DP^3: Differentially Private Prompt Perturbation for Multi-turn LLM Inference

Lele Zheng, Chao Zhang, Feiyang Yuan, Ke Cheng, Tao Zhang*, Anxiao Song, Yulong Shen
School of Computer Science and Technology, Xidian University, Xi'an, China
{zhenglele, taozhang}@xidian.edu.cn

Abstract

Large language models (LLMs) are widely used for text understanding and generation, with increasing deployment in applications involving sensitive user inputs. This raises significant privacy concerns, motivating the adoption of differential privacy (DP) to protect prompts during LLM inference. However, most existing DP methods assume single-turn interactions, whereas real-world usage often relies on multi-turn dialogue. Consequently, these single-turn-based methods break down in multi-turn settings, where recurring tokens repeatedly consume the privacy budget under DP, leading to accumulated privacy loss and degraded cross-turn semantic coherence. To address these challenges, we propose DP^3, a differentially private prompt perturbation framework for multi-turn LLM inference. DP^3 constructs a perturbation mapping table to reuse perturbations for recurring tokens, reducing redundant privacy costs. It also defines a context-aware utility function that combines embedding distance with attention-based contextual representations to maintain semantic consistency across turns. Additionally, DP^3 introduces a two-stage bucketed exponential mechanism to manage long-tail phenomena in large candidate spaces. Experimental results on multi-turn dialogue tasks demonstrate that DP^3 offers a better privacy-utility trade-off and stronger resistance to inference attacks compared to existing methods. Our code is publicly available at https://github.com/XidianNSS/DP3.

1 Introduction

Large language models (LLMs) (Radford and Narasimhan, 2018; Brown et al., 2020; Zhang et al., 2025), represented by systems such as ChatGPT, have been widely adopted for text understanding and generation tasks. In practice, users rarely obtain satisfactory solutions from a single query and instead engage in multi-turn dialogue to refine the problem formulation, introduce additional constraints, correct earlier misunderstandings, and build upon intermediate results. Such multi-turn interaction is crucial for high-quality reasoning (Chen et al., 2025; Zhang et al., 2018), as later turns rely on accumulated context to maintain coherence and support consistent inference.

In most commercial deployments, LLMs are accessed through black-box APIs (Sun et al., 2024), where users submit prompts and receive generated responses without visibility into the underlying model internals. This deployment paradigm introduces significant privacy risks, as user prompts may contain sensitive personal information, confidential business data, or proprietary knowledge (Ahmadian and Marinescu, 2018). Moreover, prior studies (Carlini et al., 2021; Kandpal et al., 2022) have shown that LLMs can memorize user inputs and may inadvertently reveal sensitive content in later dialogue turns or future interactions. These risks are further amplified by increasingly stringent data privacy regulations, which impose strict requirements on how user data is collected, processed, and stored. Consequently, protecting the privacy of user prompts has become a critical requirement for the trustworthy deployment of LLM systems.

Differential privacy (DP) (Dwork, 2006) provides a provable framework for protecting sensitive user information. Some works have explored applying DP (Yan et al., 2025) to safeguard user prompts in LLM systems. In black-box inference settings, DP can be achieved by locally perturbing user prompts on the client side, enabling plug-and-play deployment without modifying the underlying model. For example, CUSTEXT (Chen et al., 2023) and InferDPT (Tong et al., 2025) perform token-level perturbation using the exponential mechanism and adjacency-based substitutions, while CAPE (Plant et al., 2021) improves utility through a context-aware utility function and buck-

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 18538–18552

eted sampling. However, existing methods typically assume a single-turn dialogue between the user and the LLM, where a satisfactory result is obtained from a single query. Under this assumption, each prompt is perturbed independently.

Compared with single-turn inputs, multi-turn dialogue exhibits distinct characteristics in practice, where prompts evolve over successive turns and often repeat or reference earlier context. Users frequently revisit key entities across turns, and successive prompts are semantically linked through shared conversational context. As a result, directly applying single-turn prompt perturbation mechanisms to multi-turn inference raises two major challenges. First, privacy loss accumulates with the number of dialogue turns. By the sequential composition property of differential privacy, independently perturbing each turn repeatedly rerandomizes recurring content, causing the overall privacy loss to grow across interactions and potentially exceed acceptable limits in long dialogues. Second, independent perturbations disrupt cross-turn semantic coherence. Since multi-turn dialogue relies on historical context to resolve references and maintain user intent (Zheng et al., 2024), perturbing each turn in isolation can weaken semantic alignment between turns, degrading reference resolution and reasoning quality in subsequent interactions.

To address the above challenges, we propose DP^3, a differentially private prompt perturbation framework for multi-turn LLM inference. DP^3 is designed to account for the cross-turn dependencies inherent in multi-turn dialogue, rather than treating each prompt in isolation. To mitigate privacy loss accumulation, DP^3 constructs a perturbation mapping table that reuses perturbations for recurring tokens across turns, preventing redundant consumption of privacy budget. To preserve cross-turn semantic coherence, DP^3 incorporates a context-aware utility function that leverages both token embeddings and attention-derived contextual representations, encouraging consistent perturbations for semantically related content. In addition, DP^3 adopts a two-stage bucketed exponential mechanism to improve the stability of perturbation selection in large candidate spaces. Extensive experiments across multiple multi-turn dialogue tasks, along with ablation studies, show that DP^3 consistently achieves a more favorable privacy–utility trade-off than existing prompt perturbation methods, particularly in long-horizon multi-turn settings, while maintaining coherent and high-quality model responses.

2 Related Work

Existing prompt protection methods typically employ privacy-enhancing technologies to safeguard sensitive user information, often facing inherent trade-offs among efficiency, utility, and privacy. Table 1 compares the respective advantages and disadvantages of various methods.

Cryptography-based Approaches. Cryptographic techniques provide strong confidentiality guarantees under standard assumptions and have been used to support secure model inference (Katz and Lindell, 2007; Bellare and Rogaway, 2005). For example, CipherGPT (Hou et al., 2023) leverages homomorphic encryption to perform Transformer-style inference directly on encrypted prompts (Brakerski and Vaikuntanathan, 2014; Brakerski et al., 2014). Despite strong privacy guarantees, such cryptographic approaches typically incur substantial computational and communication overhead, which can make real-time or large-scale deployment challenging.

Client-server Hybrid Architectures. Another line of work adopts a client-side split inference (Shen et al., 2025; Mai et al., 2024), where part of the model is executed on the user device. TextObfuscator (Zhou et al., 2023) and DP-Forward (Du et al., 2023) exemplify this approach by injecting calibrated noise into continuous representation space to preserve privacy. Such continuous space perturbations often retain more semantic information than discrete token substitution, leading to higher utility. However, these methods require partial access to model components or parameters, which is typically unavailable under proprietary, black-box API deployments.

Method	Black-box	Inference Efficiency		Multi-turn
CipherGPT	✓	✓	○	✗
TextObfuscator	✗	✓	◐	✗
DP-Forward	✗	✓	◐	✗
SANTEXT	✓	✗	●	✗
CUSTEXT	✓	✗	●	✗
InferDPT	✓	✓	●	✗
CAPE	✓	✓	●	✗
DP^3	✓	✓	●	✓

Table 1: Comparison of different methods. ●, ◐, and ○ represent high, medium, and low levels, respectively, while ✓ denotes that the corresponding capability is supported.

