Open LLMs are Necessary for Current Private Adaptations and Outperform their Closed Alternatives

Vincent Hanke, Tom Blanchard, Franziska Boenisch,
Iyiola Emmanuel Olatunji, Michael Backes, Adam Dziedzic*
CISPA Helmholtz Center for Information Security

Abstract

While open Large Language Models (LLMs) have made significant progress, they still fall short of matching the performance of their closed, proprietary counterparts, making the latter attractive even for the use on highly *private* data. Recently, various new methods have been proposed to adapt closed LLMs to private data without leaking private information to third parties and/or the LLM provider. In this work, we analyze the privacy protection and performance of the four most recent methods for private adaptation of closed LLMs. By examining their threat models and thoroughly comparing their performance under different privacy levels according to differential privacy (DP), various LLM architectures, and multiple datasets for classification and generation tasks, we find that: (1) all the methods leak query data, i.e., the (potentially sensitive) user data that is queried at inference time, to the LLM provider, (2) three out of four methods also leak large fractions of private training data to the LLM provider while the method that protects private data requires a local open LLM, (3) all the methods exhibit lower performance compared to three private gradient-based adaptation methods for *local open LLMs*, and (4) the private adaptation methods for closed LLMs incur higher monetary training and query costs than running the alternative methods on local open LLMs. This yields the conclusion that, to achieve truly *privacy-preserving LLM adaptations* that yield high performance and more privacy at lower costs, taking into account current methods and models, one should use open LLMs.

1 Introduction

Recently, there has been the trend of releasing open Large Language Models (LLMs), such as LLama [21, 60], Vicuna [11], or Mistral [27] as an alternative to their proprietary closed counterparts, such as GPT from OpenAI [2], Claude from Anthropic [4], or Gemini from Google [57]. Despite the significant progress in improving open LLMs, they are still outperformed in multiple tasks by closed LLMs [12], making the latter attractive even for learning tasks from highly *private* data.

Since it was shown that private data can leak from the adaptations of LLMs [16, 17], in the last few months alone, an array of new methods for privacy-preserving adaptation of closed LLMs has been proposed by the machine learning community at multiple conferences (NeurIPS'23 [16] and ICLR'24 [25, 56, 63]). Given the lack of access to the closed LLMs parameters—which renders parameter-tuning based adaptations infeasible—they all rely on the generation of privacy-preserving discrete prompts. We detail their operational setup in Figure 1 (left).

In this work, we ask the simple yet impactful question of whether these efforts actually lead into the right direction towards the goal of achieving *truly privacy-preserving LLM adaptations*. Therefore, we thoroughly analyze the proposed methods both conceptually and empirically and compare them to

---

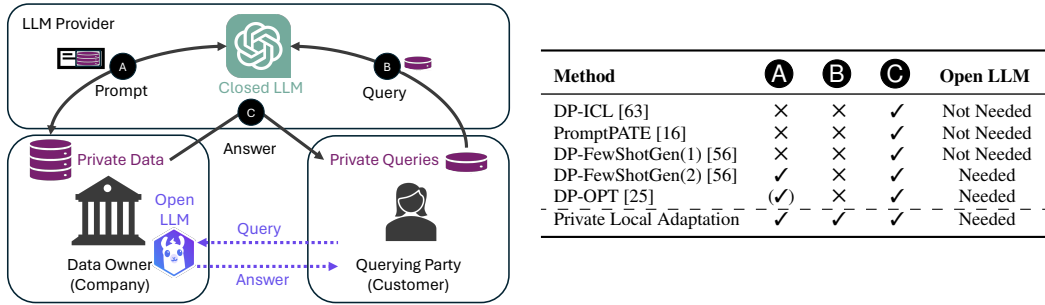*Correspondance to adam.dziedzic@cispa.de

Figure 1: **Setup for Privacy Protection with Open vs Closed LLMs**. The three parties involved are (1) an LLM provider who hosts the proprietary LLM, (2) a data owner, such as a company that owns private data, for example, of their customers' previous transactions, and (3) a querying party, *i.e.,* a customer of the company who wants to perform a new private transaction. There are three steps where privacy leaks: **A** During the creation of the discrete prompt, the data owner's private data leaks to the LLM provider. **B** The private query of the querying party leaks to the LLM provider. **C** Private information from the data owner leaks to the querying party through the returned answers of the prompted LLM [17]. Prior methods for closed LLMs [16, 56, 63] only provide protection against **C** . None of them protects against **B** . To prevent leakage through **A** , they require access to a (powerful) local open LLM. As an alternative (dashed purple lines), the data owner could privately adapt the open LLM locally and let the querying party interact with this LLM, protecting against **A** , **B** , **C** .

The table in Figure 1:

Method	A	B	C	Open LLM
DP-ICL [63]	✗	✗	✓	Not Needed
PromptPATE [16]	✗	✗	✓	Not Needed
DP-FewShotGen(1) [56]	✗	✗	✓	Not Needed
DP-FewShotGen(2) [56]	✓	✗	✓	Needed
DP-OPT [25]	(✓)	✗	✓	Needed
Private Local Adaptation	✓	✓	✓	Needed

alternatives that rely on privately adapting *open local LLMs*. In particular, we study each approach's threat space, assumptions, and methodological limitations and perform extensive experiments using ten state-of-the-art open and closed LLMs of various sizes, including Vicuna, Llama 3, Open LLaMa, BERT, RoBERTa, the Pythia suite of models, Claude, two versions of GPT3 (Babbage and Davinci), and GPT4 Turbo —applied to multiple datasets both for classification and generation tasks. Our analyses cover the axes of privacy protection, performance in terms of privacy-utility trade-offs, and monetary costs for training and queries.

Our results provide the following insights: (1) All current methods for adapting closed LLMs leak private query data (intended for the data owner) at inference time to the LLM provider. (2) Three out of the four methods studied also leak large fractions of the private training data to the LLM provider. The approaches that do not, require an additional locally deployed open LLM for prompt engineering. (3) All methods for closed LLMs yield lower final downstream performance than privacy-preserving local adaptations on open LLMs—even when the local methods rely on significantly smaller LLMs than their closed counterparts. (4) The training and query costs of the private adaptations of closed LLMs (API access costs imposed by the LLM provider) are significantly higher than the costs for private open LLM adaptations (estimated as the costs of training and querying on cloud-based hardware). We provide a condensed summary of our results in Figure 1 (right Table above), and Table 1.

Overall, our results indicate that, from the perspective of effective privacy-preservation, current adaptations of open LLMs are strictly preferable over their closed LLM alternatives, since they are more private, more performant, and less expensive. Going beyond the concrete existing methods studied [16, 25, 56, 63], we then analyze the reasons behind the underwhelming results of privacy-preserving closed LLM adaptations and discuss potential directions for improvements.

On the way, to further strengthen private adaptations for open LLMs, we demonstrate how to locally apply privacy-preserving prompt-based methods to train generation tasks with high-performance— claimed impossible by prior work [35]. In particular, we show for the first time that private prompt tuning for text generation tasks PromptDPSGDGen can achieve comparable performance to private (full) fine-tuning and private low-rank adaptations (LoRA). Additionally, we demonstrate that ensemble-based few-shot prompts PromptPATEGen can privately generate high-quality text at a low privacy cost.

In summary, we make the following contributions:

Table 1: **Comparison of privacy protection, performance, and cost between private adaptations for closed vs open LLMs.** We select the top-performing adaptations. For closed LLMs, we use DP-ICL [63] and leverage PrivateLoRA [64] on open LLMs for both tasks. We consider sentiment classification on SST2 and the dialog summarization on SAMSum. The training data is denoted by $\mathcal{D}_T$ and the test queries by Q. *Reveals* represents which data are exposed to the LLM provider. The methods were trained with DP guarantees: $\varepsilon = 8$ and $\delta = 1/N$, where N is the number of examples in $\mathcal{D}_T$. We report the *Performance* (higher is better) on test data (where *Acc* denotes the classification accuracy). The cost (in $) is computed separately for training (Train) and for answering 10k test queries (Query). Note, the (estimated) number of parameters for closed LLMs is 1.76T for GPT4 Turbo and 175B for GPT3 Davinci, while Llama3 has only 8B and BART-Large is significantly smaller with 355M parameters. The adaptation of the open LLMs is more expensive on SST2 than on SAMSum due to the larger training data size for SST2 and a larger model. DP-ICL's query cost is high due to the usage of an ensemble of 100 prompts to answer each query. *In summary, open local LLM adaptations are more private, more performant, and less expensive.*

Adaptation	LLM Type	Model	Task	Reveals	Performance↑	Train($)	Query($)
DP-ICL [63]	Closed	GPT4 Turbo	SST2	$\mathcal{D}_T+Q$	Acc=$95.9_{\pm 0.1}$%	0	138.00
PrivateLoRA [64]	Open	Llama3-8B(instruct)	SST2	*None*	Acc=$96.0_{\pm 0.1}$%	27.60	0.78
DP-ICL [63]	Closed	GPT3 Davinci	SAMSum	$\mathcal{D}_T+Q$	RougeL=$31.8_{\pm 0.3}$	0	665.91
PrivateLoRA [64]	Open	BART-Large	SAMSum	*None*	RougeL=$39.1_{\pm 0.2}$	3.63	0.80

1. We perform a thorough conceptual and experimental study on existing privacy-preserving closed and open LLM adaptations, analyzing their threat space, assumptions, and achieved results.

2. Our extensive experiments on various open and closed LLMs and on multiple classification and generation tasks show that the local (gradient-based) adaptations outperform their current closed (discrete prompt-based) counterparts in terms of privacy, performance, and cost efficiency.

3. We propose differentially private prompts for text generation tasks that, for the first time, reach performance comparable to private LoRA or private fine-tuning.

2 Background and Related Work

Differential Privacy. Differential Privacy (DP) [19] is a mathematical framework that provides privacy guarantees by implementing the intuition that an algorithm $\mathcal{A} : I \rightarrow R$, executed on two neighboring datasets D, D' that differ in only one data point (we adopted the definition of *neighboring* based on addition/removal. [35, 45]), will yield approximately the same output, *i.e.,* $\Pr[\mathcal{A}(D) \in R] \leq e^{\epsilon} \cdot \Pr[\mathcal{A}(D') \in R] + \delta$. While ε specifies by how much the output can differ, δ specifies the probability of failure. There are two prevalent DP algorithms for training machine learning models. The first one is the differential private stochastic gradient descent algorithm (**DPSGD**) [3] where the impact of each private training data point is limited during training through gradient clipping, and privacy guarantees are integrated through the addition of calibrated amounts of stochastic noise. The second algorithm is the private aggregation of teacher ensembles (**PATE**) [50] where first, an ensemble of teacher models is trained on disjoint subsets of the private data, and then a noisy knowledge distillation is performed to a student model using public data. Another general mechanism for implementing DP is the exponential mechanism (**EM**) [44]. The EM selects an output r from a set of possible outputs based on a scoring function $q(D, r)$ that measures the quality of r for dataset D. Let Δq be the sensitivity of the scoring function. The EM chooses r with probability proportional to $\exp\left(\frac{\epsilon q(D,r)}{2\Delta q}\right)$.

