Privacy Preserving Prompt Engineering: A Survey

KENNEDY EDEMACU, Electrical Engineering and Computer Science, University of Arkansas, FAYETTEVILLE, United States and Computer Science, College of Staten Island, Staten Island, USA
XINTAO WU, Electrical Engineering and Computer Science, University of Arkansas Fayetteville, Fayetteville, United States

Pre-trained language models (PLMs) have demonstrated significant proficiency in solving a wide range of general natural language processing (NLP) tasks. Researchers have observed a direct correlation between the performance of these models and their sizes. As a result, the sizes of these models have notably expanded in recent years, persuading researchers to adopt the term *large language models* (LLMs) to characterize the larger-sized PLMs. The size expansion comes with a distinct capability called *in-context learning* (ICL), which represents a special form of prompting and allows the models to be utilized through the presentation of demonstration examples without modifications to the model parameters. Although interesting, privacy concerns have become a major obstacle in its widespread usage. Multiple studies have examined the privacy risks linked to ICL and prompting in general, and have devised techniques to alleviate these risks. Thus, there is a necessity to organize these mitigation techniques for the benefit of the community. In this survey, we provide a systematic overview of the privacy protection methods employed during ICL and prompting in general. We review, analyze, and compare different methods under this paradigm. Furthermore, we provide a summary of the resources accessible for the development of these frameworks. Finally, we discuss the limitations of these frameworks and offer a detailed examination of the promising areas that necessitate further exploration.

CCS Concepts: • **Security and privacy → Privacy protections**; • **Computing methodologies → Machine learning;**

Additional Key Words and Phrases: Pre-trained language models, large language models, prompting, in-context learning

ACM Reference Format:
Kennedy Edemacu and Xintao Wu. 2025. Privacy Preserving Prompt Engineering: A Survey. *ACM Comput. Surv.* 57, 10, Article 255 (May 2025), 36 pages. https://doi.org/10.1145/3729219

1 Introduction

The recent advancements in **pre-trained language models (PLMs)** have demonstrated significant capabilities across a wide array of **natural language processing (NLP)** tasks such as text classification, question answering, sentiment analysis, information retrieval, and summarization [10, 76, 155, 163]. A number of these models have recently been introduced and are consistently

This work was supported in part by the National Science Foundation under awards 1920920 and 1946391.
Authors' Contact Information: Kennedy Edemacu, Electrical Engineering and Computer Science, University of Arkansas, FAYETTEVILLE, Arkansas, United States and Computer Science, College of Staten Island, Staten Island, New York, USA; e-mail: edemacu.kennedy@gmail.com; Xintao Wu, Electrical Engineering and Computer Science, University of Arkansas Fayetteville, Fayetteville, Arkansas, United States; e-mail: xintaowu@uark.edu.

ACM 0360-0300/2025/05-ART255
https://doi.org/10.1145/3729219

gaining considerable popularity. For example, the number of users for OpenAI's ChatGPT [2] has surpassed 180 million [33]. Examples of other common advanced models are Meta's LLaMA [129], OPT [164], OPT-IML [59], BigScience's BLOOM [143], BLOOMZ [92], and Databricks' Dolly [22].

Generally, these models are huge with parameter sizes of hundreds of billions, and require enormous amounts of computational resources for their training and storage. The term **large language models (LLMs)** is employed to delineate these large-sized PLMs [113, 125, 140]. Furthermore, these models are primarily pre-trained using diverse open-text resources sourced from the web, books, and Wikipedia, among others. For the rest of this work, we shall use the terms *PLMs* and *LLMs* interchangeably. While general-purpose LLMs have proven adept at comprehending and solving general NLP tasks, they occasionally demand greater depth and nuance in addressing domain-specific tasks and adapting to specific objectives. Fine-tuning techniques that adjust learnable model parameters have been proposed to tailor the models for domain-specific downstream tasks and to adapt them to specific goals [53, 55, 74]. However, challenges such as high computational resource demand, risks of overfitting, concerns about catastrophic forgetting, and model stability are commonly associated with the process of model fine-tuning [167]. Consequently, caution must be exercised when performing fine-tuning. Other model adaptation techniques include instruction tuning, prompt tuning, alignment tuning, and so forth [166].

An emerging capability of LLMs is known as *prompting*. Through prompting, an LLM can generate anticipated outputs for a given query when provided with natural language instructions and/or demonstration examples, without necessitating updates to the model parameters. The simplest type of prompt is a direct prompt (also known as zero-shot) where users phrase the instruction as a question and provide no examples to the LLM. **In-context learning (ICL)** is another form of prompting proposed along with GPT-3 [10] and includes a few demonstration examples in the prompt. ICL serves as an efficient and effective method for leveraging pre-trained or adapted LLMs to address a variety of downstream tasks without the need to modify the model parameters for each task.

However, the aforementioned LLM utilization technique invariably entails the use of data that may be deemed private and harbor sensitive information. For example, consider using ICL to predict if an individual earns at least $50,000 in a year. To help the LLM form a context and make a better prediction, it is prompted with a demonstration example that might contain sensitive information such as age, salary, and SSN. Sensitive information of this nature could potentially be accessed by either an untrusted LLM server or an adversarial entity capable of bypassing the API provided by the LLM service provider. In addition, vulnerabilities in Redis client open source library have previously led to incidents such as ChatGPT leaking users' chat history [97]. Similarly, the aforementioned approach is vulnerable to inference-based privacy violations, where LLMs might deduce sensitive information from the unstructured text provided to them during inference. Recently, Staab et al. [120] studied the capabilities of pre-trained LLMs to infer personal attributes from text and showed that common mitigations such as text anonymization and model alignment are currently ineffective at protecting user privacy against LLM inference. While privacy challenges such as training or fine-tuning data memorization, and their subsequent recovery through model inversion and **membership inference attacks (MIAs)** [90, 135, 162], have been noticed [31], they are fundamentally distinct from the privacy challenges posed by ICL. Therefore, addressing the privacy challenges associated with ICL specifically, as well as prompting in general, is a matter of urgency.

We focus on privacy concerns when utilizing LLMs with users' sensitive data incorporated into prompts. Progresses dedicated to mitigating these privacy challenges have been made [30, 31, 50, 124, 146]. These studies have adopted various privacy protection mechanisms such as **differential privacy (DP)**, sanitization, lattice, encryption, and ensembling in designing their

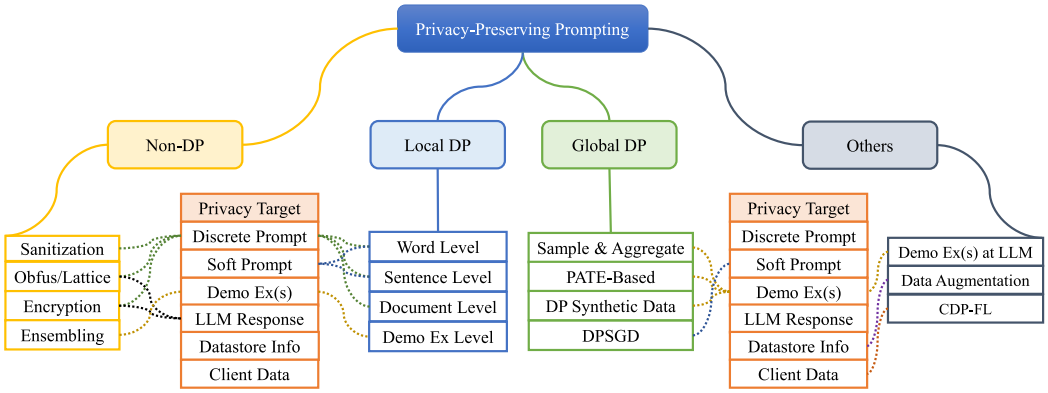

Fig. 1. The layout of privacy mechanisms employed for privacy-preserving prompting. Each privacy mechanism protects at least one privacy target. We elaborate on this by creating links between the mechanisms and the privacy targets. Demo Ex(s) denotes demonstration example(s), Obfus denotes obfuscation, and CDP-FL denotes client data protection via federated learning.

privacy protection frameworks. In this survey, we review the generic approaches that preserve privacy during prompting in general. We characterize them by their privacy models, summarize them, and draw links between them. Figure 1 provides a concise overview of the categorization of privacy-preserving prompting. We classify most privacy frameworks into four main categories: non-DP, **local DP (LDP)**, **global DP (GDP)**, and other scenarios. Within each category, we specify privacy mechanisms/definitions and emphasize their respective privacy objectives.

From a broader perspective, privacy protection for LLMs can be categorized into two families: protecting personal privacy in the pre-training or fine-tuning corpus, and protecting sensitive information in a user's prompt input data. There have been a number of surveys focusing on privacy issues in PLMs. Our primary focus in this article is on the privacy protection aspects during ICL in specific and prompting in general. The selection of the studies considered in this work was guided by this question: Which privacy mechanisms are most suitable for preserving the privacy of text data? Besides this, we added other questions to obtain relevant answers on applying the selected mechanisms in prompting LLMs. We searched using Google Scholar and included studies between 2020 and 2024 for our analysis because ICL [10] was introduced along with GPT-3 in 2020. In particular, we included papers published on top-tier CS conferences (i.e., those included in CSRankings) and also manually chose related and potentially high-impact papers from arXiv.

Similar surveys exist in the literature. Hu et al. [54] specifically explored privacy preservation in NLP using DP. Their work mainly focused on how to privately train and release a language model without leaking information about training and fine-tuning data. Yao et al. [152] investigated how LLMs could alter the cybersecurity world. The authors explored the merits, risks, and vulnerabilities of using LLMs. Neel and Chang [94] presented privacy issues in LLMs from training to inference, with limited coverage of the prompting methods. Sun et al. [122] thoroughly investigated the trustworthiness of LLMs across different dimensions including truthfulness, safety, fairness, robustness, privacy, and machine ethics. However, the privacy aspect centers on assessing privacy awareness within LLMs and the potential disclosure of private information from the training dataset in the responses generated by LLMs. Das et al. [24] examined the security and privacy challenges associated with LLMs, encompassing considerations for both training data and users. The study evaluated the vulnerabilities of LLMs, explored emerging security and privacy threats targeting LLMs, and provided a review of potential defense mechanisms, although with

limited coverage for prompting privacy. In contrast, our work holistically focuses on a review of the literature on privacy protection in LLM usage with prompting methods. To the best of our knowledge, we are the first to systematically organize such literature.

Article Organization. We organize the rest of the article as follows. Section 2 presents the preliminary section, introducing language models and privacy models. Section 3 presents the review of the efforts for privacy-preserving prompting. In Section 4, we present the available resources for privacy-preserving prompting. Section 5 presents the limitations of the existing frameworks and future prospects. We conclude the article in Section 6.

2 Preliminaries

2.1 Language Models

2.1.1 Pre-Trained Language Models. With the abundance of extensive unlabeled corpora and the rise of Transformers [133], the research community has crafted universal PLMs employing self-supervised learning methods [80]. The term *LLMs* is introduced to distinguish PLMs characterized by vast parameter sizes, typically comprising tens or hundreds of billions of parameters. Through pre-training on extensive corpora, LLMs develop the capability to comprehend and generate natural languages proficiently.

Generally, these language models are designed to output the probability distribution of a token sequence [166]. Given a text sequence $s = \langle w_1, w_2, \ldots, w_n \rangle$, where $w_i \in \mathcal{V}$ and $\mathcal{V}$ denotes the vocabulary space. The likelihood of s using the chain rule of probability is $\Pr(s) = \prod_{i=1}^{n} \Pr(w_i | w_{<i})$. An autoregressive language model takes a sequence of tokens $\langle w_1, w_2, \ldots, w_{i-1} \rangle$ as input and outputs a probability distribution for the next token w_i as $\Pr(w_i | w_{<i})$. Choosing the next token can be achieved through various techniques, including greedy search, beam search, top-k sampling, and nucleus sampling, among others. Several surveys [26, 47, 79, 105, 107, 166, 167] have extensively covered the technical details of these LLMs.

