Locally Differentially Private In-Context Learning

Chunyan Zheng, Keke Sun, Wenhao Zhao, Haibo Zhou,
Lixin Jiang, Shaoyang Song, Chunlai Zhou
Renmin University of China, Beijing, CHINA
jany.zh666@gmail.com, {skk2020, zhaowh, zhoub21,lixinjiang,songshaoyang,czhou}@ruc.edu.cn

Abstract

Large pretrained language models (LLMs) have shown surprising In-Context Learning (ICL) ability. An important application in deploying large language models is to augment LLMs with a private database for some specific task. The main problem with this promising commercial use is that LLMs have been shown to memorize their training data and their prompt data are vulnerable to membership inference attacks (MIA) and prompt leaking attacks. In order to deal with this problem, we treat LLMs as *untrusted in privacy* and propose a *locally differentially private framework of in-context learning* (LDP-ICL) in the settings where labels are sensitive. Considering the mechanisms of in-context learning in Transformers by gradient descent, we provide an analysis of the trade-off between privacy and utility in such LDP-ICL for classification. Moreover, we apply LDP-ICL to the discrete distribution estimation problem. In the end, we perform several experiments to demonstrate our analysis results.

Keywords: In-context learning, local differential privacy, LLM

1. Introduction

Large language models (LLMs) have exhibited surprising emergent abilities for in-context learning (Brown et al., 2020a). With a few input-label pairs as exemplars, they can predict the label for an unseen input without additional parameter modifications. Although the training data for an LLM is usually assumed to be public and non-private, the demonstration pairs in in-context learning for the downstream task may contain private information about individual users and are often considered to be *sensitive*. After Samsung leaked private data by using the LLM ChatGPT (Mitchell, 2023) and Italy banned the use of ChatGPT due to the concern about the exposure of personal information, it becomes imminent to study the privacy-preservation for the LLMs.

In this paper, we propose a *locally differentially private in-context learning* (LDP-ICL). Differential privacy (DP) is now a gold standard of privacy-preserving which addresses the paradox of learning nothing about an individual while learning useful information about a population (Dwork et al., 2006). There are two kinds of DP models: one is the central model and the other is the local model (Kasiviswanathan et al., 2011; Duchi et al., 2013; Warner, 1965). In the central model, the original private data are aggregated by the curator and then are perturbed by a DP mechanism before publishing. On the other hand, in the local model, the private data of each individual get randomized locally according to a DP mechanism and then are aggregated by the curator. The main difference between these two models is that the local model treats the data curator *untrusted* while the central model believes in the curator. In the in-context learning, LLMs are usually the data aggregator and LLMs have been shown to memorize their training data (Biderman et al., 2023; Carlini et al., 2019), and their prompt data are vulnerable to membership inference attacks (MIA) (Duan et al., 2023) and prompt leaking attacks (Perez and Ribeiro, 2022). In this sense we consider LLMs as untrusted. Our first contribution is to propose a locally differentially private mechanism for protecting individuals' privacy in ICL. In this paper, we focus on the classification problem, especially the binary classification. For each input-label pair in the demonstration set, we consider the input as an identifier and hence nonsensitive but regard the label as *sensitive* (Dinur and Nissim, 2003). We employ the well-known LDP mechanism *k-ary randomized response (k-RR)* to perturb each label and obtain an input-label pair with a noisy label (Kairouz et al., 2016; Wang et al., 2017). An adversary can query the LLM with x_{test} prepended by such a perturbed demonstration set. Due to the noises in the labels in the demonstration set, the adversary obtains a corresponding noisy label as response to the input x_{test}, from which he cannot reliably tell the true label of any input in the private demonstration set. This implies that the privacy in the labels are protected. The process of such an LDP-ICL is illustrated in Figure 1.

Our second contribution is to propose a formula to represent the prediction output probability for the noisy label of the query in the LDP-ICL (Eq. (7)) by considering ICL for classification as an *implicit* gradient-descent based optimization, which is the dual form of the Transformer attention in ICL (Irie et al., 2022a; Von Oswald et al., 2023;

LREC-COLING 2024, pages 10686–10697

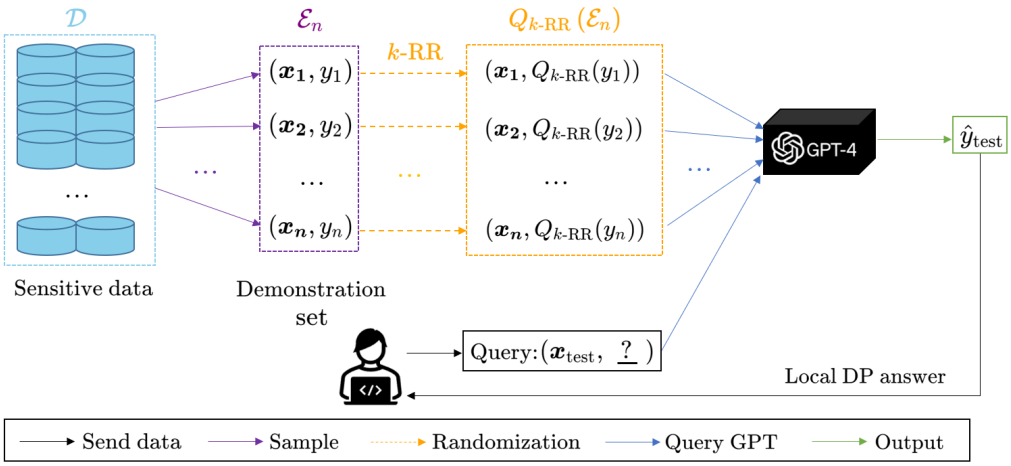

Figure 1: The framework of LDP-ICL: We first sample a few input-label pairs from the original private database to form the demonstration set. Next we employ the k-ary randomized response mechanism $Q_{k\text{-RR}}$ to perturb the labels and then perform the ICL with a given query $\boldsymbol{x}_{\text{test}}$ prepended by the noisy demonstration set. At the end, the response is returned to the adversary.

Dai et al., 2022). From this formula, we obtain the trade-off between privacy-preservation and utility, which is measured by the accuracy rate of the query answers. When the privacy-preservation gets stronger, i.e., ϵ in k-RR gets smaller, the accuracy becomes smaller. Moreover, we run experiments on several datasets for the classification task and demonstrate the trade-off effects (Figure 3). In order to support our understanding of LDP-ICL, we apply it to a *touch-stone* LDP problem: the discrete distribution estimation problem (Kairouz et al., 2016; Wang et al., 2017). We design an algorithm with LDP-ICL to perform distribution estimation of sensitive labels in the original private database (Algorithm 2) and compare the results with the classic Warner's mechanism for the same task (Warner, 1965). Our results show that our algorithm performs better than Warner's mechanism in the privacy-utility trade-off for the high-privacy region.