LLM Adaptations. LLMs are pre-trained on large amounts of public data and then adapted to downstream tasks using private data. We divide existing methods for private LLM adaptations into *private tuning* methods that rely on access to the LLM gradients, and *private in-context learning* (ICL) which requires only API (black-box) access to the LLM. While private tuning is only applicable to open LLMs, private ICL can, in principle, be applied to both open and closed LLMs. We note that all private LLM adaptations rely in their core on the three DP algorithms introduced above and summarize existing methods, their setup, and their assumptions in Table 2.

Private Tuning for Open LLMs. There exist three main ways for private tuning. **1) Prompt-based adaptations** adds a small number of parameters (usually <1% of the total number of parameters)

Table 2: **Comparison of properties between private LLM adaptations.** The in-context learning (ICL) optimizes instructions and shots (demonstrations). Many privacy techniques include the ones designed for multi-label PATE (denoted as MLPATE) [67], exponential mechanism (EM) [44], joint exponential mechanism (JEM) [22], Gaussian Mechanism (GM), Report-Noisy-Max Mechanism (RNM), Propose-Test-Release (PTR) [20], sample-and-aggregate (SAA) [48], Limited Domain Algorithm (LDA) [18].

Property Adaptation	Privacy Algorithms	Optimization Strategy	Privatize	Inference Type	Require
PromptDPSGD [16]	DPSGD	Gradient-based	Soft Prompt/Prefix	Multi-task	Open LLM
PrivateLoRA [64]	DPSGD	Gradient-based	Added parameters	Single-task	Open LLM
DP-FineTune [35]	DPSGD	Gradient-based	all LLM parameters	Single-task	Open LLM
DP-ICL [63]	RNM,GM,JEM,PTR,MLPATE	ICL	Answers	Limited Queries	None
PromptPATE [16]	PATE	ICL	Shots	Multi-task	Public Data
DP-FewShotGen [56]	GM,RNM,EM	ICL	Shots	Multi-task	Public Labels,Open LLM
DP-OPT [25]	SAA,LDA	ICL	Instructions+Shots	Multi-task	Validation Data,Open LLM

only in the model input space, either on the level of token embeddings (soft prompts [39, 40]), or also to every LLM layer (prefix-tuning [30, 33]). Duan et al. [16] presented **PromptDPSGD**, which adapts the DPSGD algorithm to soft prompts. The main advantage of prompt-based adaptations is that they enable multi-task batch processing, *i.e.,* many soft prompts for different users and tasks can be processed in the same mini-batch during LLM training or inference. **2) Parameter efficient fine-tuning-based adaptations** such as LoRA [26] add a relatively small number of parameters (<10% of total number of parameters) within the model, usually in each block of a transformer architecture [61]. These added parameters are then tuned while the pre-trained original parameters remain frozen. **PrivateLoRA** [64] extends LoRA with DP guarantees by building on the DPSGD algorithm. **3) Full fine-tuning-based adaptations** either fine-tune the whole model or only a few last layers. The **DP-FineTune** [35], again based on the DPSGD algorithm, shows that full fine-tuning with DP optimization can provide strong privacy guarantees and good performance. The general trend, when choosing an adequate method, suggests that the more difficult the task, the higher the number of adaptation parameters required [16]. Thus, for simple downstream tasks, PromptDPSGD [16] is sufficient, while DP-LoRA [64] is recommended for medium-difficulty tasks, and the full fine-tuning [35] for complex tasks.

Private ICL for Closed LLMs. Recently, many new methods were proposed for private in-context learning with closed LLMs. All of them leverage discrete (hard) prompts and rely on a voting mechanism for privacy protection, similar to PATE [49, 50] and CaPC [13]. We divide the existing methods into the following four categories: **(1) Private Question Answering:** The work on **DP-ICL** [63] proposed to answer queries based on the private dataset. Following the PATE setup, the private data is divided into non-overlapping partitions and then each partition is prepended with an instruction to form a private teacher prompt. The prompts form an ensemble of private teachers (prompted LLMs). Since DP-ICL does not implement the idea of a student model from PATE, all the teachers (usually 100) are required to answer each query, rendering the method expensive when executed on a closed LLM. Moreover, each query incurs additional privacy cost, such that the method can answer only a limited number of queries for a given privacy budget. **(2) Private Student Prompt: PromptPATE** [16] tackles the problem of the high costs and the limited number of answered queries in DP-ICL by creating a student prompt. PromptPATE uses an ensemble of teacher prompts (usually around 200) to label public data. Then it selects the most performant shots for the student prompt from these newly labeled examples. **(3) Private Prompt Generation: DP-FewShotGen** [56] is similar to PromptPATE but eschews the assumption about the public data for labeling and, in turn, starting from a public label, generates each output token privately to obtain a private shot. **(4) Private Prompt Engineering:** Finally, **DP-OPT** [25] privatizes prompt engineering based on the Deep Language Network (DLN) method [54]. While DP-ICL, PromptPATE, and DP-FewShotGen assume a generic instruction and emphasize the protection of the direct leakage from the shots only, DP-OPT [25] proposed to privately generate shots and instructions since either can leak information about the private training set. To overcome the problem that PATE-based approaches face with large output spaces (here equal to the vocabulary size of around 50k), DP-ICL [63] and DP-OPT [25] incorporate the EM and its improved versions [18, 22, 67] to privately release a token with the maximum count based on the voting from teacher prompts.

3 Prompt-based Private Adaptations for Text Generation

While PromptDPSGD and PromptPATE [16] were designed for classification tasks only, we further extend them to text generation tasks. Having prompt-based generation holds the advantage that, in contrast to fine-tuning based approaches, they support mixed-task inference [30, 33, 37], *i.e.,* they require one frozen model for multiple tasks rather than a separate model copy for each of them. This reduces storage and offers greater flexibility and efficiency.

PromptDPSGDGen. We observe that an adequate choice of hyperparameters is sufficient for adjusting PromptDPSGD [16] to generation tasks. This is in line with prior work highlighting that the challenge of prompt tuning is that it requires experimenting with various hyperparameter choices to achieve good performance [37]. In particular, we observe that increasing the number of parameters in the soft prompt from 0.1% of the total LLM parameters, as done for classification [16], to 10% of total model parameters, by enabling prefix projection, yields a significant increase in generation performance. Additionally, we observe the need for an increased learning rate, compared to other tuning methods, to generate more precise outputs. Otherwise, the hyperparameters are dependent on the data the model is trained on.

PromptPATEGen. Adjusting PromptPATE [16] to generation tasks (where more than one output token is generated) is challenging due to 1) the large output space (equivalent to the number of tokens in the vocabulary) and 2) the privacy costs incurred by generating multiple tokens through the teacher ensemble. To overcome this challenge and support generation tasks with an unlimited number of queries, we extended PromptPATE by combining the training of the student prompt from [16] with the privacy techniques used in [63] and call the result PromptPATEGen. In particular, PromptPATEGen uses the private generation in DP-ICL to obtain longer output sequences for some public data inputs. The outputs sequences can then be treated as a "label" for the public data and can be deployed as a form of student prompt, just like in PromptPATE [16].

4 Comparing Open and Closed LLM Adaptations

We perform a thorough conceptual and empirical study to compare the adaptation of both open LLMs with private tuning (PromptDPSGD [16], PrivateLoRA [64], and DP-FineTune [35]) and closed LLMs with private ICL (DP-ICL [63], PromptPATE [16], DP-FewShotGen [56], and DP-OPT [25]). Our comparison spans the axes of privacy protection, performance, and cost.

4.1 Comparing Privacy Protection

All the considered methods offer privacy guarantees according to DP. Thereby, they ensure that the final prompted LLM's predictions will not leak more than the specified tolerated privacy budget ε to any party who queries the LLM or gets access to the final private prompt. Yet, the threat model of multiple private ICL methods for closed LLMs does not include providing privacy against the LLM provider. Those methods that do might still occasionally experience leakage. We analyze the result of this lack of consideration for the goal of truly privacy-preserving LLM adaptations. In our analysis, we distinguish between the leakage of private training data and the leakage of test data queried at inference time, which might also be sensitive.

Private Training Data. PromptPATE [16], DP-ICL [63], and DP-FewShotGen [56] (without using an open LLM) disclose (large parts of) their private training set to the LLM provider in the form of shots in their teacher prompts and their engineering. This leakage is inherent in their design. To avoid such leakage, DP-OPT [25] tunes the prompt locally with DP guarantees and then exposes it to the LLM provider. Thereby, the data that the prompt was generated from is protected towards the LLM provider with the DP guarantees that also protect against leakage to a querying party. While the experimental evaluation in [25] suggests that at higher ε, the locally generated DP prompts might still contain generated data close to the private training data, this is a step towards the right direction. However, to generate the private prompt, DP-OPT [25] requires a powerful open LLM deployed locally. Looking at Figure 1, it becomes obvious that any private tuning method executed on that open LLM would, conceptually, improve privacy protection since the LLM provider would neither be involved in the adaptation nor in the use of the adapted LLM, yielding absolute privacy against them.

5

Private Query Data. DP does not aim at protecting query data. Hence, none of the studied private ICL methods attempt to protect that data against the LLM provider. While the protection of query data is often considered as an orthogonal research direction, we note that all the private tuning-based adaptations of the open local LLMs do naturally prevent leakage of the query data to the LLM provider. This is because the querying party directly interacts with the data owner (see Figure 1)—making the use of open models inherently more suited for truly privacy-preserving application than relying on closed models.

4.2 Comparing Performance

We look at privacy-utility trade-offs to compare the performance of private tuning on open LLMs vs. private ICL on their closed counterparts. We depart from analyzing the trade-offs and required assumptions conceptually and then present our thorough experimental evaluation.

Private Tuning Outperforms Private ICL Conceptually. Previous work [37] has shown for the non-private settings that gradient based tuning methods (used for open LLMs) offer better accuracy and significantly lower computational costs than ICL (used for closed LLMs) since the adaptations can leverage the internal behavior of the LLM. This benefit holds also in the privacy regime. Moreover, the tuning based methods do not make additional assumptions, such as the availability of public data (required by PATE-based methods, such as PromptPATE [16]), making them inherently more practical.

Private Tuning outperforms Private ICL Experimentally. To assess the performance of private tuning vs. private ICL, we perform extensive experimental evaluation. We use various LLM architectures and multiple datasets for classification and text generation tasks.

4.2.1 Experimental Setup

Text Classification. We follow the setup from [25] and use four datasets for the evaluation: SST2 from the GLUE benchmark [62], Trec [34], Mpqa [42] and Disaster [5]. SST2 and Mpqa are two-class sentiment analysis datasets. SST2 includes 67.3k training samples and 872 test samples, while Mpqa contains 8.6k training samples and 2k test samples. Trec is a six-class question-type classification dataset with 5.4k training samples and 500 test samples. Finally, the Disaster dataset involves determining whether a sentence is relevant to a disaster scenario or not and includes 4.4k training and 1000 test samples.

Text Generation. We use three different datasets: SAMSum, a dialog summarization [23] (14732 train, 818 val, and 819 test samples), PFL-DocVQA, question answering [59] (85k train and 10k test samples), and MIT Movies trivia10k13, movie extraction on directors (MIT-D with 1561 train and 415 test samples) and genre (MIT-G with 2953 train and 780 test samples) [38].

