Comparing Differential Privacy in Practice: A Cross-Domain Analysis of Differentially Private-Offsite Prompt Tuning and Google's Differential Privacy Library

Yurui Zheng
Supervisor(s): Dr. Zeki Erkin, Dr. Roland Kromes
[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2025

Name of the student: Yurui Zheng
Final project course: CSE3000 Research Project
Thesis committee: Dr. Zeki Erkin, Dr. Roland Kromes, Dr. Xucong Zhang

Abstract

Differential Privacy (DP) has become one of the most used approaches to protect individual data. However, its implementation can vary significantly depending on the context we are using it. In this study, we aim to compare two such implementations of DP: Google's Differential Privacy, an open-source library used for structured data analytics, and Differentially Private Offsite Prompt Tuning (DP-OPT), a tool used for the adaptation of machine learning models.

The main research question of this study is the following: "How do DP-OPT and Google's Differential Privacy Library compare when accounting for different factors in different contexts?" We aim to conduct this research by doing a literature-based study where no empirical experiments will be performed. This is because of the underlying complexity of both tools, especially DP-OPT, and the time constraints posed in this project.

The main results of this study show that Google DP is a tool that benefits from its interpretability and speed when processing data analysis, while DP-OPT shows a higher accuracy when training large language models (LLMs). Meaning that there is no single mechanism for each problem, but instead it depends on factors like the underlying task complexity, current available resources, and the final goal of the task.

By comparing these tools side-by-side, this research aims to provide more insights into how each tool behaves and performs under different contexts and tasks. We aim to guide developers and researchers to make better and more informed decisions when choosing a tool for their desired task.

1 Introduction

How can we analyze sensitive data while still guaranteeing the individual's privacy? For many years, researchers, computer scientists, and developers have tried to answer this question with very little success. Early attempts to preserve data privacy either failed to protect it under scrutiny or rendered the data unusable [21]. A formal breakthrough came in 2006 by Dwork et al. [10] who introduced Differential Privacy (DP). DP changes the problem entirely, rather than hiding the data from attackers, it makes sure that the output of any process is statistically indistinguishable. Meaning that no matter how much an adversary may know about the process or the dataset itself, it cannot confidently determine if any individual's data was used to take part in the process [11]. As a result, DP has become one of the most robust frameworks that is used nowadays to protect an individual's privacy.

However, DP is still only a theoretical concept that needs to be applied in practice. In this study, we will analyze two of such implementations, the first one, Google's Differential Privacy Library [14], which is an open-source tool used for tabular data analysis. It adds carefully calibrated noise to statistical results while using mechanisms that allow limiting user contributions and tracking privacy budget. The second one is Differentially Private Offsite Prompt Tuning [17], a technique used for adapting machine learning models. It generates locally a private discrete prompt that is later used for adaptation without exposing raw data.

Although both tools are based on the same theoretical principle of DP, they vary in the domain where they are applied. One is used for more statistical purposes, while the other focuses more on privately adapting language models. Therefore, comparing these two tools is not about finding a universally superior one, but instead about understanding how each tool performs in different contexts, under different parameters, and for different tasks. Nowadays, more and more applications are incorporating both statistical analysis and machine learning

models in their tasks, therefore, the need for a cross-domain comparison has become even more relevant for this purpose. To the best of our knowledge, there is currently no systematic study that compares such divergent tools across different settings. Most of the existing work either focuses on DP libraries for analytics only or on DP applications in machine learning, but rarely on both in the same study.

Our contribution is based on analyzing the cross-domain applicability of DP implementations. Specifically, we compare two such implementations. Google's DP library, which is used for privatized statistical queries, and DP-OPT, which privatizes prompt generation for LLMs. This evaluation aims to show how DP principles translate into trade-offs between data utility, computational performance, and their different privacy accounting mechanisms in diverse contexts. By going from single-domain to cross-domain comparisons, this work seeks to contribute to the growing literature on applied DP and provide better guidance for future developers and researchers to make better informed decisions.

To guide this investigation, we focus on the following research questions:

- How do DP-OPT and Google's Differential Privacy Library compare in their privacy-budget accounting mechanisms?
- What are the performance trade-offs (runtime, memory) of each tool on representative ML and analytics tasks?
- How does the output utility of DP-OPT compare to that of Google's Differential Privacy Library across different use cases?

To answer these questions, we have adopted a literature-based approach rather than conducting our own empirical experiments, where we analyze existing results from peer-reviewed publications, benchmarks, and other technical reports. This is done mainly for two reasons. First, because of the underlying complexity of both tools, particularly DP-OPT. Second, because there already exist evaluations and benchmarks from which meaningful insights can be drawn. Therefore, by systematically reviewing existing publications, this paper aims to provide a grounded analysis based on real-world data, while still trying to remain feasible within the time and constraints of this project.

The structure of the paper is as follows. Section 2 provides background on the mathematical foundations and core mechanisms of DP. Section 3 surveys existing literature on DP tools and their evaluation. Section 4 introduces the criteria and metrics used to assess the two implementations. Section 5 presents a detailed comparison based on privacy accounting, runtime performance, memory usage, and utility. Section 6 discusses the responsible research practices followed throughout this study. Section 7 reflects on the results and their implications. Finally, Section 8 concludes with a summary of findings and directions for future research.