The rest of the paper is organized as follows. In Section 2, we present the definition of in-context learning. In Section 3, we consider ICL as implicit gradient-descent optimization which is a dual form of Transformers attention mechanism and propose LDP-ICL. And we analyse the trade-off between privacy and utility in the LDP-ICL and further deal with the discrete distribution estimation problem with the analysis. We perform experiments to support our analysis in Section 4 and conclude with related works and further problems in Section 5. Besides, we will add the effects with demonstration sets of different sizes in the extended version, which also include some experiment details and proofs.

2. In-context Learning

In this paper, we focus on in-context learning(ICL) for classification tasks using large language models (LLMs) (Brown et al., 2020a). In-context learning is a paradigm that allows large language models to learn tasks given only a few examples in the form of demonstrations, which is an emergent ability for LLMs. Essentially, it gauges the probability of a prospective answer based on the provided demonstrations, leveraging a well-trained large language model. For a classification task, given a query input text $\boldsymbol{x}_{\text{test}}$ and a candidate answer set $Y = \{y_1, y_2, \ldots, y_M\}$, we need to predict a label $\hat{y}_{\text{test}} \in Y$ conditional on n demonstration examples $\mathcal{E}_n = \{I, s(\boldsymbol{x_1}, y_1), \ldots, s(\boldsymbol{x_n}, y_n)\}$ or $\mathcal{E}_n = \{s(\boldsymbol{x_1}, y_1), \ldots, s(\boldsymbol{x_n}, y_n)\}$, where I is an optional task instruction and $s(\boldsymbol{x_i}, y_i)(1 \le i \le n)$ is an example written in natural language texts according to the task. Formally, given a GPT3.5 model $\mathcal{M}$, we calculate the probability for each answer $y_j : P_{\mathcal{M}}(y_j \mid \mathcal{E}_n, \boldsymbol{x}_{\text{test}})$. Then, the ultimate predicted label $\hat{y}_{\text{test}}$ is given by the candidate answer with the highest probability: $\hat{y}_{\text{test}} = \text{LLM}(\mathcal{E}_n, \boldsymbol{x}_{\text{test}}) = y_{\arg\max_j P_{\mathcal{M}}(y_j|\mathcal{E}_n, \boldsymbol{x}_{\text{test}})}$. For example, we could predict the class label in a binary sentiment classification by comparing the prediction probability of the two labels: 0 and 1. The following are some characteristics which make ICL an important form of learning method: without optimizing any parameters, ICL directly performs predictions on the pretrained language models; by altering the demonstration and templates, it is easier to incorporate human knowledge into LLMs (Wei et al., 2022); ICL is a training-free learning framework and can be easily applied to large-scale real-world tasks.

In this paper, we work with ICL for *large* lan-

guage models, which can override semantic priors from pretraining when presented with in-context demonstrations that contradict priors (Wei et al., 2023). It performs both *task recognition* for identifying tasks and *task learning* for learning new input-label mappings from demonstrations (Pan et al., 2023). So its performance improves consistently with more demonstrations.

3. Locally Differentially Private ICL

This section is our main contribution. First we adapt some previous results in the literature and obtain a formula for the prediction probability of the query answer in ICL by considering ICL as a dual form of gradient-descent-based optimization. Next we formulate locally differentially private ICL and use the above prediction formula to analyze the trade-off between privacy-preservation and prediction accuracy. After that, we design an algorithm with LDP-ICL for the discrete distribution estimation problem and compare with the standard Warner's mechanism (Warner, 1965).

3.1. In-context Learning by Gradient Descent

Mathematically, let $\boldsymbol{W}_0, \Delta\boldsymbol{W} \in \mathbb{R}^{N\times M}$ be the initialized weight and update matrix for a given classification task, respectively, where update is performed across few demonstrations: $\mathcal{E}_n = \{(\boldsymbol{x_i}, \boldsymbol{y_i})\}_{i=1}^n$ comprising input representations $\boldsymbol{x_i} \in \mathbb{R}^N$ and corresponding labels $\boldsymbol{y_i} \in \mathcal{Y} = \{y_1, \cdots, y_M\} \subseteq \mathbb{R}^M$. $\boldsymbol{W}_0\boldsymbol{x}_{\text{test}}$ is the answer of the zero-shot learning, i.e., ICL with *no demonstrations* and hence serves as an important reference. In this paper, we will fix this formalization. According to (Irie et al., 2022b), transformer attention has a dual form of gradient descent. Gradient descent use back-propagation algorithm to calculate $\Delta\boldsymbol{W}$, by summing the outer products of $\{\boldsymbol{x_i}\}_{i=1}^n$ with their corresponding error signals $\mathbf{e}_i \in \mathbb{R}^{N\times M}$

$$\Delta\boldsymbol{W} = \sum_{i=1}^{n} \mathbf{e}_i \otimes \boldsymbol{x}_i = \sum_{i=1}^{n} \mathbf{e}_i \boldsymbol{x}_i^T \tag{1}$$

Then, given a specific query $\boldsymbol{x}_{\text{test}}$, we obtain its prediction

$$\hat{\boldsymbol{y}}_{\text{test}} = (\boldsymbol{W}_0 + \Delta\boldsymbol{W})\, \boldsymbol{x}_{\text{test}} \tag{2}$$

Combining (1) and (2), we derive the dual form of gradient decent

$$\begin{aligned}\hat{\boldsymbol{y}}_{\text{test}} &= \boldsymbol{W}_0\boldsymbol{x}_{\text{test}} + \sum_{i=1}^{n} \mathbf{e}_i \boldsymbol{x}_i^T \boldsymbol{x}_{\text{test}} \\ &= \boldsymbol{W}_0\boldsymbol{x}_{\text{test}} + \sum_{i=1}^{n} \mathbf{e}_i \left(\boldsymbol{x}_i^T \boldsymbol{x}_{\text{test}}\right) \\ &= \boldsymbol{W}_0\boldsymbol{x}_{\text{test}} + \text{LinearAttn}\left(\boldsymbol{E}, \boldsymbol{X}, \boldsymbol{x}_{\text{test}}\right)\end{aligned}$$

where Linear Attention operation is performed over error signal matrix $\boldsymbol{E}$, demonstration set $\boldsymbol{X}$ and query $\boldsymbol{x}_{\text{test}}$ representing values, keys and query, respectively.