Closed Models. We follow the setup and choice of models originally proposed in the respective previous papers to evaluate the four private ICL methods for closed LLMs [16, 25, 56, 63]. The GPT3-Babbage and GPT3-Davinci models cited in [56, 63] were discontinued in early 2024[2] and replaced by their second versions (babbage-002 and davinci-002). Therefore, we use the newer versions here. The (estimated) number of parameters for the closed models is: 1.3B for GPT3 Babbage, 175B for GPT3 Davinci, 1.76T for GPT4 Turbo, and 200B for Claude 2.1.

Open Models. We consider various open LLMs with differing pre-training sets and numbers of parameters to simulate the choices a data owner can make for their local LLM. We select the following models: Pythia [6], OpenLLaMA [21], Vicuna [11], Mixtral [28], Bart [31], and RoBERTA [41], whose sizes vary from 160M to 45B parameters.

4.2.2 Performance of Private Adaptations for Classification

We show that the private adaptations on local open LLMs outperform the private methods for closed LLMs for classification tasks. In Table 3, we analyze the performance differences. We follow the evaluation in [25] (Table 2) and average the accuracy across the tasks (denoted as Average). Our analysis follows the standard practice and sets the privacy budget as $\varepsilon = 8$ and $\delta = 1/|D|$ where $|D|$ is the training size [16, 25]. Among the methods for closed LLMs, DP-OPT was tested on

---

[2]https://platform.openai.com/docs/deprecations

Table 3: **Private local adaptations on open LLMs outperform their closed alternatives for classification tasks.** The default privacy budget is set to $\varepsilon = 8$, except for PromptPATE [16], where the performance plateaus after $\varepsilon = 0.3$. The best result for a given task is bolded, and the 2nd best is underlined. **T($)** is training cost while **Q($)** is query cost for 10k queries (SST2), **All($)** is total cost.

Method	LLM Type	Model	SST2	Trec	Mpqa	Disaster	Average	T($)	Q($)	All($)
0-shot ($\varepsilon = 0$) [25]	Closed	GPT3 Davinci	$92.4_{\pm0.0}$	$51.8_{\pm0.2}$	$84.5_{\pm0.1}$	$76.4_{\pm0.2}$	76.3	0	6.00	6.00
DP-OPT (original) [25]	Closed	GPT3 Davinci	$92.2_{\pm0.8}$	$68.7_{\pm6.5}$	$85.8_{\pm0.7}$	$78.9_{\pm0.3}$	81.4	2.10	6.00	8.10
ICL ($\varepsilon = \infty$) [25]	*Closed*	*GPT3 Davinci*	$94.7_{\pm0.4}$	$79.1_{\pm0.5}$	$88.8_{\pm0.1}$	$69.0_{\pm5.9}$	82.9	0	6.00	6.00
PromptPATE [16]($\varepsilon = \infty$)	Closed	GPT3 Babbage	93.8	58.7	83.0	64.3	75.0	8.66	1.72	10.38
PromptPATE [16]($\varepsilon < 0.3$)	Closed	GPT3 Babbage	$88.8_{\pm2.3}$	$52.8_{\pm1.5}$	$79.0_{\pm0.5}$	$58.0_{\pm0.5}$	69.6	9.72	1.72	11.44
PromptPATE [16]($\varepsilon < 0.3$)	Closed	Claude 2.1	$95.7_{\pm1.4}$	$79.3_{\pm1.2}$	$\mathbf{92.1_{\pm0.6}}$	$71.0_{\pm0.8}$	84.5	48.24	5.36	53.6
DP-FewShotGen(1) [56]	Closed	GPT3 Babbage	$72.8_{\pm7.7}$	$51.3_{\pm5.8}$	$73.4_{\pm8.5}$	$59.2_{\pm2.5}$	64.2	0.86	1.10	1.96
DP-ICL [63]	Closed	GPT3 Babbage	$92.8_{\pm0.9}$	$26.3_{\pm5.6}$	$80.6_{\pm0.9}$	$50.6_{\pm1.1}$	62.6	0	17.2	17.2
DP-ICL [63]	Closed	GPT4 Turbo	$95.9_{\pm0.1}$	$16.2_{\pm1.7}$	$90.4_{\pm0.1}$	$70.3_{\pm0.4}$	68.2	0	138.00	138.00
PromptDPSGD [16]	Open	RoBERTA Large	$92.3_{\pm0.5}$	$54.5_{\pm2.5}$	$50.0_{\pm0.0}$	$77.8_{\pm0.6}$	68.6	7.59	0.40	7.99
DP-FineTune [35]	Open	RoBERTA Large	$93.5_{\pm0.3}$	$93.7_{\pm0.8}$	$88.2_{\pm0.4}$	$\mathbf{82.2_{\pm0.3}}$	89.4	5.75	0.40	6.15
PrivateLoRA [64]	Open	RoBERTA Large	$93.6_{\pm0.3}$	$93.9_{\pm0.6}$	$87.7_{\pm0.8}$	$81.8_{\pm0.2}$	89.3	3.45	0.40	3.85
PrivateLoRA [64]	Open	Vicuna 7B	$94.8_{\pm0.5}$	$\mathbf{97.3_{\pm0.1}}$	$87.8_{\pm0.5}$	$81.3_{\pm0.5}$	**90.3**	13.80	0.78	14.58
DP-OPT (local) [25]	Open	Vicuna 7B	$89.5_{\pm2.6}$	$65.3_{\pm4.3}$	$80.7_{\pm3.3}$	$65.6_{\pm0.3}$	75.3	2.10	0.78	2.88
PrivateLoRA [64]	Open	Pythia 6.9B	$92.2_{\pm0.5}$	$96.3_{\pm0.8}$	$87.2_{\pm0.3}$	$82.1_{\pm0.2}$	89.4	13.80	0.78	14.58
PrivateLoRA [64]	Open	Pythia 160M	$80.4_{\pm0.7}$	$82.5_{\pm3.2}$	$77.9_{\pm0.3}$	$\underline{73.6}_{\pm0.2}$	78.6	1.60	0.50	2.1
PrivateLoRA [64]	Open	Llama3-8B(Instruct)	$\mathbf{96.0_{\pm0.1}}$	$\underline{96.8_{\pm0.2}}$	$87.3_{\pm0.2}$	$80.8_{\pm0.1}$	$\underline{90.2}$	27.60	0.78	28.38

the strongest Davinci model (with 175B parameters) from the GPT3 family. Across all the tasks, DP-OPT is outperformed by both DP-FineTune and PrivateLoRA by a large margin (even >26% absolute on Trec), even though DP-FineTune and PrivateLoRA were trained on RoBERTa Large with only 355M parameters (500X fewer than for GPT3 Davinci). Furthermore, we show that PrivateLoRA outperforms DP-OPT even when using Pythia-6.9B, which guarantees that the open LLM for PrivateLoRA was not pre-trained on any of the downstream datasets. For a fair comparison, we also train PrivateLoRA on Vicuna 7B, which was used in DP-OPT as the local model to find the transferable prompts and show that PrivateLoRA is also significantly better than DP-OPT applied either directly to Vicuna 7B or when run on GPT3 Davinci. This suggests that the data owners, rather than using their local LLM to tune prompts for DP-OPT, should privately tune it with PrivateLoRA (in this case on RoBERTA Large) since it yields stronger performance and privacy at a lower cost.

For PromptPATE, the performance plateaus after around $\varepsilon = 0.3$, since it creates a public prompt using only a few shots, and the selection of the demonstrations from a large pool of publicly labeled examples has a negligible gain on the final performance. In the limit, we also show that PromptPATE even with an infinite privacy budget ($\varepsilon = \infty$) for GPT3 Babbage (with 1.3B parameters) performs worse than PrivateLoRA or DP-FineTune on RoBERTA Large (3.6X fewer parameters). In the same setup of models, PrivateLoRA and DP-FineTune on RoBERTA Large also outperform DP-ICL tested on GPT3 Babbage on all tasks. Additionally, PrivateLoRA adapted on Pythia-160M (with even fewer parameters) performs much better than DP-FewShotGen on GPT3-Babbage (8X more parameters).

We also run DP-ICL with GPT4 Turbo. The resulting accuracies are high for sentiment classification with SST-2 and Mpqa. However, it has the lowest accuracy on Trec (with 6 classes), caused by a small number of output probability tokens released for a query (only 20 vs 100 for GPT3, which might not contain the correct class label token) while being the most expensive option. Similar trends are observed for PromptPATE on Claude, however, it has more consistent performance and emerges as the most performant closed model on the tested tasks (while being the 2nd most expensive one). In contrast, Private LoRA with Vicuna 7B performs the best on Trec and on *average*. It is the best of all tested adaptations while incurring around 3.7 and 9.5 times lower costs than Claude and GPT4 Turbo, respectively. In general, the open models have the highest average performance at a much lower cost.

We further analyze the privacy-utility trade-off for classification tasks across different privacy budgets ($\varepsilon \in [0, 8)$) in Figure 2. We show that even under tight privacy constraints ($\varepsilon < 1.0$), the privacy-preserving adaptation for open LLMs performs significantly better than the one for closed LLMs. Specifically, we analyze the differences between PrivateLoRA for open LLMs vs PromptPATE for closed LLMs. The performance for PromptPATE plateaus after around $\varepsilon = 0.3$ and only for one out of four datasets, namely for MPQA, we observe that the crossover point between PromptPATE and PrivateLoRA (PromptPATE performs better than PrivateLoRA until $\varepsilon = 0.6$). For the smallest $\varepsilon = 0.1$ values that we analyzed, the performance of PrivateLoRA is better by 0.6% on SST2, by

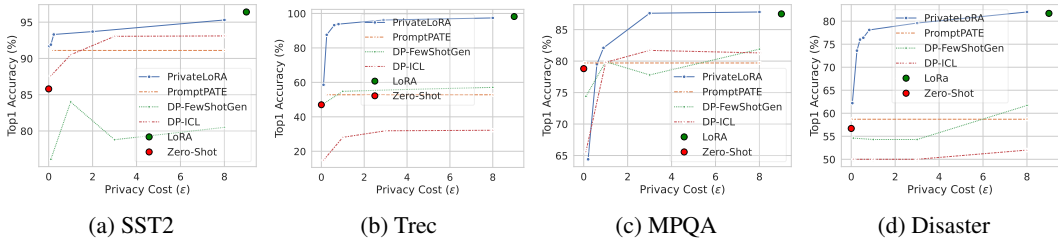(a) SST2   (b) Trec   (c) MPQA   (d) Disaster

Figure 2: **Privacy-utility trade-off for classifications tasks.** We use PrivateLoRA to adapt Vicuna-7b to the downstream tasks, PromptPATE, DP-ICL, and DP-FewShotGen with GPT3 Babbage. We analyze the privacy costs ε in the range $[0, 8]$ (see corresponding Figure 3 for text generation tasks).

Table 4: **Evaluation on Dialog Summarization with SAMSum** for $\varepsilon = 8$. **T($)** is training cost while **Q($)** is query cost for 10k queries, **All($)** is total cost.