DP-based Approaches. In black-box settings, differential privacy methods are widely used for prompt protection, as they can be applied on the client side without requiring access to or modification of the underlying model. However, most existing approaches are designed for single-turn interactions, which limits their effectiveness in multi-turn dialogues. SANTEXT (Yue et al., 2021) relies on metric-LDP with the entire vocabulary as the candidate space, leading to severe utility degradation, while CUSTEXT (Chen et al., 2023) and InferDPT (Tong et al., 2025) reduce the candidate space via adjacency lists at the cost of weaker privacy guarantees. CAPE (Plant et al., 2021) improves utility through context-aware perturbation and bucketed sampling, but applying such single-turn methods independently across multiple turns results in repeated perturbations and cumulative privacy loss. In contrast, DP^3 introduces a perturbation mapping table and a context-aware utility function, effectively reducing global privacy consumption while preserving semantic consistency in multi-turn interactions.

3 Preliminaries

3.1 Differential Privacy

Differential privacy (DP) (Dwork, 2006) provides a rigorous framework for protecting sensitive user information by limiting the influence of any individual input on the output of a randomized mechanism. We focus on ε-local differential privacy (ε-LDP), which is particularly suitable for black-box LLM inference settings. Under ε-LDP, each user locally perturbs their prompt using a randomized mechanism $\mathcal{M}(\cdot)$ before sending it to an untrusted server, ensuring privacy protection without requiring access to or modification of the underlying model.

Definition 1. (ε-Local Differential Privacy (Raskhodnikova et al., 2008)) *Given a privacy parameter $\varepsilon \geq 0$, a randomized mechanism $\mathcal{M}$ is said to satisfy ε-local differential privacy if for any two inputs $x, x' \in \mathcal{X}$ and any possible output $y \in \mathcal{Y}$, the following holds:*

$$\frac{\Pr[\mathcal{M}(x) = y]}{\Pr[\mathcal{M}(x') = y]} \leq e^{\varepsilon}, \tag{1}$$

where ε denotes the privacy budget. A smaller value of ε indicates stronger privacy protection, but it also typically reduces the utility of the output.

Definition 2. (Exponential Mechanism (EM) (McSherry and Talwar, 2007)). *Given a utility function $u : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, a randomized mechanism $M(\cdot)$ satisfies ε-LDP if it follows the rule:*

$$\Pr[y \mid x] \propto \exp\left(\frac{\varepsilon \cdot u(x, y)}{2\Delta_u}\right), \tag{2}$$

where Δ_u is the sensitivity of the utility function $u(x, y)$, defined as:

$$\Delta_u = \max_{x, x' \in \mathcal{X}, y \in \mathcal{Y}} \left|u(x, y) - u(x', y)\right|. \tag{3}$$

The sensitivity Δ_u measures how much the utility function can change when the input changes.

3.2 Problem Setup and Threat Model

Problem Setup. We consider the problem of privacy-preserving multi-turn LLM inference in a black-box setting. A multi-turn dialogue consists of a sequence of turns $X = [x_1, x_2, ..., x_T]$, where x_t denotes the user prompt at turn t. Each prompt is composed of a sequence of tokens drawn from a vocabulary $\mathcal{V}$. In multi-turn dialogue, prompts are not independent: later prompts often reuse tokens, entities, or expressions from earlier turns, and their semantics are conditioned on the accumulated dialogue context.

Threat Model. We assume a black-box inference setting, where the underlying LLM is accessed via an external API. Prompt perturbation is performed on the client-side, and the perturbed prompt is sent to the server only after the local perturbation process is completed. The server hosting the model is considered untrusted and may observe only the perturbed prompts submitted by the user, but does not reveal model internals, gradients, or parameters. We do not assume any control over the model or its training process, and privacy protection must be achieved without modifying the LLM itself.

Goal. Our goal is to design a prompt perturbation mechanism for multi-turn LLM inference that provides formal ε-local differential privacy guarantees while preserving the utility of multi-turn dialogue. Specifically, the mechanism should (1) protect user prompts under a black-box inference setting without modifying the underlying LLM, (2) avoid excessive privacy loss accumulation across dialogue turns, and (3) maintain cross-turn semantic coherence to support consistent and high-quality multi-step reasoning. The resulting mechanism should be practical for real-world deployment and applicable to long-horizon multi-turn interactions.

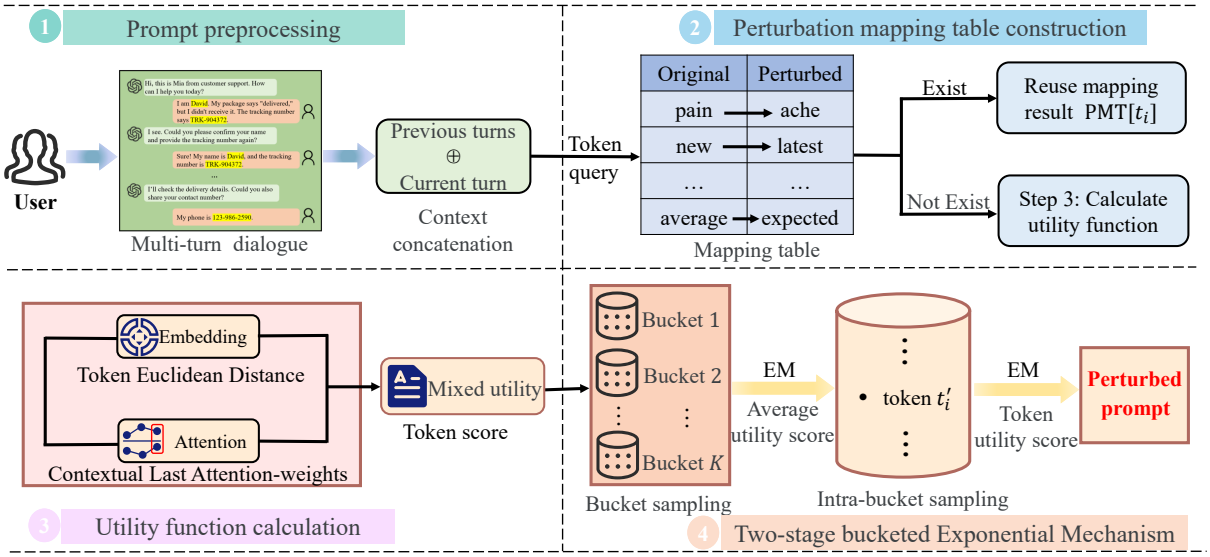


Figure 1: An overview of the proposed DP^3 framework. For each dialogue turn, DP^3 concatenates the current prompt with the dialogue history, computes utility scores for unseen tokens via PMT lookup, and applies two-stage bucketed sampling to generate a privatized prompt for black-box LLM inference.