Most LLMs, such as GPT-4 from OpenAI and Germini from Google, only support API-based accesses. Users submit their prompts via LLMs' standard APIs that provide the input/output functions of the models. A limited number of LLMs such as LLaMA from Meta are open source and provide users all details of system architecture, parameters, and code. For these white-box LLMs, users can retrain or fine-tune models locally for their application task. While pre-training equips LLMs with the capacity to solve diverse NLP tasks, research indicates that their proficiency can be tailored to specific tasks with techniques such as fine-tuning, prompt tuning, instruction tuning, and alignment tuning [166, 167]. Our emphasis lies not in tuning/adaptation, but rather in leveraging the LLMs post pre-training and/or adaptation using prompting.

2.1.2 Prompting. A major approach to efficiently and effectively utilize LLMs to solve specific downstream tasks is through designing suitable prompting strategies. A lot can be achieved with well-designed prompts. Manually crafting prompts can be time consuming and prone to errors. Improperly constructed prompts can lead to poor performance. As a result, a series of efforts have been aimed at automating the optimization of *discrete/hard* and *continuous/soft* prompts [74, 115]. A *discrete prompt* typically consists of a sequence of natural language text and can be formulated as

$$LLM(prompt) \rightarrow response, \tag{1}$$

where *prompt* encompasses elements such as task description, input data, contextual information, and prompt style. Those elements are essential for guiding LLMs to generate an appropriate *response*. Formally, we can describe *prompt* as a token sequence $s = \langle w_1, w_2, \ldots, w_l \rangle$ with length l. The optimization process in discrete prompts searches for prompts in the discrete text space.

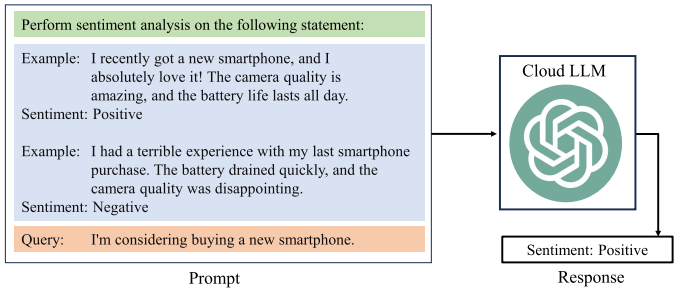

Fig. 2. An illustration of ICL. In ICL, the prompt consists of a task description (light green), demonstration examples (light blue), and a query (light orange).

Common categories of discrete prompt optimization methods include gradient-based, reinforcement learning based, edit-based, and LLM-based approaches [166].

A *continuous prompt* consists of a continuous set of task-specific embeddings. Note that all discrete input tokens to LLMs are internally transformed into continuous input embeddings that the LLM then processes. In the white-box setting, LLMs further support the user's access via submitting the embeddings of the prompt. The formulation can be modified as

$$LLM(embeddings) \rightarrow response. \tag{2}$$

We can choose a word embedding model $\phi : \mathcal{V} \mapsto \mathbb{R}^d$ to derive the prompt embeddings at the token level, $\langle \phi(w_1), \phi(w_2), \ldots, \phi(w_l) \rangle$, or use an encoder $\mathbf{r} = \text{Enc}(s)$ that returns a vector representation $\mathbf{r}$ for the whole prompt. The continuous prompts can also be considered trainable parameters and can be learned with optimization strategies such as employing supervised learning to minimize cross-entropy loss using sufficient downstream task data, and engaging in prompt-based transfer learning [166].

2.1.3 In-Context Learning. ICL exemplifies a form of prompting method [10, 27] and has become a new learning paradigm where LLMs make predictions only based on contexts presented through a few demonstration examples. In ICL, prompts are composed of task descriptions and/or demonstration examples presented in natural language text. The key idea of ICL is to learn from analogy. An illustration of ICL is presented in Figure 2. A query is concatenated to the demonstration examples usually written in natural language templates. Once done, the combination of the task description, demonstration examples, and the query are submitted to a remote LLM (mostly hosted in the cloud). The LLM is expected to learn the patterns hidden in the demonstration and make the prediction directly without conducting parameter updates. This is different from supervised learning requiring a training stage that often uses backward gradients to update model parameters. Using appropriate demonstration examples, the LLM can be steered to correctly answer the query. Let $D_k = \{(x_1, y_1), \ldots, (x_k, y_k)\}$ denote a set of k demonstration examples. We denote $g(x_k, y_k)$ a prompt function (e.g., a template) that transforms the k-th demonstration example into natural language text. For the task description $inst$, a set of demonstration examples D_k, and a test query x_{k+1}, the prediction output $\hat{y}_{k+1}$ from LLMs can be formally formulated as [166]

$$LLM(inst, \underbrace{g(x_1, y_1), \ldots, g(x_k, y_k)}_{\text{demonstration examples}}, g(\underbrace{x_{k+1}}_{\text{query}}, \underbrace{\_\_\_\_}_{\text{answer}})) \rightarrow \hat{y}_{k+1}, \tag{3}$$

where the true answer y_{y+1} is left as a blank to be generated by the LLM. The advantage of this paradigm is that a single trained model can be used to efficiently solve myriad downstream tasks

in an unsupervised manner [79]. The performance of ICL depends on the appropriate construction of prompts—in particular, how to select those k demonstration examples. ICL has an inherent connection with instruction tuning as both utilize natural language to construct the task or instances. The main difference is that instruction tuning needs to fine-tune LLMs for adaption, whereas ICL only prompts LLMs for utilization. We suggest the readers refer to the survey paper of Dong et al. [27] for a comprehensive review of ICL.

Improving ICL can involve implementing strategies such as chain-of-thoughts [140] that incorporates intermediate reasoning steps within prompts, and planning [168] which decomposes complex tasks into sub-tasks and devises plans to address them one by one. These strategies prove beneficial for models equipped with extensive knowledge about the given task. Users may as well enrich their prompts with private external datastores. Such a strategy is referred to as database augmentation and allows the introduction of knowledge from external sources such as training corpus, external data, and unsupervised data [8, 136, 154]. This can help improve the context base within the prompt. Generally, a database augmentation model consists of a retriever and a generator [167]. Here, the retriever can retrieve information from external private sources based on a user query, and the generator combines the retrieved information with instructions to form a prompt.

2.2 Privacy Models

Privacy is a multifaceted term that is often used to refer to a wide and disparate group of related topics, and therefore it is important to clarify the exact interpretations of "privacy" that we consider in this survey. Solove [118] creates a taxonomy of privacy that classifies actions that lead to privacy violations into four different groups: (1) information collection, (2) information processing, (3) information dissemination, and (4) invasion. Depending on the use case and operating conditions, ICL can be classified into the first three categories. First, in many cases, it is not clear to the user who has access to the prompts that they submit to an LLM, which can result in covert surveillance (i.e., information collection) that can bring harm to a user. For example, in the future, we may see criminal cases that use a person's prompt data as incriminating evidence. Second, it is often not clear how the prompt data is aggregated or used in addition to the intended use case of supplementing a query to the LLM (i.e., information processing). For instance, some LLMs do not clearly state that user submitted prompts are used to improve model performance or are sold to third parties (i.e., secondary use). Finally, it has been shown that prompted LLMs (e.g., using ICL) exhibit a high risk to disclose the membership of their private prompt data [32], which can lead to the disclosure of private information that results in harm to the users (i.e., information dissemination). In this work, we aim to discuss strategies proposed to enhance privacy of ICL relative to these three taxonomy groups.

2.2.1 Differential Privacy. DP has been a de facto standard for preserving privacy in myriad machine learning tasks. There are two popular variants of DP: GDP and LDP. *GDP*[35] assumes the existence of a trusted data curator. The curator has access to all individuals' raw data and processes it using a randomized algorithm $\mathcal{A}$.

Definition 2.1 (Global DP). A randomized algorithm $\mathcal{A}$ satisfies (ε, δ)-GDP, if for any two neighboring datasets D and D' which differ one single record, and for any output $O \subseteq \text{Range}(\mathcal{A})$, the following holds:

$$\Pr(\mathcal{A}(D) \in O) \leq exp(\varepsilon) \Pr(\mathcal{A}(D') \in O) + \delta,$$

where ε is the privacy budget and controls the level of privacy guarantee. The smaller the value, the stronger the privacy guarantee (i.e., more noise is added) and vice versa. δ is a small error

probability. If $\delta = 0$, $\mathcal{A}$ is ε-DP. GDP guarantees output of an algorithm be insensitive to the presence or absence of one record in a dataset.

LDP [64] does not require the existence of a trusted central data curator. Individuals locally perturb their data using a randomized algorithm $\mathcal{A}$ before sending them to the curator for analysis.

Definition 2.2 (Local DP). A randomized algorithm $\mathcal{A}$ satisfies ε-LDP, if for any two inputs $x, x' \in \mathcal{X}$, and for any output $O \subseteq \text{Range}(\mathcal{A})$, the following holds:

$$\Pr(\mathcal{A}(x) \in O) \leq exp(\varepsilon) \Pr(\mathcal{A}(x') \in O).$$

Different from GDP, the inequality holds for all elements x and x' instead of all adjacent pairs of the dataset. Essentially, the LDP ensures that an adversary is unable to infer the input values of any target individual from the output values obtained. To achieve the LDP in statistical analysis, mechanisms such as **randomized response (RR)**, histogram encoding, unary encoding, or local hashing are applied during the collection of user data that are categorical in nature [137].

LDP is a strong privacy notion because it aims homogeneous protection over all input pairs—that is, no matter how unrelated two inputs x and x' are, their output distributions must be similar. To improve utility, metric LDP [3] was proposed where the indistinguishability of output distributions is further scaled by the distance between the respective inputs.

Definition 2.3 (Metric LDP). A randomized algorithm $\mathcal{A}$ satisfies ε-metric LDP, if for any two inputs $x, x' \in \mathcal{X}$, and for any output $O \subseteq \text{Range}(\mathcal{A})$, the following holds:

$$\Pr(\mathcal{A}(x) \in O) \leq exp(\varepsilon \cdot d(x, x')) \Pr(\mathcal{A}(x') \in O),$$

where $d(\cdot, \cdot)$ is a distance metric. When $d(\cdot, \cdot) = 1$, metric LDP is equivalent to LDP.

2.2.2 DP Properties. Several important properties of differentially private mechanisms arise from the preceding DP definitions [37]. In the following sections, we describe those commonly used in NLP.

(a) *Post-processing property*: Post-processing an output of a differentially private algorithm cannot reverse its privacy protection. In other words, the output of (ε, δ)-DP mechanism remains (ε, δ)-DP after post-processing.

Proposition 2.4 (Post-Processing Property). *Let $\mathcal{A}(\mathcal{X})$ satisfy (ε, δ)-DP. Then, for any (randomized) algorithm f, $f \circ \mathcal{A}(\mathcal{X})$ satisfies (ε, δ)-DP.*

(b) *Composition property*: The composition property guarantees DP privacy when releasing multiple outputs of DP mechanisms on the same data.

Proposition 2.5 (Composition Property). *Suppose $\mathcal{A}_1, \mathcal{A}_2 \ldots, \mathcal{A}_k$ are each $(\varepsilon_i, \delta_i)$-DP algorithms. An algorithm $\mathcal{A} = \mathcal{A}_1 \circ \mathcal{A}_2 \circ \cdots \circ \mathcal{A}_k$ that runs $\mathcal{A}_i$ sequentially satisfies (ε, δ)-DP where $\varepsilon = \sum_i^k \varepsilon_i$ and $\delta = \sum_i^k \delta_i$.*

(c) *Privacy amplification via subsampling property*: Subsampling further helps to improve sample secrecy by introducing additional randomness.