Method	LLM Type	Model	Rouge-1	Rouge-2	Rouge-L	T($)	Q($)	All($)
DP-ICL [63]	Closed	GPT3 Davinci	$41.2_{\pm0.6}$	$16.3_{\pm0.4}$	$31.8_{\pm0.3}$	0	665.91	665.91
DP-ICL [63]	Closed	GPT3.5 Turbo	$42.6_{\pm0.2}$	$18.9_{\pm0.3}$	$33.8_{\pm0.5}$	0	449.16	449.16
DP-ICL [63]	Closed	GPT4 Turbo	$41.8_{\pm0.2}$	$17.3_{\pm0.3}$	$33.4_{\pm0.2}$	0	3419.42	3419.42
PromptPATEGen	Open	Vicuna 7B	$41.3_{\pm0.3}$	$18.0_{\pm0.4}$	$32.8_{\pm0.2}$	3.29	2.74	6.03
PromptPATEGen	Open	OpenLLaMA 13B	$43.4_{\pm0.3}$	$19.7_{\pm0.5}$	$34.2_{\pm0.4}$	18.63	0.80	19.43
PromptDPSGDGen	Open	BART-Large	$46.1_{\pm0.4}$	$21.3_{\pm0.1}$	$37.4_{\pm0.0}$	1.73	0.40	2.13
PrivateLoRA [64]	Open	BART-Large	$48.8_{\pm0.6}$	$23.5_{\pm0.5}$	$39.1_{\pm0.2}$	2.90	0.69	3.59
PrivateLoRA [64]	Open	Pythia 410M	$40.4_{\pm0.1}$	$16.6_{\pm0.3}$	$33.0_{\pm0.4}$	3.45	1.34	4.79
PromptDPSGDGen	Open	Pythia 1B	$41.2_{\pm0.2}$	$17.8_{\pm0.1}$	$33.7_{\pm0.1}$	4.83	0.95	5.78
DP-FineTune [35]	Open	Pythia 1B	$42.5_{\pm0.7}$	$18.4_{\pm0.3}$	$33.9_{\pm0.3}$	9.84	1.08	10.92
PrivateLoRA [64]	Open	Pythia 1B	$42.3_{\pm0.6}$	$18.4_{\pm0.7}$	$34.7_{\pm0.5}$	4.24	1.00	5.24
PrivateLoRA [64]	Open	Pythia 6.9B	$45.6_{\pm0.3}$	$21.4_{\pm0.3}$	$37.4_{\pm0.5}$	10.18	6.57	16.75
PrivateLoRA [64]	Open	Vicuna 7B	$48.6_{\pm3.5}$	$24.8_{\pm2.6}$	$40.2_{\pm3.4}$	11.28	6.19	17.47
PrivateLoRA [64]	Open	OpenLLaMA 13B	$48.5_{\pm1.1}$	$24.2_{\pm0.8}$	$40.1_{\pm0.9}$	19.46	8.05	27.51
PrivateLoRA [64]	Open	Mixtral 8x7B	$\mathbf{52.8}_{\pm0.4}$	$\mathbf{29.6}_{\pm0.2}$	$\mathbf{44.7}_{\pm0.2}$	57.96	9.99	67.95

Table 5: **Evaluation on Question Answering with PFL-DocVQA** for $\varepsilon = 8$.

Method	LLM Type	Model	Rouge-1	BLEU	Levenshtein	T($)	Q($)	All($)
DP-ICL [63]	Open	OpenLLaMA 13B	$60.7_{\pm0.6}$	$23.9_{\pm0.5}$	$52.5_{\pm1.1}$	0	641.32	641.32
PromptPATEGen	Open	Vicuna 7B	$31.7_{\pm1.5}$	$26.6_{\pm0.7}$	$35.7_{\pm0.4}$	2.28	0.57	2.85
PromptDPSGDGen	Open	Pythia 1B	$57.3_{\pm0.9}$	$40.1_{\pm1.1}$	$66.8_{\pm0.7}$	37.26	0.96	38.22
DP-FineTune [35]	Open	Pythia 1B	$\mathbf{70.2}_{\pm0.2}$	$\mathbf{55.7}_{\pm0.3}$	$\mathbf{78.3}_{\pm0.3}$	137.06	1.32	138.38
PrivateLoRA [64]	Open	Pythia 1B	$64.2_{\pm0.7}$	$43.2_{\pm0.8}$	$73.4_{\pm1.3}$	44.16	1.28	45.44
PrivateLoRA [64]	Open	Pythia 6.9B	$64.4_{\pm0.1}$	$47.9_{\pm0.2}$	$73.3_{\pm0.2}$	293.25	5.80	299.05
PrivateLoRA [64]	Open	OpenLLaMA 13B	$63.1_{\pm1.1}$	$22.2_{\pm1.3}$	$70.7_{\pm2.1}$	358.80	9.02	367.82

Table 6: **Evaluation on information extraction with MIT-D and MIT-G** for $\varepsilon = 8$.

Method	LLM Type	Model	MIT-D	MIT-G	T($)	Q($)	All($)
DP-FewShotGen [56]	Closed	GPT3 Davinci	80.6	64.1	0.42	2.36	2.78
PromptPATEGen	Open	Vicuna 7B	$74.1_{\pm0.6}$	$41.7_{\pm1.6}$	0.52	0.73	1.25
PromptPATEGen	Open	OpenLLaMA 13B	$70.9_{\pm0.5}$	$33.4_{\pm1.3}$	3.11	0.80	3.91
PrivateLoRA [64]	Open	Pythia 410M	$74.3_{\pm8.3}$	$64.3_{\pm2.8}$	0.06	0.50	0.56
PromptDPSGDGen	Open	Pythia 1B	$89.8_{\pm0.3}$	$69.1_{\pm1.7}$	0.17	0.25	0.42
DP-FineTune [35]	Open	Pythia 1B	$92.2_{\pm1.1}$	$71.6_{\pm1.1}$	0.94	0.50	1.44
PrivateLoRA [64]	Open	Pythia 1B	$90.2_{\pm0.1}$	$68.8_{\pm0.8}$	0.08	0.31	0.39
PrivateLoRA [64]	Open	Vicuna 7B	$\mathbf{95.0}_{\pm0.2}$	$74.4_{\pm1.2}$	0.52	5.92	6.44
PrivateLoRA [64]	Open	OpenLLaMA 13B	$94.0_{\pm0.8}$	$\mathbf{76.4}_{\pm0.9}$	1.04	6.21	7.25
PrivateLoRA [64]	Open	Mixtral 8x7B	93.0	69.7	1.52	9.47	10.99

4.4% on Trec, and by 3.5% on Disaster. Overall, the private adaptations for open LLMs outperform the ones for closed LLMs in most privacy regimes.

4.2.3   Performance of Private Adaptations for Text Generation

The evaluation of the three text generation tasks demonstrates superior performance of private adaptations on open vs closed LLMs. We consider the privacy-preserving ICL methods of DP-ICL and DP-FewShotGen on closed LLMs, since only these methods were executed for generative tasks. For the SAMSum datasets in Table 4, the first three adaptations (including our PromptPATEGen) are based on few-shot in-context learning (using discrete prompts), while the remaining results are for the private gradient-based adaptations. For the discrete prompts, our PromptPATEGen runs on local open Vicuna 7B and outperforms other discrete prompt-based methods from closed LLMs. Our PromptDPSGDGen performs on par with the other private tuning method (PrivateLoRA) run on Pythia 1B. Note that only PromptDPSGDGen and ICL adaptations (PromptPATEGen and DP-ICL) support multi-task inference.

8

We additionally leverage BART-Large (with 355M parameters) [1] that was fine-tuned on the XSum summarization task [47] (which does not include SAMSum). This specialized open model outperforms other LLMs apart from Vicuna with 7B parameters, OpenLLaMA with 13B parameters, and Mixtral with 45B parameters. Crucially, PrivateLoRA on BART-Large outperforms DP-ICL run on GPT3 Davinci, despite using the model with around 500X fewer parameters. This further indicates that we can leverage a large selection of open models to solve a specific task at lower cost and with better privacy protection without resorting to general-purpose closed LLMs. We also use PrivateLoRA on larger models from different families (Vicuna 7B, OpenLLama 13B, and Mixtral 8x7B) and observe that its performance and cost steadily increase with more parameters.

The evaluation on PFL-DocVQA in Table 5 shows that PrivateLoRA on open LLMs outperforms DP-ICL (which was run also only on OpenLLaMA 13B in the original paper [63] due to the cost constraints). We also evaluate both MIT-D and MIT-G in Table 6 on the accuracy of predicted vs target labels following the metrics in DP-FewShotGen. The adaptations of open LLMs with privacy-preserving gradient-based methods outperform DP-FewShotGen on the significantly larger GPT3 Davinci, for example, on MIT-D by 13.4% and on MIT-G by 22.3% absolute, respectively by PrivateLoRA on OpenLLaMA 13B.Further analyses of privacy-utility trade-offs for text generation are presented in Figure 3 in the Appendix.

4.3 Comparing Costs

We compare the costs of obtaining a private predictor for a given downstream task using open vs closed LLMs. We use the wall clock time to capture the running time of methods for local open LLMs, which we then translate to the monetary cost that would be incurred if we ran the method on cloud-based hardware. For the adaptations of closed LLMs, we count the number of tokens used in the queries and obtained outputs from the APIs. The pricing from cloud providers and OpenAI forms the basis for the cost estimations, and we show the selected values in Table 22 in the Appendix. Further details on how the costs were calculated for each private ICL methods are presented in Appendix E. Based on the estimated costs in Tables 1,3,4,5, and 6, the privacy-preserving methods for open LLMs require much lower costs (and perform better) than for closed LLMs in the considered scenarios. The costs for classification tasks are relatively low, especially for closed LLMs, since the tasks are simple and the number of tokens (particularly for outputs) is small. However, the costs increase substantially for generation tasks, especially for the closed LLMs, where DP-ICL is around 150X more expensive than PrivateLoRA for dialog summarization. While larger models often incur higher costs, they do not necessarily imply higher performance. For example, smaller models like RoBERTA Large for classification or BART-Large for dialog summarization can obtain one of the highest performances at the lowest prices.

5 Discussion and Future Work

In summary, our results highlight that from the perspective of providing truly privacy-preservation adaptations, open LLMs are strictly preferable over closed LLMs, since their adaptations are more private, more performant, and more cost-effective. Going beyond the concrete existing methods studied in this work [16, 25, 56, 63], in the following, we analyze the general reasons behind the underwhelming results of privacy-preserving closed LLM adaptations.

Privacy Leakage. The enhanced privacy protection from adapting open LLMs is a major benefit: users' private training data and queries to adapted open LLMs are never revealed to third parties. On the contrary, the leakage of private query data to the LLM provider is to date an inherent problem with closed LLMs, since no methods to provide formal guarantees for the query data are currently known. Potential solutions might involve private inference for LLMs, where a model performs inference on encrypted queries, however, it is still in its nascency [10, 24, 32] for the scale of closed LLMs [7].

Performance. We argue that the lower performance of closed LLM adaptations stems from the fact that they have to rely on discrete prompts and that engineering such prompts for the closed LLMs is highly challenging. This is because 1) prompts, in general, have been shown to exhibit an unstable performance and to require a large number of trials and errors or discrete optimization while still underperforming gradient-based approaches [37]. Additionally, 2) when the prompts (for privacy reasons) are not tuned on the closed LLM but on an open LLM surrogate model, additional performance decrease is incurred through the prompt transfer, since it has been shown that transferred

prompts cannot reach the performance of prompts directly tuned on a given LLM [55]. While the latter problem might be mitigated through the design of more performant prompt transfer techniques, the former one seems to be a more fundamental limitation [37].