4 Method

4.1 Method Overview

We present DP^3, a differentially private prompt perturbation framework in multi-turn dialogue, designed to address privacy budget accumulation and cross-turn semantic inconsistency caused by independently perturbing each turn. As shown in Fig. 1, DP^3 processes each dialogue turn by jointly considering the dialogue history and the current user input. A persistent *perturbation mapping table* (PMT) is maintained across turns to ensure that recurring tokens are perturbed only once and consistently reused in subsequent turns, thereby avoiding redundant privacy consumption. For tokens that appear for the first time, DP^3 selects perturbations on demand using a token-level, context-aware utility function that integrates semantic similarity in the embedding space with attention-informed contextual relevance. To improve robustness when the candidate space is large and long-tailed, DP^3 further adopts a two-stage bucketed exponential mechanism, which stabilizes sampling by first selecting a bucket and then sampling within it. By perturbing only newly introduced tokens while preserving consistent mappings for recurring ones, DP^3 effectively balances formal privacy guarantees with semantic coherence in multi-turn LLM inference.

4.2 Perturbation Mapping Table Construction

In multi-turn dialogue, directly applying a single-turn ε_0-LDP perturbation at each turn leads to privacy budget accumulation under sequential composition. When prompts are perturbed independently across turns, tokens that recur in multiple turns are repeatedly randomized. By the sequential composition theorem, if a dialogue lasts for T turns and each turn applies an ε_0-LDP mechanism, the overall privacy budget is upper-bounded by $\varepsilon_{T,trad} \leq T \cdot \varepsilon_0$, which can quickly exceed acceptable limits in long dialogues.

To mitigate this issue, we introduce a *first-occurrence perturbation* scheme based on a *perturbation mapping table* (PMT). Instead of perturbing tokens at every turn, DP^3 perturbs each token only at its first occurrence in the dialogue and stores the resulting mapping in the PMT. In subsequent turns, recurring tokens directly reuse their stored mappings, while only newly introduced tokens are perturbed and added to the PMT. This design prevents repeated privacy spending on overlapping content and decouples privacy loss from the number of dialogue turns.

Formally, for a token t_i appearing in the current prompt, its perturbed output $\tilde{t}_i$ is determined as

$$\tilde{t}_i = \begin{cases} \text{PMT}[t_i], & \text{if } t_i \in \text{PMT}, \\ R(t_i), & \text{if } t_i \notin \text{PMT}. \end{cases} \tag{4}$$

If t_i has appeared before, its perturbed form is directly retrieved from the perturbation mapping table. Otherwise, when t_i appears for the first time, it is perturbed via an operation $R(\cdot)$, and the resulting mapping is stored in the PMT for future

reuse. Specifically, $R(\cdot)$ consists of two steps: (1) Attention-Driven Candidate Generation, which constructs a context-aware candidate set for the token, and (2) a Two-stage Bucketed Exponential Mechanism, which selects the final perturbed token while balancing privacy and utility. This design ensures that privacy perturbation is performed only once per token, thereby avoiding redundant privacy consumption while maintaining cross-turn consistency.

Under this scheme, tokens that appear only once incur the same privacy cost as in standard single-turn perturbation. For tokens that recur across turns, perturbation is performed only once, and subsequent reuse does not introduce additional randomization. As a result, the privacy loss contributed by each token is bounded by $R(\cdot)$, and the overall privacy budget depends on the number of *distinct* tokens perturbed during the dialogue rather than the number of turns.

Theorem 4.1 (Privacy Bound of PMT)**.** *Consider a multi-turn dialogue with T turns, where a baseline approach perturbs each turn independently using an ε_0-LDP mechanism. By the sequential composition property of differential privacy, the resulting privacy budget is $\varepsilon_{T,\,trad} = T \cdot \varepsilon_0$. Under the proposed PMT scheme, in which each distinct token is perturbed at most once and subsequent occurrences reuse the same perturbed output, the overall mechanism satisfies $\varepsilon_{T,\,PMT}$-local differential privacy with*

$$\varepsilon_{T,\,PMT} \leq \varepsilon_{T,\,trad}. \tag{5}$$

The proof is provided in Appendix A.1.

4.3 Attention-Driven Candidate Generation

In multi-turn dialogue, token semantics are inherently context-dependent and may evolve across turns. However, many existing candidate generation strategies rely solely on static embedding similarity, which is insufficient for preserving cross-turn semantic coherence. To address this limitation, we introduce an attention-weighted, context-aware candidate generation mechanism that leverages Transformer-based contextual representations.

At the dialogue turn j, we concatenate the dialogue history $C_j = [x_1, x_2, \ldots, x_{j-1}]$ with the current prompt x_j to form a contextualized input sequence $C_j \oplus x_j$. We then obtain contextual token representations using a lightweight local Transformer encoder and extract the last-layer attention tensor $\mathbf{A}^{(L)} \in \mathbb{R}^{H\times n\times n}$, where H denotes the number of attention heads and n is the sequence length. To derive a stable measure of contextual relevance, we average attention weights across heads and normalize them:

$$\bar{\mathbf{A}}_{ij} = \frac{1}{H}\sum_{h=1}^{H} \mathbf{A}^{(L)}_{h,ij}, \quad \alpha_{ij} = \frac{\exp(\bar{\mathbf{A}}_{ij})}{\sum_n \exp(\bar{\mathbf{A}}_{in})}. \tag{6}$$

Based on the normalized attention weights, we construct a *contextual aggregation representation* for token t_i as follows:

$$CE(t_i) = \sum_{j=1}^{n} \alpha_{ij} \cdot h_j, \tag{7}$$

where h_j denotes the hidden state of token t_j. Unlike static token embeddings, this representation captures the contextual semantics that the model attends to when interpreting the token, thereby reflecting cross-turn dependencies.

In addition to contextual relevance, we compute a static embedding similarity s_e between token t_i and any candidate token $t_i' \in v$ using the Euclidean distance between their embeddings:

$$s_{\mathrm{e}}(t_i, t_i') = \frac{1}{1 + \|e_{t_i} - e_{t_i'}\|_2}. \tag{8}$$

where $e_{(\cdot)}$ denotes the token embedding, v denotes the set of all candidate tokens. We further define a context-aware similarity score based on the cosine similarity between their contextual aggregation representations:

$$s_{\mathrm{ctx}}(t_i, t_i') = \cos\left(CE(t_i), CE(t_i')\right). \tag{9}$$

Finally, we combine static semantic similarity and dynamic contextual relevance into a unified utility function:

$$u_t(t_i, t_i') = \alpha \cdot s_{\mathrm{e}}(t_i, t_i') + (1-\alpha) \cdot s_{\mathrm{ctx}}(t_i, t_i'), \tag{10}$$

where α controls the balance between embedding-level semantics and contextual information.

4.4 Two-stage Bucketed EM

When the candidate space spans the full vocabulary, directly applying the exponential mechanism (EM) can lead to degraded utility under long-tailed utility distributions. In such cases, many low-utility candidate tokens have individually negligible probabilities, but their collective probability mass can be substantial, causing probability to be allocated

to semantically implausible candidates and reducing the expected utility of the sampled output. This issue is common in large-vocabulary LLM inference. To address this challenge, we adopt a two-stage bucketed exponential mechanism that performs hierarchical sampling by first selecting a utility bucket and then sampling within it, thereby limiting the cumulative influence of low-utility candidates while preserving semantic plausibility.

