Differentially Private Summarization with Large Language Models

Ivan Gutierrez
Harvard College
ivangutierrez@college.harvard.edu

Hadi Khalaf
Harvard University
hadikhalaf@g.harvard.edu

Han Qi
Harvard University
hqi@g.harvard.edu

Abstract

Large Language Models (LLMs) achieve state-of-the-art performance in text summarization but pose significant privacy risks when applied to sensitive data. Standard defenses, such as private fine-tuning (DP-SGD), are often computationally prohibitive and unstable. In this work, we propose two inference-time strategies for differentially private summarization that operate with frozen LLMs. The first method aggregates hidden states of documents at each generation step, allowing for flexible decoding at the cost of high privacy budget consumption. To address this limitation, we introduce a *one-shot mechanism* that computes a single noisy aggregate representation of the private inputs. We then utilize a soft prompt adapter to decode this representation into a summary, ensuring the privacy cost remains constant regardless of output length. We evaluate these approaches on Amazon product reviews, demonstrating that the one-shot mechanism yields superior utility-privacy tradeoffs compared to iterative decoding.

1 Introduction

Automated summarization is a fundamental goal of natural language processing (NLP) [1, 2]. The exponential growth of digital content has made this task even more critical. Organizations and individuals are overwhelmed with vast amounts of unstructured text, creating an urgent need for tools that distill this data into actionable insights. For instance, e-commerce platforms use summarization to extract feedback from thousands of product reviews [3, 4]. In healthcare, clinicians may need to rely on aggregated notes to track patient history [5, 6].

Historically, organizations have relied on human annotators or classical models like BART and T5. However, human annotation is expensive to scale while classical NLP models often struggle to produce coherent or distinct summaries [7, 8, 9]. Modern Large Language Models (LLMs) have achieved state-of-the-art performance in summarization and related generation tasks [2, 10, 11, 12]. Yet, applying them to sensitive data introduces significant risks. LLMs are known to memorize training data, leading to potential extraction or reconstruction attacks [13, 14, 15, 16, 17].

To deploy these models safely, we require rigorous guarantees such as Differential Privacy (DP). The standard approach to integrating DP with LLMs is differentially private stochastic gradient descent (DP-SGD), which involves fine-tuning the model weights [1, 18, 19, 20]. Unfortunately, private fine-tuning is computationally expensive and notoriously unstable [21, 22, 23]. Furthermore, it requires white-box access to model gradients, which is rarely available when using commercial APIs [13, 15].

Preprint.

A promising alternative is to enforce privacy strictly at inference time. We treat the LLM as a frozen model and do not update its weights on private data. Several recent works explore black-box or inference-time privacy approaches for text generation and prediction [11, 12, 13, 14]. In this work, we explore two distinct strategies for inference-time DP summarization. We leverage the fundamental property of LLMs as next-token predictors.

The first method constructs a differentially private decoder that "pays for privacy" at every step. For each token to be generated, we aggregate the hidden vectors of all text documents and add noise before generation. This allows for flexible decoding but consumes the privacy budget with the summary length, mirroring limitations observed in prior DP decoding and generation methods [14, 16, 22].

To address the budget constraints of the iterative approach, we propose a second method. This "one-shot" mechanism interacts with the private data only once to compute a single noisy mean vector. We then use a *soft prompt*, a small embedding trained solely on public data, to guide the frozen LLM in decoding this vector into a summary. Soft prompts and prompt-based adapters have been shown to effectively steer large models with small parameter budgets in a variety of settings [2, 4, 10, 11, 18]. This ensures the privacy cost remains constant regardless of the output length and avoids the instability of DP fine-tuning.

Outline. Our paper is organized as follows. Section 2 discusses related work on differential privacy and summarization in the context of LLMs. Section 3 provides the readers with the necessary background on differential privacy and large language models. Section 4 discusses our two proposed methods in detail. Lastly, Section 5 provides our experimental results with their analysis. We provide more details on the experimental setup in the Appendix and we open-source our code here.

Contributions. Our contributions are as follows:

- We present two black-box methods for differentially private summarization with LLMs.
- We provide an empirical study on Amazon-style reviews, comparing token-level and one-shot mechanisms under matched privacy budgets.

2 Related Work

Research on privacy-preserving text generation spans three main areas: (1) training-time differential privacy using DP-SGD, (2) inference-time mechanisms that privatize decoding or prediction from a frozen model, and (3) privacy for text embeddings and aggregate representations. Our work draws inspiration from all three lines while focusing specifically on differential privacy for summarizing multiple documents using a black-box LLM.

DP-SGD and Training-Time Private Generation. A substantial body of work applies DP-SGD [24] to train or fine-tune language models with formal guarantees. SeqPATE [1] adapts the PATE framework to sequential generation by combining teacher votes from disjoint shards. Yue et al. [2] and Kurakin et al. [10] show that DP-SGD can produce useful synthetic corpora when applied to generative models. Recent efforts extend DP-SGD to instruction tuning [19, 20] and to generating private prompts for in-context learning [18]. While DP-SGD remains a standard tool, it is often computationally demanding, sensitive to clipping hyperparameters, and difficult to stabilize for large models. In addition, DP-SGD requires full gradient access, which limits applicability when working with closed-source LLM APIs [13, 15]. These observations motivate efforts to explore privacy at inference time instead of during training.

