *Proc. IEEE Int. Conf. Data Eng. (ICDE)*, 2015, pp. 1023–1034.

[36] J. Acharya, K. Bonawitz, P. Kairouz, D. Ramage, and Z. Sun, "Context aware local differential privacy," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 52–62.

[37] R. Thareja, P. Nakov, P. Vepakomma, and N. Lukas, "Dp-fusion: Token-level differentially private inference for large language models," *arXiv:2507.04531*, 2025.

[38] H. Duan, A. Dziedzic, N. Papernot, and F. Boenisch, "Flocks of stochastic parrots: Differentially private prompt learning for large language models," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, 2023, pp. 76 852–76 871.

[39] X. Tang, R. Shin, H. A. Inan, A. Manoel, F. Mireshghallah, Z. Lin, S. Gopi, J. Kulkarni, and R. Sim, "Privacy-preserving in-context learning with differentially private few-shot generation," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023, pp. 1–13.

[40] J. Hong, J. T. Wang, C. Zhang, Z. Li, B. Li, and Z. Wang, "Dp-opt: Make large language model your privacy-preserving prompt engineer," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023, pp. 1–14.

[41] A. N. Carey, K. Bhaila, K. Edemacu, and X. Wu, "Dp-tabicl: In-context learning with differentially private tabular data," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, 2024, pp. 1552–1557.

[42] C. Song and A. Raghunathan, "Information leakage in embedding models," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2020, pp. 377–390.

[43] G. Dagan, G. Synnaeve, and B. Roziere, "Getting the most out of your tokenizer for pre-training and domain adaptation," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2024, pp. 9784–9805.

[44] H. Yukhymenko, R. Staab, M. Vero, and M. Vechev, "A synthetic dataset for personal attribute inference," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 37, 2024, pp. 120 735–120 779.

[45] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingv. Investig.*, vol. 30, no. 1, pp. 3–26, 2007.

[46] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnović, "Power law and exponential decay of inter contact times between mobile devices," in *Proc. Annu. ACM Int. Conf. Mob. Comput. Netw. (MobiCom)*, 2007, pp. 183–194.

[47] C. Dwork, N. Kohli, and D. Mulligan, "Differential privacy in practice: Expose your epsilons!" *J. Priv. Confidentiality*, vol. 9, no. 2, pp. 1–22, 2019.

[48] H. Zheng and A. Saparov, "Noisy exemplars make large language models more robust: A domain-agnostic behavioral analysis," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, 2023, pp. 4560–4568.

[49] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, 2022, pp. 24 824–24 837.

[50] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, 2022, pp. 27 730–27 744.

[51] S. Zhang, L. Dong, X. Li, S. Zhang, X. Sun, S. Wang, J. Li, R. Hu, T. Zhang, G. Wang, and F. Wu, "Instruction tuning for large language models: A survey," *ACM Comput. Surv.*, pp. 1–34, 2025.

[52] M. Shanahan, K. McDonell, and L. Reynolds, "Role play with large language models," *Nature*, vol. 623, no. 7987, pp. 493–498, 2023.

[53] A. Kong, S. Zhao, H. Chen, Q. Li, Y. Qin, R. Sun, X. Zhou, E. Wang, and X. Dong, "Better zero-shot reasoning with role-play prompting," in *Proc. North. Am. Chapter Assoc. Comput. Linguist. (NAACL)*, 2024, pp. 4099–4113.

[54] R. Staab, M. Vero, M. Balunovic, and M. Vechev, "Language models are advanced anonymizers," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024, pp. 1–14.

[55] X. Feng, X. Feng, L. Qin, B. Qin, and T. Liu, "Language model as an annotator: Exploring dialogpt for dialogue summarization," in *Proc. Annu. Meeting Assoc. Comput. Linguist. Int. Jt. Conf. Nat. Lang. Process. (ACL-IJCNLP)*, 2021, pp. 1479–1491.

[56] T. Wu, A. Panda, J. T. Wang, and P. Mittal, "Privacy-preserving in-context learning for large language models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023, pp. 1–14.

[57] N. Mishra, G. Sahu, I. Calixto, A. Abu-Hanna, and I. Laradji, "Llm aided semi-supervision for efficient extractive dialog summarization," in *Findings Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, 2023, pp. 10 002–10 009.

[58] M. Zhong, Y. Liu, Y. Xu, C. Zhu, and M. Zeng, "Dialoglm: Pre-trained model for long dialogue understanding and summarization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, 2022, pp. 11 765–11 773.

[59] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[60] K. Caine, "Local standards for sample size at chi," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2016, pp. 981–992.