We now illustrate that our ICL for classification tasks can be realized through self-attention mechanism *followed by an activation function such as softmax or sigmoid*, also interpretable as an implicit gradient descent step on the cross-entropy(CE) loss. For simplicity and illustration, we focus on binary classification, where the 2-dimensional one-hot label vector can be treated as a real number in $\{0, 1\}$. Given demonstrations: $\mathcal{E}_n = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^n, y_i \in \{0, 1\}$, the binary CE loss measures the dissimilarity between the predicted probability with the true binary labels

$$L\left(\boldsymbol{W}\right) = \sum_{i=1}^{n} \left[y_i \ln\left[\sigma\left(\boldsymbol{W}\boldsymbol{x_i}\right)\right] + \left(1 - y_i\right) \ln\left[1 - \sigma\left(\boldsymbol{W}\boldsymbol{x_i}\right)\right]\right]$$

where $\sigma(z) \triangleq \frac{1}{1+e^{-z}}$ denotes sigmoid function and $\boldsymbol{W} \in \mathbb{R}^N$ is weight matrix. Applying a single gradient descent iteration to the loss function L with learning rate η yields the weight change

$$\Delta\boldsymbol{W} = -\eta \nabla_{\boldsymbol{W}} L\left(\boldsymbol{W}\right) = -\eta \sum_{i=1}^{n} \left(\sigma\left(\boldsymbol{W}\boldsymbol{x_i}\right) - y_i\right) \boldsymbol{x_i}^T \tag{3}$$

Let $\hat{p}_{\text{test}}$ be the prediction probability of the true label for the query $\boldsymbol{x}_{\text{text}}$ in the zero-shot learning. Consequently, this alteration in weights will result in an update in the prediction $\hat{p}_{\text{test}}$ for query $\boldsymbol{x}_{\text{test}}$

$$\begin{aligned}\begin{pmatrix}\boldsymbol{x}_{\text{test}} \\ \hat{p}_{\text{test}}\end{pmatrix} &= \begin{pmatrix}\boldsymbol{x}_{\text{test}} \\ \sigma\left(\boldsymbol{W}_0\boldsymbol{x}_{\text{test}}\right)\end{pmatrix} \\ \leftarrow \begin{pmatrix}\boldsymbol{x}_{\text{test}} \\ \sigma\left(\left(\boldsymbol{W}_0 + \Delta\boldsymbol{W}\right)\boldsymbol{x}_{\text{test}}\right)\end{pmatrix} &= \begin{pmatrix}\boldsymbol{x}_{\text{test}} \\ \sigma\left(\boldsymbol{W}_0\boldsymbol{x}_{\text{test}} + \Delta\boldsymbol{W}\boldsymbol{x}_{\text{test}}\right)\end{pmatrix}\end{aligned} \tag{4}$$

Combining (3) and (4), we rewrite the updated prediction as

$$\begin{aligned}\hat{p}_{\text{test}}^{(\text{u})} &\triangleq \sigma\left(\boldsymbol{W}_0\boldsymbol{x}_{\text{test}} + \Delta\boldsymbol{W}\boldsymbol{x}_{\text{test}}\right) \\ &= \sigma\left(\boldsymbol{W}_0 x_{\text{test}} - \eta \sum_{i=1}^{n} \left(\sigma\left(\boldsymbol{W}_0\boldsymbol{x_i}\right) - y_i\right) \boldsymbol{x_i}^T \cdot \boldsymbol{x}_{\text{test}}\right)\end{aligned} \tag{5}$$

Note that the gradient-descent step is performed on the inner transformer attention mechanism. Following (Zhmoginov et al., 2022), self-attention mechanism can emulate gradient descent on a classification task:

Proposition 3.1 *Given previous token: $\mathcal{E}_n = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^n$, we can construct key, query and value matrices $\boldsymbol{W_k}, \boldsymbol{W_q}, \boldsymbol{W_v}$ as well as the projection matrix $\boldsymbol{P}$ such that a 1-head linear attention operation on the matrix $X := [\mathcal{E}_n, \boldsymbol{x}_{\mathbf{test}}]$ followed by sigmoid(or softmax) yields the same results $\hat{p}_{\text{test}}^{(\text{u})}$ as induced by gradient descent*

$$\hat{p}_{\text{test}}^{(\text{u})} = \sigma(\boldsymbol{W}_0\boldsymbol{x}_{test} + \boldsymbol{P}\,\text{LinearAttn}(\boldsymbol{W_v}X, \boldsymbol{W_k}X, \boldsymbol{W_q}X))$$

Proof. For simplicity, we don't specify the sizes of different matrices. The context will determine the sizes.

We define matrix P and operator σ^- by

$$P = \eta I, \text{ and } \sigma^-(\begin{bmatrix} \boldsymbol{W}_0 x_i \\ y_i \end{bmatrix}) = \begin{bmatrix} 0 \\ \sigma(\boldsymbol{W}_0 x_i) - y_i \end{bmatrix}$$

where I is the identity matrix. Note that σ^- is just a sigmoid function followed by a subtraction. Define

$$\boldsymbol{W}_K = \boldsymbol{W}_Q = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}, \boldsymbol{W}_V = \begin{bmatrix} \boldsymbol{W}_0 & 0 \\ 0 & I \end{bmatrix}$$

Consider

$$P = \sum_{i=1}^{n} \left[\sigma^- \left(\begin{bmatrix} \boldsymbol{W}_0 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \right) \otimes \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \right) \right] \cdot \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_{\text{test}} \\ p_{\text{test}} \end{bmatrix} \right)$$

We can compute the above expression and obtain that it is equal to $\eta \sum_{i=1}^{n} (\sigma(\boldsymbol{W}_0 x_i) - y_i)) \boldsymbol{x_i}^T \boldsymbol{x}_{\text{test}}$, which is just $-\Delta \boldsymbol{W} x_{\text{test}}$. Then the matrices $\boldsymbol{W_v}, \boldsymbol{W_k}$ and $\boldsymbol{W_q}$ can be constructed from σ^- and the above matrices $\boldsymbol{W}_V$, $\boldsymbol{W}_K$ and $\boldsymbol{W}_Q$ respectively.

QED

This proposition can explain well why the performance of ICL for classification improves with more demonstrations with *true* labels. We see from the formula Eq. (5) that any demonstration with true label will increase the prediction probability of the query $\boldsymbol{x}_{\text{test}}$ and any exemplary with false label will decrease the prediction probability. For the binary classification, when $\boldsymbol{x}_i^T \boldsymbol{x}_{\text{test}} > 0$ and $y_i = 1$, then $(\sigma(\boldsymbol{W}_0 \boldsymbol{x_i}) - y_i) \boldsymbol{x_i}^T \cdot \boldsymbol{x}_{\text{test}}) < 0$. Since σ is an increasing function, the demonstration $(\boldsymbol{x}_i, y_i)$ contribute to increase the prediction probability. Other cases can be analyzed similarly. This may explain well the emergent ability of *task learning* of the LLMs, especially the increasing ability of the in-context learning with more exemplaries with true labels.

In the following, we will use Eq. (5) to formulate our following framework for ICL with local differential privacy.