Inference-Time Differential Privacy. Another line of work studies how to privatize the decoding process itself, leaving the underlying model unchanged. Majmudar et al. [14] propose adding Gaussian noise to logits at each decoding step. Flemings et al. [16] analyze how the sensitivity of next-token probabilities propagates through the softmax. Utpala et al. [22] investigate locally private decoding strategies using zero-shot prompting. Related approaches focus on private prediction rather than full text generation. For example, Ginart et al. [13] aggregate hidden representations across shards to reduce sensitivity, and Amin et al. [4] develop black-box private prediction mechanisms for sampling from a frozen LLM. More recent work uses clustering and median aggregation to improve

robustness under noise [12]. Vinod et al. [23] study cost-effective private text generation using a black-box model. Our token-level method relates to these approaches through the use of noisy, clipped hidden-state aggregation, while our one-shot method connects to inference-time privacy by privatizing a single latent representation used to steer generation.

Privacy for Embeddings and Aggregate Representations. A complementary set of methods focuses on privatizing intermediate text representations. Feyisetan and Kasiviswanathan [6] propose mechanisms for releasing text embeddings under local DP. Xu et al. [7, 8] develop perturbation-based techniques that preserve semantic similarity while satisfying differential privacy. Yue et al. [5] study local DP for sentence embeddings and show that clipped, noise-perturbed vectors can retain semantically meaningful structure. These works motivate operating in the hidden-state space of LLMs: hidden representations are lower dimensional than logits and exhibit smoother semantic geometry, reducing noise requirements. Both of our methods adopt this perspective by aggregating clipped hidden states before decoding.

Summarization and Multi-Document Aggregation. Classical summarization systems such as BART and T5 provide strong baselines for conditional text generation, though they can struggle with long inputs and domain adaptation. Modern LLMs achieve substantially stronger summarization performance, especially in multi-document settings. However, directly concatenating sensitive documents into a prompt raises privacy concerns. Existing work on private text generation primarily focuses on synthesizing standalone text or privatizing next-token prediction, rather than summarizing multiple private inputs.

3 Preliminaries

3.1 Differential Privacy

We use the standard (ε, δ)-DP definition under the replace-one adjacency relation. A randomized mechanism $\mathcal{M}$ is (ε, δ)-DP if for all adjacent $D \sim D'$ and all measurable sets S,

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

Our mechanisms rely on the Gaussian mechanism, which adds Gaussian noise to a function of the data. Repeated privacy loss is tracked using Rényi Differential Privacy (RDP), which composes additively across decoding steps and is converted back into an equivalent (ε, δ) guarantee. In our token-level method, RDP accounts for T queries (i.e. T max tokens); in our one-shot method, privacy is spent only once.

3.2 LLM Basics

Large language models decode text autoregressively. Given a sequence of tokens, the model produces a last-layer hidden state $h \in \mathbb{R}^d$, which the LM head maps to a next-token distribution via a linear projection and softmax. Generation proceeds by appending each sampled token to the context and repeating the process until an End-of-Sequence (EOS) token is generated or the maximum number of tokens T is reached.

A soft prompt is a sequence of embedding vectors prepended to the LLM's input to steer generation. While standard soft prompts are optimized as static parameters, we employ a dynamic approach using a *soft prompt adapter*. This adapter is a lightweight network—trained on public data—that functions as a mapping from latent space to input space. Specifically, it projects an embedding vector into a sequence of tokens. These are then fed into the frozen LLM, enabling it to decode a summary based solely on the private aggregate representation.

4 Methodology

We consider the problem of generating a summary sequence $Y = (y_1, \ldots, y_T)$ from a private dataset $D = \{x_1, \ldots, x_n\}$ containing n textual inputs. We assume black-box access to a pre-trained Large Language Model (LLM), parameterized by a function $f_\theta : \mathcal{V}^* \to \mathbb{R}^d$ that maps a sequence of tokens

to a hidden vector representation, and a language modeling head $\phi : \mathbb{R}^d \to \Delta_{|\mathcal{V}|}$ that maps hidden vectors to a probability distribution over the vocabulary $\mathcal{V}$.

In the following, we consider an *add-remove adjacency* relation. Two datasets D and D' are considered neighbors (denoted $D \sim D'$) if D' can be obtained by adding or removing a single review x from D. A randomized summarization mechanism $\mathcal{M}$ satisfies (ε, δ)-Differential Privacy if for all adjacent datasets $D \sim D'$ and all possible sets of outputs S:

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in S] + \delta. \tag{1}$$

Our goal is to ensure that the distribution of generated summaries is indistinguishable regardless of whether any specific user's review is included in the input.

4.1 Token-Level Private Decoding

To generate a consensus summary of multiple textual inputs, we must aggregate the model's predictions across the inputs at each token step. Conceptually, this aggregation can occur at two distinct stages: the high-dimensional output logits (the vocabulary scores) or the internal hidden representations (the latent space).

We choose to perform aggregation in the hidden space for two primary reasons. First, the dimensionality of the hidden space is significantly smaller than the vocabulary size ($d \ll |\mathcal{V}|$, e.g., $4,096$ vs. $128,000$). Second, the hidden space is semantically continuous. The average of two latent vectors often represents a coherent intermediate concept. In contrast, logits are unnormalized scores, sensitive to the non-linear softmax function, where averaging noisy values can unpredictably skew the probability mass.

Inference Process Our approach generates the summary auto-regressively by aggregating these semantic representations. Let $y_{<t}$ denote the summary prefix generated prior to step t. For each private input $x_i \in D$, we construct a context sequence $c_{i,t} = [x_i; y_{<t}]$ and compute the final hidden state $h_{i,t} = f_\theta(c_{i,t})$.

To satisfy differential privacy, we employ the Gaussian Mechanism on the aggregate of these hidden states. To bound the sensitivity, we enforce a clipping norm C. Let $\Pi_C(v) = v \cdot \min(1, C/\|v\|_2)$ denote the projection of a vector onto the ℓ_2-ball of radius C. The private empirical mean at step t is computed as:

$$\bar{h}_t \;=\; \frac{1}{n} \sum_{i=1}^{n} \Pi_C(h_{i,t}) \;+\; z_t, \quad \text{where } z_t \sim \mathcal{N}(0, \sigma^2 I_d). \tag{2}$$

The noisy vector $\bar{h}_t$ is then projected back to the vocabulary space via the fixed head ϕ, yielding a distribution $p_t = \phi(\bar{h}_t)$ from which the next token y_t is sampled. The complete procedure is summarized in Algorithm 1.