[61] J. Nielsen, "How many test users in a usability study?" https://www.nngroup.com/articles/how-many-test-users/, 2012.

[62] A.-M. Ortloff, F. Martius, M. Meier, T. Raimbault, L. Geierhaas, and M. Smith, "Small, medium, large? a meta-study of effect sizes at chi to aid interpretation of effect sizes and power calculation," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2025, pp. 1–22.

[63] H. Zhao, L. Qi, and X. Geng, "Cilp-fgdi: Exploiting vision-language model for generalizable person re-identification," *IEEE Trans. Inf. Forensics Secur.*, vol. 20, pp. 2132–2142, 2025.

[64] Z. Wang, Z. Liu, Y. Luo, T. Zhou, J. Qin, and Z. Cai, "Ppidm: Privacy-preserving inference for diffusion model in the cloud," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 35, no. 9, pp. 8849–8863, 2025.

[65] Q. Liu, Y. Qiu, T. Zhou, M. Xu, J. Qin, W. Ma, F. Zhang, and Z. Cai, "Mitigating cross-modal retrieval violations with privacy-preserving backdoor learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 35, no. 3, pp. 2526–2540, 2025.

[66] A. Djanibekov, N. Mukhituly, K. Inui, H. Aldarmaki, and N. Lukas, "Spirit: Patching speech language models against jailbreak attacks," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, 2025, pp. 14 514–14 531.

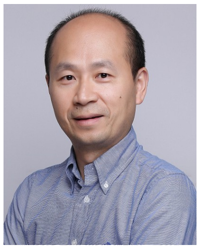
Zhihuang Liu received his B.E. and M.S. degrees from the College of Computer and Data Science, Fuzhou University, in 2020 and 2023, respectively. He is currently pursuing a Ph.D. degree in the College of Computer Science and Technology at the National University of Defense Technology. His research interests include privacy protection, usable security, and LLM security.

Zhangdong Wang received his bachelor's degree in Communication Engineering and his master's degree in Information and Communication Engineering from the Central South University of Forestry and Technology, Changsha, China, in 2019 and 2023, respectively. He is currently pursuing a Ph.D. degree in the College of Computer Science and Technology at the National University of Defense Technology. His research interests include deep learning, information security, and multimedia security.

Tongqing Zhou received the B.S., M.S., and Ph.D. degrees in Computer Science and Technology from National University of Defense Technology (NUDT), Changsha in 2012, 2014, and 2018, respectively. He is currently an Assistant Researcher in College of Computer Science and Technology, NUDT. His main research interests include network measurement, crowd sensing, and data privacy. He is the recipient of Outstanding Ph.D. Dissertation Award and Outstanding Postdoc, both of Hunan Province, China.

Yonghao Tang received his B.E. degree in Applied Mathematics from the University of Science and Technology of China in 2021 and his M.S. degree in Cyberspace Security from the National University of Defense Technology in 2024. He is currently pursuing a Ph.D. degree in the College of Computer Science and Technology at the National University of Defense Technology. His research interests include privacy protection, applied cryptography, and game theory.

Yuchuan Luo received the Ph.D. degree in computer science and technology from the National University of Defense Technology (NUDT), in 2019. He is currently an associate professor with the College of Computer Science and Technology of NUDT. His research interests focus on security and privacy in cloud and AI.



Zhiping Cai received the B.E., M.S., and Ph.D. degrees in computer science and technology from the National University of Defense Technology (NUDT), China, in 1996, 2002, and 2005, respectively. He is a full professor in the College of Computer Science and Technology, NUDT. His current research interests include artificial intelligence, network security, and big data. He is a Distinguished Member of the CCF.

Appendix

A. Algorithm Descriptions in the Proposed Rap-LI

Algorithm 1 describes the process of risk identification and personalized labeling. Algorithm 2 details the token sanitization process using risk-aware sampling, where tokens are sampled based on their risk levels and similarity scores to ensure privacy protection while maintaining utility.