Costs. The high costs incurred by some closed LLM adaptations result from the fact that they rely on ensemble-based approaches to yield DP guarantees and the fact that they incur continuous query costs at inference time. The former one could be solvable by designing more efficient DP schemes for discrete prompts, however, the latter is inherent to the nature of closed LLMs.

We hope that implementing the above-mentioned solutions will shrink the gap between private adaptations of open and closed LLMs. However, it remains unclear whether it is worth the community's effort, given the effectiveness of private adaptations for open LLMs.

Acknowledgments

References

[1] Bart large xsum. URL `https://huggingface.co/facebook/bart-large-xsum`.

[2] Openai, https://openai.com. URL `https://openai.com/`.

[3] 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.

[4] Antropic. Introducing claude. Antropic Website. URL `https://www.anthropic.com/index/introducing-claude`. 2023-03-14, `https://www.anthropic.com/index/introducing-claude`.

[5] Trapit Bansal, Rishikesh Jha, and Andrew McCallum. Learning to few-shot learn across diverse natural language classification tasks. arXiv preprint arXiv:1911.03863, 2019.

[6] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In International Conference on Machine Learning, pages 2397–2430. PMLR, 2023.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901, 2020.

[8] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In USENIX Security Symposium, 2021. URL `https://arxiv.org/abs/2012.07805`.

[9] Nicholas Carlini, Daniel Paleka, Krishnamurthy Dj Dvijotham, Thomas Steinke, Jonathan Hayase, A. Feder Cooper, Katherine Lee, Matthew Jagielski, Milad Nasr, Arthur Conmy, Itay Yona, Eric Wallace, David Rolnick, and Florian Tramèr. Stealing part of a production language model. In International Conference on Machine Learning, 2024.

[10] Tianyu Chen, Hangbo Bao, Shaohan Huang, Li Dong, Binxing Jiao, Daxin Jiang, Haoyi Zhou, Jianxin Li, and Furu Wei. THE-X: Privacy-preserving transformer inference with homomorphic encryption. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, Findings of the Association for Computational Linguistics: ACL 2022, pages 3510–3520, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.277. URL `https://aclanthology.org/2022.findings-acl.277`.

[11] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL `https://lmsys.org/blog/2023-03-30-vicuna/`.

[12] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating llms by human preference, 2024.

[13] Christopher A Choquette-Choo, Natalie Dullerud, Adam Dziedzic, Yunxiang Zhang, Somesh Jha, Nicolas Papernot, and Xiao Wang. Capc learning: Confidential and private collaborative learning. In International Conference on Learning Representations, 2021.

[14] Mostafa Dehghani, Anurag Arnab, Lucas Beyer, Ashish Vaswani, and Yi Tay. The efficiency misnomer. arXiv preprint arXiv:2110.12894, 2021.

[15] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. arXiv preprint arXiv:2305.14314, 2023.

[16] Haonan Duan, Adam Dziedzic, Nicolas Papernot, and Franziska Boenisch. Flocks of stochastic parrots: Differentially private prompt learning for large language models. In Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS), 2023.

[17] Haonan Duan, Adam Dziedzic, Mohammad Yaghini, Nicolas Papernot, and Franziska Boenisch. On the privacy risk of in-context learning. In The 61st Annual Meeting Of The Association For Computational Linguistics, 2023.

[18] David Durfee and Ryan M Rogers. Practical differentially private top-k selection with pay-what-you-get composition. Advances in Neural Information Processing Systems, 32, 2019.

[19] Cynthia Dwork. Differential privacy. In Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II 33, pages 1–12. Springer, 2006.

[20] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science, 9(3–4):211–407, 2014.

[21] Xinyang Geng and Hao Liu. Openllama: An open reproduction of llama, May 2023. URL `https://github.com/openlm-research/open_llama`.

[22] Jennifer Gillenwater, Matthew Joseph, Andres Munoz, and Monica Ribero Diaz. A joint exponential mechanism for differentially private top-k. In International Conference on Machine Learning, pages 7570–7582. PMLR, 2022.

[23] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In Proceedings of the 2nd Workshop on New Frontiers in Summarization, pages 70–79, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5409. URL `https://www.aclweb.org/anthology/D19-5409`.

[24] Meng Hao, Hongwei Li, Hanxiao Chen, Pengzhi Xing, Guowen Xu, and Tianwei Zhang. Iron: Private inference on transformers. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, Advances in Neural Information Processing Systems, 2022. URL `https://openreview.net/forum?id=deyqjpcTfsG`.

[25] Junyuan Hong, Jiachen T. Wang, Chenhui Zhang, Zhangheng LI, Bo Li, and Zhangyang Wang. DP-OPT: Make large language model your differentially-private prompt engineer. In The Twelfth International Conference on Learning Representations, 2024. URL `https://openreview.net/forum?id=Ifz3IgsEPX`.

[26] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685, 2021.

[27] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. arXiv preprint arXiv:2310.06825, 2023.

[28] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

[29] Simon Lermen and Charlie Rogers-Smith. LoRA fine-tuning efficiently undoes safety training in llama 2-chat 70b. In ICLR 2024 Workshop on Secure and Trustworthy Large Language Models, 2024. URL https://openreview.net/forum?id=Y52UbVhglu.

[30] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, November 2021. URL https://aclanthology.org/2021.emnlp-main.243.

[31] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. CoRR, abs/1910.13461, 2019. URL http://arxiv.org/abs/1910.13461.

[32] Dacheng Li, Hongyi Wang, Rulin Shao, Han Guo, Eric Xing, and Hao Zhang. MPCFORMER: FAST, PERFORMANT AND PRIVATE TRANSFORMER INFERENCE WITH MPC. In The Eleventh International Conference on Learning Representations, 2023. URL https://openreview.net/forum?id=CWmvjOEhgH-.

[33] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 4582–4597, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353.

[34] Xin Li and Dan Roth. Learning question classifiers. In COLING 2002: The 19th International Conference on Computational Linguistics, 2002. URL https://www.aclweb.org/anthology/C02-1150.

[35] Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. In International Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=bVuP3ltATMz.

[36] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-1013.

[37] Haokun Liu, Derek Tam, Muqeeth Mohammed, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, Advances in Neural Information Processing Systems, 2022. URL https://openreview.net/forum?id=rBCvMG-JsPd.

[38] Jingjing Liu, Scott Cyphers, Panupong Pasupat, Ian Mcgraw, and Jim Glass. A conversational movie search system based on conditional random fields. 13th Annual Conference of the International Speech Communication Association 2012, INTERSPEECH 2012, 3, 01 2012.

[39] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. arXiv preprint arXiv:2110.07602, 2021.

[40] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 61–68, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.8. URL https://aclanthology.org/2022.acl-short.8.

[41] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Ro{bert}a: A robustly optimized {bert} pretraining approach, 2020. URL https://openreview.net/forum?id=SyxS0T4tvS.

[42] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. arXiv preprint arXiv:2104.08786, 2021.

[43] Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schoelkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, Findings of the Association for Computational Linguistics: ACL 2023, pages 11330–11343, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.719. URL https://aclanthology.org/2023.findings-acl.719.

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

[45] Ilya Mironov, Kunal Talwar, and Li Zhang. R\'enyi differential privacy of the sampled gaussian mechanism. arXiv preprint arXiv:1908.10530, 2019.

[46] Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In International Conference on Machine Learning, pages 24950–24962. PMLR, 2023.

[47] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1206. URL https://aclanthology.org/D18-1206.

[48] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In Proceedings of the thirty-ninth annual ACM symposium on Theory of computing, pages 75–84, 2007.

[49] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian J. Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, 2017.

[50] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with PATE. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, 2018.

[51] Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. Bleu: a method for automatic evaluation of machine translation. pages 311–318, 2002.

[52] Matt Post. A call for clarity in reporting BLEU scores. In Proceedings of the Third Conference on Machine Translation: Research Papers, pages 186–191, Belgium, Brussels, October 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-6319.

[53] Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In The Twelfth International Conference on Learning Representations, 2024. URL `https://openreview.net/forum?id=zWqr3MQuNs`.

[54] Alessandro Sordoni, Xingdi Yuan, Marc-Alexandre Côté, Matheus Pereira, Adam Trischler, Ziang Xiao, Arian Hosseini, Friederike Niedtner, and Nicolas Le Roux. Deep language networks: Joint prompt training of stacked llms using variational inference. arXiv preprint arXiv:2306.12509, 2023.

[55] Yusheng Su, Xiaozhi Wang, Yujia Qin, Chi-Min Chan, Yankai Lin, Huadong Wang, Kaiyue Wen, Zhiyuan Liu, Peng Li, Juanzi Li, et al. On transferability of prompt tuning for natural language processing. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3949–3969, 2022.

[56] Xinyu Tang, Richard Shin, Huseyin A Inan, Andre Manoel, Fatemehsadat Mireshghallah, Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, and Robert Sim. Privacy-preserving in-context learning with differentially private few-shot generation. In The Twelfth International Conference on Learning Representations, 2024. URL `https://openreview.net/forum?id=oZtt0pRnOl`.

[57] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805, 2023.

[58] Zhiliang Tian, Yingxiu Zhao, Ziyue Huang, Yu-Xiang Wang, Nevin L Zhang, and He He. Seqpate: Differentially private text generation via knowledge distillation. Advances in Neural Information Processing Systems, 35:11117–11130, 2022.

[59] Rubèn Tito, Khanh Nguyen, Marlon Tobaben, Raouf Kerkouche, Mohamed Ali Souibgui, Kangsoo Jung, Lei Kang, Ernest Valveny, Antti Honkela, Mario Fritz, and Dimosthenis Karatzas. Privacy-aware document visual question answering. arXiv preprint arXiv:2312.10108, 2023.

[60] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971, 2023.

[61] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

[62] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL `https://aclanthology.org/W18-5446`.

[63] Tong Wu, Ashwinee Panda, Jiachen T. Wang, and Prateek Mittal. Privacy-preserving in-context learning for large language models. In The Twelfth International Conference on Learning Representations, 2024. URL `https://openreview.net/forum?id=x4OPJ7lHVU`.

[64] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. Differentially private fine-tuning of language models. In International Conference on Learning Representations, 2022. URL `https://openreview.net/forum?id=Q42f0dfjEC0`.

[65] Jiachen ZHAO, Zhun Deng, David Madras, James Zou, and Mengye Ren. Learning and forgetting unsafe examples in large language models, 2024. URL `https://openreview.net/forum?id=hkQOYyUChL`.

[66] Chujie Zheng, Fan Yin, Hao Zhou, Fandong Meng, Jie Zhou, Kai-Wei Chang, Minlie Huang, and Nanyun Peng. On prompt-driven safeguarding for large language models. In Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024. OpenReview.net, 2024. URL `https://openreview.net/forum?id=ugxGpOEkox`.

[67] Yuqing Zhu and Yu-Xiang Wang. Adaptive private-k-selection with adaptive k and application to multi-label pate. In International Conference on Artificial Intelligence and Statistics, pages 5622–5635. PMLR, 2022.