2 Background

2.1 Differential Privacy Fundamentals

Differential Privacy (DP) is a mathematical framework for quantifying and controlling the privacy risk of individuals in statistical databases [10]. Formally, a randomized algorithm $\mathcal{M}$ is said to be (ε, δ)-differentially private if, for all adjacent datasets D and D' (differing

by a single individual), and for all possible outputs S, the following equation holds:

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

Here, ε (the privacy budget) measures the strength of the privacy guarantee, while δ allows for a small probability of failure.

The insight of DP is that the output of a computation should not substantially change depending on whether any single individual's data is included. This implies that even an adversary with external knowledge cannot confidently infer the participation of an individual. DP can be achieved through mechanisms such as the Laplace and Gaussian mechanisms, which inject noise calibrated to the sensitivity of the query function. In practical systems, cumulative privacy loss across multiple queries is tracked through privacy accounting techniques, such as composition theorems [12].

2.2 Google Differential Privacy Library

The Google Differential Privacy Library is one of the most used and accessible frameworks that apply DP to real-world data analytics [15]. It is an open-source software that uses different mechanisms to guarantee the indistinguishability of sensitive data within a dataset. The main functionality of Google DP can be expressed in three components: contribution bounding, noise addition, and privacy budget accounting. These components together guarantee that user information is safe, even when multiple queries are issued over the same data or on overlapping datasets.

2.2.1 Contribution Bounding

Before making any privacy changes, the system limits how much a single user can contribute to the data. This is done in two ways: (1) cross-partition bounding, which limits the number of groups or partitions a single user can contribute to, (2) per-partition contribution bounding, which limits the number of times a single user can contribute to a single partition. We do this to ensure that no single user can overly influence the outcome, which potentially leads to more noise in the outcome or even a privacy leakage.

In addition to this, Google DP also applies a probabilistic suppression mechanism, this threshold is used to remove partitions that do not have sufficient user contributions. Partitions that do not meet a specific threshold τ, often computed with a randomized mechanism, are not considered for the final outcome with high probability. Since partitions with very few user contributions tend to leak private information, this thresholding essentially stops that [20, 38].

2.2.2 Noise Addition

Once the user contributions have been bound, the library applies randomized noise to the results through the use of either the Laplace mechanism or the Gaussian mechanism. These algorithms add calibrated noise depending on the amount of global sensitivity of the query, meaning that the amount of noise depends on "the maximum change in the output that can result from modifying a single individual's data". Formally, for a given query function f with sensitivity Δf, the Laplace mechanism computes the noise from a distribution with scale $\Delta f/\varepsilon$, whereas the Gaussian mechanism computes the noise from $\Delta f \cdot \sqrt{2\log(1.25/\delta)}/\varepsilon$ [12]. This added randomness ensures that any single user's inclusion or exclusion produces a statistically negligible change in the query result.

2.2.3 Privacy Budget Accounting

In the final component, we talk about privacy budget accounting. As mentioned before, Google DP tracks how much privacy the mechanism has spent over multiple queries or repeated executions. Since each operation that accesses the data consumes a fraction of the privacy budget, defined as (ε, δ). The total privacy loss is computed using advanced composition theorems [19] that provide a tighter bound on the total privacy loss compared to basic summation, for example.

2.3 Differentially Private Offsite Prompt Tuning

Differentially Private Offsite Prompt Tuning (DP-OPT) is a more novel technique designed to use large language models (LLMs) under sensitive data without altering the original model weights [17]. Unlike other fine-tuning methods that often require changing thousands of model parameters or gradient-based approaches, DP-OPT focuses on adjusting privately a discrete prompt that is later used for inference on the desired model. This is particularly useful specially if such a model is closed-source, it has restricted access or simply sharing sensitive data with a model in the cloud is not a viable solution.

DP-OPT works on the premise that discrete prompts, when generated effectively, can direct LLMs to perform a specific task without the need to alter the model's internal weights [32]. However, it has been shown that prompt tuning methods such as DLN-1 [33] often times leave private information from the training samples [8, 23]. DP-OPT addresses this issue by incorporating DP into the two main processes of DNL-1, prompt generation and prompt selection.

The first phase is based on the sample-and-aggregate paradigm [29]. Given a training set of inputs, their corresponding outputs, and an initial model prompt, the mechanism first splits the data into disjoint subjects without any overlapping data. Then for each subset, the mechanism uses the prompt to predict the output, knowing the results, the model learns from it and generates a new token from each subset. Since often times the output space is very large and we may end up with a wide range of tokens, the naive Exponential Mechanism [26] would not be appropriate to use since the result would end up with a high variance and poor utility [7]. The answer to this would be to use a mechanism that limits the amount of token-level votes, which is exactly what LimitedDomain does [9]. This mechanism, as the name explicitly says, limits the amount of candidate tokens to the top-k vote counts, and in the case where the vote distribution is very similar, the algorithm either is rerun with a new set of tokens or stops the process.