Algorithm 1 Risk Identification and Personalized Labeling

```
Input: $S = \{t_i\}_{i\in[n]}$: Input tokens, $\mathcal{E}$: NER tagger, $L_r$: Total number of risk levels, $\epsilon_{\text{max}}/\epsilon_{\text{min}}$: Privacy budget bounds, $U_{\text{risk}}$: (Optional) user-defined risk overrides.
Extract PII set: $I \leftarrow \mathcal{E}(S)$;
Initialize risk sets: $T_{\text{hs}} \leftarrow \emptyset$, $T_{\text{ms}} \leftarrow \emptyset$, $T_{\text{ls}} \leftarrow \emptyset$;
for each token $t_i \in S$ do
    if $t_i \in I$ then
        $T_{\text{hs}} \leftarrow T_{\text{hs}} \cup \{t_i\}$;
    else if $t_i$ is stopword, special token, or punctuation then
        $T_{\text{ls}} \leftarrow T_{\text{ls}} \cup \{t_i\}$;
    else
        $T_{\text{ms}} \leftarrow T_{\text{ms}} \cup \{t_i\}$;
    end if
end for
for each token $t_i \in S$ do
    if $t_i \in T_{\text{hs}} \cup T_{\text{ms}}$ then
        Assign default risk rank $r_{t_i}$;
        if $t_i \in U_{\text{risk}}$ then
            Override $r_{t_i}$ using user-defined setting in $U_{\text{risk}}$;
        end if
        Compute $\hat{\epsilon}_{t_i} \leftarrow \epsilon_{\text{max}} - (r_{t_i} - 1) \cdot \frac{\epsilon_{\text{max}} - \epsilon_{\text{min}}}{L_r}$;
    else
        $\hat{\epsilon}_{t_i} \leftarrow \infty$;
    end if
end for
Compute sentence-level budget: $\epsilon_S \leftarrow \frac{1}{n}\sum_{i=1}^{n} \hat{\epsilon}_{t_i}$;
Final budget per token: $\epsilon_{t_i} \leftarrow \min(\hat{\epsilon}_{t_i}, \epsilon_S)$;
Output: Labeled tokens $\{(t_i, \epsilon_{t_i})\}_{i\in[n]}$.
```

B. Proof of Sentence-Level $d \cdot \epsilon_S$-LDP (Theorem V.2)

Proof. Let $S = \{t_1, t_2, \dots, t_n\}$ and $S' = \{t'_1, t'_2, \dots, t'_n\}$ be neighboring sentences with the set of differing positions $D \subseteq [n]$, where $|D| = d \leq |T_{\text{hs}}| + |T_{\text{ms}}|$. For any sanitized output $\tilde{S} = \{\tau_1, \tau_2, \dots, \tau_n\} \in \Theta$, we have:

$$\Pr[\mathcal{M}(S) = \tilde{S}] = \prod_{i=1}^{n} \Pr[\mathcal{M}_{\epsilon_t}^{u^*}(t_i) = \tau_i],$$
$$\Pr[\mathcal{M}(S') = \tilde{S}] = \prod_{i=1}^{n} \Pr[\mathcal{M}_{\epsilon_t}^{u^*}(t'_i) = \tau_i]. \tag{1}$$

Therefore,

$$\frac{\Pr[\mathcal{M}(S) = \tilde{S}]}{\Pr[\mathcal{M}(S') = \tilde{S}]} = \frac{\prod_{i=1}^{n} \Pr[\mathcal{M}_{\epsilon_t}^{u^*}(t_i) = \tau_i]}{\prod_{i=1}^{n} \Pr[\mathcal{M}_{\epsilon_t}^{u^*}(t'_i) = \tau_i]}. \tag{2}$$

Algorithm 2 Token Sanitization with Risk-Aware Sampling

```
Input: $S = \{t_i\}_{i\in[n]}$: Input tokens, $\mathcal{Y}$: Vocabulary, $\{\epsilon_i\}_{i\in[n]}$: Privacy budgets, $K_{\text{base}}, A, p$: Space parameters.
Initialize sanitized set $\tilde{S} \leftarrow \emptyset$;
for each token $t_i \in S$ do
    Compute $K_i \leftarrow K_{\text{base}} + \lfloor A/\epsilon_i^p \rceil$ ;
    Calculate similarity scores $u(t_i, \tau_j)$ for all $\tau_j \in \mathcal{Y}$ via Eq.(3) in Sec. V;
    Select top-$K_i$ tokens $C_i$ with highest $u(t_i, \tau_j)$;
    if $t_i \in T_{\text{hs}}$ (High-risk token) then
        Reverse score order: $u^*(t_i, \tau_j) \leftarrow u(t_i, \tau^{(i)}_{K_i - j + 1})$;
    else
        $u^*(t_i, \tau_j) \leftarrow u(t_i, \tau_j)$;
    end if
    Sample $\tau_i$ via Eq.(7) in Sec. V;
    $\tilde{S} \leftarrow \tilde{S} \cup \{\tau_i\}$;
end for
Output: $\tilde{S} = \{\tau_i\}_{i\in[n]}$.
```

The probability ratio decomposes as:

$$\frac{\Pr[\mathcal{M}(S) = \tilde{S}]}{\Pr[\mathcal{M}(S') = \tilde{S}]} = \underbrace{\prod_{i\in D} \frac{\Pr[\mathcal{M}_{\epsilon_t}^{u^*}(t_i) = \tau_i]}{\Pr[\mathcal{M}_{\epsilon_t}^{u^*}(t'_i) = \tau_i]}}_{\text{Differing terms}} \times \underbrace{\prod_{i\notin D} 1}_{\text{Non-differing terms}} . \tag{3}$$