Privacy Composition and Noise Calibration A central challenge in this setup is determining the noise scale σ such that the cumulative leakage over T steps satisfies the global (ε, δ)-DP constraint. We utilize the framework of Gaussian Differential Privacy (GDP) for exact composition.

A mechanism is denoted as μ-GDP if the hypothesis testing problem of distinguishing adjacent datasets is equivalent to distinguishing $\mathcal{N}(0, 1)$ from $\mathcal{N}(\mu, 1)$. The Gaussian mechanism applied to a query with ℓ_2-sensitivity Δ and noise standard deviation σ is μ-GDP with $\mu = \Delta/\sigma$. Under the replace-one adjacency relation, the sensitivity of the sum of vectors clipped to norm C is $\Delta = 2C$. Thus, a single generation step is μ_0-GDP with $\mu_0 = \frac{2C}{n\sigma}$.

By the composition theorem of GDP, the independent composition of T homogeneous μ_0-GDP mechanisms is μ_{total}-GDP, where $\mu_{\text{total}} = \sqrt{T}\mu_0$. To ensure the total mechanism satisfies (ε, δ)-DP, μ_{total} must satisfy the following duality relation:

$$\delta(\varepsilon) \geq \Phi\left(-\frac{\varepsilon}{\mu_{\text{total}}} + \frac{\mu_{\text{total}}}{2}\right) - e^{\varepsilon}\Phi\left(-\frac{\varepsilon}{\mu_{\text{total}}} - \frac{\mu_{\text{total}}}{2}\right), \tag{3}$$

where Φ is the standard normal CDF. We solve Eq. (3) numerically to find the maximum permissible μ_{total} given ε and δ. The required noise scale per token is then derived as:

$$\sigma = \frac{2C\sqrt{T}}{n \cdot \mu_{\text{total}}}. \tag{4}$$

Algorithm 1 Token-Level DP Summarization with GDP Composition

Require: Dataset $D = \{x_i\}_{i=1}^n$, Target privacy (ε, δ), Clipping norm C, Max tokens T.

Calibration: Compute noise scale σ satisfying Eq. (3).
Initialize $y_0 \leftarrow$ "".
for $t = 1$ **to** T **do**
  $S_t \leftarrow \mathbf{0}$
  for $i = 1$ **to** n **do**
    Extract hidden state $h_{i,t} \leftarrow f_\theta([x_i; y_{t-1}])$
    Clip $\tilde{h}_{i,t} \leftarrow h_{i,t} \cdot \min(1, C/\|h_{i,t}\|_2)$
    $S_t \leftarrow S_t + \tilde{h}_{i,t}$
  end for
  Sample noise $z_t \sim \mathcal{N}(0, \sigma^2 I_d)$
  Compute private mean $\bar{h}_t \leftarrow \frac{1}{n} S_t + z_t$
  Compute logits $p_t \leftarrow \phi(\bar{h}_t)$
  Sample $y_t \sim p_t$ and append to sequence
end for
return y_T

Algorithm 2 One-Shot DP Summarization via Soft Prompts

Require: Dataset $D = \{x_i\}_{i=1}^n$, Target privacy (ε, δ), Clipping norm C, Soft prompt length m.

Calibration: Compute noise scale σ satisfying the Gaussian mechanism requirement for (ε, δ)-DP.
$S \leftarrow \mathbf{0}$
for $i = 1$ **to** n **do**
  Extract global hidden state $h_i \leftarrow f_\theta(x_i)$
  Clip $\tilde{h}_i \leftarrow h_i \cdot \min(1, C/\|h_i\|_2)$
  $S \leftarrow S + \tilde{h}_i$
end for
Sample noise $z \sim \mathcal{N}(0, \sigma^2 I_d)$
Compute private mean $\bar{h}_{\text{priv}} \leftarrow \frac{1}{n} S + z$
Compute soft prompt $S_{\text{soft}} \leftarrow g_\phi(\bar{h}_{\text{priv}})$, where $S_{\text{soft}} \in \mathbb{R}^{m \times d}$
Generate summary $y \leftarrow \text{LLM\_decode}(S_{\text{soft}})$
return y

This calibration ensures that we exactly exhaust the privacy budget at step T.

Discussion The efficacy of $\bar{h}_t$ depends on the semantic alignment of the individual vectors $h_{i,t}$. If inputs diverge (e.g., reviews focusing on disparate features), the mean vector may lie in a low-probability region of the language model's latent space. To mitigate this, we constrain the generation using a rigid prompt template (e.g., "*The product is a [TYPE]...*"). This forces the internal states of the LLM into a shared subspace at each step. Moreover, this type of prompt-engineering is a common trick to improve utility of small language models. We realize that this method is very expensive: each summary requires n model calls. However, text summarization is not a user-facing task where low latency is essential. We explain in Section 6 some potential ways to improve this method.

4.2 One-Shot Summarization via Soft Prompts

While the token-level approach allows for adaptive decoding, it incurs a privacy cost that scales with the summary length T. This creates a bottleneck: to generate longer, more detailed summaries, one must either increase the global privacy budget ε or accept higher noise levels per token, which degrades coherence. To address this, we propose a *one-shot* mechanism that interacts with the private data exactly once, incurring a constant privacy cost regardless of the output length.