After generating the candidate prompts from the original template and the tokens, each prompt is then tested to see its accuracy. Since this step is also DP, we will use the Exponential Mechanism to choose the final prompt. This is done by evaluating each prompt on its utility and assigning a probability depending on how good each prompt is, meaning that better prompts have higher chances of being chosen. The algorithm then randomly chooses a prompt based on this probabilistic distribution.

Regarding the privacy accounting method, DP-OPT uses the Rényi Differential Privacy (RDP) [28] to track the privacy loss across both phases. RDP uses a more advanced composition mechanism that tracks the privacy loss across multiple iterations of the prompt generation phase and the prompt selection phase. However, RDP still allows for some interpretability since, in the end, it can be converted to the traditional (ε, δ) guarantee.

3 Related Work

There exists several studies that have performed evaluations on open-source tools for Differential Privacy (DP). Among the ones that seem most comprehensive is the work done by Zhang et al. [41], which benchmarks a wide range of DP libraries. Their evaluation focuses on three main aspects, namely, the privacy guarantees, the utility of the data, and how each tool performs computationally.

However, the work by Zhang et al. is limited only to more traditional tools in DP applications, namely, related to statistical queries and relatively simple machine learning tasks. Their review does not seem to extend to more recent techniques where DP is applied, particularly in the context of training and deployment of large language models (LLMs). These newer use cases have shown to raise new challenges in the way we manage our computer resources, account for the privacy budget, and also how the overall utility is affected [17]

Nevertheless, there does exist other works that have used DP in the context of language models. Examples of them are Carlini et al. [6] and Duan et al. [8], who have shown in their work that even small prompt-based fine-tuning can lead to a privacy leakage, motivating the use of private prompt tuning methods. It was not until early 2024 that Hong et al. introduced Differentially Private Offsite Prompt Tuning (DP-OPT) [17].

In relation to this last development, we have seen in the literature that most work evaluating DP tools often focuses solely on accuracy and privacy guarantees, but often forgets the importance of practical factors such as runtime and memory consumption of such tools. However, in one particular study done by Hanke et al. [16], they analyze these aspects for DP-OPT in 4 different contexts, revealing important trade-offs between privacy budget, runtime overhead, and memory consumption of such tools. However, such evaluations still remain in the same domain and rarely extend to compare across different fields of DP.

To the best of our knowledge, no prior work has systematically compared classical tools, such as Google DP, to a more novel and recent one in the context of LLMs, such as DP-OPT. By focusing on both systems' utility, performance overhead, and privacy accounting mechanisms, our study aims to contribute a cross-domain analysis that has been missing so far in the literature.

4 Methodology

The literature review of this research was done using the snowballing method, where, starting with a set of initial papers, we would then identify other relevant information to the research through citations and references [39]. This method was chosen over other ones, such as PRISMA [31], due to the relatively small number of available studies done regarding Differentially Private Offsite Prompt Tuning (DP-OPT). This method made it easier and, in our opinion, faster to identify relevant sources without the need to use explicit inclusion or exclusion criteria. Which makes it a better fit for this project in particular.

The evaluation and analysis of Google DP and DP-OPT are based on three criteria that are directly influenced by real-world applications and scenarios. Namely: data utility, performance overhead, and privacy budget accounting. In addition to this, a table of symbols can be found in Table 1 for better context across the paper.

Utility refers to the output quality or usefulness after applying DP mechanisms. In Google DP, this refers to how close the outcome is to the same query without having applied DP. Whereas in DP-OPT it refers to the level of accuracy a certain model has performed on a specific task.

Performance overhead refers to the runtime and memory overhead that each tool requires during performance. Even though both tools have been theoretically described, these metrics show how each mechanism scales and performs under real-world settings.

Privacy budget accounting refers to how each method tracks and allocates the total privacy budget over multiple iterations of the process. In the case of Google DP, it is a more traditional (ε, δ)-composition that works by using a built-in ledger. For the case of DP-OPT, it uses the more advanced Rényi Differential Privacy (RDP) [28], which allows it to have a tighter composition after iterations. Tracking the privacy budget is essential to avoid its overconsumption, which can lead to less private guarantees of information leakage.

Symbol	Definition
D	Original dataset
D'	Neighboring dataset differing by one individual
$\mathcal{M}$	Randomized mechanism (e.g., a differentially private algorithm)
ε	Privacy loss parameter (controls the strength of privacy guarantee)
δ	Probability of a privacy breach (used in (ε, δ)-differential privacy)
τ	Contribution threshold (used in Google DP to limit partitions)
C_u	User contribution limit
$RMSPE$	Relative Mean Square Percentage Error (a utility metric)

Table 1: Table of Symbols Used in This Paper

5 Comparison of Methods

5.1 Google Differential Privacy Library Results

The Google Differential Privacy Library includes several tunable parameters that significantly influence the balance between privacy guarantees and data utility [2]. Among the most critical are the dataset size, the privacy parameters ε and δ, the limits on user contribution, both to a single partition and across multiple partitions, and the clamping of values within specified bounds.