Proposition 2.6 (Amplification Effect of Sampling [72]). *Suppose $\mathcal{A}$ is (ε, δ)-DP and $\mathcal{B}$ is constructed as follows. Given a dataset $D = \{x_1, x_2, \ldots, x_n\}$, first, we create a sub-sampled dataset D_s. The probability $x_i \in D_s$ is q. Next, we run $\mathcal{A}$ on D_s. Then $\mathcal{B}(D) = \mathcal{A}(D_s)$ is $(\tilde{\varepsilon}, \tilde{\delta})$-DP, where $\tilde{\varepsilon} = \ln(1 + (e^{\varepsilon} - 1)q)$ and $\tilde{\delta} = q\delta$.*

2.2.3 DP Mechanisms. In the following, we introduce several commonly used mechanisms to achieve DP. The mechanisms of achieving DP mainly include the classic approach of adding Laplacian noise [36], the exponential mechanism [86], the sample and aggregate framework [95], the **Private Aggregation of Teacher Ensembles (PATE)** framework [100], the functional perturbation approach [15], and **differentially private stochastic gradient descent (DP-SGD)** [1].

Dwork et al. [36] proved that using the Laplace mechanism can preserve DP by calibrating the standard deviation of the noise according to the sensitivity of the query function. The sensitivity measures the maximum possible change in the function's output when one record in the dataset changes.

Proposition 2.7 (Laplace Mechanism). *Given a dataset D and a query f, a mechanism $\mathcal{A}(D) = f(D) + \boldsymbol{\eta}$ satisfies ε-DP, where $\boldsymbol{\eta}$ is a random vector drawn from $Lap(S_f(D)/\varepsilon)$ where the sensitivity $S_f(D)$ is defined as $S_f(D) = \max_{D,D'} ||f(D) - f(D')||_1$.*

McSherry and Talwar [86] proposed the exponential mechanism to guarantee DP in non-numeric sensitive queries by sampling according to a mapping function instead of adding noise. For a given dataset D and privacy budget ε, the quality function induces a probability distribution over the output domain, from which the outcomes are exponentially chosen. It favors higher scoring classes while guaranteeing ε-DP.

Proposition 2.8 (Exponential Mechanism). *Given a dataset D, let $q : D \rightarrow R$ be a quality function that scores each output class $r \in R$. The sensitivity of this function is defined as*

$$S(q(D,r)) = \max_{D,D',r \in R} ||q(D',r) - q(D,r)||_1. \tag{4}$$

The exponential mechanism $\mathcal{A}$ randomly selects a potential outcome r based on the following probability, then the mechanism $\mathcal{A}^{\epsilon}_{q,S(q)}(D,R)$ is ε-differentially private:

$$\Pr(r \in R \text{ is selected}) \propto \exp\left(\frac{\varepsilon q(D,r)}{2S(q)}\right). \tag{5}$$

Nissim et al. [95] introduced the sample and aggregate framework. It calibrates instance-specific noise based on a smooth sensitivity to achieve rigorous DP. The private database is randomly split into multiple partitions. The arbitrary function f is computed exactly, without noise, independently on each partition. The intermediate outcomes are then combined via a differentially private aggregation mechanism—for example, standard aggregations followed by noise perturbation. Because any single element can affect at most one partition, changing the data of any individual can change at most a single input to the aggregation function.

Papernot et al. [100] developed the PATE framework that incorporates both a private labeled dataset and a public unlabeled dataset and ensures DP by employing a teacher-student knowledge distillation framework. The student model acquires knowledge from the private dataset through knowledge distillation facilitated by the multiple teacher models. Specifically, the private dataset is first randomly divided into m disjoint subsets, each of which is used to train a teacher model. For each record in the public dataset, the label outputs from all teacher models are aggregated. The student model is then trained on the public dataset using the label guidance provided by the aggregated teacher models. To achieve the DP protection of the private dataset, the noisy majority votes as labels are adopted in the classification task.

Chaudhuri et al. [15] proposed an objective perturbation approach by perturbing the objective function which is convex and doubly differentiable. Zhang et al. [159] further proposed a functional mechanism to enforce DP on general optimization-based models, such as linear regression and logistic regression. Abadi et al. [1] proposed a DP-SGD technique, which has been another

popular mechanism to achieve DP in deep learning. The procedure of deep learning model training is to minimize the output of a loss function through numerous **stochastic gradient descent (SGD)** steps. DP-SGD uses a clipping bound on the l_2 norm of the gradient from an individual input, aggregates the clipped updates, and then adds Gaussian noise to the aggregate. The clipping truncation controls the sensitivity of the sum of gradients as the sensitivity of gradients and the scale of the noise would otherwise be unbounded. Abadi et al. [1] further proposed a moment accounting mechanism which calculates the aggregate privacy bound when performing SGD for multiple steps. The moments accountant is tailored to the Gaussian mechanism and computes tighter bounds for the privacy loss compared to the standard composition theorems.

3 Prompting with Privacy Protection

In this section, we systematically organize the privacy preservation methods employed during prompting. Generally, prompts can leak sensitive information and can be accessed by adversarial entities. Drawing from the formulations outlined in Equations (1) through (3), our attention centers on safeguarding the privacy of key components: the prompts (including demonstration examples) and the outputs produced by LLMs. We categorize the noteworthy methods into non-DP, LDP, GDP, and other scenarios in Table 1. We identify six major privacy targets: discrete prompt, soft prompt, demonstration examples, LLM response, datastore information, and client data. For each combination of privacy type and privacy target, we present the appropriate privacy protection mechanisms, such as sanitization, encryption, and DP synthetic data, in Table 1. Each method has its own LLM applicability requirements—that is, black-box, black-box (with access to softmax probabilities), and white-box LLMs. We also include an analysis of the **privacy-utility guarantee (PUG)** for each method based on its scheme design, entities involved, and assumptions made. Most privacy preserving prompt engineering methods covered in this survey are randomization-based or anonymization-based techniques, and the utility guarantee (U) here refers to the method's ability to balance privacy protection with maintaining the utility (or effectiveness) of the learning model. This utility relates to the model's accuracy, generalization ability, or performance in its intended tasks. Randomization-based privacy-preserving methods, such as LDP and GDP, can provide privacy guarantees (P) by introducing randomness into the data or model parameters. However, they do not inherently guarantee utility in the strictest sense. For those methods that aim to achieve a balance between privacy and utility, we label them with the **privacy-utility (P&U)** guarantee. For those methods that only focus on privacy protection or are generally infeasible in practice (e.g., due to high computational cost), we only label them with privacy guarantee (P).

3.1 Non-DP Methods

3.1.1 Sanitization Methods. Data sanitization methods aim to protect the privacy of the prompt in Equation (1). In general, data sanitization techniques aim to identify and eliminate sensitive attributes that contain **personally identifiable information (PII)** from the data [58]. Quite often, a machine learning model is employed to perform the identification of PIIs and other sensitive attributes. Kan et al. [63] proposed to use a local LLM to sanitize privacy-sensitive user inputs before using the sanitized texts for prompting a cloud LLM. The proposed Privacy-Preserving via Text Sanitization (PP-TS) framework consists of three modules: a pre-processing privacy protection module that conducts de-identification, an LLM invocation module, and a post-processing privacy recovery module that recovers the original sensitive information. The local LLM is presented with text examples and rewrites requirements in the form of an instruction. This steers the local LLM to sanitize sensitive attributes in the input. Once the sanitized text is generated, the local LLM checks its reasonability. If it is considered reasonable, then the user employs it to prompt a cloud LLM. Additionally, the original and sanitized versions of the text are

Table 1. Overview of Privacy-Preserving Prompting

Type	Ref-Year	Method	Privacy Target	LLM Applicability	PUG	Task
Non-DP	[63]-2023	Sanitization	Prompt (Equation (1))	BB, BB-probs, WB	U	
	[19]-2023			BB, BB-probs, WB	U	
	[160]-2024			BB, BB-probs, WB	U	
	[31]-2023	Ensembling	Demo Ex(s) (Equation (3))	WB	U	
	[153]-2024	Obfuscation/ Lattice	Prompt & Response (Equation (1))	BB, BB-probs, WB	P	
	[161]-2023			BB-probs, WB	P	
	[77]-2024	Encryption	Prompt & Response (Equation (1))	BB, BB-probs, WB	-	
	[51]-2023			WB	P	
	[48]-2022			WB	P	
	[18]-2022			WB	P	
LDP	[83]-2020	Word Level	Prompt (Equation (1))	BB, BB-probs, WB	P	
	[103]-2021			BB, BB-probs, WB	P	
	[17]-2023			BB, BB-probs, WB	P	
	[128]-2023			BB, BB-probs, WB	P	
	[40]-2020			BB, BB-probs, WB	P	
	[149]-2020			BB, BB-probs, WB	P	
	[14]-2023			BB, BB-probs, WB	P	
	[156]-2021			BB, BB-probs, WB	P&U	
	[171]-2023		Prompt Embeddings (Equation (2))	WB	P	
	[28]-2023	Sentence Level	Prompt (Equation (1))	BB, BB-probs, WB	P	
	[29]-2023		Prompt Embeddings (Equation (2))	WB	P	
	[131]-2023	Document Level	Prompt (Equation (1))	BB, BB-probs, WB	P	
	[12]-2024	Demo Ex Level	Demo Ex(s) (Equation (3))	BB, BB-probs, WB	P&U	
GDP	[124]-2023	Sample and Aggregate	Demo Ex(s) (Equation (3))	BB, BB-probs, WB	P&U	
	[50]-2023			BB, BB-probs, WB	P	
	[30]-2023	PATE-Based	Demo Ex(s) (Equation (3))	BB, BB-probs, WB	P&U	
	[126]-2022			BB, BB-probs, WB	P&U	
	[75]-2023			BB, BB-probs, WB	P&U	-
	[157]-2022	DP Synthetic Data	Demo Ex(s) (Equation (3))	BB, BB-probs, WB	P	-
	[41]-2024			BB, BB-probs, WB	P	-
	[69]-2023			BB, BB-probs, WB	P	-
	[13]-2023			BB, BB-probs, WB	P	-
	[147]-2024			BB, BB-probs, WB	P&U	-
	[12]-2024			BB, BB-probs, WB	P&U	
	[87]-2022		Prompt (Equation (1))	BB, BB-probs, WB	P	
	[30]-2023	DPSGD	Prompt Embeddings (Equation (2))	WB	P	
Others	[144]-2023	Demo Ex(s) at LLM	Demo Ex(s)	BB, BB-probs, WB	P	
	[56]-2023	Data Augmentation	Datastore	WB	U	
	[146]-2023			BB, BB-probs, WB	P&U	
	[5]-2023			BB, BB-probs, WB	P&U	
	[150]-2023	CDP-FL	Client Data	WB	U	
	[121]-2024			WB	U	
	[46]-2023			WB	U	

Demo Ex(s) denotes demonstration example(s), CDP-FL denotes client data protection via federated learning, LLM Applicability is the types of LLMs that the technique applies to, BB is black-box, BB-probs denotes black-box with access to softmax probabilities, WB is white-box, PUG is privacy-utility guarantee, P denotes privacy guarantee, U is utility guarantee, P&U denotes both privacy and utility guarantees, $T_1, \ldots, T_{11}$ denote downstream tasks. T_1: classification, T_2: information extraction, T_3: generation, T_4: translation, T_5: creative writing, T_6: recommendation, T_7: tabular data analysis, T_8: question answering, T_9: multi-hop question answering, T_{10}: summarization, T_{11}: recognition.

kept locally as a Plaintext-Ciphertext pair. This is used to recover and restore sanitized attributes included in the response returned by the cloud LLM in the post-processing stage.