Private Representation Aggregation Instead of aggregating at every decoding step, we aggregate the semantic representations of the full input texts. For each private review x_i, we perform a single

forward pass through the LLM to extract a global representation vector $h_i \in \mathbb{R}^d$ (specifically, the hidden state of the last token). We apply the same clipping and Gaussian mechanism described previously. This vector $\bar{h}_{\text{priv}}$ represents the "average sentiment" or semantic content of the private dataset. Since $\bar{h}_{\text{priv}}$ satisfies differential privacy, any subsequent computation performed on it is safe by the post-processing property of DP.

Decoding via Soft Prompts The challenge lies in decoding this abstract vector $\bar{h}_{\text{priv}}$ into fluent text. Since $\bar{h}_{\text{priv}}$ is an average of hidden states, it does not correspond to a valid token sequence in the LLM's vocabulary. To bridge this gap, we assume access to a small *public dataset* $D_{\text{pub}} = \{(X^{(j)}, y^{(j)})\}$ consisting of review sets $X^{(j)}$ and reference summaries $y^{(j)}$.

We introduce a *soft prompt adapter*, a lightweight parametric function $g_\phi : \mathbb{R}^d \to \mathbb{R}^{m \times d}$ (e.g., a multi-layer perceptron). This adapter maps a single hidden vector to a sequence of m continuous embedding vectors (soft tokens). We optimize the parameters ϕ on the public data to minimize the standard language modeling loss:

$$\mathcal{L}(\phi) = - \sum_{(X,y) \in D_{\text{pub}}} \log P_{\text{LLM}}(y \mid g_\phi(\bar{h}_{\text{agg}}(X))), \tag{5}$$

where $\bar{h}_{\text{agg}}(X)$ is the un-noised mean representation of the public inputs. The LLM parameters remain frozen. Only the adapter ϕ is trained to learn how to steer the model based on an aggregate vector.

Inference At test time on private data, we compute the noisy aggregate $\bar{h}_{\text{priv}}$ and project it into a soft prompt $S = g_\phi(\bar{h}_{\text{priv}})$. We then prepend S to the LLM's input embedding layer. The model generates the summary autoregressively conditioned on S.

5 Experimental Results

5.1 Setup

We evaluate our differentially private summarization methods on Amazon product reviews. Each product in our dataset contains between 10 and 100 customer reviews, along with a ground truth summary provided by the website which we use as the reference for evaluation. Given limited computational constraints, we consider 100 products in our experiments. We rerun each experiment with three different seeds for statistical significance.

We use `Llama-3.2-3B-Instruct` as our base language model. For each product, we generate summaries using a structured template that produces concise sentences following the pattern:

> "The product is a [TYPE]. Customers praise its [ASPECT_1] and [ASPECT_2], but some complain about [ISSUE_1] and [ISSUE_2]."

This templated approach improves summary quality and aligns with standard practices in e-commerce. The model is prompted to replace bracketed placeholders with specific phrases extracted from the customer reviews. We target global privacy parameters (ε, δ) where $\delta = 10^{-6}$ and ε ranges from 10 to 120 over a maximum number of tokens $T = 50$. We evaluate the quality of generated summaries using two complementary metrics: (1) **ROUGE** (F1 scores for ROUGE-1 and ROUGE-L) to measure n-gram overlap with the ground truth summary, and (2) **BERTScore** to capture semantic similarity using contextualized embeddings. For each metric, we report average scores across all products and seeds, with 95% bootstrap confidence intervals computed by resampling products.

5.2 Results

5.2.1 Token-Level Private Decoding

We present the utility as a function of the privacy budget in the following figures, with utility measured with **ROUGE-1** in Figure 1, with **ROUGE-L** in Figure 2, and with **BERTScore** in Figure 3. Each figure contains three cases with varied sampling temperatures. A larger sampling temperature

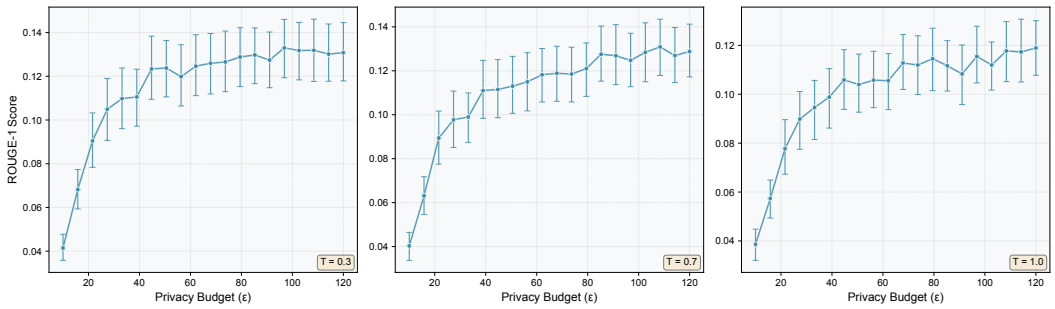

Figure 1: Utility-privacy tradeoffs with utility measured using **ROUGE-1**, with token-level DP

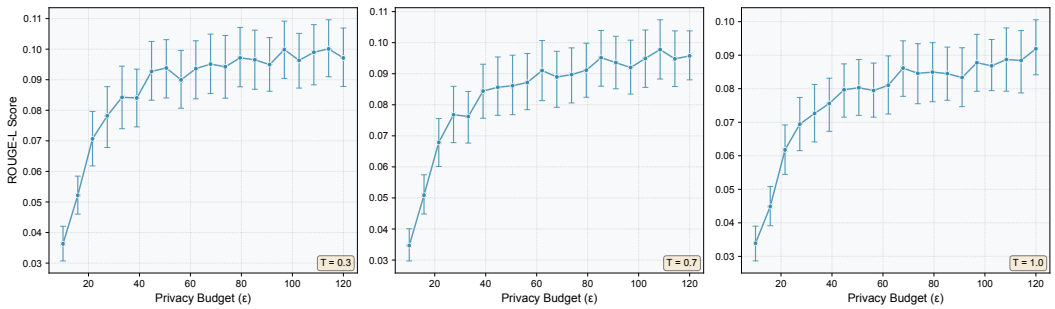

Figure 2: Utility-privacy tradeoffs with utility measured using **ROUGE-L**, with token-level DP

indicates a higher randomness in selecting a next token. We present example generation responses in the Appendix (see Figures 7, 8, and A).