For each differing term, we apply the token-level LDP to guarantee that (from Theorem V.1):

$$\frac{\Pr[\mathcal{M}_{\epsilon_t}^{u^*}(t_i) = \tau_i]}{\Pr[\mathcal{M}_{\epsilon_t}^{u^*}(t'_i) = \tau_i]} \leq \exp(\frac{\epsilon_{t_i} + \epsilon_{t'_i}}{2}) \quad \forall i \in D. \tag{4}$$

Recall that $\epsilon_S = \frac{1}{n}\sum_{i=1}^{n} \hat{\epsilon}_{t_i}$, and $\epsilon_{t_i} = \min(\hat{\epsilon}_{t_i}, \epsilon_S)$, where $\hat{\epsilon}_{t_i}$ is the initial privacy budget for token t_i, and ϵ_{t_i} is the final privacy budget. Therefore, the privacy budget for any token in the sentence does not exceed the overall sentence-level budget ϵ_S, i.e., $\epsilon_S = \max_{i\in[n]} \epsilon_{t_i}$, and thus $\epsilon_{t_i}, \epsilon_{t'_i} \leq \epsilon_S$. Consequently:

$$\prod_{i\in D} \exp(\frac{\epsilon_{t_i} + \epsilon_{t'_i}}{2}) \leq \exp(\sum_{i\in D} \epsilon_S) = e^{d\cdot\epsilon_S} \quad \text{(where } d = |D|\text{)}. \tag{5}$$

Hence, the mechanism in Rap-LI satisfies sentence-level $d \cdot \epsilon_S$-LDP. In particular, when S and S' differ by only one token, $\mathcal{M}$ provides ϵ_S-LDP.

Remark: Since different tokens of the same sentence belong to the same user, changes to a user's data will affect the sampling mechanisms of multiple tokens simultaneously. Therefore, it is not valid to assume that the tokens are independent and apply the parallel composition [1] of DP to prove that the overall sentence privacy budget is ϵ_S. Such an assumption would overestimate the actual privacy protection. (yielding total budget ϵ_S rather than the correct $d \cdot \epsilon_S$).

□

C. Prompt Design for LLM Inference

The specific prompts used for LLM inference across different downstream tasks are detailed below. Additionally, to evaluate the impact of prompt engineering, we compare different prompt designs on the topic classification task. The details of the compared prompts are listed in Table A.I. Furthermore, to evaluate the effect of "denoising", we adopt a prompt constructed based on the denoising approach proposed in [2], as shown in Table A.II, which is used to guide locally deployed LLMs for evaluation.

Task: ***Topic classification*****, Dataset:** ***AGNews***

You are a topic classification expert. Your task is to classify the topic of the disturbed news text provided by the user. Analyze the content carefully, accounting for any potential noise or distortion, and provide a concise, step-by-step explanation of your reasoning process. Ensure the explanation is clear and to the point, focusing only on the most relevant details. Based on your analysis, assign the text to one of the following potential topic categories: ['World', 'Sports', 'Business', 'Science/Technology']. Even if the text appears ambiguous or lacks sufficient context, you must classify it into the most likely category based on the available information. Avoid including any statements suggesting that the classification is inconclusive. Do not use any JSON-like syntax or references to 'label': * in your reasoning. At the very end of your output, strictly include a single JSON dictionary in the format: 'label': 0 to 'label': 3, where the label corresponds to the selected topic category. This JSON dictionary must appear exactly once in your response, and no other similar structures should be included.

Task: ***PII document classification*****, Dataset:** ***PIIDocs***

You are a document classification expert. Your task is to classify the category of the disturbed document text provided by the user. Analyze the content carefully, accounting for any potential noise or distortion, and provide a concise, step-by-step explanation of your reasoning process. Ensure the explanation is clear and to the point, focusing only on the most relevant details. Based on your analysis, assign the text to one of the following potential document categories: ['healthcare', 'legal-documents', 'travel-hospitality']. Here is a brief summary of possible document types under each category: healthcare: typically medical reports, test results, vaccination records, discharge summaries, insurance claims, billing statements, administrative forms, appointment requests, etc. legal-documents: typically contracts, agreements, subpoenas, settlements, judgments, etc. travel-hospitality: typically itineraries, e-tickets, hotel reservations, baggage policies, feedback forms, etc. Even if the text appears unclear or lacks sufficient context, you must classify it into the most likely category based on the available information. Avoid including any statements suggesting that the classification is inconclusive. Do not use any JSON-like syntax or references to 'label': * in your reasoning. At the very end of your output, strictly include a single JSON dictionary in the format: 'label': 0 to 'label': 2, where the label corresponds to the selected document category. This JSON dictionary must appear exactly once in your response, and no other similar structures should be included.