Instead of using a single local LLM for sanitization, Chen et al. [19] developed a framework called *Hide and Seek* (HaS) that comprises two models, hiding private entities (Hide-Model) for anonymization, and seeking private entities (Seek-Model) for de-anonymization. The training phase uses an LLM through prompt engineering to generate a training dataset. This training

dataset is used to train Hide-Model and Seek-Model. During the inference phase, a user-submitted text is anonymized using the Hide-Model stored in a terminal device such as a mobile phone, tablet, or laptop. The anonymized text can then be used to prompt a cloud LLM. At the same time, the anonymized text together with the original text are sent to the Seek-Model deployed locally. Similarly, the output returned from the cloud LLM is also fed into the Seek-Model. Finally, the Seek-Model de-anonymizes the output returned by the cloud LLM for user consumption. However, training two LLMs presents multiple challenges ranging from computational issues to model effectiveness, thus making this framework appear infeasible.

Zhang et al. [160] introduced another approach, called *mixed-scale model collaboration*, that combines the capabilities of a large model in the cloud with a small model deployed locally. This blends public text with private data to personalize usage, thus providing a logical solution to the privacy challenge. The developed CoGenesis framework has two variants: sketch-based and logit-based variants. CoGenesis operates under the assumption that a user's instruction comprises two sections: a general (privacy-insensitive) section and a personal (privacy-sensitive) section. In sketch-based CoGenesis, a user first prompts the cloud LLM with the general instruction and receives a content sketch as a response. By employing the sketch-then-fill approach, the user integrates the sketch with both the general and personal instruction sections, allowing for content personalization using the small local LLM. The logit-based variant of CoGenesis integrates the logits from both the cloud LLM and the small LLM to determine the subsequent tokens. The foundation of achieving privacy in this framework lies in not exposing the personal section of the instruction to the cloud LLM, but rather utilizing it locally.

3.1.2 Ensemble-Only Methods. Ensemble-only methods focus on safeguarding the privacy of demonstration examples during ICL outlined in Equation (3). Prompted LLMs may incur a high risk of disclosing private information of demonstration examples used in prompts. Duan et al. [31] first studied this privacy leakage through the lens of MIAs. The adversary assumes to have access to the prediction probabilities of each possible target class of the test example. The attacker instantiates the MIA to determine whether his input was part of the demonstration examples by examining the prediction probabilities y. Duan et al. [31] compared the MIA risk of prompted and fine-tuned models and observed that the privacy risk of prompting is significantly higher than fine-tuning. To mitigate the privacy risk exposed by prompt membership, Duan et al. [31] proposed to aggregate the prediction probability vectors over multiple independent prompted models into an ensemble prediction. In this work, K models are prompted, each with a disjoint subset of the private data. For an input x_i, each prompted model generates its own output probability $y_i^{(k)}$. To generate the final output, two ensemble methods, Avg-Ens and Vote-Ens, were developed. In Avg-Ens, the final output $\hat{y}_i$ for the input x_i is generated by computing the average of the raw probability vectors of each of the K prompted models as $\hat{y}_i = \frac{1}{K}\sum_{k=1}^{K} y_i^{(k)}$. Meanwhile, Vote-Ens relies on a majority vote of all prompted models to generate the final output. For input x_i, each prompted model outputs a token prediction from the vocabulary $\mathcal{V}$ with the highest logit value. Supposing that n_v denotes the number of prompted models that predict token v as the output for the input x_i, the final output $\hat{y}_i$ is computed as $\hat{y}_i = \arg\max_{v \in \mathcal{V}}(n_v)$.

3.1.3 Obfuscation/Lattice Methods. Obfuscation and lattice methods aim to safeguard the privacy of both the prompt and the response described in Equation (1). In the current user-LLM interaction paradigms, the user provides the LLM server with a prompt and the LLM generates a response, with both the prompt and the generated response being accessible to the potentially untrustworthy LLM server. There exist application scenarios in which both user prompts and the generated contents need obfuscation because both can directly affect the user's decisions.

Yao et al. [153] introduced the **Instance-Obfuscated Inference (IOI)** framework to safeguard both the raw input and output in the black-box inference setting. During inference, IOI obfuscates input instances, ensuring that the raw decision distribution does not disclose any sensitive information, while still enabling the user to recover the true decision. In other words, IOI obfuscates the raw input instance using obfuscators designed to intentionally influence the cloud LLM's inference decision, thereby preventing adversaries from deducing the true decision without knowledge of the resolution method and parameters. Instead of sending the real instance, IOI combines it with dummy instances and utilizes a Privacy-Preserving Representation Generation (PPRG) method [170] to transform the plaintext combination into privacy-preserving representation. The cloud LLM generates its response based on this combination. Subsequently, IOI employs its Privacy-Preserving Decision Resolution algorithm to extract the true response from the LLM's returned response. This prevents adversaries including the PLM service provider from discovering the true instance and response.

Zhang et al. [161] developed the LatticeGen framework that enables a collaborative generation of tokens between the user and the LLM server. On each timestep, the user and server conduct the generation token-by-token cooperatively instead of letting the server alone handle the generation process. Here, the user sends the server N tokens, only one of which is the true token and the rest act as noise. For a round t, the true token is denoted as w_t^1 and the noise tokens are denoted as $[w_t^2, \ldots, w_t^N]$. To prevent the server from knowing the true token, the client permutes the tokens as $[\tilde{w}_t^1, \ldots, \tilde{w}_t^N]$. Prior to sending them to the server, a linear operation is then performed on the permuted tokens as

$$\text{linearize}\left(\tilde{W}_T^N\right) = [< bos >] + \text{concat}_{t=1}^{T}\left(\left[\tilde{w}_t^1, \ldots, \tilde{w}_t^N\right]\right), \tag{6}$$

where T is lattice length and $< bos >$ denotes the beginning of a sentence token. After receiving the linearized tokens, the server generates the next token for each of the N token options and returns them to the user. The user then performs reverse permutation mapping to obtain the true token from the returned tokens.

3.1.4 Encryption Methods. Encryption-based methods follow the same principles as obfuscation and lattice methods, aiming to safeguard the privacy of both the prompt and the response in Equation (1). Lin et al. [77] introduced an emoji encryption-based framework called *EmojiCrypt*. Here, a user's sensitive prompt inputs are encrypted into emojis before sending them to the cloud LLM. This attempts to render the original input unreadable to humans while retaining enough information for the LLM to perform effectively. In a way, this method is similar to the sanitization method, and instead of replacing sensitive text with other non-sensitive phrases, the emoji encryption replaces them with emojis.

Hou et al. [51] developed the CipherGPT framework for secure two-party inference. Secure inference is a two-party cryptographic protocol running inference to achieve the goal that the server learns nothing about the client's input and the client learns nothing about the model except the final inference results. Generally, the protocol proceeds by having the server and client run the encrypted model over the encrypted input through cryptographic techniques such as **homomorphic encryption (HE)** and secret sharing. CipherGPT introduces a series of novel protocols, including a secure matrix multiplication that is customized for GPT inference, a protocol for securely computing GELU, and a protocol for top-k sampling, to support secure GPT inference. Other frameworks, such as Iron [48] and THE-X [18], also employ cryptographic techniques to securely perform inference in transformer-based models. Specifically, Iron adopts HE and secrete sharing to build efficient protocols for matrix multiplication and other non-linear operations such as Softmax, GELU, and LayerNorm used in single transformer LLMs. THE-X replaces complex non-linear

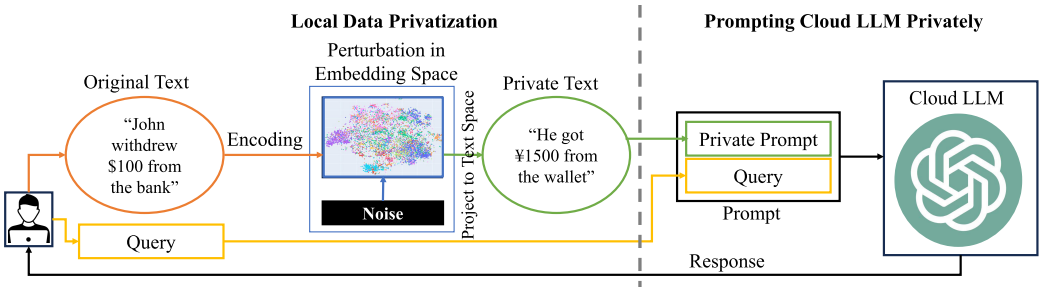

Fig. 3. An illustration of privatizing sensitive local data with LDP before using it to prompt a cloud LLM privately.

operations in transformer-based LLMs with HE-friendly operations. For instance, it replaces GELU with ReLU, and Softmax with a combination of ReLU and polynomials. Although these techniques can be used for privacy-preserving text generation tasks theoretically, practical applications are limited because of high computational and communication costs.

3.2 LDP Methods

Simple sanitation techniques discussed in Section 3.1.1 fail to provide rigorous privacy protection. LDP mechanisms have been explored to allow users to sanitize their sensitive data locally before sending the sanitized prompt to the untrusted LLM server. Generally speaking, these methods introduce randomness to text by privatizing the embedding vector for each token, word, sentence, or the whole prompt. The text data is first transformed into a representation vector via embedding methods and some DP mechanism is then applied to privatize representations. The perturbed vector representations are projected back into the text space to find some appropriate text. Due to the post-processing property of DP, mapping privatized representations back to text also preserves DP. After perturbation, the LDP perturbed prompt is then sent to the remote black-box LLM. The LLM then returns the generation result to the user. Figure 3 illustrates this procedure. The original text ("John withdrew $100 from the bank") is first encoded to generate its embeddings. LDP noise is then added to the embeddings to generate a perturbed version ("He got ¥1500 from the wallet") of the text. This private text is then used in the prompt to guide the cloud LLM to generate a response to the query.

Most LLMs only support black-box access where users submit text prompts via LLMs' standard interfaces. However, a limited number of LLMs also support white-box access. Here, we present LDP methods that can be employed to safeguard the text prompts described in Equation (1), soft prompts described in Equation (2), and demonstration examples described in Equation (3). We categorize them according to their privacy levels as follows.

3.2.1 Word-Level Perturbation. To achieve privacy protection for each word (or token) in the text prompt, Lyu et al. [83] proposed an LDP-based framework that privatizes each word's representation vector and sequentially replaces sensitive words in the text with semantically similar words. The key idea is to map each real value of the embedding vector into a binary vector with a fixed size and then privatize the vector via a variant of the unary encoding mechanism. Plant et al. [103] developed the context-aware private embeddings (CAPE) approach that extracts the representation of each token from the final representation layer of a pre-trained model, normalizes it with sequence, and adds Laplace noise. Zhou et al. [171] further introduced TextObfuscator that obscures word information while maintaining word functionality through random perturbations applied to clustered representations. Clusters are created by identifying prototypes

for each word and promoting word representations to be close to their respective prototypes. After training, words of similar functionality are close to the same prototype. Random perturbations are applied to these clustered representations to protect privacy. They also introduced techniques for identifying prototypes for both token-level and sentence-level tasks by leveraging semantic and task-related information. However, in scenarios where the architecture and parameters of the LLM are known, there is no need to project the perturbed input into text space.