Across all three metrics, we observe a distinct logarithmic growth pattern. In the low-budget regime ($10 \leq \varepsilon \leq 30$), utility increases sharply. The noise scale σ required to satisfy the tight privacy constraints significantly perturbs the hidden states, pushing the aggregate vector away from the semantic manifold of valid tokens. This is evident when looking at sample generations: with low ε, the LLM generated non-English tokens (see Figure 8). Notably, we observe diminishing returns as the budget increases beyond $\varepsilon = 80$. At this stage, the performance plateaus (e.g., **BERTScore** stabilizes around 0.84 and **ROUGE-L** around 0.10). The dominant error source shifts to the inherent capability of the base LLM and the constraints of the rigid template.

A comparison between **BERTScore** and **ROUGE** scores reveals a discrepancy in absolute performance magnitudes. Our method achieves high semantic similarity, indicating that the generated summaries successfully capture the core meaning of the reviews (e.g., correctly identifying the product type and sentiment). In contrast, the ROUGE scores are relatively low. **This is an expected artifact of our templated approach.** Since ROUGE measures exact n-gram overlap, it penalizes our method heavily since the ground truth summary does not follow our strict "*The product is a [TYPE]...*" structure, even if the information content is identical. The high **BERTScore** validates that our method preserves the semantic utility of the private dataset.

5.2.2 One-Shot Summarization via Soft Prompts

We next present with soft prompt, the utility as a function of the privacy budget in the following figures, with utility measured with **ROUGE-1** in Figure 4, with **ROUGE-L** in Figure 5, and with **BERTScore** in Figure 6.

Similar to the token-level private decoding results, we observe a clear tradeoff between privacy budget and model performance across ROUGE-1, ROUGE-L, and BERTScore. However, the soft-prompt approach exhibits several striking advantages. Because it privatizes the data only once, it entirely avoids the length-dependent privacy–utility bottleneck inherent to token-level decoding. Even under

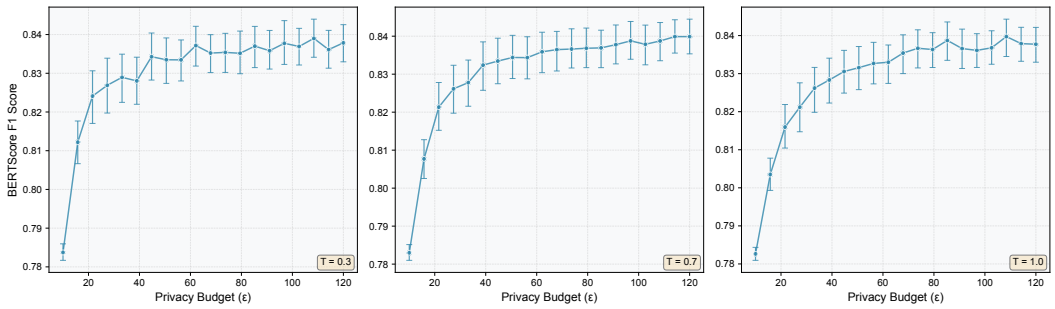

Figure 3: Utility-privacy tradeoffs with utility measured using **BERTScore**, with token-level DP

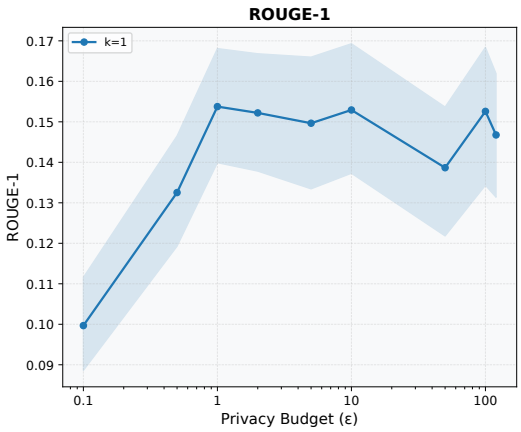

Figure 4: Utility-privacy tradeoffs with utility measured using **ROUGE-1**, with soft prompts

strict privacy settings, the soft prompt mechanism maintains similar to substantially better stability and higher overall utility.

While soft prompting offers a fixed-cost alternative to token-level decoding, it also introduces several limitations. The effectiveness of the soft prompt depends on how well the adapter, trained only on public data, maps this noisy representation into a meaningful steering signal. In addition, soft prompting provides fewer opportunities for controllability or correction during generation, since the model cannot refine its representation once the prompt is produced. To remedy this, we could extend the existing algorithm into a k-Shot algorithm by generating k soft prompts with $(\epsilon/k, \delta/k)$-DP, and averaging them into a single soft prompt (which becomes post-processing under DP). More experimentation would be needed to discover if slight increases to k could increase stability within the already high-variance scores like ROUGE-1 and ROUGE-L. Finally, a single averaged hidden vector may struggle to capture diverse or conflicting information across inputs, limiting the expressiveness and utility of the one-shot approach compared to the more adaptive and dynamically grounded token-level mechanism. Often, this lack of expressiveness would lead to the summaries leaking aspects of the prompt instructions (e.g., mentioning "This product is a [TYPE]," as opposed to describing the actual product type).