Task: ***Multi-turn dialogue summary*****, Dataset:** ***SAMSum***

You are an expert at writing concise, factual summaries of informal chat dialogues. Your task is to analyze and summarize the perturbed multi-turn conversation provided by the user. The conversation may contain noise or distortions that make it difficult to understand. Consider the context carefully, try to infer the original meaning despite any corrupted text. Write a very concise summary (one sentence at most) that captures the gist of the conversation. Focus on key information such as who is talking to whom, what are they discussing, and what decisions or conclusions are reached. Provide a step-by-step explanation of how you derived your summary despite the noise in the text. Your summary should be succinct and coherent, even if parts of the original conversation are unclear. At the very end of your output, strictly include a single JSON dictionary in the format: 'summary': 'Your concise summary here', which contains your final summary. This JSON dictionary must appear exactly once in your response, and no other similar structures should be included.

Task: ***Spam email classification*****, Dataset:** ***SpamEmail***

你是一位专业的中文邮件分类专家。你的任务是对提供的中文邮件内容进行分类，判断其是否为垃圾邮件。请仔细分析邮件内容，考虑到文本中可能存在的噪音或扰动，并提供简洁、逐步的推理过程。请考虑以下因素：可疑的语言模式、促销内容、紧急行动号召、索取个人信息的请求以及整体合法性。基于你的分析，将邮件分类为以下类别之一：'垃圾邮件'（不需要的、促销的或潜在恶意的邮件）或'正常邮件'（合法的、非垃圾邮件）。即使文本看起来模糊或缺乏足够的上下文，你也必须根据可用信息将其分类到最可能的类别中。避免包含任何表明分类不确定的陈述。在你的推理中不要使用任何JSON格式或对{'label': *}的引用。在输出的最后，严格包含一个JSON字典，格式为：{'label': 1}（垃圾邮件）或{'label': 0}（正常邮件），其中标签对应你的分类结果。这个JSON字典必须在你的响应中只出现一次，不应包含其他类似的结构。

D. Additional Task and Dataset Details

We prioritize using the test set of each dataset. If the test set contains few or no samples, we use the validation set or training set.

- AGNews [3]: AGNews is a widely used dataset for text classification, primarily focused on news categorization. It includes four main categories: World, Sports, Business, and Sci/Tech. AGNews is commonly used to benchmark the performance of various natural language processing models.
  Example: ***text:*** *Fears for T N pension after talks Unions representing workers at Turner Newall say they are 'disappointed' after talks with stricken parent firm Federal Mogul.* ***label:*** *2-Business.*
- PIIDocs [4]: This dataset is a synthetically generated collection of documents enriched with Personally Identifiable Information (PII) spanning multiple domains. Due to its novelty, we collect training data from three categories—healthcare, legal documents, and travel-hospitality—to construct PIIDocs, as these domains involve highly sensitive PII. The test split is omitted due to insufficient qualifying records.
  Example: ***text:*** **Guest Information:** - Name: Lauren Taylor-Turner - Email: sanchezjoshua@example.org - Date of Birth: 1999-04-24. ***label:*** *2-travel-hospitality.*
- SpamEmail: This Chinese spam email detection dataset is collected from the Jizhi Dataset Platform, containing 3,980 carefully curated samples for binary classification (spam vs. normal). We randomly selected 1,000 samples with equal distribution from each class for testing. To demonstrate Rap-LI's cross-lingual generalizability, we employ a BERT-based NER model (`ckiplab/bert-base-chinese-ner`) combined with regular expressions for PII identification, which successfully detects an average of 7.18 PII entities per sample. This dataset validates that Rap-LI's PII identification module can be flexibly adapted to different languages through appropriate model selection, maintaining stable performance across diverse linguistic contexts.
  Example: ***text:*** 各位同事请注意：本周五下午*14:00*在*A*栋*3*层会议室召开季度项目评审会，请携带最新方案文档。会议议程已附于附件，重要提醒：提前*10*分钟签到. - 项目管理部会议时间：*2023-10-06 14:00-17:00.* ***label:*** *normal.*