Bucket Construction and Utility Aggregation. Given a token t_i, we partition its candidate tokens into K buckets based on their utility values $u_t(t_i, t_i')$. Specifically, the utility range $[u_{\min}, u_{\max}]$ is uniformly divided into K intervals, and each candidate token t_i' is assigned to a bucket $\mathcal{B}_k$ according to its utility score. This bucketization provides a coarse-grained organization of the candidate space that separates high-utility regions from the long tail.

For each bucket $\mathcal{B}_k$, we define a bucket-level utility score as the average utility of the tokens within the bucket:

$$U(\mathcal{B}_k) = \frac{1}{|\mathcal{B}_k|} \sum_{t_i' \in \mathcal{B}_k} u_t(t_i, t_i'). \tag{11}$$

Bucket-level Sampling. We first select a target bucket $\mathcal{B}^*$ using the exponential mechanism based on the bucket-level utilities:

$$\Pr[\mathcal{B}^* = \mathcal{B}_k] \propto \exp\left(\frac{\varepsilon_1 \cdot U(\mathcal{B}_k)}{2\Delta U}\right), \tag{12}$$

where ε_1 denotes the privacy budget allocated to bucket selection and ΔU is the global sensitivity of the bucket utility function. This stage concentrates probability mass on semantically plausible regions of the candidate space and reduces the cumulative effect of a large number of low-utility candidates.

Intra-bucket Sampling. Conditioned on the selected bucket $\mathcal{B}^*$, we apply the exponential mechanism again to sample the final perturbed token:

$$\Pr[\tilde{t}_i = t_i' \mid \mathcal{B}^*] \propto \exp\left(\frac{\varepsilon_2 \cdot u_t(t_i, t_i')}{2\Delta u}\right), \tag{13}$$

where ε_2 is the privacy budget for intra-bucket sampling and Δu denotes the global sensitivity. Restricting sampling to a single bucket preserves fine-grained semantic relevance while avoiding interference from the long tail.

Overall, the two-stage procedure mitigates utility degradation caused by long-tailed candidate distributions through hierarchical sampling. By allocating the privacy budget such that $\varepsilon_1 + \varepsilon_2 = \varepsilon_T$, the mechanism satisfies the sequential composition property of differential privacy.

Theorem 4.2 (Differential Privacy Guarantees)**.** *The proposed two-stage bucketed exponential mechanism satisfies $(\varepsilon_1 + \varepsilon_2)$-differential privacy.*

The proof is provided in Appendix A.2.

5 Experiments

5.1 Experiment Setup

Datasets. For conversational question answering, we use the MT-Bench-101 (MT) benchmark (Bai et al., 2024), which includes Mathematical Reasoning (MR) and General Reasoning (GR) tasks for evaluating multi-turn dialogue capabilities of LLMs. In addition, we adopt the hard version of CHARP (hCHARP) (Ghaddar et al., 2024), which assesses a model's ability to leverage dialogue history and mitigate hallucinations. MR and GR focus on collaborative multi-turn problem solving in mathematical and general reasoning domains, respectively, while hCHARP is designed to evaluate historical reasoning in dialogue systems. The dataset can be found in Appendix B.7.

Metrics. We use the widely adopted pre-trained model Qwen3-Max as a black-box LLM for zero-shot answering and also use Qwen3-Max to evaluate the accuracy of the model's responses. Importantly, Qwen3-Max is not used to subjectively judge answer quality. Instead, it serves as a *semantic equivalence checker* that compares the model's response against the task-provided reference answer. Concretely, we compute utility as the *accuracy* of whether the response is semantically consistent with the gold answer under a fixed verification prompt (Appendix B.1). Additionally, we evaluate privacy under KNN and BERT inference attacks. The attack success rate is denoted as r_{ats}, and the privacy protection level of the different mechanisms is defined as $1 - r_{\text{ats}}$. Detailed definitions and implementations are provided in Appendix B.4 and Appendix B.5.

Baseline. Since existing differential privacy mechanisms for prompt perturbation are primarily developed for single-turn interactions, we compare DP^3 against several representative single-turn privacy-preserving methods that are adapted

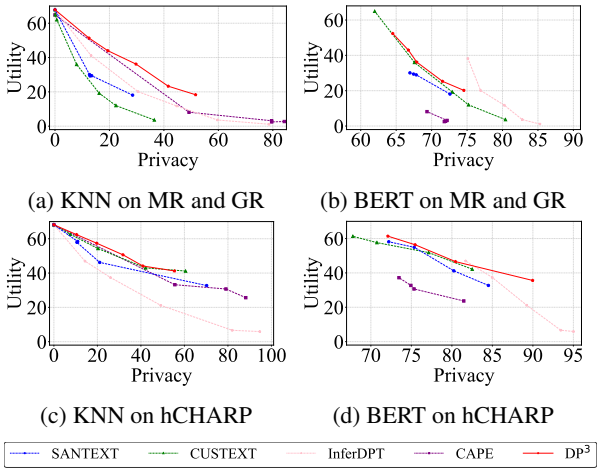


Figure 2: The privacy-utility trade-off between privacy attacks and accuracy, with the privacy budget ε varied within the interval $[1, 20]$, and privacy quantified by privacy scores obtained under empirical attack scenarios.

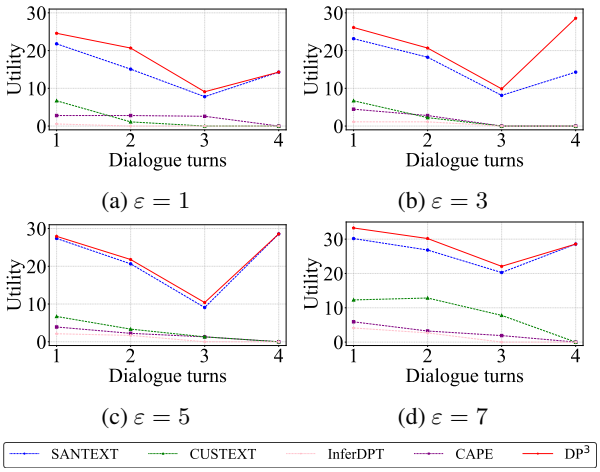


Figure 3: The relationship between model accuracy and the number of dialogue turns on MR and GR, under fixed privacy budgets $\varepsilon \in \{1, 3, 5, 7\}$.

to multi-turn dialogue settings. Specifically, we include SANTEXT (Yue et al., 2021), CUSTEXT (Chen et al., 2023), InferDPT (Tong et al., 2025), and CAPE (Plant et al., 2021). For a fair comparison, all baseline methods are applied under the same black-box inference setting and privacy budget constraints.

Implementation. All baseline methods are evaluated using their default configurations. For DP^3, we set the balance factor $\alpha = 0.7$ and the bucket size $K = 50$ in the experiments unless otherwise specified. All results are averaged over 10 independent runs to reduce randomness and improve reliability. More detailed experimental settings and implementation details are provided in Appendix B.6.