A Broader Impacts

Our comparative study of open and closed LLMs has significant implications for privacy adaptation. While our research advocates for the use of open LLMs for privacy adaptation, we acknowledge that any research, including ours, could potentially be misused.

However, we stress that our goal is not to discredit closed LLMs, but to highlight the potential privacy benefits and cost-effectiveness associated with the use of open LLMs. Through thorough evaluations, we demonstrated in our paper that adapting open LLMs with private parameter efficient fine-tuning methods results in higher performance and mitigates open privacy risks of in-context learning with closed models. This not only leads to better performance but also reduces costs, making privacy adaptation on open LLMs a more viable option for many applications.

Moreover, our work can serve as a baseline for future private learning methods for LLMs, furthering research in this area. We believe that an open dialogue about the strengths and weaknesses of both open and closed LLMs is crucial for the advancement of privacy-preserving LLMs. We hope that our research will serve as a catalyst for further investigations into privacy adaptation in LLMs, ultimately leading to the development of models that effectively balance the need for both openness and privacy, all while ensuring user privacy remains uncompromised.

B Limitations

Hyperparameter tuning. Due to limited computational resources, we were not able to tune the hyperparameters for all open LLM private tuning methods. This is especially true for the generation tasks with longer input length, PFL-DocVQA for example, due to their long training time. Therefore, we still see the potential for an increase in performance. But, as our results for private tuning already show that these methods outperform the private in-context learning approaches, the used hyperparameters suffice for these comparisons.

Pretraining set as privacy risk. Closed models do not provide much information about their pretraining process. This also includes the non-disclosure of the pretraining set. For this reason, we purposefully chose open LLMs where the pretraining set is known and can be easily downloaded. But, even though the pretraining set is available, and a user could check it directly against their own data for any privacy leakage, we recognize that this process is also costly. This is why, despite having potential access to the pretraining set, a user might not be able to cover the privacy risk fully.

Cost estimation. To show the potential cost of running the training of privately tuning an open LLM, we used the cost of $0.69 per compute hour from RunPod. This represents the average cost of running such training, and the cost might vary for different users on machines of different cloud providers. Additionally, we ran our training on our own machines, therefore, do not have the exact same server setup as the one given by RunPod. This could lead to additional variations in training time and therefore training costs.

Budgetary limits. In this paper we looked at different closed models (GPT3 Davinci, GPT3 Babbage, GPT4 Turbo, and Claude 2.1) and methods to add private in-context learning. Due to limitations to our budget, we were not able to look at a higher variety of closed models and also extend the experiments with GPT4 Turbo.

Limited model sizes. In our experiments for the private tuning of open LLMs, we tried a variety of different models and model sizes. To handle the models with a higher amount of parameters, we needed to use 4-bit loading to be able to run the training on our GPUs. Unfortunately, due to some type-inconsistencies with Opacus, the Python library we use to run DP-SGD, and 4-bit loading from

bitsandbytes, we were not able to do training on our machines for some of the models. This included 13B models for PrivateLoRA for the classification tasks and 7B+ models for PromptDPSGDGen and DP-FineTune for the text generation tasks.

C   Further Details on the Related Work

C.1   Private Adaptations for Closed LLMs

We present the methods for the private adaptations of closed LLMs with a few more details.

PromptPATE [16] prompts an LLM with different prompts containing disjoint examples from the private training dataset, each prompt corresponding to a teacher. To label the public data for the knowledge transfer, PromptPATE [16] infers the next token prediction of each teacher on public text sequences and interprets them as labels. Instead of training a student model from scratch, PromptPATE [16] creates a student prompt. It utilizes the data efficiency of discrete prompts by selecting examples for the student prompt from the labeled public sequences.

DP-ICL [63]. For the generation tasks, it proposes the Embedding Space Aggregation(ESA), which involves mapping each sentence produced by the LLM for a given exemplar-query pair onto the embedding space and then reconstructing a sentence from the noisy mean of these embeddings. This process depends on the quality of the text-to-embedding models and the zero-shot examples employed to map the noisy mean embedding back to the sentence, potentially leading to suboptimal outputs. The other approach proposed in DP-ICL is keyword space aggregation (KSA). It creates a *keyword space* by segmenting each output sentence into keywords to form a histogram. The keywords with the highest counts are selected privately using either the Propose-Test-Release (PTR) or Joint Exponential Mechanism (JointEM) [22]. The selected private keywords are then used to create a prompt and query the LLM.

DP-FewShotGen [56] introduces a method for text generation of public prompts. In this method, tokens are individually generated using disjoint subsets of the private data and then noisily aggregated based on the frequency of the generated tokens to predict the next token. The drawback of this approach is that the generation process is conditioned on the label. Consequently, despite being a text generation task, it necessitates the assignment of a public label to the private data.

DP-OPT [25] is currently the only private ICL method that uses discrete prompt tranferrability to create a private prompt on a local open model, which can be used to infer a closed model. Based on the approach of deep language networks [54], multiple initial prompts with different private examples are optimized through separate back- and forward passes such that a prompt is created that gives good performance on the downstream task. To add privacy, they use the exponential mechanism to sample each generated token from all different initial prompts. Currently, their proposed method is only shown to work with classification tasks.

C.2   Private Text Generation based on PATE

SeqPATE [58] safeguards the privacy of individual training samples and sensitive phrases in the training data of a language model. To adapt PATE for text generation, SeqPATE creates pseudo-contexts, simplifying the sequence generation task to a next-word prediction problem. To manage the extensive output space, SeqPATE introduces a candidate filtering strategy that dynamically narrows the output space and enhances the teacher aggregation in PATE to avoid low agreement caused by voting among a large number of candidates. Additionally, to further minimize privacy losses, it employs knowledge distillation to reduce the number of teacher queries.

C.3   Membership Inference Attacks

Threshold based: Carlini et al. (2021) [8] Extracting training data from large language models Min-K% Prob: Shi et al. (2024) [53] Detecting pretraining data from large language models Perturbation based: Mattern et al. (2023) [43] Membership inference attacks against language models via neighbourhood comparison DetectGPT: Mitchell et al. (2023) [46] Detectgpt: zero-shot machine-generated text detection using probability curvature zlib Ratio: Carlini et al. (2021) [8] Extracting training data from large language models Reference based: Carlini et al. (2021) [8] Extracting training data from large language models

D Additional Details on our Setup

In this section, we present the detailed (hyper-)parameters used to evaluate all the tasks that were used for the different Open and Closed LLMs privacy-preserving training methods.

D.1 Text classification

Detailed information about the datasets. We expose the different statistics of each dataset used for text classification evaluation in Table 7. For SST2, the validation set was used as the test set, as the original test set is only provided with unknown labels for each sample.

Table 7: **Stastistics of the 4 evaluated tasks** related to text classification.

Task	#Train	#Test	#Class	Task description
SST2	66,674	872	2	Sentiment analysis on movie reviews
Trec	5,452	500	6	Question type classification
Mpqa	8,603	2,000	2	Sentiment analysis on short ensembles
Disaster	4,430	1,000	2	Relevance of sentence to a disaster

Private Tuning. We detail the hyperparameters used to fine-tune the models with private LoRA in Table 8, for DP-FineTune in Table 9 and for PromptDPSGD in Table 10. All the experiments were conducted on 3 different seeds. Note that unlike LoRA or Full-Finetune, PromptDPSGD requires a precise tuning of hyperparameters. A total of 50 trials over 100 epochs were necessary to tuned them. For the Mpqa sentiment analysis task, no converging set of hyperparameters was found.

Table 8: **Hyperparameters for PrivateLoRA [64]** on evaluated classification datasets for $\varepsilon = 8$.

Hyperparameters	Datasets			
	SST2	Trec	Mpqa	Disaster
bs	128	128	128	128
lr	1e-3	1e-3	1e-3	1e-3
max grad clip	0.1	0.1	0.1	0.1
epochs	10	40	20	20
lora rank	4	4	4	4
δ	$\frac{1}{|D|}$	$\frac{1}{|D|}$	$\frac{1}{|D|}$	$\frac{1}{|D|}$
GradClip	0.1	0.1	0.1	0.1

Table 9: **Hyperparameters for DP-FineTune [35]** on evaluated classification tasks with Roberta-Large for $\varepsilon = 8$.

Hyperparameters	SST2	Trec	Mpqa	Disaster
LR	1e-4	1e-4	1e-4	1e-4
BS	128	128	128	128
Epoch	10	40	40	50
δ	$\frac{1}{|D|}$	$\frac{1}{|D|}$	$\frac{1}{|D|}$	$\frac{1}{|D|}$
GradClip	0.1	0.1	0.1	0.1

Table 10: **Hyperparameters for PromptDPSGD [16].** The hyperparameters for SST2 datasets are directly extracted from the paper and are evaluated on Roberta-Large for $\varepsilon = 8$. LR = learning rate, BS = batch size, GRAD = per sample gradient clipping. P-length = length of the prepended prompt in number of tokens. The trainings are all performed with prefix-tuning and not soft-prompt. Those are the hyperparameters of the best performing prompt on the test set of each dataset, and the accuracy of this prompt is reported in the table.

Hyperparameters	SST2	Trec	Mpqa	Disaster
LR	0.01	0.001	-	0.01
BS	32	32	32	32
GRAD	4	0.3	-	1.0
Epochs	22	100	100	100
P-length	1	10	10	10
Best accuracy	92.8	58.0	50.0	78.6

Private in-context learning. The respective set of hyperparameters for DP-FewShotGen, Prompt-PATE and DP-ICL are listed in Table 11, Table 12 and Table 13. For the used hyperparameters for DP-OPT, see [25] since the results of Table 3 are directly extracted from the paper. The accuracy results for DP-FewShotGen were computed for 5 different generated prompts following the method from the paper. For the PromptPATE method, experiments were only conducted for MPQA and Disaster datasets as we used already made evaluation from the original paper PromptPATE [16] for SST2 and Trec datasets on using GPT3-Babbage. All hyperparameters here are extracted directly from the previous paper.

Table 11: **Hyperparameters for DP-FewShotGen [56]** for the evaluation of new datasets with $\varepsilon = 8$ on GPT3-Babbage. M = Number of private prompts used for meta prompt generation. N = number of private shots per prompt. σ = noise relative to wanted ε using the Gumbel mecanism. T_{max} = max number of tokens of the generate prompt.

Hyperparameters	SST2	Trec	Mpqa	Disaster
σ ($\varepsilon = [0.1, 1, 3, 8]$)	[1.0,0.61, 0.48,0.34]	[3.0,0.83, 0.59,0.44]	[2.0,0.77, 0.57,0.41]	[3.5,0.93, 0.64,0.46]
MN	80	80	80	80
M	20	20	20	20
T_{max}	50	50	50	50

Table 12: **Hyperparameters for PromptPATE [16]** for the evaluation of new datasets with $\varepsilon = 8$ on GPT3-Babbage.Those parameters are common to all 4 tasks.

Hyperparameters	Claude	GPT3-babbage
train set	400	400
student set	200	300
num shots	2	1

Table 13: **Hyperparameters for DP-ICL [63]** for the evaluation of the text classification datasets with $\varepsilon = 8$ on GPT3-Babbage.