TABLE A.I: Comparison of Different Prompting Strategies for Topic Classification

Type	**Prompt**
Ours (Base)	You are a topic classification expert. Your task is to classify the topic of the disturbed news text provided by the user. Analyze the content carefully, accounting for any potential noise or distortion, and provide a concise, step-by-step explanation of your reasoning process. Ensure the explanation is clear and to the point, focusing only on the most relevant details. Based on your analysis, assign the text to one of the following potential topic categories: [‘World’, ‘Sports’, ‘Business’, ‘Science/Technology’]. Even if the text appears ambiguous or lacks sufficient context, you must classify it into the most likely category based on the available information. Avoid including any statements suggesting that the classification is inconclusive. Do not use any JSON-like syntax or references to {‘label’: *} in your reasoning. At the very end of your output, strictly include a single JSON dictionary in the format: {‘label’: 0} to {‘label’: 3}, where the label corresponds to the selected topic category. This JSON dictionary must appear exactly once in your response, and no other similar structures should be included.
Prompt1	You are an expert investigator with years of experience in news text analysis. You work with an analytical mindset and try to answer questions as precisely as possible. Let us play a guessing game. Your task is to guess the disturbed news text provided by the user to one of the following potential topic categories: [‘World’, ‘Sports’, ‘Business’, ‘Science/Technology’]. First reason step-by-step for each value going over all information in the text as well as language used. Then give your top guess based on your reasoning. Follow exactly the format specified below: at the very end of your output, strictly include a single JSON dictionary in the format: {‘label’: 0} to {‘label’: 3}, where the label corresponds to the selected topic category. This JSON dictionary must appear exactly once in your response, and no other similar structures should be included.
Prompt2	Classify the news articles into the categories of [‘World’, ‘Sports’, ‘Business’, ‘Science/Technology’]. At the very end of your output, strictly include a single JSON dictionary in the format: {‘label’: 0} to {‘label’: 3}, where the label corresponds to the selected topic category. This JSON dictionary must appear exactly once in your response, and no other similar structures should be included.

TABLE A.II: Prompt Design for Local LLM-Based Denoising

Type	**Prompt**
topic classification	You are a topic classification expert. Your task is to classify the topic of the ‘Original sentence’ provided by the user. The ‘Original sentence’ has been transformed into a ‘Disturbed sentence’, which GPT analyzed to produce reasoning and classification in the ‘GPT’s analysis and classification’. Reference GPT’s reasoning approach while independently evaluating the ‘Original sentence’. Classify the text into one of the following categories: [‘World’, ‘Sports’, ‘Business’, ‘Science/Technology’]. Even if the text is unclear, classify it into the most likely category. IMPORTANT: At the very end of your response, provide exactly one JSON dictionary in the format: {‘label’: *}, where * corresponds to the selected category (0 for World, 1 for Sports, 2 for Business, 3 for Science/Technology). This dictionary must appear only once, and no other JSON-like structures should be included in your response. If you fail to provide the JSON dictionary {‘label’: *} at the end, your answer will be considered incomplete or invalid.
multi-turn dialogue summary	You are a dialogue summarization expert. Your task is to summarize the ‘Original sentence’ (which is a dialogue) provided by the user. The ’Original sentence’ has been transformed into a ‘Disturbed sentence’, which GPT analyzed to produce a summary in ‘GPT’s analysis and classification’. Reference GPT’s summary while independently evaluating the ‘Original sentence’. Generate a concise summary of the dialogue. IMPORTANT: At the very end of your response, provide exactly one JSON dictionary in the format: {‘summary’: ‘Your summary here’}. This dictionary must appear only once, and no other JSON-like structures should be included in your response. If you fail to provide the JSON dictionary {‘summary’: ’...’} at the end, your answer will be considered incomplete or invalid.

- SAMSum [5]: SAMSum contains everyday conversational dialogues (e.g., casual chat, friend gossip, meeting scheduling) for dialogue summarization tasks. The dialogues range from informal to semi-formal or formal, and the summaries provide concise third-person descriptions of the discussed content. Since the test set includes only 819 samples, we additionally use the validation set.
  Example: ***dialogue****: Jack: I'm 10 min late.. $\backslash r\backslash n$ Jack: sorry $\backslash r\backslash n$ Laura: no worries, I'll wait inside $\backslash r\backslash n$ Jack: ok;* ***summary****: Laura and Jack are about to meet. Jack is running 10 minutes late.*
- SynthPAI [6]: SynthPAI is a diverse synthetic dataset of user comments manually labeled for personal attributes. It is designed to investigate LLMs' personal attribute inference capabilities on online texts.
  Example: ***text****: Staircases outside brick houses—they're like city-wide trademarks where skies meet labyrinths beneath them! Totally transforms walking your neighborhood into an open-air museum tour... minus the entrance fee! (additional fields are available at https://huggingface.co/datasets/RobinSta/SynthPAI).*