3.2. Locally Differentially Private ICL

Here we describe a threat model and emphasize the importance of local differential privacy in the preservation of individual privacy. In this model, an organization owns a fully private database for some specific task (for example, presidential voting data, the school students' health records) and hosts large language models (LLMs) via an API endpoint, allowing users to query the LLM for answers based on the private data. Sometimes, we assume that LLMs are frozen without any update of parameters (Panda et al., 2023; Duan et al., 2023). We know that, in this scenario, privacy leakage occurs under a canonical private attack called membership inference attack (MIA) which assesses whether a data point is used in the prompts appended to the inputs of a trained LLM (Duan et al., 2023). The above formula Eq. (5) explains well the membership inference attack. Given a query $q = (\boldsymbol{x}_{\text{test}}, ?)$, an adversary tries to distinguish whether it is within an demonstration set $\mathcal{E}_n = \{(\boldsymbol{x_1}, y_1), \cdots, (\boldsymbol{x_n}, y_n)\}$. Without loss of generality, we assume that $\boldsymbol{x}_{\text{test}} = \boldsymbol{x_1} \in \mathcal{X}_n$ where $\mathcal{X}_n = \{\boldsymbol{x} : (\boldsymbol{x}, y) \in \mathcal{E}_n \text{ for some } y\}$, i.e., $\boldsymbol{x}_{\text{test}}$ is within the demonstration set $\mathcal{E}_n$. Let $\mathcal{E}'_n := \mathcal{E}_n \setminus \{(\boldsymbol{x_1}, y_1)\} \cup \{(\boldsymbol{x'}, y')\}$ where $(\boldsymbol{x'}, y') \notin \mathcal{E}_n$. In particular, Since $\boldsymbol{x}_{\text{test}} \neq \boldsymbol{x'}$, the similarity $\boldsymbol{x}_{\text{test}}^{\boldsymbol{T}} \boldsymbol{x}_{\text{test}}$ is usually much bigger than the similarity $\boldsymbol{x}_{\text{test}}^{\boldsymbol{T}} \boldsymbol{x'}$. It follows from Eq. (5) that $P(y_{\text{test}} | \mathcal{E}_n, \boldsymbol{x}_{\text{test}}) > P(y_{\text{test}} | \mathcal{E}'_n, \boldsymbol{x}_{\text{test}})$. In other words, the prediction probability of the answer y_{test} to the query $\boldsymbol{x}_{\text{test}}$ in the in-context learning with the demonstration set $\mathcal{E}_n$ should be larger than the prediction of the answer to the same query with the demonstration set $\mathcal{E}'_n$. So the adversary may easily use the query $\boldsymbol{x}_{\text{test}}$ to distinguish $\mathcal{E}_n$ and $\mathcal{E}'_n$ especially when n is small, and hence distinguish between membership and non-membership. However, when n gets larger (say 32), the difference between these two probabilities $P(y_{\text{test}} | \mathcal{E}_n, \boldsymbol{x}_{\text{test}})$ and $P(y_{\text{test}} | \mathcal{E}'_n, \boldsymbol{x}_{\text{test}})$ is relatively very small. Then it is not easy to distinguish between membership and non-membership.

Moreover, LLMs are shown to memorize individual data from the original training data and to retain users' data from smaller private datasets used to fine-tune them for downstream tasks (Mireshghallah et al., 2022; Zhang et al., 2021; Ippolito et al., 2022; McCoy et al., 2023). If prompts contain sensitive information, the LLM might expose privacy during queries as in Samsung privacy leakage (Mitchell, 2023; Duan et al., 2023). In this case, we may regard LLMs as an *untrusted* aggregator in in-context learning.

Our goal is to employ LLMs to provide accurate answers to different queries from users but protect the privacy of individual data in the database. In this paper, we focus on the scenarios where the LLMs are untrusted in privacy and the *labels* in in-context learning are sensitive information. We employ local differential privacy for these settings. Let $\mathcal{A}$ be the *private* input alphabet set and $\mathcal{O}$ be a finite output alphabet set.

Definition 3.2 A randomized mechanism Q from $\mathcal{A}$ to $\mathcal{O}$ is called ϵ-*locally differentially private* if, for any two inputs a and a', and any output event $O \subseteq \mathcal{O}$, the

following inequalities holds

$$Q(O|a) \leq e^{\epsilon} Q(O|a') \tag{6}$$

◁

It formulates the privacy requirement that, by observing the same outcome O, an adversary cannot reliably distinguish whether the conditioned input is a or a'. So the privacy in the input alphabet is preserved. When the privacy index ϵ is smaller, it is more difficult for the adversary to tell the two inputs apart and hence the privacy-preserving is better. Differential privacy (central or local) satisfies two important properties that are crucial for the practical uses of DP mechanisms. The first one is called *composition*. Multiple DP mechanisms can be adaptively composed and applied to the same dataset. The second is called *postprocessing*. If a mechanism is differentially private, then any post-processing applied to the output of that mechanism is also differentially private.

In particular, we use the k-ary randomized response mechanism (k-RR for short) to protect the privacy with labels (Kairouz et al., 2016; Wang et al., 2017). Let $\mathcal{Y} = \{y_1, \cdots, y_M\}$ be the label set and $\mathcal{E}_n := \{(\boldsymbol{x_1}, y_1), \cdots, (\boldsymbol{x_n}, y_n)\}$ be the set of demonstration examples where $y_1, \cdots, y_n \in \mathcal{Y}$. The *$k$-ary randomized response* on the label set $\mathcal{Y}$ is a randomized mechanism which maps $\mathcal{Y}$ stochastically to itself as follows:

$$Q_{k\text{-RR}}(y'|y) = \frac{1}{M - 1 + e^{\epsilon}} \begin{cases} e^{\epsilon} & \text{if } y' = y, \\ 1 & \text{if } y' \neq y. \end{cases}$$

Note that here $k = M$. In $\mathcal{E}_n$, y_i is considered to be the *true* label of the input $\boldsymbol{x_i}$ $(1 \leq i \leq n)$. If we apply the k-RR mechanism $Q_{k\text{-RR}}$ to protect the privacy in labels, we obtain the perturbed demonstration set $Q_{k\text{-RR}}(\mathcal{E}_n) = \{(\boldsymbol{x_1}, Q_{k\text{-RR}}(y_1)), \cdots, (\boldsymbol{x_n}, Q_{k\text{-RR}}(y_n)\}$ where $Q_{k\text{-RR}}(y_i)(1 \leq i \leq n)$ is a perturbation of the true label y_i. When $k = 2$, the k-RR mechanism is just the well-known Warner's mechanism. In this paper, since we mainly focus on the binary classification problem in in-context learning, we use Warner's mechanism to protect the privacy in labels. For simplcity, we denote $\mathcal{Y} = \{0, 1\}$. The obfuscation from Warner's mechanism is illustrated in Figure 2.

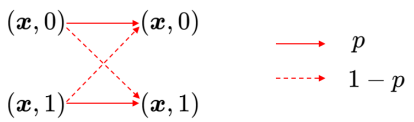

Figure 2: Obfuscation in labels, where $p = \frac{e^{\epsilon}}{e^{\epsilon}+1}$

.