Feyisetan et al. [40] developed a text perturbation approach to achieve the metric LDP. The metric LDP inherits the idea of LDP to ensure that the outputs of any two adjacent inputs are indistinguishable to protect the original input from being inferred. Furthermore, the metric LDP also aims to preserve the utility of sanitized texts by assigning higher probabilities to words that are semantically closer to the original ones. The developed approach [40] applies the generalized planar Laplace mechanism [145] to perturb each token embeddings and further post-processes them to sanitized text via nearest neighbor search. To improve the performance, Xu et al. [149] used the Mahalanobis norm to replace the Euclidean norm adopted in the work of Feyisetan et al. [40] to measure the semantic similarities between words. Carvalho et al. [14] proposed an improved perturbation method via the truncated Gumbel noise. Yue et al. [156] further proposed utility-optimized metric LDP based on the observation that different inputs have different sensitivity levels to achieve higher utility. The developed SANTEXT+ approach divides all the text into a sensitive token set and an insensitive token set and allocates different privacy budget to each set. After deriving token vectors, it samples new tokens via metric LDP when the original tokens are in the sensitive set.

Injecting DP noise into representation vectors directly may significantly distort the semantics of the original text. Several research works [17, 128] leveraged the exponential mechanism to privately replace each word or token in the raw prompt with semantically similar alternatives. Chen et al. [17] developed a customized text sanitization mechanism (CusText) that assigns each input token a customized output set of a small size and adopts the exponential mechanism to sample the output for each input token. The designed scoring function for exponential mechanism takes into account the semantic similarities between tokens during sampling. Tong et al. [128] developed the InferDPT framework and proposed RANdom adjacency for Text perturbation (RANTEXT) that introduces random adjacency for token-level perturbation of uploaded prompt. To provide strong protection for the raw prompt, its perturbation module utilizes the exponential mechanism to sequentially replace each word or token in the raw prompt with semantically similar alternatives from the adjacency list. It also adopts the Laplace mechanism to dynamically determine the size of the adjacency list for each token.

One limitation with the word-level protection is its lack of contextualization. The text generated by the perturbed prompt is often partially inconsistent and semantically incoherent with the raw prompt. This is because word-wise perturbation is conducted independently. Research has been proposed to address this challenge. For example, the InferDPT framework developed in the work of Tong et al. [128] incorporates an extraction module that employs a local language model to extract text from the perturbed generation results so that the final reconstructed output is coherent, consistent, and aligns with the raw prompt. Note that the local language model is smaller than the remote LLMs and does not pose any privacy leakage.

3.2.2 Sentence-Level Perturbation. The word-level privacy (i.e., having each word indistinguishable with similar words) may not hide higher-level concepts in the prompt. There have been a few research works studying sentence-level private embeddings. Sentence embeddings can be aggregated from token embeddings, for example, via mean pooling, or extracted from language models like BERT [25]. Du et al. [28] initiated a study of sanitizing sentence embeddings to achieve

the metric LDP. They proposed two instantiations from the Euclidean and angular distances. The former directly draws replacements from a distribution defined on a sphere and utilizes the Purkayastha mechanism [139] based on the angular distance. The latter is to post-process the output of the noisy embedding by the Euclidean distance based planar Laplace mechanism [145].

Du et al. [29] considered sentence-level privacy for private fine-tuning. The proposed DP-Forward directly perturbs embedding matrices in the forward pass of PLMs and ensures the standard LDP for test sequences. They considered transformer-based LLMs which contain two categories of layers: embedding layers and task layers. The authors proposed an **analytic matrix Gaussian mechanism (aMGM)** to draw a non-iid DP noise from a matrix Gaussian distribution. DP-Forward adds aMGM noise to embeddings output by layers preceding the task layers. During inference, a user with an unlabeled sequence accesses the embedding layers to derive embedding matrices and perturbs them with aMGM noise. The user then sends the noisy embeddings to the LLM service provider. To make predictions, the LLM service provider runs the task layer functions on the noisy embeddings. With known LLM architectures and parameters, the perturbed embeddings can be directly utilized in white-box scenarios. Although their original purpose is to fine-tune models, these methods can also be applied for inference.

3.2.3 Document-Level Perturbation. Instead of focusing only on word-level and sentence-level privacy, another work considers document-level privacy. To protect the privacy of the entire prompt during the inference process, Utpala et al. [131] devised DP-Prompt to attain document-level DP. DP-Prompt takes a private document and generates a paraphrased version using zero-shot prompting on a local PLM. The process of sequentially generating text from the language model is regarded as a problem of selecting tokens at each step. To make the generation step differentially private, DP-Prompt replaces the sequence with a differentially private version of the selection process. The resulting paraphrased document is then released as a sanitized document to prompt cloud LLMs. Specifically, given a private document alongside a designated prompt template instructing the language model, DP-Prompt generates text in a differentially private manner, producing a paraphrased version of the private document. The exponential mechanism is utilized for token selection during the sequential text generation process.

3.2.4 Demonstration Example Level Perturbation. The capabilities of LLMs have been extended to include tabular data analysis, leveraging the principles of ICL and prompt tuning [49, 77]. In performing this task, the common practice involves first serializing the tabular data into text before using them to prompt the LLM. To protect the privacy of demonstration examples, Carey et al. [12] investigated the application of DP mechanisms for private tabular ICL via data privatization prior to serialization and prompting. They introduced the LDP-TabICL framework that employs the RR LDP technique [138] to perturb attribute values of each sample at the each end user side. Generating private demonstration examples within the LDP-TabICL involves reconstructing the DP-protected data obtained from users, selecting k samples from the reconstructed data, and serializing the k samples into texts. Similarly, concatenating the serialized texts with a query to facilitate ICL. The top left part of Figure 4 depicts the private demonstration example generation in LDP-TabICL.

3.3 GDP Methods

In ICL, users often have a private local dataset $D_{priv} = \{(x_i, y_i)\}_{i=1}^{n}$, from which a set of k demonstration examples D_k is chosen to be included in the prompt as shown in Equation (3). However, the simple inclusion of D_k in the prompt certainly incurs the privacy disclosure. The cloud LLM is privacy untrusted and may try to gain private information from the user's prompt. GDP-based algorithms can be adopted here to ensure that each individual sample in D_{priv} cannot be inferred

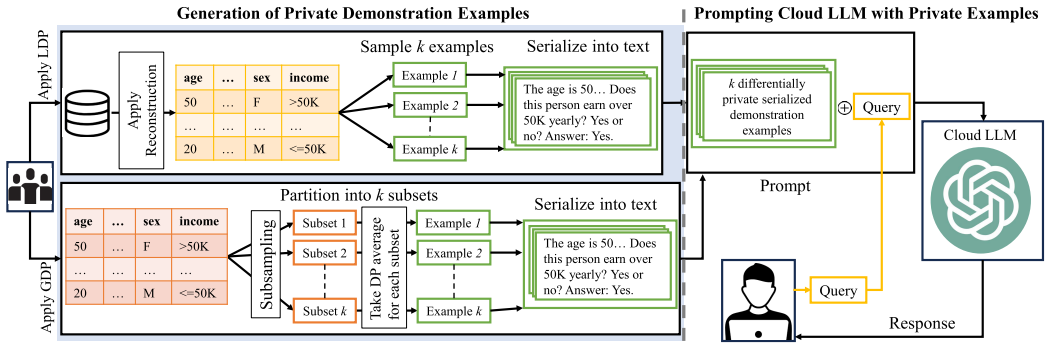

Fig. 4. Privacy-preserving demonstration example generation with LDP-TabICL and GDP-TabICL approaches. For LDP-TabICL (top left), users perturb their data with the RR LDP mechanism before being collected. The collected data is then reconstructed to recover the original data distribution. The k samples are selected and serialized into text. Meanwhile, for GDP-TabICL (bottom left), user data is collected in clear. Then the collected data is partitioned into k disjoint subsets. GDP averages for each attribute in each subset are generated. The generated noisy attributes are then serialized into text. During ICL, an LLM is prompted with k demonstration examples selected from the serialized text and a query from a user. LLM's response is generated and sent to the user.

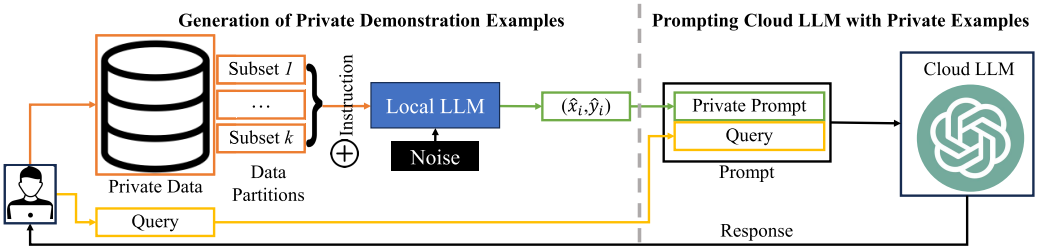

Fig. 5. An illustration of privacy-preserving demonstration examples generated with a local LLM, then leveraging the examples with a query to perform ICL using a cloud LLM.

with high confidence from the prompt sent to untrusted LLMs. One common solution is to generate the differentially private demonstration examples (denoted as $\hat{D}_k = \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^{k}$) before employing them to perform ICL in the cloud LLM. Achieving GDP here means the presence or absence of any example (x_i, y_i) in D_{priv} would not have significant impact on the produced $\hat{D}_k$ which will be used as demonstrations in ICL. Several methodologies have surfaced for generating differentially private demonstration examples: sample and aggregate based approach, PATE-based approach, DP synthetic data generation approach, and soft prompt generation via DP-SGD.

3.3.1 Sample and Aggregate Based Approach. Figure 5 presents an illustration of the concept. In broad terms, the process entails partitioning the private data into distinct subsets. These subsets are then submitted alongside instructions, guiding the local LLM to sequentially generate data resembling the private data token-by-token. At each token generation step, DP noise is introduced to the token probability. Consequently, the resultant generated data is DP private. This iterative process is repeated multiple times. Upon completion, the DP-protected examples are utilized to prompt the more powerful (yet untrusted) cloud LLM.

Tang et al. [124] studied how to conduct ICL with LLMs on private datasets and focused on the privacy protection of demonstration examples used in the prompt. Their developed algorithm generates synthetic differentially private few-shot demonstrations from the original private dataset, and uses the generated samples as demonstrations in ICL during inference. The approach leverages the capabilities of local trusted LLMs in terms of generating a data sample similar to the ones in the original dataset. To generate a differentially private demonstration example for a given label y, the algorithm generates one token at a time from an empty list. At each token generation, disjointed subsets are extracted from the private training dataset D_{priv}. Each subset is appended with previously generated tokens and is fed into a local LLM to generate the next token. Next-token generation probabilities obtained from each subset are then privately aggregated. Both the Gaussian mechanism and the report-noisy-max with exponential mechanism are adopted in the algorithm. Finally, the next token is produced and appended to previously generated tokens. This process is continued until the end of sequence token is produced. To reduce the effect of noise, the algorithm limits the vocabulary to the tokens present in top-K indices of the next-token probability coming from only the instruction without any private data. The generated private demonstration examples are leveraged to perform ICL on the LLM in the cloud (a less trusted environment). Note that the generated samples can be used for an infinite number of queries without incurring any additional privacy costs.

Similarly, to generate private demonstration examples locally, Hong et al. [50] developed a framework called **Differentially Private Offset Prompt Tuning (DP-OPT)**. Given the private training dataset D_{priv}, DP-OPT uses a few samples as demonstrations to guide a local LLM to generate private prompts. The prompt generation process is facilitated by a differentially private ensemble of ICL with disjoint private demonstration subsets. DP-OPT adopts the *forward-backward* approach of the Deep Language Network (DLN) [119]. This approach mimics the gradient-based optimization method that uses forward and backward passes to train prompts on a training dataset. Given the private training dataset D_{priv}, in the *forward* pass of DP-OPT, the local LLM is prompted to predict the labels on a sample batch of training samples $S \subset D_{priv}$ by submitting a forward template with a task instruction π. The output from the forward pass $\hat{y}_i$ is then used together with S in a *backward* template to guide the local LLM. This backward pass attempts to regenerate the task instruction π. The regeneration of π is performed one token at a time. During each token generation, token candidates are generated and aggregated with aggregates perturbed using differential noise. The best performing $\{(x_i, y_i, \hat{y}_i)\}$ are then privately selected and used for performing ICL in the cloud LLM.