In 2023, an evaluation of such a library was conducted [41], using two datasets that were carefully selected due to its realistic data and to their compatibility with the library functionalities: the Parkinson's Telemonitoring dataset [36] and the Massachusetts Health Reform Survey [24]. In their study, the authors primarily varied the dataset size and the privacy budget ε to analyze their effects on utility and performance. Regarding the rest of the parameters, δ was held fixed at a small value (effectively zero for pure DP mechanisms), similarly, the clipping bounds were often derived from the actual minimum and maximum values of the data, and the user contribution limit was kept at the library's default of one contribution per user.

The evaluation consisted of subsampling the datasets to create scenarios of different data sizes (e.g., subsets of 7k, 8k, 9k, and the full 9358 records in the health dataset) to see how the dataset size would affect utility. As expected, larger dataset sizes resulted in a better utility, which meant fewer errors in the query outputs. It was shown that for simpler queries such as SUM, COUNT, and AVG, the increase of data size would spread the noise. Particularly, we could see how Google DP's relative mean square percentage error

(RMSPE) ranged from approximately 0.1-20% compared to that of other tools, which had shown RMSPEs between 0.2 and 350% [41].

Regarding the privacy budget ε, it was seen that, as expected, smaller values of ε, meaning stricter privacy constraints, resulted in a higher RMSPE, meaning lower data utility. The evaluation varied ε over a range of values based on prior research and industry recommendations [27, 3]. The results showed that at low ε values (e.g., $\varepsilon \approx 0.1$), noise levels were high and errors increased significantly ($RMSPE \approx 3.5\%$), especially in more complex queries ($RMSPE \approx 20\%$). On the other hand, when ε was raised (e.g. $\varepsilon \approx 3$), the errors dropped consistently across all query types ($RMSPE \approx 1\%$ for simpler queries and $RMSPE \approx 15\%$ on more complex ones) [41].

Although [41] maintained fixed values for δ, clipping bounds, and contribution limits, further insights into these parameters are provided in [38]. Using the TPC-H benchmark [35], they analyzed how these settings affect utility and output completeness. For δ, which bounds the probability of privacy failure due to rare or small-count groups, relaxing the parameter resulted in more complete outputs. It has been seen that increasing δ reduced the number of suppressed partitions nearly linearly, while stricter values led to higher error due to more aggressive result filtering.

Regarding clipping bounds, it was seen that they were also of relevance. In their study, they showed that tighter bounds ($UpperBound \approx 80$) would yield higher expected errors ($RMSPE \approx 0.4\%$) due to its risk of moving read data, while too wide bounds ($UpperBound \approx 200$) would also show some errors ($RMSPE \approx 0.3\%$) since they would add excessive noise. The best results were shown by limiting only extreme outliers ($UpperBound \approx 120$, resulting in $RMSPE \approx 0.2\%$), where the main goal here is to find the balance between reducing noise and minimizing bias.

In addition to this, user contribution limits also show some important trade-offs. [38] had shown that depending on the data distribution, some errors would decrease faster than the others, but as a general pattern, as the contribution limits increased ($10 \rightarrow 100C_u$), the median percent error decreased somewhat linearly ($1 \rightarrow 0\%$).

In terms of performance overhead, the most influential parameter was dataset size. In contrast, changes to ε had minimal impact on runtime or memory usage. This is thought to be because ε just affects the scale of added noise without altering the underlying computational structure. As reported in [41], the runtime for a given query remained relatively stable across different ε values, and memory usage was mostly driven by static allocations and data processing.

Interestingly, larger datasets not only improved utility but also showed to slightly reduce the relative runtime overhead. For example, the runtime for DP queries was approximately 100-130% of the non-private runtime on the largest datasets, compared to 110-150% on smaller ones.

Memory overhead on the other hand, measured using Docker container tracking [5], showed variable behavior. Although larger datasets generally required more memory due to the volume of data processed, the relationship that was seen from the results were not linear. Instead, memory usage fluctuated across dataset sizes, with DP functionality adding only a small percentage (up to 3%) over baseline memory consumption.

5.2 Differentially Private Offsite Prompt Tuning Results

Differentially Private Offsite Prompt Tuning (DP-OPT) uses a smaller local language model (e.g., Vicuna-7B) to generate a private prompt for a specific task, which is later used to query

a larger, potentially closed-source model (e.g., GPT-4). Therefore, the performance of DP-OPT is influenced by several metrics, including the privacy budget ε, the capacity of the model, and the nature of the downstream task.

In the original paper we can see several experiments that test how the utility, in this case the models accuracy to perform a task, varies under different privacy conditions [17]. The authors show that Vicuna-7B [7] can yield an accuracy similar to the non-private baselines when used for private prompt generation under the SST-2 classification task [37]. For instance, as ε is reduced to 1, the accuracy was shown to drop from the low-90s to approximately 86%. This decline is thought to be because of the LimitedDomain mechanism [9], which limits the inclusion of certain data values when the privacy budget is too restrictive. However, larger models, such as LLaMA-2-13B or LLaMA-2-70B [34], have been shown to perform better with a small ε, maintaining accuracy above 90% even under strict privacy settings [17]. This can suggest that the privacy-utility trade-off is significantly mitigated when using stronger language models.