5.2 Experiment Evaluation

5.2.1 Privacy-Utility Trade-off

Fig. 2 shows the privacy-utility trade-off of DP^3 and baseline methods under the KNN and BERT inference attacks. The X-axis represents privacy, which is denoted by $1 - r_{\text{ats}}$ under the corresponding attack. A curve closer to the top-right corner indicates a better trade-off. Across all settings, DP^3 consistently outperforms prior baselines. Under the KNN privacy metric in Fig. 2a and Fig. 2c, DP^3 achieves the best privacy-utility trade-off among perturbation methods. Under the BERT inference attack in Fig. 2b and Fig. 2d, DP^3 also performs favorably in most cases. The BERT-based evaluation is sensitive to vocabulary and tokenization mismatches across methods, including subword tokens used in InferDPT, word-level tokens in SANTEXT and CUSTEXT (using GloVe), and the vocabulary $\mathcal{V} = cl100\_embeddings$ used in DP^3. These mismatches may distort the attack success rate and limit cross-method comparability. Detailed privacy-utility experiments across different dialogue rounds are available in Appendix B.2.

Fig. 3 shows utility over dialogue turns under fixed theoretical privacy budgets for each method. As dialogue turns increase, utility declines monotonically for all methods, reflecting the accumulation of perturbation-induced semantic drift in multi-turn contexts. DP^3 consistently achieves higher utility and slower degradation compared to baselines, indicating better preservation of contextual coherence and a more favorable privacy–utility trade-off in long-horizon dialogues. Examples of perturbations under different privacy budgets for various methods can be found in Appendix B.3.

5.2.2 Influence of Privacy Budget

In this section, we investigate the influence of varying privacy budgets on model performance using the hCHARP dataset. We evaluate the semantic similarity between the original and perturbed prompts by calculating the average Rouge-L F1 score, as reported in Table 2. The results show that all methods achieve higher accuracy as the privacy budget ε increases, with DP^3 consistently outperforming SANTEXT, InferDPT, and CAPE.

Specifically, the privacy budget relationships for each method are as follows: For SANTEXT, which relies on ε-metric-LDP, the effective privacy budget scales with the number of tokens, where

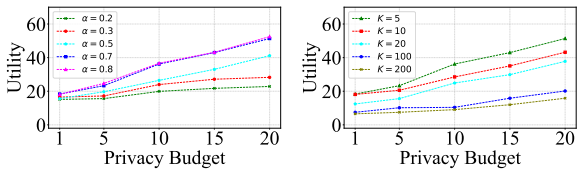
$\varepsilon' = \varepsilon \cdot d_{max}$, with $d_{max} \approx 14.86$. In InferDPT, a random adjacency list is generated using Laplace noise, with a default budget of approximately 9, leading to an actual privacy budget of $\varepsilon' \sim \varepsilon + 9$. CAPE operates with $\varepsilon' = \varepsilon + \ln\left(\max_{i,j} \frac{|b_i|}{|b_j|}\right) \sim \varepsilon + 8$. In contrast, DP^3 evenly splits the privacy budget between bucket sampling and intra-bucket sampling, yielding an effective privacy budget of $\varepsilon' \sim \varepsilon_1 + \varepsilon_2$, where $\varepsilon_1 = \varepsilon_2 = \frac{\varepsilon'}{2}$. CUSTEXT uses a static adjacency list of size 20, making it difficult to quantify the actual privacy budget, and therefore it is excluded from the comparison. Overall, DP^3 achieves a better privacy-utility trade-off, showing superior utility across various privacy budgets.

Method	Rouge-L (F1) ↑			
	$\varepsilon = 10$	$\varepsilon = 14$	$\varepsilon = 16$	$\varepsilon = 18$
SANTEXT	74.30	79.48	81.45	83.15
InferDPT	19.93	21.57	29.67	37.27
CAPE	77.06	77.17	77.32	77.98
DP^3	81.74	83.80	85.23	86.39

Table 2: Sentence-level similarity of various methods.

5.2.3 Ablation Studies

We further evaluate DP^3 on the MT dataset using different parameter configurations. Specifically, we adjust the balance factor α in the token utility function and the number of buckets K during bucket sampling. Privacy budget ε is used as the privacy metric, and we evaluate model performance using answer accuracy as the utility metric.



(a) $\alpha \in \{0.2, 0.3, 0.5, 0.7, 0.8\}$ (b) $K \in \{5, 10, 20, 100, 200\}$

Figure 4: Ablation results under various parameter configurations with $\varepsilon \in [1, 20]$.

Effect of the balance factor α. To evaluate the impact of α, which balances Euclidean embedding distance and contextual attention in the utility function, we fix the bucket size at $K = 5$ and vary $\alpha \in \{0.2, 0.3, 0.5, 0.7, 0.8\}$. As shown in Fig. 4a, utility improves with increasing α, indicating that embedding similarity plays a significant role in preserving task performance. However, after α reaching approximately 0.7, further increases yield only marginal gains and lead to a rapid rise in attack success rate, while the utility only improves slightly. This suggests that $\alpha = 0.7$ provides an optimal balance between privacy protection and system utility, without diminishing semantic coherence.

Effect of the bucket size K. We also explore the effect of bucket size K by varying it in the range $K \in \{5, 10, 20, 100, 200\}$. As shown in Fig. 4b, smaller bucket sizes lead to significantly higher utility, as they restrict sampling to more semantically coherent regions of the candidate space. Larger K values reduce the effectiveness of bucket sampling, as they allow more low-utility candidates to contribute to the overall probability mass, diminishing the benefits of fine-grained sampling. These results demonstrate the importance of bucket granularity in mitigating long-tail effects and stabilizing perturbation quality, highlighting the critical role of bucket size in the performance of DP^3.

Client-side computational overhead. We further evaluate the client-side computational overhead of DP^3, which employs a lightweight local Transformer to generate context-aware representations for unseen tokens. As reported in Table 3, large candidate vocabularies result in comparable runtime memory consumption across most methods. Specifically, SANTEXT, CUSTEXT, CAPE, and DP^3 require between 30 and 32 GB of memory, whereas InferDPT consumes substantially more at 42.67 GB. GPU memory usage ranges from 5487.7 to 5767.1 MB for the majority of methods, increasing to 8386 MB for InferDPT. The Perturbation Time Ratio (PTR) quantifies the proportion of local prompt perturbation time relative to the total end-to-end runtime. Due to the need for extensive computations over large candidate spaces, SANTEXT, CUSTEXT, and InferDPT incur considerable perturbation overhead, yielding PTRs of 82.71%, 95.54%, and 90.60%, respectively. In contrast, both CAPE and DP^3 leverage a local BERT-base model with approximately 110 million parameters. CAPE achieves the lowest PTR at 12.38%, while DP^3 maintains a moderate ratio of 58.21%.

Effect of applying PMT to baselines. We further investigate whether simply applying PMT to baseline methods can match the gains of DP^3. Table 4 reports each method's accuracy with and without PMT on MT-Bench-101 under $\epsilon = 1$ and $K = 5$. Adding PMT yields marginal utility im-

Method	Memory	VRAM	PTR
SANTEXT	31.89GB	5504.5MiB	82.71%
CUSTEXT	30.86GB	5487.7MiB	95.54%
InferDPT	42.67GB	8386MiB	90.60%
CAPE	30.84GB	5487.7MiB	12.38%
DP^3	31.44GB	5767.1MiB	58.21%

Table 3: Comparison of computational overhead across different methods.