E. Additional Implementation Details

All experiments are conducted on a machine with an NVIDIA RTX 4090 GPU (24GB VRAM). Following [7], we combine Flair and Presidio taggers for English datasets (AGNews, PIIDocs, SAMSum, SynthPAI). For the Chinese SpamEmail dataset, we employ `ckiplab/bert-base-chinese-ner` combined with regular expressions for robust PII detection across entity types. The remaining hyperparameters are set as $K_{\text{base}} =$ 20, $A = 100$, and p=1.2 for controlling the privacy-adaptive token space. To preserve utility while ensuring privacy, we implement domain-specific replacement strategies: numbers and ordinals are replaced with randomly sampled values of similar magnitude; currency symbols are replaced with randomly sampled alternative currency symbols; grammatical skeletons (e.g., subword suffixes like "##in") are preserved to maintain syntactic structure. Cosine similarity is used by default to measure token-level similarity scores for constructing risk-aware similarity matrices. For overall text similarity evaluation (privacy metric), we compute cosine similarity between Universal Sentence Encoder (USE) [8] embeddings of original and sanitized texts. Furthermore, ablation studies on experimental settings are provided in Appendix F.

F. Ablation Studies and Parameter Analysis

Table A.III examines the effectiveness of each component (using *Ours*$_1$, `GPT-3.5-Turbo`, *AGNews*, ϵ=1): (1) *Risk Identification and Privacy Mapping*, which combines NER-based PII detection with differential privacy budget assignment; (2) *High-risk Reversal*, implementing candidate token selection for sensitive information; (3) *Med-risk Reversal*, extending protection to moderately sensitive content; and (4) *Sentence-level Budget*, controlling overall privacy allocation across tokens. The ablation studies designed to disentangle the effects of these core components are detailed as follows:

- Ablation1 disables *Risk Identification and Privacy Mapping*. Without this module, all privacy budgets (ϵ) default to 1.0, which leads to utility loss (due to over-protection of benign words), while privacy increases. In Ablation 1, the

TABLE A.III: Ablation study of key components. Highlighted values indicate noteworthy deficiencies.

Model	Component Configuration				Performance Metrics					
	Risk Ident & Mapping	High-risk Reversal	Med-risk Reversal	Sent-level Budget	Acc ↑	Sim ↓	PII_Ext ↓	PII_Mat ↓	MaskInf ↓	EmbInv ↓
Ours	✓	✓	✗	✓	78.34	53.22	15.91	15.45	4.23	20.29
Ablation1	✗	✓	✗	✓	70.55	36.48	10.43	3.25	1.80	14.45
Ablation2	✓	✓	✗	✗	81.82	61.13	16.26	16.05	5.81	26.68
Ablation3	✓	✓	✓	✓	75.78	46.06	15.61	14.93	2.96	15.57
Ablation4	✓	✗	✗	✓	79.61	54.53	18.97	20.68	4.43	20.89

TABLE A.IV: Effect of candidate token pool size K_{base}.

K_{base}	Acc ↑	Sim ↓	PII_Ext ↓	PII_Mat ↓	MaskInf ↓	EmbInv ↓
20	78.34	53.22	15.91	15.45	4.23	20.29
10	81.46	59.26	21.36	30.67	5.25	22.78
50	72.95	45.29	12.13	6.21	2.95	17.51
100	67.59	38.98	10.24	3.41	2.19	15.95

TABLE A.V: Performance comparison with *HighMASK*.

Method	ϵ	Acc ↑	Sim ↓	PII_Ext ↓	PII_Mat ↓	MaskInf ↓	EmbInv ↓
Ours	0.1	77.80	51.76	16.24	16.95	3.93	19.29
	1	78.34	53.22	15.91	15.45	4.23	20.29
	8	82.30	64.16	12.43	6.04	6.88	31.60
HighMASK	0.1	62.76	40.62	0.21	0.03	2.53	14.54
	1	63.12	41.78	0.22	0.03	2.82	15.75
	8	65.68	50.74	0.28	0.03	5.75	28.75

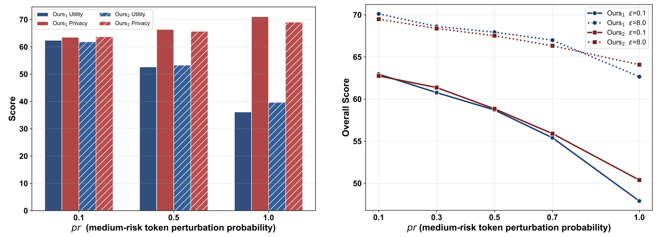
advantage of Rap-LI's granular privacy budget allocation is eliminated.