In addition to this, we can also see how across several classification tasks, DP-OPT's accuracy was generally very close to the non-private prompt tuning baseline. For example, when using DaVinci-003 (text generation version of GPT3.5) [30] , DP-OPT was seen to obtain an average accuracy of 81.4% compared to 82.9% of the one without privacy [17], meaning there was a minimal utility loss.

Furthermore, DP-OPT consistently outperformed PromptDPSGD, a baseline applying DP-SGD [1] to soft prompts [8], due to its ability to transfer the learned prompt to a more powerful model [1].

Nevertheless, DP-OPT does not always show to match the performance of full DP fine-tuning methods, especially when applied to open-source models. A recent benchmark study published in late 2024 by Hanke et al. had shown that DP-OPT underperformed DP fine-tuning techniques such as Private LoRA [40] and DP-FineTune [22] on several tasks [16]. For instance, on the TREC classification benchmark [25], DP-OPT was seen to achieve over 26% lower accuracy compared to the other approaches, and on average, Private LoRA applied to Vicuna-7B was seen to achieve 90.3% across four tasks, whereas DP-OPT with the same model only reached 75.3%.

As already mentioned, the model's capacity is not the only factor that determines the outcome's utility, but it is also dependent on the type of task to be performed. To this end, the experiments had worked with DP-OPT across four different cases: SST-2 [37], TREC [25], MPQA [25], and Disaster [4]. During their experiments, it was shown that for some particular tasks, namely, the binary classification ones such as SST-2 and MPQA, DP-OPT outcome is very close to the non-private model under moderate privacy levels (e.g., $\varepsilon \approx 8$). At this exact setting, it was also seen how DP-OPT achieves 92.2% accuracy on SST-2 versus 92.4% without DP, and 85.8% on MPQA, equaling the non-private result. A similar trend is observed for the Disaster classification task, where DP-OPT scored 78.9% compared to 79.0% in the non-private case [17].

When talking about computational performance, on one hand, DP-OPT has been shown to require fewer resources compared to other gradient-based DP methods regarding runtime. According to the study done by Hanke et al. [16], the total compute cost of DP-OPT, including both training and inference, was significantly lower than that of DP fine-tuning techniques. Specifically, using the Vicuna 7B model, training with DP-OPT costs approximately \$2.10, compared to \$13.80 for Private LoRA, and including inference, the total cost for DP-OPT was about \$2.90, while Private LoRA required \$14.60. In terms of actual measurable time, DP-OPT required less than one hour on a single GPU for prompt genera-

tion, while DP fine-tuning took several hours under the same conditions. Furthermore, when used to adapt GPT-3.5, DP-OPT had shown to add only a slight overhead compared to non-private prompt tuning, with prompt generation costing approximately $2.10 in additional compute.

On the other hand, when talking about memory overhead, DP-OPT has been shown to be on the lower side of it. Since the method avoids backpropagation, it no longer needs to store gradients or optimizer states. As explicitly mentioned in [17], DP-OPT is "much more memory efficient than any gradient-based method, including soft prompt tuning." This is thought to be because prompt generation only involves forward passes, meaning the memory it uses during training is kept constant and at a minimum.

6 Responsible Research

6.1 Data and Privacy

This project did not involve the collection of new data from human subjects. All data used in the evaluation was obtained from publicly available datasets that had already undergone appropriate anonymization and ethics approval. Regarding the data that was used (e.g., the Massachusetts Health Reform Survey [24] and the Parkinson's Telemonitoring dataset [36]), these were accessed through reputable repositories [18] and handled in accordance with the F.A.I.R. principles (Findability, Accessibility, Interoperability, and Reusability) [13]. Since no identifying or sensitive data was collected or stored during this project, there were no special storage or retention considerations required. As the project only used existing data and did not generate new datasets, there is no new data to be made available apart from referencing the sources.

6.2 Research Integrity

Throughout the project, proper academic integrity was maintained by citing all relevant sources, including research papers, code libraries, datasets, and benchmarks used for analysis. All externally sourced material has been clearly referenced, and licensing terms of the tools and datasets were respected. Sections of the report that were supported by AI tools (such as for drafting, checking grammar, coherence, etc) have been reviewed and revised to ensure accuracy, ownership, and integrity. All presented results have been presented as they are from the original sources, with no fabrication or manipulation.

6.3 Replicability and Reproducibility

Regarding reproducibility, all evaluation results were based on peer-reviewed or publicly released benchmarks. The original sources for both the Google Differential Privacy Library and DP-OPT were verified and are available to the public. The methodology followed a transparent process, with clear references to the parameters and setups described in the literature. Although we did not create our own implementation for this project, future researchers can replicate the analysis by following the documentation and benchmarks cited. The tools and datasets referenced in the study are open-source or freely available.

6.4 Bias

In terms of bias, this has likely occurred from the selection of sources and benchmarks. Since they use their own datasets, it is very likely that they are biased inherently. To mitigate this, during the gathering of results, discussion, and conclusion sections, we have aimed to base our analysis on multiple sources, making sure the results are as consistent as possible. Furthermore, it is very important to also see that unnoticed but existing biases could still remain in the study, and therefore extend to other contexts where organizations make use of our analysis. We would like to address this in future works, where we can empirically obtain new results, making sure to account for bias as much as possible.