Fix the demonstration set $\mathcal{E}_n := \{(\boldsymbol{x_1}, y_1), \cdots, (\boldsymbol{x_n}, y_n)\}$. We apply the ϵ-LDP k-RR to $\mathcal{E}_n$ and obtain the perturbed demonstration set $Q_{k\text{-RR}}(\mathcal{E}_n)$ for the following ICL. For any given query $\boldsymbol{x}_{\text{test}}$, we perform the ICL by querying the LLM GPT3.5 with the demonstration set $Q_{k\text{-RR}}(\mathcal{E}_n)$. From the above analysis, we obtain the prediction probability of the true label y_{test} for the query $\boldsymbol{x}_{\text{test}}$ as follows:

$$\begin{aligned} &P(y_{\text{test}}|Q_{k\text{-RR}}(\mathcal{E}_n), \boldsymbol{x}_{\text{test}}) \\ &= \sigma(\boldsymbol{W_0 x}_{\text{test}} - \eta \sum_{i=1}^{n} (\sigma(\boldsymbol{W_0 x_i}) - Q_{k\text{-RR}}(y_i)) \boldsymbol{x_i}^T \boldsymbol{x}_{\text{test}}) \end{aligned} \tag{7}$$

The privacy of the labels is preserved from *both* the untrusted LLMs and the observers of the query answers. By abuse of notion, we use $Q_{k\text{-RR}}^{(+\boldsymbol{x}_{\text{test}})}(\mathcal{E}_n)$ for LLM$(Q_{k\text{-RR}}(\mathcal{E}_n), \boldsymbol{x}_{\text{test}}) = \arg\max_{l \in \{0,1\}} P(\hat{y}_{\text{test}} = l | Q_{k\text{-RR}}(\mathcal{E}_n), \boldsymbol{x}_{\text{test}})$ to emphasize that the input of the randomized algorithm $Q_{k\text{-RR}}^{(+\boldsymbol{x}_{\text{test}})}$ is $\mathcal{E}_n$. The privacy of the true labels in the private set $\mathcal{E}_n$ is preserved by the randomized mechanism $Q_{k\text{-RR}}$. Indeed, given a prediction output probability for the label 1, the untrusted LLMs cannot reliably tell in the perturbed demonstration set $Q_{k\text{-RR}}(\mathcal{E}_n)$ which one of 0 and 1 is the true label of the input $\boldsymbol{x_i} (1 \leq i \leq n)$. For example, although we may know that $(\boldsymbol{x}, 0)$ is in $Q_{k\text{-RR}}(\mathcal{E}_n)$, the LLM cannot be certain of the true label of $\boldsymbol{x}$ in the private demonstration set $\mathcal{E}_n$ because $(\boldsymbol{x}, 0)$ can be the output of both $(\boldsymbol{x}, 0)$ and $(\boldsymbol{x}, 1)$ under the randomized mechanism $Q_{k\text{-RR}}$ (Figure 2). Let $\mathcal{E}_n^-$ be $\mathcal{E}_n \backslash \{(\boldsymbol{x_1}, y_1)\} \cup \{(\boldsymbol{x_1}, 1 - y_1)\}$. In other words, $\mathcal{E}_n^-$ is obtained from $\mathcal{E}_n$ by flipping the label of the first element. For any label $l \in \{0, 1\}$, $e^{-\epsilon} P(Q_{k\text{-RR}}^{(+\boldsymbol{x}_{\text{test}})}(\mathcal{E}_n) = l) \leq P(Q_{k\text{-RR}}^{(+\boldsymbol{x}_{\text{test}})}(\mathcal{E}_n^-) = l) \leq e^{\epsilon} P(Q_{k\text{-RR}}^{(+\boldsymbol{x}_{\text{test}})}(\mathcal{E}_n) = l)$. In this sense, the privacy in the true label is preserved by our $Q_{k\text{-RR}}$ against any observer of the query outcomes.

We propose LDP-ICL (Algorithm 1), a new framework for protecting private in-context learning demonstration examples. We randomly sample n demonstration examples from the private dataset $\mathcal{D}$, perturb their labels, and then send these perturbed samples concatenated with the query to a LLM to predict the answer. The algorithm is illustrated in Figure 1.

Algorithm 1: **LDP-ICL**

Input: Private data $\mathcal{D}$, query q, model **LLM**, privacy budget ϵ, number of demonstration examples n.

Output: Model prediction $O(q)$

1: **Subsample** of size n **from** $\mathcal{D}$ and obtain $\mathcal{E}_n$
2: **Perturb** $\mathcal{E}_n$ using k-RR and obtain $Q_{k\text{-RR}}(\mathcal{E}_n)$
3: **Concatenate** query and form $I(q) = Q_{k\text{-RR}}(\mathcal{E}_n) \cup q$
4: Obtain model output $O(q) =$ **LLM**$(I(q))$

There is a *trade-off* between the privacy and utility in LDP-ICL, which is characterized by the above

formula (Eq. (7)). From this formula, we can theoretically compute the expected prediction probability of $Q_{k\text{-RR}}^{(+\boldsymbol{x}_{\text{test}})}(\mathcal{E}_n)$ and its variance. In the private $\mathcal{E}_n$, when more (true) labels y_i are flipped, the terms $\sigma(\boldsymbol{W_0 x_i}) - y_i$ will flip the signs and hence the prediction output probabilities will decrease. When all the labels are flipped, the probability is the smallest. This also implies that, when ϵ in the randomization mechanism $Q_{k\text{-RR}}$ gets close to 0, the accuracy rate will decrease with the output probability in expectation. We run some experiments on several datasets with different ϵ. The results show a trade-off between the privacy index ϵ and the prediction accuracy. If we want a better privacy-preservation for the sensitive labels, then ϵ must get smaller and hence the prediction accuracy decreases. The results are illustrated in Figure 3. The privacy cost accumulates with more demonstrations and more queries according to the composition property of DP.

3.3. Discrete Distribution Estimation

Now we apply the above LDP-ICL to a touchstone problem in DP: discrete distribution estimation problem. Assume that $\mathcal{D}$ is a given private database for some specific classification task (for example, the 2016 US Presidential Election Data) where each individual classification label is sensitive. Now we want to estimate the discrete distribution of different labels, i.e., the proportion of data points with each label in the whole database. For simplicity and illustration, we choose the binary classification. Without loss of generality, let the label set be $\{0, 1\}$, π_0 and π_1 be their unknown prior proportion in the database. Now we select a finite set $\mathcal{D}_n$ of input-label pairs from the database $\mathcal{D}$ whose label distribution is the same as that of the original dataset $\mathcal{D}$. Let $\mathcal{X}_{\mathcal{D}_n} := \{\boldsymbol{x} : (\boldsymbol{x}, y) \in \mathcal{D}_n \text{ for some } y\}$. Now we use the above LDP-ICL to perturb the answer to each query from $\mathcal{X}_{\mathcal{D}_n}$. We choose a demonstration set for each query and perturb the labels in the demonstration set with LDP mechanism. With this noisy demonstration set for in-context learning, the answer of each query is also perturbed with a certain associated probability without affecting the prediction accuracy much. The noisy answer is regarded as an privacy-preserving estimation of the true label. By collecting the noisy answers to all queries from $\mathcal{X}_{\mathcal{D}_n}$, we can estimate π_1 with local differential privacy.