Method	without PMT	with PMT
SANTEXT	15.83%	18.10%
CUSTEXT	3.62%	4.07%
InferDPT	1.13%	1.13%
CAPE	2.49%	2.71%
DP^3	1.99%	18.33%

Table 4: Comparison of baseline methods with and without the PMT mechanism on MT-Bench-101.

provements for most baselines: SANTEXT improves from 15.83% to 18.10%, CUSTEXT from 3.62% to 4.07%, and InferDPT shows no gain. In contrast, DP^3 improves substantially from 1.99% to 18.33% with PMT, demonstrating PMT's greater importance when combined with the context-aware utility function and two-stage bucketed EM. The effectiveness of PMT heavily relies on the quality of the underlying perturbation: if the initial perturbation selects appropriate semantically similar candidates, PMT can effectively maintain cross-turn semantic consistency.

6 Conclusion

In this paper, we present DP^3, a differentially private prompt protection framework for black-box multi-turn LLM inference. DP^3 addresses privacy budget accumulation and cross-turn semantic inconsistency by reusing perturbations through a perturbation mapping table and guiding token selection with an attention-informed, context-aware utility, together with a two-stage bucketed exponential mechanism for stable sampling. Experiments on multi-turn dialogue reasoning benchmarks show that DP^3 achieves a more favorable privacy–utility trade-off and stronger resistance to inference attacks than existing baselines, with ablation studies validating the contribution of each component. Overall, DP^3 provides a practical foundation for deploying privacy-preserving and trustworthy multi-turn conversational LLM systems.

Limitations

This work has several limitations that suggest directions for future research.

PMT Robustness. The effectiveness of the perturbation mapping table (PMT) relies on the reuse of the same token forms across dialogue turns. When users frequently paraphrase, switch languages, or express entities in diverse forms, such reuse becomes less reliable. In these cases, PMT may not be able to match tokens correctly, reducing its ability to save privacy budget and weakening its advantage over turn-wise independent perturbation.

Error Propagation. The reuse mechanism in PMT may also propagate early mistakes. If a token is replaced with an inappropriate alternative at its first occurrence, the same replacement will be reused in later turns. This can gradually change the meaning of the dialogue and lead to inconsistencies across turns, especially in longer conversations where later responses depend on earlier context.

Computational Overhead. DP^3 uses a lightweight local Transformer to obtain contextual representations and guide token replacement. While this helps maintain consistency across turns, it introduces additional computation on the client side. In practice, the performance of this component may vary depending on the local model, tokenizer, and type of input data.

Security Considerations. The deterministic token-level mapping in PMT may introduce potential risks under long-term observation. For example, if the same tokens are consistently mapped to the same outputs, an attacker may infer patterns based on frequency statistics. Although such attacks require long dialogue histories and are less common in practice, this remains a potential limitation of the current design.

Acknowledgments

This paper is supported in part by the National Natural Science Foundation of China (62572373, 62220106004), in part by the Fundamental Research Funds for the Central Universities (QTZX26042).

References

Mohammad Ahmadian and Dan C Marinescu. 2018. Information leakage in cloud data warehouses. *IEEE Transactions on Sustainable Computing*, 5(2):192–203.

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, and Wanli Ouyang. 2024. MT-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7421–7454, Bangkok, Thailand. Association for Computational Linguistics.

Mihir Bellare and Phillip Rogaway. 2005. Introduction to modern cryptography. *Ucsd Cse*, 207:207.

Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. 2014. (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):1–36.

Zvika Brakerski and Vinod Vaikuntanathan. 2014. Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on computing*, 43(2):831–871.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association.

Jiawei Chen, Xinyan Guan, Qianhao Yuan, Mo Guozhao, Weixiang Zhou, Yaojie Lu, Hongyu Lin, Ben He, Le Sun, and Xianpei Han. 2025. Consistentchat: Building skeleton-guided consistent multi-turn dialogues for large language models from scratch. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 8426–8452.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Minxin Du, Xiang Yue, Sherman SM Chow, Tianhao Wang, Chenyu Huang, and Huan Sun. 2023. Dp-forward: Fine-tuning and inference on language models with differential privacy in forward pass. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 2665–2679.

Cynthia Dwork. 2006. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer.

Abbas Ghaddar, David Alfonso-Hermelo, Philippe Langlais, Mehdi Rezagholizadeh, Boxing Chen, and Prasanna Parthasarathi. 2024. CHARP: Conversation history AwaReness probing for knowledge-grounded dialogue systems. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1534–1551, Bangkok, Thailand. Association for Computational Linguistics.

Xiaoyang Hou, Jian Liu, Jingyu Li, Yuhan Li, Wen-jie Lu, Cheng Hong, and Kui Ren. 2023. Ciphergpt: Secure two-party gpt inference. *Cryptology ePrint Archive*.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*, pages 10697–10707. PMLR.

Jonathan Katz and Yehuda Lindell. 2007. *Introduction to modern cryptography: principles and protocols*. Chapman and hall/CRC.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

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

Frank McSherry and Kunal Talwar. 2007. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE.

Richard Plant, Dimitra Gkatzia, and Valerio Giuffrida. 2021. CAPE: Context-aware private embeddings for private language learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7970–7978, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Sofya Raskhodnikova, Adam Smith, Homin K Lee, Kobbi Nissim, and Shiva Prasad Kasiviswanathan. 2008. What can we learn privately. In *Proceedings of the 54th Annual Symposium on Foundations of Computer Science*, pages 531–540.

Xicong Shen, Yang Liu, Yi Liu, Peiran Wang, Huiqi Liu, Jue Hong, Bing Duan, Zirui Huang, Yunlong Mao, Ye Wu, and Sheng Zhong. 2025. Sap: Privacy-preserving fine-tuning on language models with split-and-privatize framework. In *International Joint Conference on Artificial Intelligence*.

Congzheng Song and Ananth Raghunathan. 2020. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 377–390.

Haotian Sun, Yuchen Zhuang, Wei Wei, Chao Zhang, and Bo Dai. 2024. Bbox-adapter: Lightweight adapting for black-box large language models. In *International Conference on Machine Learning*.

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

Jun Yan, Yijun Zhang, Laifeng Lu, Yi Tian, and Yihui Zhou. 2025. A graph generating method based on local differential privacy for preserving link relationships of social networks. *Journal of Networking and Network Applications*, 4(4):145–156.

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

Tao Zhang, Chao Zhang, Feiyang Yuan, Lele Zheng, and Yiyun Guo. 2025. Alternating aggregation low-rank adaptation approach for federated large models. In *International Conference on Advanced Data Mining and Applications*, pages 418–425. Springer.

Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018. Modeling multi-turn conversation with deep utterance aggregation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3740–3752, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Chunyan Zheng, Keke Sun, Wenhao Zhao, Haibo Zhou, Lixing Jiang, Shaoyang Song, and Chunlai Zhou. 2024. Locally differentially private in-context learning. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10686–10697.

Xin Zhou, Yi Lu, Ruotian Ma, Tao Gui, Yuran Wang, Yong Ding, Yibo Zhang, Qi Zhang, and Xuan-Jing Huang. 2023. Textobfuscator: Making pre-trained language model a privacy protector via obfuscating word representations. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5459–5473.