6.5 Beyond the Project

We believe that this research may pose potential implications in the way organizations use differential privacy. One risk could potentially be that organizations misinterpret the results, for example, they could assume that one tool always performs better than the other without taking into account other considerations. To mitigate this, we tried to emphasize that choosing a tool should depend on many factors. Furthermore, we believe that extending the use of differential privacy, rather than causing harm, it helps to protect the privacy of individuals ethically. Finally, we suggest that readers should not extend the results to other contexts that have not been covered in this study, especially those involving highly sensitive data or situations with high risk.

7 Discussion

7.1 Utility Trade-offs

The comparison between Google's Differential Privacy (DP) Library and Differentially Private Offsite Prompt Tuning (DP-OPT), shows that the utility function not only depends on how the privacy is handled, but also on the **system architecture** of the mechanism.

For the case of Google DP, the utility is mainly influenced by its tunable parameters, the dataset size, clipping bounds, user contribution limits, and the privacy budget. As shown in [41], for queries like SUM, COUNT, and AVG for example, increasing dataset size has shown to reduce the relative error (RMSPE), as well as by also tuning the clipping and contribution limits.

On the other hand, DP-OPT works its way around utility degradation by **outsourcing generalization** to a larger language model at inference time. This allows DP-OPT to use more powerful, and even closed-source, language models (e.g., GPT-3.5 or LLaMA-2-70B) [17] to perform the required task. This created a new paradigm in the context of privacy preservation in the context of machine learning. This was demonstrated in [16], where DP-OPT has been seen to outperform other techniques on text classification tasks, maintaining high utility even when ε was very low (≈ 1).

This shows that the privacy-utility trade-off depends heavily on the context: while Google DP relies on parameter tuning and statistical aggregation, DP-OPT takes advantage of more powerful ML models, as well as on the type of task it is required to perform. Therefore, we would like to highlight the difference in the **system architecture**, which, to the best of our knowledge, has not been directly compared in existing literature.

7.2 Performance Considerations

Regarding the performance characteristics of both tools, it has been seen that in both cases, they have all shown to achieve a competitive performance compared to other mechanisms that work in their same context.

Google DP shows low runtime and memory overhead ($10 - 30\%$ and $\leq 3\%$ respectively) [41], which we believe is not only because of how the mechanism is designed, but also due to the low-dimensional data it operates on and the inherently lightweight queries.

On the other side, DP-OPT inherently requires a greater computational cost in absolute terms, but this is believed to be normal due to the underlying complexity of the task this performs. It has been seen that DP-OPT has achieved a significantly lower training cost compared to other machine learning-based DP approaches like DP-SDG or Private LoRA [16]. In terms of memory, thanks to the mechanism design, it does not need any backpropagation, resulting in big savings in memory consumption.

These findings indicate that the performance should not be evaluated in terms of absolute terms, but rather depending on the **complexity class** of the underlying task. This criterion is also highly important for organizations when making a choice, to not only make it based on privacy or utility, but also on system resources.

7.3 Privacy Accounting and Interpretability

Utility and performance consumptions are not the only difference between these two tools, privacy budget tracking has also been shown to be different for the two, and not just in technical terms, but also regarding its usability.

On one hand, Google DP uses the traditional (ε, δ) accounting, which is well understood, easy to audit, and directly supports formal compliance reporting. This is particularly useful in organizational contexts where interpretability and reproducibility are the main priority.

On the other hand, DP-OPT uses a more complex accounting mechanism, namely Rényi Differential Privacy (RDP) [28]. This offers a tighter composition that is especially useful for the iterative training process. Despite the fact that the result can be translated to (ε, δ) in the end and that it results in more accurate cumulative privacy guarantees, the middle steps of composition are less transparent and harder to understand for non-experts.

These two differences have shown that, as well as the other factors, privacy budget accounting does not have a universal solution, but instead, the algorithm chosen should be evaluated on its usability, ability to audit, and interpretability, depending on the user's requirements.

7.4 Implications and Contextual Guidance

The key takeaway of this study is that no single DP tool has been seen to offer universally optimal performance across all use cases. On one hand, Google DP can be better suited for analytics systems that require transparent, efficient, and auditable privacy guarantees. Google DP has been seen to perform best on low-dimensional data and simple statistical queries, where utility loss can be minimized through careful parameter tuning.

DP-OPT, on the other hand, is tailored for modern machine learning tasks, particularly in scenarios involving personalized or federated data. It allows private adaptation of powerful language models without requiring raw data to leave the local device. While it demands more in terms of computational resources and expertise, it compensates with higher flexibility and accuracy guarantees when performing in different types of tasks.

As a result, our comparison is not meant to say that we should only choose one tool and forget about the other. Instead, it aims to highlight a question that organizations tend to face more nowadays: *where in the pipeline should differential privacy be applied?* In systems that include both statistical analysis and machine learning, developers may need to choose between protecting the data early (via DP analytics), protecting it later (via DP model tuning), or applying DP at multiple stages, where each decision comes with trade-offs, in utility, interpretability, and cost.