6 Limitations and Future Work

Limitations We do note some limitations of our work. First, we derive our (ε, δ) guarantees solely from the added noise to the mean hidden vector, treating the subsequent token sampling step as a post-processing operation. Since we only release the sampled token and not the mean vector, we can improve our bounds by also considering the randomness of the language modeling head itself. Second, the discrepancy between our high **BERTScore** results and lower **ROUGE** scores highlights the limitation of our rigid prompt template. Our current prompt template yielded the best results after

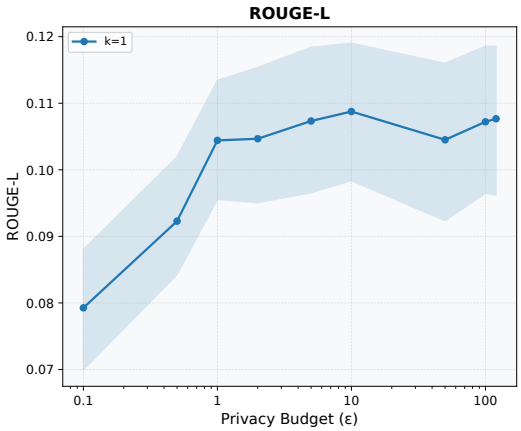

Figure 5: Utility-privacy tradeoffs with utility measured using **ROUGE-L**, with soft prompts

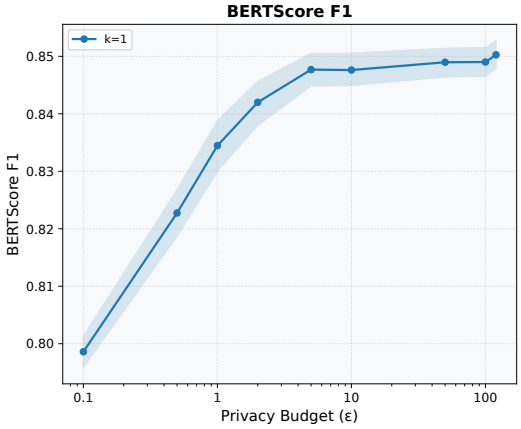

Figure 6: Utility-privacy tradeoffs with utility measured using **BERTScore**, with soft prompts

a lengthy process of prompt optimization and trials. While the template enforces semantic stability, it restricts the syntactic diversity of the output. Consequently, if the ground truth summary utilizes a free-form narrative style, our method is penalized by n-gram metrics. Developing methods to relax this template constraint without causing the aggregated hidden states to diverge is a critical next step. Lastly, as noted in Section 4, our approach requires n forward passes through the LLM for every generated token. While acceptable for offline batch processing as in our setting, this linear scaling with dataset size prohibits real-time applications.

Future Work Several promising directions may extend this work. First, the prompt template used to stabilize latent representations restricts stylistic flexibility. Future methods may explore cases with less constraints for templating. Second, the token-level approach requires n forward passes per generated token, limiting scalability. More efficient DP-aware strategies such as sub-sampling or sharding could reduce this overhead.

References

[1] Zhiliang Tian, Yingxiu Zhao, Ziyue Huang, Yu-Xiang Wang, Nevin L. Zhang, and He He. Seqpate: Differentially private text generation via knowledge distillation. In *Advances in Neural Information Processing Systems*, 2022.

[2] Xiang Yue, Huseyin Inan, Xuechen Li, Girish Kumar, Julia McAnallen, Hoda Shajari, Huan Sun, David Levitan, and Robert Sim. Synthetic text generation with differential privacy: A simple and practical recipe. In *Proceedings of the 61st Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), pages 1321–1342, Toronto, Canada, July 2023. Association for Computational Linguistics.

[3] Yuan Sun, Zhiqi Huang, Bo Shan, Xiaoxuan Zhu, and Tianhao Wang. Evaluating differentially private generation of domain-specific text. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*. Association for Computing Machinery, 2025.

[4] Kareem Amin, Alex Bie, Weiwei Kong, Alexey Kurakin, Natalia Ponomareva, Umar Syed, Andreas Terzis, and Sergei Vassilvitskii. Private prediction for large-scale synthetic text generation. *arXiv preprint arXiv:2407.12108*, 2024.

[5] Xiang Yue, Huseyin Inan, Xuechen Li, Girish Kumar, Julia McAnallen, Hoda Shajari, Huan Sun, David Levitan, and Robert Sim. Sanitizing sentence embeddings (and labels) for local differential privacy. In *Proceedings of the ACM Web Conference 2023*. Association for Computing Machinery, 2023.

[6] Oluwaseyi Feyisetan and Shiva Kasiviswanathan. Private release of text embedding vectors. In *Proceedings of the First Workshop on Trustworthy Natural Language Processing*, pages 15–27, Online, June 2021. Association for Computational Linguistics.

[7] Zekun Xu, Abhinav Aggarwal, Oluwaseyi Feyisetan, and Nathanael Teissier. A differentially private text perturbation method using a regularized mahalanobis metric. *arXiv preprint arXiv:2010.11947*, 2020.

[8] Zekun Xu, Abhinav Aggarwal, Oluwaseyi Feyisetan, and Nathanael Teissier. On a utilitarian approach to privacy preserving text generation. *arXiv preprint arXiv:2104.11838*, 2021.

[9] Erion Çano and Ivan Habernal. Differentially-private text generation degrades output language quality. *arXiv preprint arXiv:2509.11176*, 2025.

[10] Alexey Kurakin, Natalia Ponomareva, Umar Syed, Liam MacDermed, and Andreas Terzis. Harnessing large-language models to generate private synthetic text. *arXiv preprint arXiv:2306.01684*, 2023.

[11] Tong Wu, Ashwinee Panda, Jiachen T. Wang, and Prateek Mittal. Privacy-preserving in-context learning for large language models. *arXiv preprint arXiv:2305.01639*, 2023.