A Proofs of Theorems

A.1 Proof of Theorem 4.1

In the PMT mechanism, for any token t_i, the output at its first occurrence is given by $R(t_i)$, while all subsequent outputs across different rounds are deterministic reuses of this randomized result:

$$\mathcal{M}(t_i) = g(R(t_i)), \tag{14}$$

where $g(\cdot)$ denotes the lookup and copy operation in the PMT table. Based on the post-processing property of differential privacy, the joint output distribution generated by PMT for the same token t_i across multiple interaction rounds still satisfies:

$$\frac{\Pr[\mathcal{M}(t_i) = y]}{\Pr[\mathcal{M}(t_i') = y]} \leq \exp(\varepsilon). \tag{15}$$

Therefore, the total privacy loss incurred by PMT for each token is at most ε, and does not accumulate due to cross-round reuse.

In contrast, under the traditional round-wise perturbation scheme, if the same token is repeatedly perturbed in T rounds, then according to the sequential composition property, the total privacy budget is given by:

$$\varepsilon_{T,trad} = T\varepsilon. \tag{16}$$

Under the PMT mechanism, we have:

$$\varepsilon_{T,PMT} = \varepsilon. \tag{17}$$

It immediately follows that:

$$\varepsilon_{T,PMT} \leq \varepsilon_{T,trad}. \tag{18}$$

A.2 Proof of Theorem 4.2

In the two-stage exponential mechanism $\mathcal{O}$, let the input token be t_i and the candidate set be v. According to the utility function $u_t(t_i, t_i')$, the candidate tokens are partitioned into K disjoint buckets $\{\mathcal{B}_1, \ldots, \mathcal{B}_K\}$, and the bucket-level utility is defined as:

$$U(\mathcal{B}_k) = \frac{1}{|\mathcal{B}_k|} \sum_{t_i' \in \mathcal{B}_k} |u_t(t_i, t_i')|. \tag{19}$$

In the first stage, the exponential mechanism is applied to sample a target bucket B^* from the bucket set with probability:

$$\Pr[\mathcal{B}^* = \mathcal{B}_k] = \frac{\exp\left(\frac{\varepsilon_1}{2\Delta U} U(\mathcal{B}_k)\right)}{\sum_{j=1}^{K} \exp\left(\frac{\varepsilon_1}{2\Delta U} U(\mathcal{B}_j)\right)}. \tag{20}$$

By the privacy guarantee of the exponential mechanism, for any neighboring bucket $\mathcal{B}_k, \mathcal{B}_k'$ and any output bucket B^*, we have:

$$\frac{\Pr[\mathcal{B}^* = \mathcal{B}_k]}{\Pr[\mathcal{B}^* = \mathcal{B}_k']} \leq \exp(\varepsilon_1), \tag{21}$$

which implies that the bucket sampling stage satisfies ε_1-LDP.

In the second stage, conditioned on the selected bucket B^*, the exponential mechanism is again applied to sample the output token $\tilde{t}_i$ within the bucket, with conditional probability:

$$\Pr[\tilde{t}_i = t_i' \mid \mathcal{B}^*] = \frac{\exp\left(\frac{\varepsilon_2}{2\Delta u} u_t(t_i, t_i')\right)}{\sum_{t_i' \in \mathcal{B}^*} \exp\left(\frac{\varepsilon_2}{2\Delta u} u_t(t_i, t_i')\right)}. \tag{22}$$

Similarly, we can get:

$$\frac{\Pr[\tilde{t}_i = t_i \mid \mathcal{B}^*]}{\Pr[\tilde{t}_i = t_i' \mid \mathcal{B}^*]} \leq \exp(\varepsilon_2), \tag{23}$$

which indicates that the intra-bucket sampling stage satisfies ε_2-LDP.

Combining the two stages, the marginal distribution of the output $\tilde{t}_i$ is given by:

$$\Pr[\tilde{t}_i = t_i'] = \sum_{k=1}^{K} \Pr[\mathcal{B}^* = \mathcal{B}_k] \Pr[\tilde{t}_i = t_i' \mid \mathcal{B}^*]. \tag{24}$$

Thus, we have:

$$\frac{\Pr[\mathcal{O}(t_i) = \tilde{t}_i]}{\Pr[\mathcal{O}(t_i') = \tilde{t}_i]} \leq \exp(\varepsilon_1) \cdot \exp(\varepsilon_2) = \exp(\varepsilon_1 + \varepsilon_2), \tag{25}$$

where the inequality follows from the fact that the two stages satisfy ε_1-LDP and ε_2-LDP, respectively, and are executed sequentially.

By the sequential composition property of differential privacy, the two-stage bucketed exponential mechanism as a whole satisfies $(\varepsilon_1 + \varepsilon_2)$-Differential Privacy.

B Experiment Results

B.1 System Prompts

We provide the prompts used for the MT-Bench-101 and hCHARP datasets in Table 5.

B.2 Experiments across Different Dialogue Turns

On the MT-Bench-101 dataset, we conduct privacy-utility evaluations across different dialogue rounds,

MT-Bench-101:
Can you answer this question for me? If there is a definitive answer, return a final conclusive sentence; if there is no definitive answer, return a final analytical conclusion. Question: {**Protected_query**}
hCHARP:
Answer the question based on the provided knowledge. If the knowledge allows for a definitive conclusion, give a final conclusive sentence. If the knowledge does not provide a definitive answer, give a final analytical conclusion. Knowledge: {**Knowledge**} Question: {**Protected_query**}
Test prompt for MT-Bench-101 and hCHARP:
You are a semantic consistency evaluation expert. Please determine whether the model-generated answer is semantically equivalent to the correct answer for the given question. Question: {**Current_query**} Correct Answer: {**Correct_answer**} Generated Answer: {**Query_response**} 1 - If they are semantically equivalent. 0 - If they are not equivalent. Return only 0 or 1.

Table 5: The detailed prompts for MT-Bench-101 and hCHARP task.

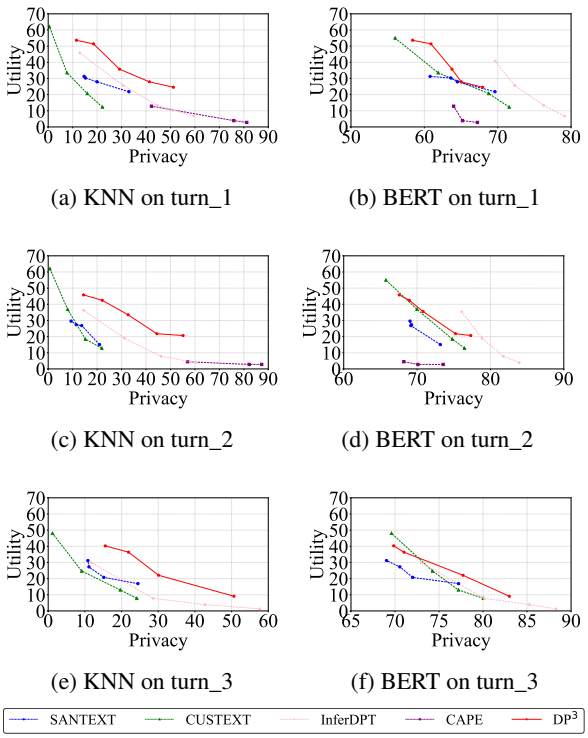
and the results are shown in Fig. 5. In the first dialogue round, DP^3 performs slightly worse than existing methods at some operating points. However, as the number of dialogue rounds increases, the overall privacy-utility curve of our DP^3 consistently rises above the baselines. This trend further demonstrates the effectiveness of our method in multi-turn dialogue settings: with more rounds, DP^3 achieves increasingly better privacy-utility trade-offs than existing methods.