The crux of this approach is to choose the demonstration sets. One possible solution is to choose *a single* set of input-label pairs from $\mathcal{D}$ and its perturbed version as the demonstration set for *all* queries from $\mathcal{X}_{\mathcal{D}_n}$. The problem with this approach is that each perturbed label in the demonstration set would expose to the untrusted LLM *many times* (precisely $|\mathcal{D}_n|$ times) so that the adversary may estimate correctly the true labels in the demonstration set via LLM. Our solution instead is to first partition $\mathcal{D}$ into different subsets of input-label pairs of relatively small size and try to pick up a different subset for each query. The perturbed version of the subset is chosen as the demonstration set for the query. In this way, we can avoid the possibility that a perturbed label might expose to LLM many times and hence we can estimate the proportion π_1 without leaking information about the true labels much. Our approach is detailed in the following Algorithm 2. Specifically, in line 1, we perform a random sampling of size R from $\mathcal{D}$ without replacement to generate the query set. Notably, it is crucial to maintain a consistent proportion of samples of labels 0 and 1 during the sampling process. In line 2, we split the original dataset into l parts of demonstrations, each containing n examples, where $l = |\mathcal{D}|/n$. Finally, we obtain predicted answers for each query and calculate estimated positive rate

$$\widehat{P}_t \triangleq \frac{\sum_{i=1}^{R} \mathbb{1}\left\{O(q^i) = 1\right\}}{R} \tag{8}$$

Algorithm 2: **LDP-ICL for distribution estimation**

Input: Private data $\mathcal{D}$, model **LLM**, privacy budget ϵ, number of demonstration examples n, number of round(queries) R

Output: Proportion estimation

1: **Subsample** of size R **from** $\mathcal{D}$, obtain $\{(\boldsymbol{x}_{\text{test}}^{\boldsymbol{i}}, y_{\text{test}}^i)\}_{i=1}^R$ and construct queries $\{q^i\}_{i=1}^R = \{(\boldsymbol{x}_{\text{test}}^{\boldsymbol{i}}, ?)\}_{i=1}^R$
2: **Partition** $\mathcal{D}$ into classes with size n: $\mathcal{D}_n^1, \ldots, \mathcal{D}_n^l \leftarrow \mathcal{D}$
3: **for** $i \in \{1, \ldots, R\}$ **do**
4: **Perturb** $\mathcal{D}_n^i$ using k-RR and obtain $Q_{k\text{-RR}}(\mathcal{D}_n^i)$
5: **Concatenate** corresponding query and form $I(q^i) = Q_{k\text{-RR}}(\mathcal{D}_n^i) \cup q^i$
6: Obtain i-th model output $O(q^i) =$ **LLM**$(I(q^i))$
7: **end for**
8: Calculate estimated rate (Eq.(8))

For each query $\boldsymbol{x}_{\text{test}}^{\boldsymbol{i}}$, since $(\boldsymbol{x}_{\text{test}}^{\boldsymbol{i}}, y_{\text{test}}^i) \in \mathcal{D}$, y_{test}^i can be regarded as the true label of $\boldsymbol{x}_{\text{test}}^{\boldsymbol{i}}$. So LLM$(Q_{k\text{-RR}}(\mathcal{D}_n^i), \boldsymbol{x}_{\text{test}}^{\boldsymbol{i}})$ is a perturbation of the true label y_{test}^i according to the LDP-ICL. Now we compare this in-context-learning approach with the well-known Warner's method on the same distribution estimation problem. According to Warner's method, for each sample $(\boldsymbol{x}_{\text{test}}^{\boldsymbol{i}}, y_{\text{test}}^i)$, we flip the true label y_{test}^i according to $Q_{2\text{-RR}}$ (Warner's mechanism). By collecting the noisy labels, we empirically estimate the proportion of the labels in the original

private database $\mathcal{D}$. The main difference between these two methods is that the LDP-ICL approach perturbs directly the labels in the *demonstration set* but the Warner's method add noise directly to the *queries*. Generally, the LDP-ICL approach performs better in the high-privacy region (when ϵ is relatively small) because the semantic prior of the LLM add some extra power to the estimation. The experimental results are shown in Figure 4.

4. Experiments

We provide empirical results to demonstrate the effectiveness of our proposed LDP-ICL under two scenarios, classification(Alg.1) and distribution estimation(Alg.2).

4.1. Experimental Setup

4.1.1. Datasets and Model

We evaluate LDP-ICL using four binary classification datasets, all obtained from Hugging Face. SST-2(Socher et al., 2013) and Subj(Pang and Lee, 2004) are for sentiment classification; Ethos(Mollas et al., 2022) is a hate speech detection dataset; and SMS_Spam(Almeida and Hidalgo, 2012) is used for recognizing spam text messages. Dataset size details are provided in Table 1. The training set of each task is considered as private data. Test samples are randomly selected from the validation set in the classification scenario, while they are drawn from the training set in label distribution estimation scenario. We choose the GPT-3.5-turbo model for all tasks which demonstrates strong performance across various natural language tasks while offering a balanced combination of performance and cost-effectiveness.

Task	Training dataset	Validation dataset
SST-2	67349	872
Subj	8000	2000
Ethos	500(998)	498(-)
SMS_Spam	4070(5570)	1500(-)

Table 1: The dataset size for each task. In the classification scenario, we partitioned the initial Ethos dataset (consisting of 998 training examples) and SMS_Spam dataset (comprising 5570 training examples) into separate training and validation sets.

4.1.2. Baselines

In the classification scenario, we compare LDP-ICL with the following baselines:

- **Non-private ICL**, i.e., $\epsilon = \infty$ in LDP-ICL, is equivalent to a non-private n-shot prediction.
- **Zero-Shot Learning (ZSL)** is the same as one-shot learning except that no demonstrations are allowed, and the model is only given a natural language instruction describing the task. ZSL performance enhances with larger model sizes, demonstrating commendable outcomes in GPT-3 175B(Brown et al., 2020b).
- **Flipped-Label ICL(FL-ICL)** is a pattern that flips all class labels in the exemplars, indicating a disagreement between semantic prior knowledge and input-label mappings. The performance accuracy is *inversely* proportional to the ability to learn input–label mappings and override semantic priors (Wei et al., 2023).

For distribution estimation, we compare LDP-ICL with simple **Coin Flipping(CF)** (Warner's) method that randomly flips the true lables in test queries.

4.1.3. Implementation Details

In our experiments, we employ a uniform template for structuring examples across all tasks. Additionally, to mitigate the influence of sensitive settings on ICL performance, we ensure that our demonstration examples meet some constraints. Specific templates for each task and constraints are provided in the Appendix B and C. By default, we set the number of demonstrations to $n = 32$. For classifying unknown test queries, we first tune the random seed for each task to find a set of demonstration examples that achieves the best validation performance based on ICL. Then, the same ordered demonstration set is used for fair LDP-ICL comparisons across various discrete budget levels($\epsilon = \{0, 0.5, 1, 2, 3, 8, \infty\}$). Finally, we utilize evaluation metric to assess model prediction accuracy on a subset of 150 test examples from the validation set. Evaluation performance is averaged over 6 runs under the same parameter configuration. In the distribution estimation scenario, we selected number of queries separately $R = 1000$ for SST-2 dataset and $R = 500$ for Ethos dataset.

4.1.4. Evaluation Metrics

In the classification scenario, we gauge our method's performance by measuring the accuracy between predicted answers with the true ones. For estimating label distribution, we calculate estimated positive rate using Eq.(8).