Because most existing studies treat analytics and machine learning separately, the field lacks a unified framework for cross-domain evaluation. By comparing these two tools, we contribute to the literature by showing that selecting a tool in the context of DP is not only a matter of algorithms, but also of **system architecture, task objectives, and user constraints**. Our results aim to offer practical guidance for developers, researchers, and engineers deciding what is the best way to integrate differential privacy into their system architecture.

8 Conclusions and Future Work

This research aimed to compare two tools that implement Differential Privacy (DP) in very different ways: Google's Differential Privacy Library, which is used for statistical analysis, and Differentially Private Offsite Prompt Tuning (DP-OPT), which applies DP to tuning language models. Although both tools rely on the same mathematical definition of privacy, they are designed for different types of tasks and have different goals. Therefore, the purpose of this research was not to find which tool is better, but to understand how DP behaves when used in very different settings.

Based on public evaluations and benchmarks, we considered three metrics: utility, performance overhead, and privacy accounting methods. The results show that Google DP is best suited for tasks where speed, low memory usage, and transparent privacy guarantees are important, since it uses standard accounting methods that are easier to track and understand. On the other hand, DP-OPT performs well in machine learning tasks, especially when privacy must be preserved during model tuning. It does this by generating private prompts locally and using a more powerful external model for inference. This design lets it keep good performance even under strong privacy constraints, but the downside is that it requires more computing power and uses a more complex privacy accounting system.

Our findings suggest that selecting a DP implementation should not be seen as a simple choice of one over the other, but rather as an informed decision about where in the data pipeline privacy protections are most effective. In many real-world systems, analytics and machine learning coexist, and organizations must take into account the trade-offs in privacy guarantees, utility loss, and operational overhead at each stage. This study contributes to the literature by providing a structured, cross-domain comparison, helping bridge the gap between DP in analytics and DP in machine learning.

Because this project had limited time and constraints, we focused on reviewing existing studies instead of running our experiments. This made it possible to compare the tools based on real-world data that was already available, but also meant that we could not evaluate them under other conditions to see how they would perform in different scenarios.

Regarding future work, it would be valuable to conduct new empirical experiments to validate the results observed in previous studies. Additionally, exploring the practical usability of both tools could show interesting insights, which include: evaluating the ease of integration into data pipelines, installation processes, and whether they are constantly up-

dated. Finally, another interesting direction would be to study how these tools perform when they are both implemented within the same pipeline, from data gathering, through analytics, and to machine learning. This could help determine whether the tools can be effectively combined or if their integration introduces any limitations or conflicts.

As nowadays more applications use both analytics and machine learning, understanding how different DP tools work in practice is becoming more important. This research takes a step in that direction by offering a side-by-side view of how DP behaves in two very different but increasingly connected areas of data science.

A Use of LLMs

ChatGPT was used to generate ideas, to gather information, and/or to assist in the writing process in the following context and prompts:

- Explain to me what X is, please do so by providing an example.
- Edit this paragraph that is in the literature review above in an academic tone, featuring a clear and succinct writing style: *Insert my own text*

References

[1] M. Abadi, H.B. McMahan, A. Chu, I. Mironov, L. Zhang, I. Goodfellow, and K. Talwar. Deep learning with differential privacy. volume 24-28-October-2016, pages 308–318, 2016.

[2] K. Amin, A. Kulesza, A.M. Medina, and S. Vassilvitskii. Bounding user contributions: A bias-variance trade-off in differential privacy. volume 2019-June, pages 388–399, 2019.

[3] Apple Inc. Differential privacy overview. `https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf`, 2017. Accessed: 2025-05-15.

[4] T. Bansal, R. Jha, and A. McCallum. Learning to Few-Shot Learn Across Diverse Natural Language Classification Tasks. pages 5108–5123, 2020.

[5] C. Boettiger. An introduction to Docker for reproducible research. volume 49, pages 71–79, 2015. Issue: 1.

[6] N. Carlini, F. Tramèr, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, Ú. Erlingsson, A. Oprea, and C. Raffel. Extracting training data from large language models. pages 2633–2650, 2021.

[7] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J.E. Gonzalez, I. Stoica, and E.P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. `https://lmsys.org/blog/2023-03-30-vicuna/`.

[8] H. Duan, A. Dziedzic, N. Papernot, and F. Boenisch. Flocks of Stochastic Parrots: Differentially Private Prompt Learning for Large Language Models. volume 36, 2023.

[9] D. Durfee and R. Rogers. Practical differentially private top-k selection with pay-what-you-get composition, September 2019. arXiv:1905.04273 [cs], https://arxiv.org/abs/1905.04273.

[10] C. Dwork. Differential privacy. volume 4052 LNCS, pages 1–12, 2006.

[11] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. volume 3876 LNCS, pages 265–284, 2006.

[12] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–487, 2013.

[13] GO FAIR Initiative. Fair principles: Findable, accessible, interoperable, reusable. https://www.go-fair.org/fair-principles/. Accessed: 2025-06-17.

[14] Google. Google differential privacy library. https://github.com/google/differential-privacy, 2020. Accessed: 2025-04-28.