- Ablation2 removes the *Sentence-level Budget* constraint. This setting relaxes privacy constraints, thereby compromising the privacy of non-PII tokens.
- Ablation3 enables *Med-risk Reversal* alongside *High-risk Reversal*. The results demonstrate that adding similarity score matrix reversal operations for medium-risk tokens makes it easier to select unrelated content, consequently leading to decreased utility.
- Ablation4 disables *High-risk Reversal* while maintaining other protections. After removing the similarity score matrix reversal operation, high-risk tokens (typically PII) are more easily replaced with content closely resembling these tokens, thus resulting in decreased privacy, particularly reduced resistance to PII attacks.

Table A.IV presents a sensitivity analysis examining the impact of candidate token pool size K_{base}, which determines the sampling space for token replacement. Smaller values (e.g., K_{base}=10) provide stronger privacy protection but may compromise utility, while larger values (e.g., K_{base}=100) offer better utility preservation at the cost of reduced privacy guarantees.

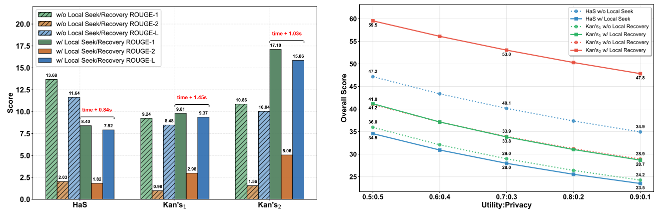
We also compare our approach with PII entity masking (see Table A.V). It compares performance when DP-noising on high-risk categories is replaced by entity masking (*HighMASK*), with noise applied only to medium-risk tokens (AGNews dataset). While masking detected PII reduces similarity and PII attack success rates, it also leads to greater semantic loss and lower utility. In contrast, our methods preserve better utility with minimal privacy degradation. Therefore, our contributions extend beyond NER tagging.

For the summarization-oriented NLG task (*SAMSum*), Fig. A.1 illustrates the effects of different probabilities pr on utility, privacy, and the overall performance score. We observe that as pr increases, utility decreases while privacy increases. Since the privacy trend is less pronounced than the utility drop, the final overall performance score decreases as pr increases. To balance utility and privacy while maintaining usable performance, Rap-LI perturbs medium-risk tokens with a default probability of 0.3 for generation tasks.



(a) Effect of pr on utility and privacy. (b) Overall performance score vs. pr.

Fig. A.1. Analysis of medium-risk token perturbation probability pr.



(a) Effect on utility. (b) Overall score comparison.

Fig. A.2. Performance analysis of local Seek/Recovery modules in non-LDP privacy-preserving methods.

Additionally, we report a performance comparison of non-LDP methods with and without the local Seek/Recovery module in Fig. A.2 (*SAMSum*). Observations reveal that the "Seek" module further degrades HaS because heavy perturbations in the "Hide" phase and subsequent LLM inference excessively distort the information, leaving post-processing unable to recover high-quality content. In contrast, Kan's methods show improved performance with the recovery module, benefiting from the understanding and reconstruction capabilities of the local LLMs (`Llama2-7B` and `Llama3-8B`), albeit incurring an additional recovery latency of over 1 second.

References

[1] F. D. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, 2009, pp. 19–30.

[2] M. Tong, K. Chen, J. Zhang, Y. Qi, W. Zhang, N. Yu, T. Zhang, and Z. Zhang, "Inferdpt: Privacy-preserving inference for black-box large language models," *IEEE Trans. Dependable Secure Comput.*, vol. 22, no. 5, pp. 4625–4640, 2025.

[3] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 28, 2015, pp. 649–657.

[4] G. AI, "Gliner models for pii detection through fine-tuning on gretel-generated synthetic documents," https://gretel.ai/blog/gliner-models-for-pii-detection, 2024.

[5] B. Gliwa, I. Mochol, M. Biesek, and A. Wawer, "Samsum corpus: A human-annotated dialogue dataset for abstractive summarization," in *Proc. 2nd Workshop New Front. Summ.*, 2019, pp. 70–79.

[6] H. Yukhymenko, R. Staab, M. Vero, and M. Vechev, "A synthetic dataset for personal attribute inference," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 37, 2024, pp. 120 735–120 779.

[7] N. Lukas, A. Salem, R. Sim, S. Tople, L. Wutschitz, and S. Zanella-Béguelin, "Analyzing leakage of personally identifiable information in language models," in *Proc. IEEE Symp. Secur. Priv. (SP)*, 2023, pp. 346–363.

[8] E. A. Rocamora, Y. Wu, F. Liu, G. Chrysos, and V. Cevher, "Revisiting character-level adversarial attacks for language models," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2024, pp. 1–12.