[12] Kareem Amin, Salman Avestimehr, Sara Babakniya, Alex Bie, Weiwei Kong, Natalia Ponomareva, and Umar Syed. Clustering and median aggregation improve differentially private inference. *arXiv preprint arXiv:2506.04566*, 2025.

[13] Antonio Ginart, Laurens van der Maaten, James Zou, and Chuan Guo. Submix: Practical private prediction for large-scale language models. *arXiv preprint arXiv:2201.00971*, 2022.

[14] Jimit Majmudar, Christophe Dupuy, Charith Peris, Sami Smaili, Rahul Gupta, and Richard Zemel. Differentially private decoding in large language models. *arXiv preprint arXiv:2205.13621*, 2022.

[15] Shufan Zhang, Jordan Liss, Mingqing Chen, and Yu-Xiang Wang. Privinfer: Accurate differential privacy inference without retuning. *arXiv preprint arXiv:2310.12214*, 2024.

[16] James Flemings, Meisam Razaviyayn, and Murali Annavaram. Differentially private next-token prediction of large language models. *arXiv preprint arXiv:2403.15638*, 2024.

[17] Md Mahadi Hasan Nahid and Sadid Bin Hasan. Safesynthdp: Leveraging large language models for privacy-preserving synthetic data generation using differential privacy. *arXiv preprint arXiv:2412.20641*, 2024.

[18] Fengyu Gao, Ruida Zhou, Tianhao Wang, Cong Shen, and Jing Yang. Data-adaptive differentially private prompt synthesis for in-context learning. In *International Conference on Learning Representations*, 2025.

[19] Da Yu, Peter Kairouz, Sewoong Oh, and Zheng Xu. Privacy-preserving instructions for aligning large language models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[20] Fengyu Gao, Ruida Zhou, Tianhao Wang, Cong Shen, and Jing Yang. Privacy-preserving instructions for aligning large language models. *arXiv preprint arXiv:2402.13659*, 2024.

[21] James Flemings. Differentially private knowledge distillation via synthetic text generation. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024. Full author list omitted for brevity; see ACL Anthology entry 2024.findings-acl.769.

[22] Saiteja Utpala, Sara Hooker, and Pin-Yu Chen. Locally differentially private document generation using zero shot prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8442–8457, Singapore, December 2023. Association for Computational Linguistics.

[23] Vishnu Vinod, Krishna Pillutla, and Abhradeep Guha Thakurta. Invisibleink: High-utility and low-cost text generation with differential privacy, 2025.

[24] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

Reference Summary:
A newer vegan cookbook with a host of unique and delicious plant-based takes on traditional comfort foods.

ϵ	**LLM Generated Summary**
10	The product is a cookbook. Customers praise its food presentation and ease of cooking overall. Some complain about portion sizes and occasional mistakes.
16	The product is a cookbook. Customers praise its food presentation and ease of cooking. Some complain about portion sizes and taste inconsistencies.
27	The product is a cookbook. Customers praise its variety of recipes and ease of use. Some find little to criticize.
50	The product is a cookbook. Customers praise its readability and accessibility. Some complain about ingredient sourcing and meal planning complexity.
68	The product is a cookbook. Customers praise its recipes and cooking methods. Some complain about lack of clear instructions and portion sizes.
108	The product is a cookbook. Customers praise its recipes and cooking methods. Some complain about lack of nutritional information and limited recipe count.

Figure 7: Example of generated responses using Method 1 with temperature of 0.3

A Experimental Details

All experiments were conducted with a single NVIDIA A100 GPU on the Harvard FASRC cluster. The implementation was built using PyTorch and the HuggingFace Transformers library. The total computational cost for the results reported in this paper (including the grid search over privacy budgets, generation across all seeds, and automated metric evaluation) was approximately 96 GPU hours. We utilized `meta-llama/Llama-3.2-1B-Instruct` and `meta-llama/Llama-3.2-3B-Instruct` as the base generators. As described in the methodology, we utilized Gaussian Differential Privacy (GDP) for exact composition. The noise calibration was implemented using the `scipy.optimize.brentq` root-finding algorithm. Automatic evaluation metrics were computed using standard Python libraries to ensure comparability:

- **ROUGE:** Calculated using the `rouge-score` library.
- **BERTScore:** Calculated using the `bert-score` library (with its default English model). To accelerate evaluation, candidate and reference summaries were processed in batches of size 256.

For the One-Shot Summarization method, we used 1000 public documents to train our adapter. During training and inference, we conditioned the summarization model on publicly available product metadata (title, and, when present, categories) to ensure that the summaries of private reviews stay on topic.

For Method 1, we present some responses with different sampling temperatures below. For Method 2, we present some responses across 3 different runs under the same settings below as well.

Reference Summary:
A newer vegan cookbook with a host of unique and delicious plant-based takes on traditional comfort foods.

ϵ	**LLM Generated Summary**
10	The product is 出了地 clinching periodener foldsoftravel guidevdsetennenhama-linewsolesoup.microsofthoottenkleannerookhaven express expinkbreblankpaper spanner twatraksales infesisawe seacapeobrafixehandlemaid vigageriks.
16	The product is a cookbook. Customers praise its flavor and ease of dining upright. Note incomplete, incomplete review: Anonymous neighbor, who sometimes helps other neighbors dig up gardening tips offered previously online, noted here last night, no punch, little anger would work ther.
27	The product is a cookbook. Customers praise its quality execution of recipes and readability, but some complain about portions being woefully empty and limited use of spices.
50	The product is a cookbook. Customers praise its colorful pictures and clear instructions, but some complain about missing pages and confusing formatting.
68	The product is a cookbook. Customers praise its recipes and taste, but some complain about the ingredients and cost.
108	The product is a cookbook. Customers praise its recipes and ease of use, but some complain about ingredients and lack of variety.