B.3 Perturbation Example

We provide perturbation examples in Table 6. Given the input prompts: *"Now there are three people A, B and C. I currently know that A is taller than B and B is taller than C. Who is the tallest currently?" "Now there are two more people, D and E. D is higher than B and E is higher than D. Who is the tallest at the moment?"* and *"Now, I know that D is higher than A. Who is the highest now?"*, we vary the perturbation mechanisms and the privacy budget.

B.4 KNN Attack

In the KNN inference attack (Song and Raghunathan, 2020), the adversary first computes the em-



(a) KNN on turn_1 (b) BERT on turn_1

(c) KNN on turn_2 (d) BERT on turn_2

(e) KNN on turn_3 (f) BERT on turn_3

Figure 5: The privacy-utility trade-off between privacy attacks and accuracy across different dialogue rounds on the MT-Bench-101 dataset, where the privacy budget ε varies within the interval $[1, 20]$.

Mechanism	ϵ	Original Prompt:	Rouge-L (F1)
		Round 1: Now there are three people A, B and C. I currently know that A is taller than B and B is taller than C. Who is the tallest currently?	
		Round 2: Now there are two more people, D and E. D is higher than B and E is higher than D. Who is the tallest at the moment?	
		Round 3: Now, I know that D is higher than A. Who is the highest now?	
SANTEXT	1	**Round 1:** Now know are the , A , B and C people I people people the A is Who Who B and B is taller three C . ? is the tallest people that,	31.27
		Round 2: Now there more two more people , more and E . D is higher than B and E is higher than people . Who is , tallest at the moment ?,	
		Round 3: . , I know that D is higher than A . Who . now now now ?,	
	5	**Round 1:** Now know are three people A , B and C , I three know that A is tallest that B and B is taller three C . Who is the tallest people ?,	37.31
		Round 2: Now there more two more people , more and E . D is higher than B and E is higher than people . Who is two tallest at the moment ?,	
		Round 3: Who , I know that D is higher than A . Who the the highest now ?,	
CUSTEXT	1	**Round 1:** Soon we being ten others A , B both C . I currently thing however A means taller expect B which B be bigger least C . Have seems way towers only ?,	15.27
		Round 2: now that those seven rather there , D but E . D This higher almost B both E is decrease expect D . Think only of towered around of mind ?,	
		Round 3: Now , I forget but D also increasing but A . Someone be rest comparable up ?,	
	5	**Round 1:** Just we should eight people A , B and C . I currently thought even A seems taller perhaps B that B is taller than C . Who only one tallest previously ?,	26.01
		Round 2: Now there be four too people , D as E . D it higher better B and E is decrease perhaps D . Are is the taller @ the when ?,	
		Round 3: Now , I remember that D being higher more A . Have is whole attained even ?,	
InferDPT	5	**Round 1:** Now gad cant jam touching A , B ALLY C . Signature sometime famous fin AS pen taller aside B fore B ILL taller WAYS I . Whatever observable marriage tallest typically ?,	12.74
		Round 2: Now reg cant triangle presence POSSIBILITY , D an E . D owl impressive THEN B agar E burgh county smaller D . Who is across tallest factory touched Tue ?,	
		Round 3: Now , RITE assess they D soles doubled versus A . came averse spring smaller bout ?,	
CAPE	5	**Round 1:** now there are three : a, b and c. i currently know that a is taller than b and a is taller than c. who is the tallest currently?,	44.12
		Round 2: now there are still more lords, d and e. d is higher than b and he is higher than d. who is the king at the moment?,	
		Round 3: now, i know that d is higher than a. who is the highest now?,	
DP^3	1	**Round 1:** proceeding there are three indices A , B and C . I ankle telephone that A is taller than B and B is taller than C . ankle is the tallest ankle ?	**44.36**
		Round 2: proceeding there are two more indices , D and E . D is indices than B and E is indices than D . ankle is the tallest at the Rating ?	
		Round 3: proceeding , I telephone that D is indices than A . ankle is the Mary now ?	
	5	**Round 1:** TODAY there are three indices A , B and C . I ankle know that A is taller than B and B is taller than C . ankle is the tallest ankle ?	**46.46**
		Round 2: TODAY there are two more indices , D and E . D is selection than B and E is selection than D . ankle is the tallest at the mouth ?	
		Round 3: TODAY , I know that D is selection than A . ankle is the higher now ?	

Table 6: Perturbation examples of varying privacy budgets ε for different methods.

bedding distance between each perturbed token in the prompt and all other tokens within the vocabulary. Subsequently, the top-10 tokens exhibiting the smallest embedding distances are selected. If the corresponding original token is present within top-10 tokens, the attack is deemed successful. Finally, the attack success rate across all tokens is denoted as r_{ats}, and the privacy protection level of

the DP mechanism is defined as $1 - r_{\text{ats}}$.

B.5 BERT Inference Attack

In the BERT inference attack (Yue et al., 2021), an adversary leverages a pre-trained BERT model to recover the original prompt x from its perturbed version $\tilde{x}$. The BERT model (110M) is trained using masked language modeling, where each token in the perturbed text is sequentially replaced with a special token "[MASK]", and the model predicts the original token by utilizing contextual information. By exploiting BERT's powerful contextual modeling capabilities, the model is able to make plausible inferences about the masked tokens. The similarity between the predicted prompt and the original prompt is evaluated using the Rouge-L F1 score (Lin, 2004). The attack success rate is denoted as r_{ats}, and the privacy protection level of the different mechanisms is defined as $1 - r_{\text{ats}}$.

B.6 Implementation

All experiments are conducted on a machine equipped with two NVIDIA GeForce RTX 3090 GPUs (24GB each). SANTEXT and CUSTEXT use their default GloVe vocabulary. InferDPT is run with its default $cl100\_embeddings$ vocabulary. CAPE uses the built-in vocabulary of BERT (Devlin et al., 2019). Our method adopts the $\mathcal{V} = cl100\_embeddings$ vocabulary.

B.7 Datasets

For conversational question answering, we use the MT-Bench-101 (MT) benchmark (Bai et al., 2024), which includes Mathematical Reasoning (MR) and General Reasoning (GR) tasks for evaluating multi-turn dialogue capabilities of LLMs. In addition, we adopt the hard version of CHARP (hCHARP) (Ghaddar et al., 2024), which assesses a model's ability to leverage dialogue history and mitigate hallucinations. MR and GR focus on collaborative multi-turn problem solving in mathematical and general reasoning domains, respectively, while hCHARP is designed to evaluate historical reasoning in dialogue systems.