Hyperparameters	SST2	Trec	Mpqa	Disaster
num shots	4	4	4	4
Ensemble	10	10	10	10
Queries	872	500	1000	1000

D.2 Text Generation

We analyze the following generative downstream tasks: SAMSum, PFL-DocVQA, and MIT Movies trivia10k13. As we did for classification tasks, we compare the methods on closed LLMs against PrivateLoRA [64], PromptDPSGD [16], and DP-FineTune [35] that are run on open LLMs. For the PrivateLoRA [64] training, we use 4-bit quantization with QLoRA [15] to reduce the occupied GPU memory, which was implemented for the adaptations of open LLMs with more than 1B parameters on PFL-DocVQA and SAMSum datasets due to their long input sequences.

Detailed information about the datasets. We show the amount of data that we utilized in the experiments in Table 14.

Table 14: **Overview of the 4 text ge tasks** related to text generation.

Task	#Train	#Test	Task description
SAMSum	14,732	819	Dialogue summarization
PFL-DocVQA	85,000	10,000	Question and answering
MIT-G	2,953	780	Extracting genres from movie reviews
MIT-D	1,561	415	Extracting directors from movie reviews

Private Tuning. In Table 15, Table 16, and Table 17, we show the hyperparameters we used to train the open models with PrivateLoRA, PromptDPSGDGen, DP-FineTune respectively. For PrivateLoRA, we were able to use the same hyperparameters for all models for each task. In the tables, the *Max Seq Length* refers to the maximum amount of tokens of the sequence the model trains on. For *Schedulers*, we chose two different options, a constant scheduler that does not change the learning rate during training, and a linear scheduler. The linear scheduler is the default scheduler of the Hugging Face implementation of the Trainer class. It linearly decreases the learning rate over the whole training. For PromptDPSGDGen, we additionally have *Prefix Projection*. If enabled, prefix projection adds two additional linear layers to the prefix encoder. This increases the amount of trainable parameters, which in turn also increases the capability of the prefix to represent tasks. The evaluations for MIT-D, MIT-G, and SAMSum were done for 3 different seeds, whereas we used 2 different seeds for PFL-DocVQA.

Table 15: **Hyperparameters for PrivateLoRA [64]** on evaluated generation tasks for $\varepsilon = 8$. The hyperparameters are the same for the used models. The tested schedulers for MIT-G and MIT-D does not make a difference during training

Hyperparameters	SAMSum	PFL-DocVQA	MIT-G	MIT-D
LR	8e-4	8e-4	8e-4	8e-4
BS	256	256	256	256
LoRA Rank	8	8	8	8
Max Seq Length	650	1500	128	128
Epoch	20	15	20	20
Scheduler	Linear	Linear	/	/
δ	$\frac{1}{|D|}$	$\frac{1}{|D|}$	$\frac{1}{|D|}$	$\frac{1}{|D|}$
GradClip	0.1	0.1	0.1	0.1

Table 16: **Hyperparameters for PromptDPSGDGen** on evaluated generation tasks for $\varepsilon = 8$. The hyperparameters are the same for the used models. The tested schedulers for MIT-G and MIT-D do not result in difference in performance.

Hyperparameters	SAMSum	PFL-DocVQA	MIT-G	MIT-D
LR	1e-3	1e-3	1e-3	3e-3
BS	256	256	256	256
P-Length	10	25	5	5
Prefix Projection	True	True	True	True
Max Seq Length	650	1500	128	128
Epoch	20	15	40	40
Scheduler	Linear	Linear	/	/
δ	$\frac{1}{|D|}$	$\frac{1}{|D|}$	$\frac{1}{|D|}$	$\frac{1}{|D|}$
GradClip	0.1	0.1	0.1	1

Table 17: **Hyperparameters for DP-FineTune [35]** on evaluated generation tasks for $\varepsilon = 8$.

Hyperparameters	SAMSum	PFL-DocVQA	MIT-G	MIT-D
LR	8e-4	2e-4	2e-4	2e-4
BS	256	256	256	256
Max Seq Length	650	1500	128	128
Epoch	20	15	20	20
Scheduler	Linear	Linear	Constant	Linear
δ	$\frac{1}{|D|}$	$\frac{1}{|D|}$	$\frac{1}{|D|}$	$\frac{1}{|D|}$
GradClip	0.1	0.1	0.1	0.1

Privacy-preserving prompt tuning. In the following, we provide the used hyperparameters for the methods for Private ICL for Closed LLMs. In detail, for DP-FewShotGen in Table 18, for DP-ICL in Table 19, and for PromptPATEGen in Table 20.

Table 18: **Hyperparameters for DP-FewShotGen [56]** on evaluated generation tasks for $\varepsilon = 8$. We used the hyperparameters given in the original paper for MIT-G and MIT-D.

Hyperparameters	SAMSum	MIT-G	MIT-D
σ	0.384	0.5	0.58
MN	80	80	80
M	20	20	20
T_{max}	50	20	20

Table 19: **Original hyperparameters for DP-ICL [63]** on evaluated generation tasks for $\varepsilon = 8$.

Hyperparameters	SAMSum	PFL-DocVQA
Model	GPT-Davinci	OpenLLaMA 13B
Ensemble	100	100
#Queries	10,000	10,000

Table 20: **Hyperparameters for PromptPATEGen** on generation tasks for $\varepsilon = 8$.

Hyperparameters	SAMSum	MIT-G	MIT-D
Model	Vicuna 7B	Vicuna 7B	Vicuna 7B
Ensemble	100	25	25
#Queries	100	100	100
#Student Prompt	10	4	4
σ	1.15	0.9	0.9

E   Cost Calculation

We provide the details on measuring the cost for different methods. The assumed costs for interacting with the model APIs per 1 million tokens and GPU cost per hour are shown in Table 22. For the open LLMs, we set the median pricing per hour (based on prices from three GPU cloud providers shown in Table 22) which is $0.69 using an A40 GPU with 48GB of memory [3], which is a popular graphics card, also used in the previous work [16]. We note that we do refrain from using other metrics than monetary cost. For example, FLOPS are not a direct measurement of real-world computational cost because latency, power usage, and other costs can vary significantly depending on hardware and other factors [14].

Costs for private-tuning-based adaptations.   The private tuning-based adaptations of open LLMs require us to adjust the model parameters or the inputs for a given task, thus, we measure the running time of the training process and then query answering.

Costs for private ICL-based adaptations.   DP-ICL does not incur any training cost but uses an ensemble of teachers for each query (the same as PromptPATE for labeling public examples), which elevates the cost by the number of teachers, which can be 10 or even 100. For PromptPATE, the generation of public student prompts is done using an ensemble of teacher prompts, thus labeling each public data point costs much more (proportional to the number of teachers) than running a query (with a single prompt). DP-FewShotGen also uses an ensemble of prompts, where the number of accesses to the API in the training process is equal to the number of tokens in a public prompt. The cost of training the public prompt for DP-OPT is through the iterative process of instructing the local model to improve the prompt and obtain better predictions, however, this part is done on a local open

---

[3]The pricing is for the RunPod Cloud Service: `https://www.runpod.io/gpu-instance/pricing`.

LLM, thus, the cost is relatively low. For ease of approximation and to the benefit of the ICL methods, we assume that the creation of the teacher prompts and the private aggregation of the outputs have negligible costs. After preparing the public prompt, PromptPATE, DP-FewShotGen, and DP-OPT, need a single access to the API to answer a query.

To obtain the cost for closed LLMs, we have to compute the average number of tokens per query. For the classification task, we can take the example of the DP-OPT method applied on the SST2 dataset. For this dataset, only one token is returned by the API provider, so the cost of the outputs is negligible. SST-2 inputs have an average length of 12.35 and the best performing prepended prompt from DP-OPT training has a length of 39 tokens. Thus, for the DP-OPT task, for each query to the API, 41.35 tokens are sent approximately. This gives a cost of $0.0006 per query for GPT-3 Davinci and the total cost of $6 for 10k queries in Table 1. The cost per query is computed similarly, depending on the size of the prepended prompt of each ICL method. Regarding the generation task, we can take the example of the SAMSum dialog summarization dataset, in which the average token length is 141 for the input and 26 for the output, hence, a single query costs $0.000333 (for GPT3-Davinci). The cost for a 0-shot inference to Davinci would therefore be $3.33 for 10k queries. As DP-ICL considers the 1-shot scenario and an ensemble of 100 teachers, we add the average input and label lengths to the input and multiply this by the size of the ensemble, which results in an overall cost of roughly $666. The exact average token count for each dataset which we used for the cost estimations can be found in Table 21.

Table 21: **Average token length** of different inputs and outputs of the used datasets. The average does not include instructions.

Dataset	SST-2	Trec	Mpqa	Disaster	MIT-D	MIT-G	SAMSum	DocVQA
Input	12.35	11.43	3.88	30.79	25.276	24.314	140.857	924.191
Output	1	1	1	1	3.877	2.301	25.620	6.384

Table 22: **Pricing** for the models and cloud options (as of May 22nd, 2024).

Model	Cost/1M tokens		Cost/hour
	Input	Output	
GPT-Babbage [4]	$0.40	$0.40	-
GPT-Davinci	$2.00	$2.00	-
GPT-3.5-turbo Instruct	$1.50	$2.00	-
GPT-4-turbo	$10.00	$30.00	-
Claude 2.1 [5]	$8	$24	-
A40 (RunPod) [6]	-		$0.69
A40 (Replicate) [7]	-		$2.07
A40 (Hyperstack) [8]	-		(starts from)$0.50

F   Additional Experiments

PrivateLoRA extensive classification results. Table 23 shows the top1 accuracies at different ε used to compute the PrivateLoRA graph for each of the 4 text classification tasks in Figure 2.

---

[4]https://openai.com/api/pricing/

[5]https://www.anthropic.com/api

[6]https://www.runpod.io/gpu-instance/pricing

[7]https://replicate.com/pricing

[8]https://www.hyperstack.cloud/gpu-pricing

Table 23: **Private LoRA [64] top1-accuracies** for the evaluated datasets given different ε.

ε	Model	SST-2	Trec	Mpqa	Disaster
8	Vicuna 7B	95.3	97.4	88.4	82.0
3	Vicuna 7B	94.4	96.2	87.6	79.6
1	Vicuna 7B	93.5	93.8	82.1	78.1
0.7	Vicuna 7B	93.4	93.2	79.5	76.4
0.3	Vicuna 7B	91.9	87.6	64.4	73.6

Privacy-utility trade-off of text generation tasks.   In the following, similar to what we did in Figure 2, we show the privacy-utility trade-off for SAMSum, MIT-G, and MIT-D in Figure 3 for varying ε between PrivateLoRA, PromptPATEGen, and DP-FewShotGen. For MIT-D and MIT-G, we trained the Pythia 1B model, and for SAMSum the BART-Large Model. It can be clearly seen, that the graphs follow the same trend that we showcased in Figure 2.