3.3.2 PATE-Based Approach. PATE [100] is a DP learning model using a teacher-student framework. The student accessing unlabeled non-sensitive data distills knowledge from the aggregated predictions of multiple teachers. Each teacher model is trained on a partition of sensitive data. Calibrated noise is added to the aggregated predictions to meet DP requirements. Duan et al. [30] proposed PromptPATE, a privacy-preserving ICL framework for discrete prompts based on the PATE [100] mechanism. It assumes the existence of a labeled private dataset with labeled examples like ("The book was great," "positive"), and an unlabeled public dataset with examples like ("I enjoyed this movie," -). The private training dataset is divided into multiple subsets to create prompts that can be deployed with the LLM as teachers. During the private knowledge transfer, for any input sequence from the unlabeled public dataset, each teacher votes for the most likely class. The consensus over the teachers' votes is determined via a noisy argmax over all teachers' vote counts. The added noise is sampled from a Gaussian distribution to satisfy the DP guarantees. The labeled public dataset is then leveraged as demonstration examples to perform ICL. The process is illustrated in Figure 6.

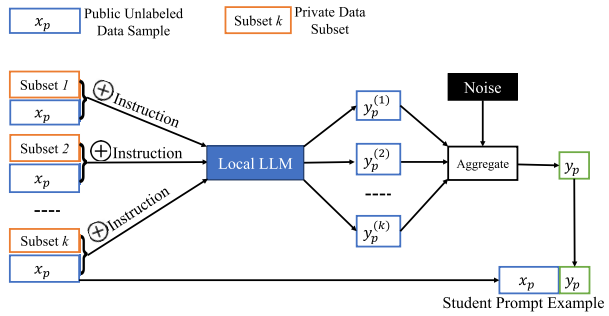

Fig. 6. Private demonstration example (student prompt example) generation using PromptPATE [30]. A public unlabeled data x_p is labeled by prompting a local LLM with subsets of private data as prompt examples. A noisy aggregation is then performed on all prompt votes to generate a label y_p for x_p. (x_p, y_p) can then be used to perform ICL publicly.

Tian et al. [126] developed the SeqPATE framework for text generation. Instead of sequential generation, SeqPATE generates pseudo-data using a PLM such that teachers only need to provide token-level supervision given the pseudo input. To address the large output space, SeqPATE aggregates teachers' outputs by interpolating their output distributions instead of voting for the final aggregate output. It also incorporates strategies to dynamically filter candidate words and only keep words with high probabilities. Specifically, for the text generation task that aims to generate the remaining part of a sentence given its prefix, SeqPATE assumes a private dataset (containing complete sentences) and a public dataset (containing input prefixes). SeqPATE trains each teacher model on one of the disjoint subsets of the private dataset and conducts student training and teacher inference on pseudo sentences generated by the LLMs based on the public data. The student model is supervised by the private aggregation of teacher output distributions.

Li et al. [75] proposed Prom-PATE that explores the benefits of **visual prompting (VP)** in generating image samples with DP noisy labels. Such generated samples can be employed to perform prompting in a DP manner. Specifically, Prom-PATE employs two steps to generate image samples with noisy labels: training re-teacher models and executing private aggregation. Training the re-teacher model involves training the soft prompts using a private dataset while maintaining the pre-trained visual model parameters unchanged. Several re-teacher models are trained using disjoint partitions of the private dataset. In the second step, Prom-PATE labels an unlabeled public image dataset by employing the PATE [100] mechanism to perform DP aggregation on responses from re-teacher models. Re-teacher responses are generated with visual prompts. This step generates noisy labels for the samples in the unlabeled public dataset. Such noisy labeled samples can then be used to freely perform prompting. However, such studies are relatively rare and more work is required to determine their effectiveness.

3.3.3 DP Synthetic Data Generation. One notable advantage of differentially private synthetic data generated from a private dataset is that the resulting text can be freely shared and utilized (including for performing private ICL) with minimal privacy concerns. Yue et al. [157] and Flemings and Annavaram [41] fine-tuned a pre-trained generative language model with DP using a private dataset, enabling the models to produce synthetic text with robust privacy protections. The studies utilized the DP-SGD [1] framework during the fine-tuning process. DP synthetic data is generated by prompting the fine-tuned model with control codes. This DP synthetic data captures the general statistical characteristics of the private text and can be utilized more freely for various downstream tasks, with minimal privacy risks. For example, Flemings and Annavaram [41] used

the DP synthetic data alongside a teacher model output distribution to transfer knowledge from the teacher model to a student model. Kurakin et al. [69] chose to perform parameter-efficient fine-tuning on the pre-trained LM using prompt tuning [71] and LoRA [53]. The authors still adopted the DP-SGD framework during the fine-tuning. In this case, DP synthetic data is generated by feeding prefix as input to the fine-tuned model. Carranza et al. [13] adopted the same approach to generate DP queries to train retrieval systems. Their work employed DP-Adafactor (Adafactor [114] that receives clipped and noised gradients as per DP-SGD [1]) to fine-tune a pre-trained LM. The DP-tuned LM is then used to generate DP synthetic queries.

Xie et al. [147] considered generating DP synthetic data with only API access to LLMs. They proposed an augmented **Private Evolution (PE)** framework referred to as AUG-PE. The concept behind AUG-PE involves initially sampling random instances from an LLM guided by instructions, followed by iterative enhancements through DP selections, focusing on those resembling the private dataset. Subsequently, the LLM is queried to generate additional samples resembling the selected ones. The entire concept can be divided into four steps, with steps 2 through 4 carried out iteratively. In step 1, an LLM is prompted to generate random samples. Subsequently, in step 2, each private sample votes for its nearest synthetic counterpart in the embedding space, followed by the addition of Gaussian noise to these votes. This produces a DP nearest neighbor histogram. With the help of this histogram, step 3 involves the selection of synthetic samples with noisy votes. Finally, in step 4, the LLM is prompted to generate new samples resembling the noisy selection in step 3.

Meehan et al. [87] proposed DeepCandidate to achieve sentence-level DP when releasing a document embedding. The document embedding is sentence private if any single sentence in the document is removed or replaced we can still have a similar probability of producing the same embedding. Thus, the sentence-level privacy is defined from the GDP perspective—that is, hiding the impact of any single sentence in a document. The privatized document embedding only stores limited information unique to any given sentence. DeepCandidate uses a sentence encoder to get sentence embeddings and adopts the exponential mechanism to sample from the candidate embeddings for each private embedding.

Emerging alternative methodologies avoid the need for local (trusted) LLM or encoder support. Carey et al. [12] embraced this approach for conducting tabular data analysis with a DP guarantee. To protect the privacy of demonstration examples, Carey et al. [12] developed the GDP-TabICL framework for private tabular ICL via data privatization prior to serialization and prompting. GDP-TabICL relies on both Poisson subsampling for privacy amplification and the Laplace mechanism to craft differentially private aggregates that represent the underlying data distribution. GDP-TabICL segments the sampled data into k disjoint subsets based on the required number of demonstration examples. DP statistics for each feature in each subset are subsequently computed. These statistics are used to generate synthetic examples and then serialized into text-based demonstration examples along with a test query to facilitate ICL. Figure 4 depicts the proposed approach. Different from works [30, 50, 124] that assume the use of local LLMs to produce differentially private prompts, DP-TabICL [12] uses reconstruction from perturbed statistics to generate DP demonstration examples and thus incurs much less computational cost.

3.3.4 Soft Prompt via DPSGD. While the preceding studies focus on discrete prompts that require only black-box access to the LLMs, approaches that privately learn soft prompts are also important. Soft prompts are additional task-specific embeddings that can be prepended to the original input embeddings before passing them through the LLMs [30]. The gradient descent method can be employed to train these embeddings using private data. For private training, DP noise is added to the gradients associated with these embeddings. Duan et al. [30] developed PromptDPSGP that

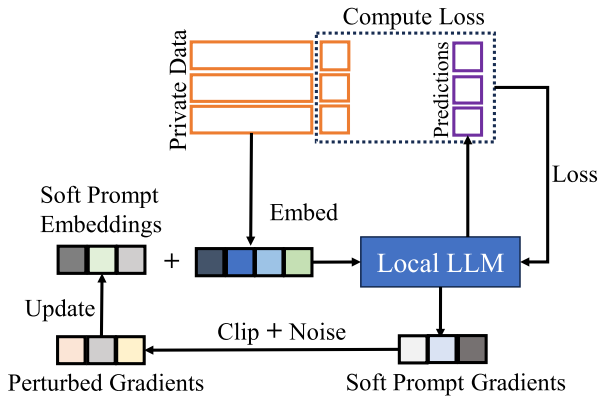

Fig. 7. An illustration of private soft prompt generation using PromptDPSGD [30]. A soft prompt is prepended to the private data embeddings. This combination is forwarded to an LLM. The LLM makes predictions $\hat{y}$ and computes loss accordingly. Next, LLM uses the loss to compute the gradients associated with the soft prompt. These gradients are then clipped, and noise is added to them before being used to update the soft prompt privately.

leverages the DP-SGD algorithm [1] to learn soft prompts that are prepended to an LLM's input with a DP guarantee. An illustration of the process is shown in Figure 7. However, soft prompts require white-box access to LLMs, so this may not be possible all the time.

3.4 Other Scenarios

3.4.1 Demonstration Examples at LLMs. An alternative point of leakage is through model outputs. An adversary can infer information about the input data from model outputs. A solution can be achieved through a noisy consensus among an ensemble of an LLM's responses. Wu et al. [144] proposed a **differentially private in-context learning (DP-ICL)** paradigm where the sensitive dataset used for demonstrations is stored in the LLM site. The server partitions the sensitive dataset into disjoint subsets, each comprising a collection of demonstration examples. The server generates these demonstration-query pairs and calls LLM to produce corresponding outputs. These outputs are aggregated through a differentially private mechanism before being returned to the user. Figure 8 depicts this framework. For text classification, Wu et al. [144] adopted Report-Noisy-Max with Gaussian noise to privately release the class that receives the majority vote. For language generation, to deal with the challenge arising from the huge output sentence space, the authors proposed (1) Embedding Space Aggregation (ESA) which projects the output sentences into a semantic embedding space and then privatizes these aggregated embeddings, and (2) Keyword Space Aggregation (KSA) which identifies frequently occurring keywords in the output and then privately selects them via propose-test-release [34] or the joint exponential mechanism [42]. Different from DP synthetic data generation algorithms, DP-ICL does not permit an infinite number of queries as each prompt query incurs privacy consumption.

3.4.2 Data Augmentation Using External Datastores. For PLM users with insufficient local data, leveraging external datastores can improve ICL performance. The final generated prompt may contains sensitive information from both the local data and external datastores. Retrieval-based techniques [8, 60, 66, 89], which have been developed to combine LLMs' output with retrieved texts from datastores, can also be applied here to help users prepare augmented prompts. However, this process incurs potential leakage of sensitive information of the external datastores. This raises

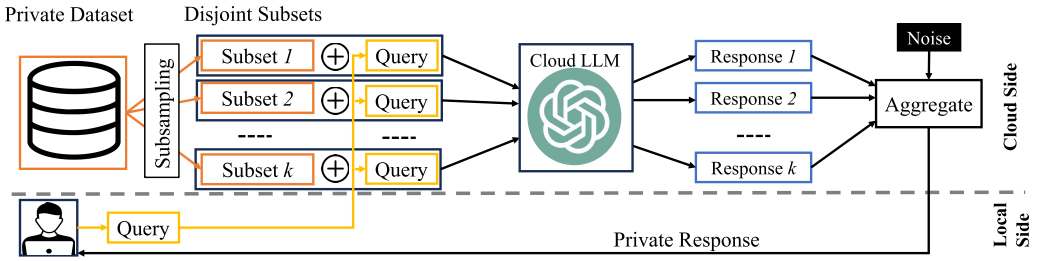

Fig. 8. DP-ICL with a private response. Disjoint subsets of a private dataset are each used as prompt examples to answer a query. Responses to each prompt are privately aggregated to generate the final response to the query.

the obvious question: *Can we introduce external knowledge in the prompt without compromising the privacy of the external datastores?* We briefly review sanitization and **information flow control (IFC)** methods that can be used to address this question.