4.2. Experimental Results

This part consists of LDP-ICLs for classification and distribution estimation.

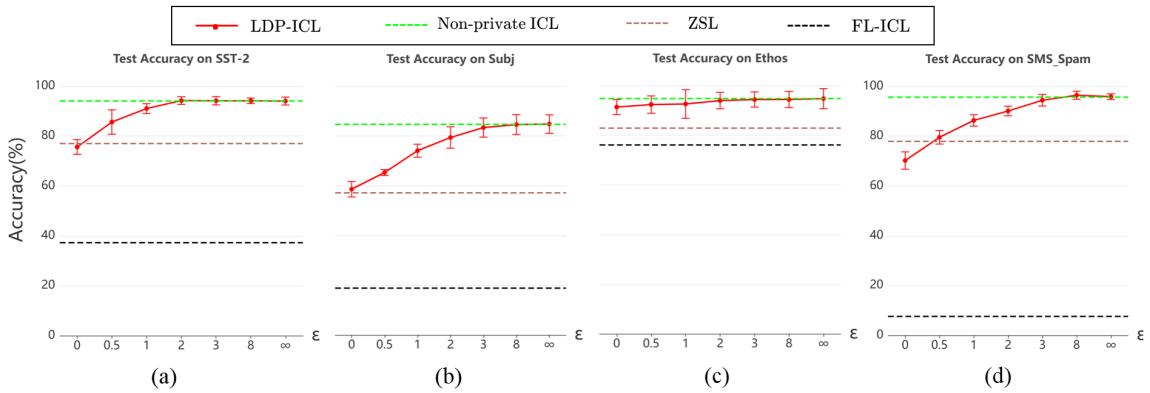

Figure 3: **Classification scenario:** Test performance on (a)SST-2, (b)Subj, (c)Ethos and (d) SMS_Spam

.

4.2.1. LDP-ICL for classification

Figure 3 shows the performance of our LDP-ICL as well as three baselines in all tasks. As can be seen, the consistently lowest accuracy of the baseline FL-ICL method, falling below 50% for all datasets(except Ethos), suggests vulnerability to perturbed class labels. This vulnerability is attributed to GPT-3.5-turbo's emergent task learning capability, enabling it to learn input-label mapping that override established semantic priors(Wei et al., 2023). Comparing against two additional baselines, namely the lower bound ZSL and the upper bound ICL, LDP-ICL demonstrates significant enhancements over ZSL and achieves competitive results similar to non-private ICL when $\epsilon \geq 3$. This observation underscores the beneficial impact of the optimization performed by LDP-ICL on downstream tasks. Furthermore, it's worth highlighting that with $\epsilon \geq 8$, the privacy protection becomes almost negligible, leading to indistinguishable performance between LDP-ICL and the non-private ICL setting. Overall, reducing budgets ϵ strengthens privacy assurance in LDP-ICL but inevitably hampers downstream task performance. Specifically, at $\epsilon = 0$, half of the demonstration example class labels are inverted, yielding performance on par with ZSL in expectation. Conversely, at $\epsilon = \infty$, there is no privacy safeguard and our LDP-ICL degrades to ICL. A more detailed analysis of those tasks reveals that to attain or approach non-private ICL performance, a slightly different budget value is needed. This implies that downstream manufacturers should select the appropriate privacy protection parameter ϵ based on task-specific needs without losing much utility.

4.2.2. LDP-ICL for distribution estimation

Figure 4 presents a comparison between the performance of LDP-ICL and CF. The results reveal that our LDP-ICL estimation aligns more closely with the true proportion and maintains a higher level of stability especially in cases of smaller ϵ, demonstrating better utility. Since we typically prefer smaller budget, which indicates stronger privacy, LDP-ICL outperforms CF in terms of utility.

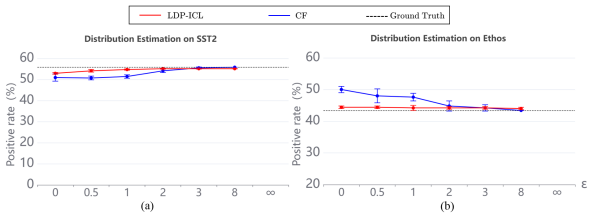

Figure 4: **Distribution estimation scenario:** Estimation results on (a)SST-2 and (b)Ethos.

.

Our initial analysis indicates that for a given privacy parameter value, a higher quantity of examples leads to a higher count of flipped examples, which implies a more powerful task-learning ability and hence a less accurate prediction rate.

4.2.3. Comparison Experiments

We have performed comparison experiments with other three representative privacy-preserving methods: DP-SGD, DP-ICL and PromptPATE and their comparison results are listed in Table 2. The results have demonstrated that locally differentially private ICL also can reach the utility level of other privacy-preserving methods.

4.2.4. Ablation study

Our intuition for choosing demonstration examples was to assess whether a model can learn input-label mappings and override semantic priors. A performance below 50% accuracy in FL-ICL indicates the model's ability to achieve this(Wei et al., 2023).

Model	Method	$\epsilon = 3$	$\epsilon = 8$	$\epsilon = \infty$
RoBERTa -large	DP-SGD	93.04	93.81	96.2
	DP-SGD	94.6	94.7	95.5
	DP-SGD	94.23	94.87	96.2
RoBERTa -base	promptPATE	86.35	92.32	93.23
GPT-3 Babbage	DP-ICL($n = 4$)	95.8	95.92	96.05
	DP-ICL($n = 16$)	91.64	96.32	96.13
GPT-3.5 Turbo	LDP-ICL($n = 16$)	94.45	94.9	95.77
	LDP-ICL($n = 32$)	94.11	94.12	94.12

Table 2: Performance comparison of DP-SGD, promptPATE, DP-ICL and LDP-ICL under various privacy budgets.

Dataset	Number of demonstration examples				
	4	8	16	32	64
SST-2	55	44	33	31	32
Subj	93	78	64	38	39

Table 3: Performance of FL-ICL over number of demonstration examples on SST-2 and Subj.

Table 3 presents the performance for the selection of $n = 32$ demonstration examples, which indicates that the accuracy rate falls below 50% and hence show LLM's capability to learn input-label mappings and override semantic priors. Additionally, we carried out ablation studies to analyze how varying the quantity of demonstration examples affects the sentiment analysis performance for the SST-2 task. The analysis was conducted under three demonstration number cases: $n = 16, 32, 64$.

As depicted in Figure 5, we find that these three curves exhibit an identical trend of change regardless of the variation in example quantities. A variation is observed when the privacy parameter falls within the range of 0 to 0.5: with an increase in the number of examples, accuracy diminishes. Our initial analysis indicates that for a given privacy parameter value, a higher quantity of examples leads to a higher count of flipped examples, which implies a more powerful task-learning ability and hence a less accurate prediction rate. Drawing on the preceding formula (Eq. (5)) in the main text), the heightened presence of flipped examples corresponds to a more pronounced influence on the accuracy.