[15] Google Inc. Enabling developers and organizations to use differential privacy. https://developers.googleblog.com/2019/09/enabling-developers-and-organizations.html, 2019. Accessed: 2025-04-28.

[16] V. Hanke, T. Blanchard, F. Boenisch, I.E. Olatunji, M. Backes, and A. Dziedzic. Open LLMs are Necessary for Current Private Adaptations and Outperform their Closed Alternatives. volume 37, 2024.

[17] J. Hong, J.T. Wang, C. Zhang, Z. Li, B. Li, and Z. Wang. DP-OPT: MAKE LARGE LANGUAGE MODEL YOUR PRIVACY-PRESERVING PROMPT ENGINEER. 2024.

[18] ICPSR, Inter-university Consortium for Political and Social Research. Icpsr: Inter-university consortium for political and social research. https://www.icpsr.umich.edu/web/pages/. Accessed: 2025-06-17.

[19] P. Kairouz, S. Oh, and P. Viswanath. The composition theorem for differential privacy. volume 2, pages 1376–1385, 2015.

[20] A. Korolova, K. Kenthapadi, N. Mishra, and A. Ntoulas. Releasing search queries and clicks privately. pages 171–180, 2009.

[21] T. Li and N. Li. On the tradeoff between privacy and utility in data publishing. pages 517–525, 2009.

[22] X. Li, F. Tramèr, P. Liang, and T. Hashimoto. Large language models can be strong differentially private learners, November 2022. arXiv:2110.05679 [cs], https://arxiv.org/abs/2110.05679.

[23] K. Liu. The entire prompt of microsoft bing chat?! (hi, sydney.), 2023. https://twitter.com/kliu128/status/1623472922374574080, Accessed: 2025-06-04.

[24] Sharon K. Long. Massachusetts health reform survey, 2018. https://doi.org/10.3886/ICPSR37411.v1, 2019. ICPSR 37411.

[25] Y. Lu, M. Bartolo, A. Moore, S. Riedel, and P. Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity, March 2022. arXiv:2104.08786 [cs], https://arxiv.org/abs/2104.08786.

[26] F. McSherry and K. Talwar. Mechanism design via differential privacy. pages 94–103, 2007.

[27] Microsoft Corporation. Differential privacy in azure machine learning. https://docs.microsoft.com/en-us/azure/machine-learning/concept-differential-privacy#differential-privacy-metrics, 2020. Accessed: 2025-05-15.

[28] I. Mironov. Rényi Differential Privacy. pages 263–275, 2017.

[29] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. pages 75–84, 2007.

[30] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C.L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. volume 35, 2022.

[31] M.J. Page, J.E. McKenzie, P.M. Bossuyt, I. Boutron, T.C. Hoffmann, C.D. Mulrow, L. Shamseer, J.M. Tetzlaff, E.A. Akl, S.E. Brennan, R. Chou, J. Glanville, J.M. Grimshaw, A. Hróbjartsson, M.M. Lalu, T. Li, E.W. Loder, E. Mayo-Wilson, S. McDonald, L.A. McGuinness, L.A. Stewart, J. Thomas, A.C. Tricco, V.A. Welch, P. Whiting, and D. Moher. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *Systematic Reviews*, 10(1), 2021.

[32] T. Shin, Y. Razeghi, R.L. Logan, IV, E. Wallace, and S. Singh. AUTOPROMPT: Eliciting knowledge from language models with automatically generated prompts. pages 4222–4235, 2020.

[33] A. Sordoni, X. Yuan, M.-A. Côté, M. Pereira, A. Trischler, Z. Xiao, A. Hosseini, F. Niedtner, and N. Le Roux. Joint Prompt Optimization of Stacked LLMs using Variational Inference. volume 36, 2023.

[34] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. Canton Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. Llama 2: Open foundation and fine-tuned chat models, July 2023. arXiv:2307.09288 [cs], https://arxiv.org/abs/2307.09288.

[35] Transaction Processing Performance Council. Tpc-h benchmark specification. http://www.tpc.org/tpch/, 2008. Accessed: 2025-05-17.

[36] A. Tsanas, M.A. Little, P.E. McSharry, and L.O. Ramig. Accurate telemonitoring of parkinsons disease progression by noninvasive speech tests. *IEEE Transactions on Biomedical Engineering*, 57(4):884–893, 2010.

[37] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S.R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. 2019.

[38] R.J. Wilson, C.Y. Zhang, W. Lam, D. Desfontaines, D. Simmons-Marengo, and B. Gipson. Differentially private sql with bounded user contribution, November 2019. arXiv:1909.01917 [cs], `https://arxiv.org/abs/1909.01917`.

[39] C. Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. 2014.

[40] D. Yu, S. Naik, A. Backurs, S. Gopi, H.A. Inan, G. Kamath, J. Kulkarni, Y.T. Lee, A. Manoel, L. Wutschitz, S. Yekhanin, and H. Zhang. DIFFERENTIALLY PRIVATE FINE-TUNING OF LANGUAGE MODELS. 2022.

[41] S. Zhang, A. Hagermalm, S. Slavnic, E.M. Schiller, and M. Almgren. Evaluation of Open-Source Tools for Differential Privacy. *Sensors*, 23(14), 2023.