As stated in Section 3.1.1, sanitization aims to create a sanitized version that would keep structural properties but remove sensitive information from the original data. A common approach used in machine learning involves masking the sensitive data attributes [7, 82, 93]. Huang et al. [56] proposed a sanitization method for replacing each privacy-sensitive phrase in the datastore with (i) < |endoftext| >, (ii) dummy text (e.g., replacing a telephone contact with "012-345-6789"), or (iii) public data (e.g., replacing a telephone contact with a publicly known telephone number). IFC is a privacy-by-design model. Wutschitz et al. [146] developed an IFC-based framework that takes into consideration metadata such as access control policies in designing machine learning systems. Private external data retrieved during the prompt augmentation is subject to satisfying security requirements (e.g., access policies). This ensures that the information presented to the users is what they are authorized to see. For example, when selecting demonstration examples from an external store, the model ensures the retrieved samples satisfy the security requirements. Under the scenarios where separate public corpora and private corpora exist, Xiong et al. [148] and Arora et al. [5] developed multi-hop retrieval-based models to protect privacy. The retrieval process is performed iteratively, limiting information flow and careful ordering of the retrieval from both public and private corpora.

3.4.3 Client Data Protection via FL. Utilizing pre-trained models through prompting has also been expanded to include visual foundational models [61, 169]. For example, Jia et al. [61] explored soft prompts by prepending them on the input tokens of pre-trained visual transformers. In this manner, the soft prompts are optimized to capture task-specific details, enabling them to be used to prompt the pre-trained model to generate suitable responses tailored to the task at hand. Consequently, there is a rising interest in studies that aim to tackle the privacy challenges associated with VP [46, 121, 150]. The majority of works in this area consider the adoption of the **federated learning (FL)** approach. FL enables clients to learn a joint model without sharing their private data [85].

Guo et al. [46] proposed the PromptFL framework which is built upon the FL mechanism. The PromptFL framework operates on the basis that each client possesses a CLIP foundation model [106]. Within PromptFL, clients keep their data locally and train shared soft prompts collaboratively by communicating gradients rather than the data. This approach facilitates the development of robust prompts while safeguarding the privacy of individual client data. Given the diversity in clients' data, personalized FL methods [16, 112] have also emerged. These approaches

enable individual clients to train personalized models tailored to their specific datasets. Su et al. [121] and Yang et al. [150] proposed a federated adaptive prompt tuning algorithm (FedAPT) and personalized FL for client-specific prompt generation (pFedPG) frameworks, respectively. These models adopt the personalized FL approach. Specifically, in FedAPT, each client trains an adaptive network and prompts using their private data along with a random key assigned by the server. A global adaptive network and global prompts are computed on the server using adaptive networks and prompts received from the clients. These global parameters are then utilized to generate personalized prompts for each client. Meanwhile, pFedPG learns a personalized prompt generator at the server, which is used to generate client-specific prompts. The framework consists of two phases: global personalized prompt generation and local personalized prompt adaptation. During the personalized prompt adaptation phase, each client trains client-specific prompts. However, personalized prompt generation is achieved by learning to derive personalized prompts for each client through the exploitation of optimization directions among clients. The principles underlying these frameworks offer valuable insights for achieving client-level privacy in NLP contexts, thus further exploration in the NLP domain is warranted.

3.5 Summary

In this section, we have categorized existing techniques of privacy preserving prompt engineering into four privacy types: non-DP, LDP, GDP, and others. We have further identified six major privacy targets: discrete prompt, soft prompt, demonstration examples, LLM response, datastore information, and client data. For each combination of privacy type and privacy target, we have shown the appropriate privacy protection mechanisms in Figure 1 and Table 1. For example, to protect the privacy of discrete prompt under non-DP, we can choose sanitization-, obfuscation/lattice-, and encryption-based methods. Similarly, to protect the privacy of demonstration examples under GDP, we can consider sample and aggregate, PATE-based, and DP synthetic data methods. Each method has its own applicability requirements and PUGs or preferences. For example, Prompt-PATE [30] is applicable with black-box (with and without access to softmax probabilities) and white-box LLMs and achieves the rigorous GDP protection of demonstration examples. However, the DP-SGD method [30] is only applicable to white-box LLMs.

Here we present a brief guide for practitioners in deciding on privacy techniques for use. The guide is based on the layout presented in Figure 1 and Table 1. We use the following scenario for further elaboration. Suppose there exist medical records on diabetes intended to be used to guide an LLM into predicting if a patient is diabetic. In this scenario, the practitioner is dealing with sensitive information and might require guidance on using the LLM while keeping privacy in mind. Our guide comprises the following six steps:

Step-1: Establish the downstream task, which is a classification task in our scenario.
Step-2: Identify the privacy type and privacy target for the task in Step-1. Suppose we want to preserve the privacy of the demonstration examples with GDP.
Step-3: Identify methods capable of preserving privacy type of the privacy target identified in Step-2. In our case, methods such as sample and aggregate, PATE-based, and DP synthetic data can be considered.
Step-4: Examine the LLM applicability requirements (i.e., BB, BB-prob, and WB) and architectures suitable for the methods generated by Step-3.
Step-5: Analyze the PUG for the given privacy target and choose techniques with matching PUG. In this case, we require the demonstration examples to achieve privacy and yet be usable (i.e., P&U is our PUG).
Step-6: Choose a technique for implementation that satisfies the criteria presented in Steps 1 through 5. In our case, the methods that satisfy all the criteria presented in Steps 1 through 5

are shown in various works [12, 30, 124]. In particular, the practitioner can choose the work by Carey et al. [12] if their data are tabular, that of Tang et al. [124] if their data contain public unlabeled records, or the work of Duan et al. [30] if they prefer to generate synthetic differentially private few-shot demonstrations from the original private dataset.

As different techniques have their own privacy targets based on the chosen privacy type, it is generally infeasible to compare them across the whole privacy spectrum. However, within the same combination of privacy type and target and for the same downstream task, it would be greatly needed to have empirical comparisons of all relevant techniques over commonly chosen benchmark datasets such that practitioners can choose appropriate models suitable for their application needs with a high level of confidence. We include the comprehensive empirical evaluation with benchmark datasets and open source software as one future prospect. Nevertheless, we present some qualitative comparisons. For *sanitization* methods, some works [19, 63] do not offer privacy guarantees, and as such, they can be usable in environments with relaxed privacy demands. However, the work by Zhang et al. [160] offers a privacy guarantee since data does not leave users' premises. Hence, it can be employed in high privacy demanding scenarios. For *encryption*-based methods, some works [18, 48, 51] have high computational demand and thus they are only usable in environments with abundant computational resources. The work by Lin et al. [77] guarantees neither privacy nor utility, thus requiring further improvements before it can be considered usable. For *word*-level LDP protection, some methods [83, 103, 171] focus on rigorous local privacy, whereas other methods [40, 149, 156] are based on the relaxed metric LDP. In particular, TextObfuscator [171] generally outperforms two other methods [83, 103] because TextObfuscator obscures word information while maintaining word functionality through random perturbations applied to clustered representations. The SANTEXT+ method [156] is also a better choice than some works [40, 149] in the metric LDP setting because SANTEXT+ adopts the utility-optimized metric LDP based on the observation that different inputs have different sensitivity levels to achieve higher utility. Nevertheless, all these methods are only suitable for scenarios where semantics are not concerning. Similarly, for *DP synthetic data* methods, some works [13, 41, 69, 157] do preserve privacy but have challenges with utility, and thus they might not be ideal for semantic sensitive scenarios. The users could consider the adoption of other works [12, 147], as these methods aim to guarantee both privacy and utility. Regarding *PATE-based* methods, all the considered methods [30, 75, 126] aim to achieve privacy and utility guarantees, but they assume the availability of unlabeled public records in addition to the labeled private dataset. Furthermore, the work of Duan et al. [30] is more suitable for use in text classification, that of Tian et al. [126] for sequential generation, and work by Li et al. [75] for VP-based image generation.

4 Resources

4.1 Datasets

In addition to developing privacy-preserving frameworks, it is crucial to identify high-quality datasets for various downstream tasks to evaluate the performance of these frameworks. This section provides an overview of widely used datasets for assessing these frameworks. Table 2 outlines the datasets, categorized according to data types and associated tasks. The datasets are grouped into three main categories. First, *text datasets* are used for various NLP tasks, including classification, information extraction, creative writing, question answering, summarization, and recommendation. Second, *tabular datasets* are commonly utilized for tabular data analysis. And third, *image datasets* are primarily used for image data analysis. All datasets included in this study are publicly available, with the exception of ACE2005. Due to space constraints, detailed descriptions of each dataset are provided in the supplementary material.

Table 2. Commonly Used Datasets for Evaluating Privacy-Preserving Prompting Frameworks

Type	NLP Task Category	Dataset	Ref	No. of Samples	Citing Publications
Text Datasets	Classification	AGNews	[165]	496835	[30, 31, 99, 124]
		DBPedia	[96]	342781	[30, 124]
		TREC	[73]	6000	[30, 31, 50, 99, 124]
		SST-2	[117]	11855	[30, 31, 50, 99]
		MPQA	[142]	10657	[50]
		Disaster	[52]	10746	[50]
		CB	[57]	556	[31]
		QNLI	[134]	110400	[30]
		QQP	[134]	404300	[30]
		MNLI	[134]	413000	[30]
		BBC	[9]	2225	[19]
	Information Extraction	MIT	[78]	4714	[124]
		Elsevier	[65]	40091	[146]
		Arxiv	[23]	About 1.7 Million	[146]
		ACE2005	[21]	About 1800	[63]
		Enron Emails	[67]	About 500000	[56]
		WikiText	[88]	Over 100 Million	[56]
	Creative Writing	WritingPromp	[38]	10700	[161]
	Question Answering	ConcurrentQA	[5]	18439	[5]
		DocVQA	[127]	46000	[99]
	Summarization	SAMSum	[43]	16369	[99]
	Recommendation	Amazon (beauty)	[116]	Over 2 Million	[77]
Tabular Datasets	Tabular Data Analysis	Adult	[6]	48842	[12, 77]
		Bank	[108]	45211	[12]
		Blood	[84]	748	[12]
		Calhousing	[98]	20640	[12]
		Car	[62]	1728	[12]
		Diabetes	[130]	768	[12]
		Heart	[39]	918	[12]
		Jungle	[132]	44819	[12]
Image Datasets	Image Data Analysis	Office-Caltech10	[44, 111]	2533	[121, 150]
		DomainNet	[102]	0.6 Million	[121, 150]
		Dermoscopic-FL	[20]	10490	[150]
		CIFAR-10	[68]	60000	[75, 150]
		CIFAR-100	[68]	60000	[75, 150]
		Blood-MNIST	[151]	17092	[75]

All datasets except *ACE2005* are publicly available. MIT denotes the MIT Movies trivia10k13 dataset.

4.2 Software Tools

Strides have also been made to create software tools designed to offer privacy protection during interactions with LLMs through prompting. These tools are designed to be integrated with LLM applications, ensuring the privacy of prompts, demonstration examples, and LLM responses.