Figure 8: Example of generated responses using Method 1 with temperature of 0.7

Reference Summary:
A newer vegan cookbook with a host of unique and delicious plant-based takes on traditional comfort foods.

ϵ	**LLM Generated Summary**
10	The product is No autoswil crud period relevant closely price Frozen chemvd-setsăm dessert Duty Nightscake Sight Plughalladder Maintenance Officer hour Immediate express Mealsilloxin blank Theory typefully twearrefer nationwide deliveredesisaweastyattoobra dinner toustbox gainsched Delivery.
16	The product is a cookbook industry. Customers praise some praises "book content greatness overall new innovations excellent specialty presses wife kids boyfriend neighbor relatives dying aging saving another cancer healthy individual lifestyle resorts shooting coast transportation stable buildings hospitals asteroid computers healthbox infinity clouds little alone injury sky disappear.
27	The product is 2-star rating, however expert reviewers claim customers prefer its concise descriptions and practical tips over anything. Rearranged with the current review: Sentence: The product is a cookbook. Customers praise its clear writing style and tasty recipes, but some complain about.
50	The product is a cookbook. Customers praise its. Customers praise its and some complain about.
68	The product is a book. Customers praise its introduction and organization, but some complain about usefulness and price.
108	The product is a cookbook. Customers praise its easy cooking methods and taste recipes, but some complain about lack of creative ideas and expensive.

Figure 9: Example of generated responses using Method 1 with temperature of 1

Reference Summary:
A newer vegan cookbook with a host of unique and delicious plant-based takes on traditional comfort foods.

ϵ	**LLM Generated Summary**
0.1	*First. &0:1,000!&<A>8.,2012.&Global |Na4|!,(2025), A.), (Tiger's!), (-Hertes for the -).-
0.5	while everyone, and all the G. Paws a Muck Ate to us on it's pasting up an F-ious Z-Mel-Paw-Gs-Awa-niv-i-pus-g-w-m-v-r-a-z-t
1	"The product is a [TYPE]. Customers praise its simplicity and ease of use, but some find it too bland for their taste."
2	Example: "The product is a cookbook, and customers praise its comprehensive recipes for [ASPECT_1] but find it lacking in terms of cooking techniques."
5	Example: "The product is a cookbook. Customers praise its recipes and cooking techniques, but some complain about the lack of [ASPECT_1] in their favorite dishes."
10	Please note that the product is not available for purchase on this platform.
50	Example: "The product is a [BOSH] series. Customers praise its simplicity and versatility, but some complain about the lack of recipes for certain ingredients."
100	Example: "The product is a [BOSH] series. Customers praise its simplicity and versatility, but some complain about the lack of recipes for certain ingredients."

Figure 10: Attempt 1 of generated responses using Method 2

Reference Summary:
A newer vegan cookbook with a host of unique and delicious plant-based takes on traditional comfort foods.

ϵ	**LLM Generated Summary**
0.1	" 'L' was, and the money for " The free", or as we'll make a global plan of our main class. In school's in Inch-Style is this year - I have been + + > = * on K>
0.5	The product is a [TYPE] and the following: "A1.2, but S_3. The way you can write an explanation for how to make one of these: * A$ *: 4. $: O_{N
1	BOSH!: Simple Recipes * Amazing Food * Categories: Books, Cooking by Ingredient Write exactly two sentences of the form: "The product is a [TYPE]. Customers praise its simplicity and ease. Some complain about it being too easy." The Bosh
2	Example: Here is the full review: Title: BOSH!: Simple Recipes * Amazing Food * Categories: Books, Cooking by Ingredient Write exactly two sentences of this form. The product has a [TYPE] and customers praise its amazing food. However some complain
5	Example: Example: "The product is a [TYPE]. Customers praise its delicious recipes and beautiful pictures, but some complain about the lack of ingredients list."
10	Example: "The product is a [BOSH] cookbook. Customers praise its simple recipes and all-plant approach, but some complain about the lack of detailed instructions."
50	Example: "The product is a [TYPE]. Customers praise its delicious recipes and beautiful photography, but some complain about the lack of ingredients list."
100	Example: "The product is a [BOSH] series. Customers praise its simplicity and versatility, but some complain about the lack of recipes for certain ingredients."

Figure 11: Attempt 2 of generated responses using Method 2

Reference Summary:
A newer vegan cookbook with a host of unique and delicious plant-based takes on traditional comfort foods.

ϵ	**LLM Generated Summary**
0.1	The product is a [TYPE] and the following: “A1.2, but S_3. The way you can write an explanation for how to make one of these: ∗ A$ ∗: 4. $: O_{N}
0.5	The product is a [TYPE]. Customers praise its [ASPECT_1] and [ASSURE_2], but some complain about it’s more like they are. For us, we believe that the customer service supports our business model. For you
1	* "The best food for [GOLDEN] children is: * Goldfish, but some prefer the more expensive and nutritious ** salmon**." +1 2 Note that these are simple recipes: 3a4b5c6
2	For example: "BOSH!: Simple Recipes is a [TYPE] that has received rave reviews for its delicious and easy-to-make recipes. However, some customers have noted issues with the ingredients being too expensive."
5	You can use this format to write a review of the product.
10	You can fill in the blanks with your own information to create a unique summary. For example: "The product is an [TYPE] that has been praised for its simplicity and ease of use, but some customers have reported issues such as difficulty finding ingredients or
50	Example: "The product is a [BOSH] series. Customers praise its simplicity and versatility, but some complain about the lack of recipes for certain ingredients."
100	Example: "The product is a [BOSH] series. Customers praise its simplicity and versatility, but some complain about the lack of recipes for certain ingredients."

Figure 12: Attempt 3 of generated responses using Method 2