(a) SAMSum   (b) MIT-G   (c) MIT-D
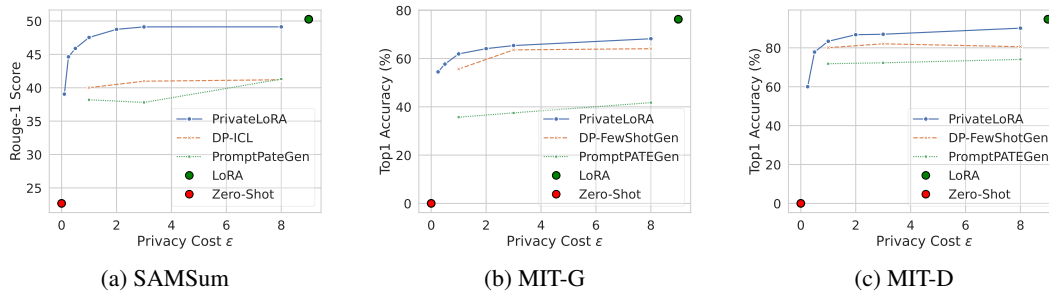
Figure 3: **Privacy-utility trade-off for generation tasks.** We analyze the privacy costs ε in the range $[0, 8]$ for the three generation tasks. PrivateLoRA for open LLMs substantially outperforms DP-ICL and DP-FewShotGen, which both utilize GPT3 Davinci. PrivateLoRA for MIT-D and MIT-G is trained on the Pythia 1B model, and for SAMSum on the BART-Large Model. PromptPATEGen uses Vicuna 7B.

Safety Evaluation of Mixtral-8x7B Instruct with and without differential privacy.   We conducted additional experiments to analyze how fine-tuning a downstream task with and without differential privacy affects the safety alignment of models. We followed the approach from [66] to evaluate Mixtral-8x7B-instruct and fine-tuned the model on SAMSum once with $\varepsilon = 8$ and once with $\varepsilon = \infty$. We selected SAMSum as it does contain conversations with unsafe language (e.g., cursing or harassment). These results are presented in Table 24. The table is divided into two sections, "Compliance on Harmful Queries" and "Refusal on Harmless Queries". The scores were generated by separately prompting our model with 100 harmful and 100 harmless queries, each repeated 20 times with different safety prompts. Finally, the outputs are categorized as complying or refusing the input by Llama Guard, giving the percentage of incorrectly handled answers in our table. Therefore, the lower the score, the better.

First, we observe that fine-tuning of any kind decreases the model's safety capability. This is easily identifiable in the table, as the original model has lower scores across the board for compliance with harmful behavior compared to both fine-tuned models. Similar results were also shown in [29] and [65], where the authors used a dataset containing unsafe samples to fine-tune safety-aligned models, which drastically increases compliance with unsafe behavior.

Second, we can observe that private fine-tuning affects the safeguards less than non-private fine-tuning. In the non-private case, the influence of individual samples is unrestricted, which can amplify the impact of unsafe samples, as shown in [29] and [65]. In contrast, differential privacy limits the influence of any single sample. Consequently, the impact of unsafe examples is minimized, resulting in a model that retains more of its safe behavior.

Table 24: **Safety Evaluations of Mixtral-8x7B Instruct** Evaluating the safety of the responses given by different models. We compare the base instruction fine-tuned model with the same fine-tuned on SAMSum with and without differential privacy. The model responses were categorized as harmful and harmless by Llama Guard.

Model	% Compliance on Harmful Queries ↓				% Refusal on Harmless Queries ↓			
Mixtral-8x7B Instruct	short	Mistral	Llama	no prompt	short	Mistral	Llama	no prompt
Base model	1	0	0	36	1	0	2	0
$\varepsilon = \infty$	38	30	39	64	2	2	3	0
$\varepsilon = 8$	9	5	7	48	2	0	1	0

Analysis of PromptPATEGen and DP-ICL [63] across different generation tasks. We observe that PromptPATEGen is particularly adept at summarization tasks, as shown in Table 4. However, when we shift focus to other task generation tasks, the performance of PromptPATEGen is not as impressive compared to DP-ICL [63]. For the MIT datasets, which are entity retrieval tasks, PromptPATEGen generates results that are semantically similar to the true values, but not identical. For instance, on the MIT-G dataset, PromptPATEGen generates "sci-fi" whereas the ground truth movie genre is "sci fi". This pattern of generating semantically similar results is also observed in the MIT-D dataset, where only one name is given as the movie director rather than the full names.

We further added evaluations using DP-ICL [63] with newer OpenAI models, such as GPT4 Turbo and GPT3.5 Turbo, for SAMSum and saw that the performance only increases slightly compared to the original evaluations on GPT3 Davinci (see Table 4). Therefore, we looked through the summarizations that were given by GPT4 Turbo and compared them to the ground truths. Similarly to the above-mentioned problem with the MIT-G and MIT-D datasets, we found, that the results were semantically correct, but not close enough to the ground truth. As an example, we show a sample generation with the corresponding dialogue and ground truth in Table 25. We can see, that the generated summary is factually correct, but too extensive compared to the ground truth. In particular, this example results in a Rouge-1 score of 31.8. Our assumption, as we get more extensive, and thus, worse samples, is the limited examples in terms of summary structure given by the keywords provided by DP-ICL [63].

The DocVQA dataset presents a unique challenge for PromptPATEGen. Given the high average token length in this dataset, only a single example can be used as the in-context example. This limitation poses a significant challenge, as a single example may not adequately represent the diverse aspects present in the dataset.

DP-FewShotGen [56] limitations on new GPT models. In addition to running DP-FewShotGen [56] on GPT Babbage and GPT Davinci, we also adapted the official code to work with newer OpenAI models, such as GPT4. The code had to be changed, as OpenAI moved from their Completion API to the new Chat Completion API. During execution, we encountered an issue that prevents us from running experiments with the Chat Completions API with DP-FewShotGen [56]. The issue arises from the parallel use of logit bias and log-probabilities, as the resulting output while using both together is unusable for further processing. We assume, this behavior is caused by a defense mechanism against a new potential model-stealing attack [9], as one of the suggested defenses was to prohibit the use of both logit bias and log-probabilities at the same time, which would explain what we have experienced. As the implementation of DP-FewShotGen [56] uses both at the same time, we are unable to get this private ICL method to work with GPT models that use the Chat Completions API.

G Generation Metrics

In this section, we briefly discuss the different metrics we use to evaluate the generation tasks.

Rouge [36]. The metrics in the Rouge, short for Recall-Oriented Understudy for Gisting Evaluation, set describe how many word-wise n-grams match between the predicted and target text. For Rouge-1,

Table 25: The dialogue, ground truth, and summary generated with **DP-ICL [63] on GPT4 Turbo** for a **SAMSum** sample

Dialogue	Ursula: Haha I got a 93 on my French exam Bob: Well done girl! Jana: Wow Jana: How did u manage to do that Ursula: I just studied hard for it Jana: omg Jana: French is so hard Vaughn: I got a 65 Vaughn: I didn't study for it haha Ursula: At least you passed Vaughn: Congrats!
Ground truth	Ursula got 93 on her French exam. Vaughn got 65, but still passed.
Generated summary	Ursula shares that she got a 93 on her French exam and Jana asks how she managed to do so well. Ursula says she studied hard for it. Vaughn admits to getting a 65 without studying and Ursula congratulates him for passing.

we look at uni-grams whereas for Rouge-2 we calculate the similarity of all 2-grams. Rouge-L refers to the similarity of the longest common subsequence between prediction and target. Important to note for Rouge-L, the grams do not need to be consecutive, but have to be in order. The scores lie between 0 and 100, where 100 is the best score.

BLEU [51]. Similar to the Rouge metric, the BLEU score, which is the abbreviation for Bilingual Evaluation Understudy, is used to evaluate the similarity of generated and reference text. To calculate the score, the precision and brevity between the two sentences have to be determined. The precision is the ratio of n-grams that match exactly between generated and reference text. Usually, n goes up to 4. Brevity, on the other hand, penalizes the score of the generated text, if it's shorter than the reference. Combining brevity and precision results in the BLEU score of the generated text. The score itself is again between 0 and 100, where higher scores are better. We use the SacreBLEU [52] version of BLEU.

Levenshtein Distance. Lastly, to evaluate PFL-DocVQA, we also use the Levenshtein Distance. This metric is used to directly compare strings on a letter by letter basis. The Levenshtein Distance calculates the minimum amount of substitutions, insertions, and deletions between two sequences. We use the normalized version to have a score between 0 and 100 independent of sequence length. As with the other metrics, the higher the score the better.

H   Abbreviations

In Table 26, we show the abbreviations we used throughout this paper for the different private in context learning methods of LLMs.

Table 26: **Abbreviations for ICL papers** and their proposed techniques.

Publication Abbreviation	Publication Name	Technique Abbreviation	Privacy Technique
DP-ICL	Privacy-Preserving In-Context Learning for Large Language Models [63]	DP-ICL Classification	DP-ICL for text classification
		ESA	Embedding Space Aggregation
		KSA	Keyword Space Aggregation
DP-OPT	DP-OPT: Make Large Language Model Your Privacy-Preserving Prompt Engineer [25]	DP-OPT	Differentially-Private Offsite Prompt Tuning
FewShotGen	Privacy-Preserving In-Context Learning with Differentially Private Few-Shot Generation [56]	FewShotGen	Differential Private Few-Shot Generation
PrivatePrompts [16]	Flocks of Stochastic Parrots: Differentially Private Prompt Learning for Large Language Models [16]	PromptDPSGD [16]	DPSGD for Private Soft Prompt Learning
		PromptPATE [16]	PATE for Privacy-Preserving Discrete Prompts

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In Section 4, we showcase the outcomes of our thorough experiments, which compare the performance, cost, and privacy protection of open and closed LLMs. Section 3 details our proposed prompt-based adaptation techniques for text generation tasks.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We presented the limitations of the paper is in the Appendix B.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: Our paper does not include theoretical proofs. For our theoretical considerations, we provide a full set of assumptions and detail the setup.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide detailed information of all the models, datasets and hyperparameter configuration used for obtaining our results in Appendix D.

Guidelines:

- The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The details of our experimental setup is in Section 4.2.1. For the closed LLMs, the models can be accessed via their respective API. For the open LLMs, they are publicly available on the HuggingFace hub. We have included citations and links for both closed and open LLM in the paper. Throughout our paper, we used standard benchmark datasets which are open-sourced datasets. The source for the datasets are well referenced in the paper. Our anonymized codes is included in the supplementary materials.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The experimental settings is presented in Section 4.2.1. All the hyperparameters used in the paper is included in Appendix D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: In our paper, we consistently provide the mean and standard deviation across multiple runs of our experiments for comprehensive understanding. However, in certain instances, we have refrained from including the standard deviation to maintain the readability of the tables. This is particularly applicable when the inclusion of standard deviation results in tables that are excessively wide, thereby posing challenges in terms of fitting into the page width and ensuring a seamless reading experience.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: In Appendix E, we provided details about the compute resources and the costs to reproduce each experiment. All experiments were conducted on A40 GPUs.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: We followed the ethics guideline and adequately anonymized our paper and code.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Our paper advocates for the more private, efficient, and cost-effective alternative of open LLMs which have significant positive societal impact. In Section 5, we summarize our findings, shedding light on the three pivotal aspects—privacy preservation, performance optimization, and cost-effectiveness. Moreover, we highlighted the broader impacts of our work in Appendix A.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our paper poses no such risks

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We included citations for all models, datasets and code used in our paper.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We used publicly available assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Our paper does not involve the use of human subject.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Our paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.