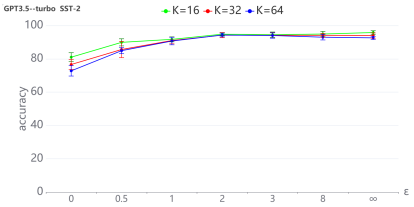

Figure 5: Performance across numbers of the examples

5. Related Works and Conclusion

In this paper, following the tradition in DP (Dinur and Nissim, 2003) literature, we treat the inputs in the input-label pairs as identifiers and hence non-sensitive but regard *only the labels as sensitive*. In this setting, we are *the first* to study the locally differentially private ICL. Our privacy-preserving of labels is different from the so-called label differential privacy (Ghazi et al., 2021), which is essentially central DP. There is a rich literature on privacy-preserving ICL. Here we only discuss some closely related to our work. There are some works which work on the privacy-preservation for ICL but mainly focus on the central DP that assume that the curator is trusted (Duan et al., 2023; Panda et al., 2023). In some sense, our approach is similar to PromptPATE in (Duan et al., 2023) that both use noisy prompts prepended to a query to perform in-context learning. The main difference is the method to add noises. In PromptPATE, the noise is added to the ensembled result in a central way while in our LDP-ICL, we add noise locally to the labels in prompts. In (Yu et al., 2021; Li et al., 2021), they deal with the DP fine tuning of the parameters of the LLMs. In contrast, we regard LLMs frozen and the in-context learning proceeds without modifications of the parameters. In (Li et al., 2023), they use text-to-text privatization while our work focuses on only the privatization of the labels. We adapt the ideas of ICL as Transformer attention mechanism by a dual implicit gradient descent optimization from (Dai et al., 2022; Von Oswald et al., 2023; Irie et al., 2022a). But those papers mainly deal with linear regression problem while we work on the classification problem. Proposition 1 in our paper is based on Appendix A in (Zhmoginov et al., 2022). It this paper, our experiments are run on some common datasets for classification which are not privacy-sensitive (probably Ethos is an exception). In the future, we will try a privacy-sensitive synthetic dataset. In this paper, the perturbation is on the labels only, which is a quite limited case. We plan to employ LDP for more general cases of demonstrations in ICL.

The selection of demonstrations is an important issue (Zhang et al., 2022; Rubin et al., 2022; Zhao et al., 2021; Dong et al., 2022) that we have not addressed yet in this paper. From our formula (Eq. (7)), we know that a good selection of demonstrations can improve the trade-off between privacy and utility of LDP-ICL. We would like to find an optimal adaptive selection algorithm for our LDP-ICL. In this paper, we treat only labels as sensitive. We also will privatize the input sentences or words with local differential privacy (Du et al., 2023; Li et al., 2023; Yue et al., 2021)

6. References

Tiago Almeida and Jos Hidalgo. 2012. SMS Spam Collection. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5CC84.

Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raf. 2023. Emergent and predictable memorization in large language models. *arXiv preprint arXiv:2304.11158*.

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

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020b. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. 2022. Why can gpt learn in-context? language models secretly perform gradient descent as meta optimizers. *arXiv preprint arXiv:2212.10559*.

Irit Dinur and Kobbi Nissim. 2003. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 202–210.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Minxin Du, Xiang Yue, Sherman SM Chow, and Huan Sun. 2023. Sanitizing sentence embeddings (and labels) for local differential privacy. In *Proceedings of the ACM Web Conference 2023*, pages 2349–2359.

Haonan Duan, Adam Dziedzic, Nicolas Papernot, and Franziska Boenisch. 2023. Flocks of stochastic parrots: Differentially private prompt learning for large language models. *arXiv preprint arXiv:2305.15594*.

John C Duchi, Michael I Jordan, and Martin J Wainwright. 2013. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 429–438. IEEE.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer.

Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, and Chiyuan Zhang. 2021. Deep learning with label differential privacy. *Advances in neural information processing systems*, 34:27131–27145.

Daphne Ippolito, Florian Tramèr, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher A Choquette-Choo, and Nicholas Carlini. 2022. Preventing verbatim memorization in language models gives a false sense of privacy. *arXiv preprint arXiv:2210.17546*.

Kazuki Irie, Róbert Csordás, and Jürgen Schmidhuber. 2022a. The dual form of neural networks revisited: Connecting test time predictions to training patterns via spotlights of attention. In *International Conference on Machine Learning*, pages 9639–9659. PMLR.

Kazuki Irie, Róbert Csordás, and Jürgen Schmidhuber. 2022b. The dual form of neural networks revisited: Connecting test time predictions to training patterns via spotlights of attention. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 9639–9659. PMLR.

Peter Kairouz, Keith Bonawitz, and Daniel Ramage. 2016. Discrete distribution estimation under local privacy. In *International Conference on Machine Learning*, pages 2436–2444. PMLR.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2011. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826.

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. 2021. Large language models can be strong differentially private learners. In *International Conference on Learning Representations*.

Yansong Li, Zhixing Tan, and Yang Liu. 2023. Privacy-preserving prompt tuning for large language model services. *arXiv preprint arXiv:2305.06212*.

R Thomas McCoy, Paul Smolensky, Tal Linzen, Jianfeng Gao, and Asli Celikyilmaz. 2023. How much do language models copy from their training data? evaluating linguistic novelty in text generation using raven. *Transactions of the Association for Computational Linguistics*, 11:652–670.

Fatemehsadat Mireshghallah, Archit Uniyal, Tianhao Wang, David Evans, and Taylor Berg-Kirkpatrick. 2022. Memorization in nlp fine-tuning methods. *arXiv preprint arXiv:2205.12506*.

Robin Mitchell. 2023. Samsung fab data leak: How chatgpt exposed sensitive information. *electropages*.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. 2022. ETHOS: a multi-label hate speech detection dataset. *Complex & Intelligent Systems*.

Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. 2023. What in-context learning" learns" in-context: Disentangling task recognition and task learning. *arXiv preprint arXiv:2305.09731*.

Ashwinee Panda, Tong Wu, Jiachen T Wang, and Prateek Mittal. 2023. Differentially private in-context learning. *arXiv preprint arXiv:2305.01639*.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.

Fábio Perez and Ian Ribeiro. 2022. Ignore previous prompt: Attack techniques for language models. In *NeurIPS ML Safety Workshop*.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642. ACL.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR.

Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. 2017. Locally differentially private protocols for frequency estimation. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 729–745.

Stanley L Warner. 1965. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. 2023. Larger language models do in-context learning differently.

Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. 2021. Differentially private fine-tuning of language models. In *International Conference on Learning Representations*.

Xiang Yue, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman SM Chow. 2021. Differential privacy for text analytics via natural text sanitization. *arXiv preprint arXiv:2106.01221*.

Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. 2021. Counterfactual memorization in neural language models. *arXiv preprint arXiv:2112.12938*.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

Andrey Zhmoginov, Mark Sandler, and Maksym Vladymyrov. 2022. Hypertransformer: Model generation for supervised and semi-supervised few-shot learning. In *International Conference on Machine Learning*, pages 27075–27098. PMLR.