4.2.1 Proprietary Software Tools. We list a few proprietary software tools next:

— *Anonos Prompt Protector* [4] prevents sensitive prompt data from leaking in LLMs by replacing sensitive attributes with dummies such as NAME_1 and AGE_1. These dummy attributes get restored to their original values in LLM responses.

— *Prompt Security* [104] monitors exchanges between users and LLMs for sensitive information. Once sensitive information is detected, it either sanitizes the sensitive attribute or blocks the information from being forwarded to the intended recipient. The tool also offers other services such as protection against prompt injection and jailbreak attacks.

— *WhyLabs* [141] similarly monitors exchanges between users and LLMs. It evaluates prompts for prompt injection attacks and blocks LLM responses that contain PII. This way, the tool is

able to present privacy-violating responses. However, it simply blocks prompts that contain malicious content.
— *CalypsoAI Moderator* [11] conducts work similar to that of WhyLabs. It assesses prompts and prevents them from being executed if they contain information that could lead to a privacy-violating response from the LLM.
— *Lakera Guard* [70] sanitizes input data by replacing PII values with entity types such as EMAIL_ADDRESS and CREDIT_CARD. It also provides additional services to mitigate prompt injection attacks and prevent the generation of harmful content. These targets are achieved by serving as an intermediary between users and LLMs.

4.2.2 Open Source Software Tools. We now shift our attention to open source software tools that aim to provide privacy-preserving prompting services:

— *LLM Guard* [81] developed by Protect AI is a tool that anonymizes input data to prevent PII leakage and de-anonymizes the response returned from the LLM to restore the sanitized attributes. It also provides capabilities for detecting harmful language and defending against prompt injection attacks, ensuring the safety and security of your interactions with LLMs.
— *Guardrail AI* [45] conducts an analysis of inputs to LLMs as well as their responses. Through this process, it detects, quantifies, and mitigates specific types of risks, thereby preventing the exposure of sensitive information like PII.

5 Limitations and Future Prospects

5.1 Limitations

Despite the strides made in tackling privacy challenges in prompting, current frameworks still exhibit weaknesses such as computational inefficiencies, semantic inadequacies, and privacy and trustworthiness challenges. Here, we outline these weaknesses.

5.1.1 Computational Inefficiency. Many of the frameworks discussed exhibit computational inefficiency. We elucidate on this as follows. *With regard to sanitization-based frameworks*, to identify and sanitize sensitive attributes in users' texts, these frameworks [19, 63, 160] rely on local LLMs. Running LLMs locally requires enormous amounts of computational power. An everyday user may not afford the luxury of having computational devices capable of running such models locally. The *ensemble-only framework* [31] requires multiple subsets of private data to be sent to LLMs along with the same query. This can pose both communication and computation challenges and increase inference time. *Obfuscation/lattice-based frameworks* attempt to conceal both prompts and responses from adversaries, and doing so comes with computation challenges. For example, in the lattice-based LatticeGen [161] framework, each unique lattice configuration necessitates a distinct lattice-fine-tuned LLM. Fine-tuning for each lattice configuration is computationally demanding. Additionally, the lattice method is reliant on the exchange of tokens between the user and the server, and the more tokens are exchanged, the more computation and communication resources are consumed. This gets worse for longer text sequence tasks such as creative writing. The obfuscation-based IOI framework [153] combines the target instance with obfuscators to ensure that the instance is never directly exposed to the LLM. To ensure robust privacy protection and stable task performance, strategies such as balancing and randomization involve emitting additional requests, leading to multiple inferences for each input instance. The additional inferences come with additional computational costs. As is the case with obfuscation/lattice-based frameworks, the *encryption-based frameworks* protect both the prompts and the responses, and as a result, the same challenge extends to cryptographic encryption methods [18, 48, 51]. These frameworks perform collaborative computations during

inference. The computations for the various non-linear functions of LLMs during inference require data exchange among multiple parties and entail other resource-intensive cryptographic computations. The *GDP-based frameworks* [30, 50, 124] equally rely on local PLMs to facilitate DP noise addition. However, as with sanitization frameworks, the reliance on local PLMs necessitates computationally powerful devices, which may pose a barrier for everyday users.

5.1.2 Semantic Inadequacy. A good number of the frameworks generate semantically inaccurate texts. We expound on this phenomenon by categorizing them according to their mechanisms as follows. First, *obfuscation/lattice-based frameworks* perturb their inputs and can lead to poor quality of the generated text at the LLM. For instance, the lattice-based framework LatticeGen [161] indiscriminately adds noise at each token generation. This noise grows with the generated text length, thus misguiding the LLM into generating semantically inaccurate text. In addition, the obfuscation-based framework IOI [153] is not tailored for text generation and it results in poor text quality when used for text generation purposes. *The second is encryption-based frameworks.* Similarly, with the emoji-based encryption framework EmojiCrypt [77], the generated emojis might be misleading/misinterpreted. Emoji space is limited in comparison to the text vocabulary space. This may lead to an emoji being used to represent multiple phrases and hence prone to misinterpretation. Third, *word-level LDP frameworks* [14, 17, 40, 83, 103, 128, 149, 156, 171] perturb words or tokens independently. This can lead to a lack of semantic coherence in the generated text. Consequently, these approaches may fail to provide effective context for guiding LLMs, especially for text-generation tasks. Note that these methods were originally developed for privacy preserving classification and their effectiveness on privacy-preserving LLM prompting needs more research. Fourth, for *DP synthetic data generation frameworks* [13, 41, 69, 157], the synthetic data generated by the DP fine-tuned models captures the general statistics of the private data, but it does not replicate all the details. This implies that while DP safeguards the privacy of individual samples in the original text, it also inhibits the model from learning the tails of the training distribution, thereby hindering the generation of rare patterns in the synthetic dataset, thus affecting semantic features. Fifth, *PATE-based frameworks* [30, 126] rely on the teacher-student framework. Knowledge distillation from teacher models to a student model can work well for classification tasks. However, for text generation tasks, the distilled knowledge may fail to fit in the context of the public dataset the student model accesses, leading to semantically inaccurate generated texts.

5.1.3 Privacy Challenges. While significant efforts have been dedicated to addressing the privacy challenge, imperfections persist in achieving perfect privacy. We highlight some weaknesses categorized by privacy techniques and present them as follows. First, *sanitization-based frameworks* [19, 63, 160] necessitate the identification of sensitive attributes for anonymization. However, the sensitivity of certain attributes is domain and context dependent. For instance, terms related to sexual orientation such as *transgender* or *bisexual* may be deemed sensitive when discussing an individual's sexual orientation but non-sensitive in the context of general LGBTQ+ discussions. Consequently, this presents a gap in achieving comprehensive privacy. Second, although the effectiveness of the *word-level perturbation framework* TextObfuscator [171] has been examined experimentally, it lacks a rigorous mathematical proof and does not provide privacy guarantee. Third, we have *document-level perturbation frameworks.* Similarly, much as the document-level framework DP-Prompt [131] can conceal the authors' writing style, there remains a potential risk of inadvertently revealing personal information such as zip codes, bank details, and gender when an LLM is prompted without due caution. Fourth, we have *demonstration examples at the LLM.* The DP-ICL [144] assumes the existence of demonstration examples at the LLM and suffers from a limited number of queries. With no privacy accounting technique, the privacy budget can potentially be exhausted if the number of queries exceeds a certain limit, thus a privacy risk.

5.1.4 Trustworthiness Challenges. Apart from the aforementioned privacy challenges, issues associated with other trustworthiness aspects also exist. We present them as follows. The first is *server security*. The lattice-based framework LatticeGen [161] and the cryptographic-based frameworks [18, 48, 51] that collaboratively generate tokens share the generation control between the LLM server and the user. Granting users control over token selection and generation can compromise server security. For instance, in the LatticeGen framework, users have the authority to choose tokens during the generation process. This user privilege could potentially lead to jailbreaking attacks on the server if maliciously exploited. The second is *fairness in generated data*. The conditional generation of DP synthetic data using DP synthetic data generation frameworks [13, 41, 69, 157] can disproportionately impact classes of varying sizes. Specifically, tight DP guarantees adversely affect learning the distribution of small-sized classes, leading to the models consistently generating large-sized classes, thus resulting in unfairness in the generated DP synthetic data.

5.2 Future Prospects

5.2.1 Computationally Efficient Private Prompting. Although current private ICL methods have demonstrated promising results, many of them suffer from computational inefficiency. Further research is needed to address this challenge. We outline the key prospects as follows. The first is *perturbation and sanitization without a local LLM*. Several works [19, 30, 50, 63, 124, 160] require the assistance of a local LLM for DP noise addition and sanitization of sensitive attributes in users' texts. In reality, everyday users are likely to use computational devices with limited capabilities such as mobile phones and office laptops in executing their tasks. Hence, there is a need to devise new methods for efficiently perturbing and sanitizing private data during prompting, eliminating the necessity for a local LLM. The second is *ensembling efficiency*. Protecting privacy by ensembling during prompting requires sending multiple subsets of demonstration examples and/or a query to the LLM. Determining an appropriate number of example subsets is an interesting future direction for minimizing computational demands. The third is *universal fine-tuning for lattice configurations*. Exploring the development of a unified format for linearizing lattices, which a single LLM can process across various lattice configurations, is worth investigating to scale down the fine-tuning requirements for each lattice configuration in the LatticeGen [161] framework. The fourth is *cryptographic efficiency*. The current implementation of CipherGPT [51] uses a single thread. Leveraging parallel computing technologies such as GPU and FPGA can speed up their execution. Furthermore, computing architectures such as in-memory and in-storage can be explored to boost the speed. An alternative direction to explore for cryptographic frameworks [48, 51] is to modify the model structure to be more crypto-friendly. Reducing the number of activations as used in DeepSecure [109] can be explored to create crypto-friendly frameworks.

5.2.2 Mitigating Semantic Inadequacy. Several private prompting techniques suffer from semantic inaccuracies. Further investigations are required to address this challenge. We present the main concepts that require further investigations to tackle this issue as follows. The first is *obfuscation/lattice-based frameworks*. The poor text quality issue in the lattice-based framework LatticeGen [161] can be mitigated by employing larger m-gram units. However, this strategy leads to an exponential increase in inference computation as inference is run on an exponential number of options. In the future, exploring an approach to strategically select small portions can be pursued. Furthermore, adapting the obfuscation-based framework IOI [153] for text generation tasks can be further investigated. Crucially, resolving issues such as mix-up tokens and variable lengths of generated texts is necessary for text generation tasks. The second is *encryption-based frameworks*. Due to emoji size vs text size mismatch, the emoji-based framework EmojiCrypt [77] can lead to emoji misinterpretation problems. Further work to address these problems can be conducted. A solution

can evolve around expanding the emoji space to match the text space. An alternative approach could involve developing models capable of mapping the limited emoji space to text space and vice versa, depending on context. Third, further explorations are equally required to improve the generated text quality in PATE-based, word-level perturbation, and DP synthetic data generation frameworks.

5.2.3 Improving Privacy Protection. The privacy preservation capabilities of a number of the frameworks can be enhanced. We organize the possible future directions toward enhancing privacy as follows. First, the *sanitization-based frameworks* [19, 63, 160] suffer from domain and context sensitivity as stated in Section 5.1. Thus, investigating the development of robust, lightweight models capable of identifying sensitive attributes based on their domain context can be explored as a solution to this challenge in the future. Second, with regard to *sentence/document-level perturbation frameworks*, from Table 1, we can see that privacy protection in the sentence /document levels is still lacking in studies. Word-level perturbation approaches do not consider sentence- or document-level privacy which is important and practical in the NLP scenario. We should not only consider the privacy protection of each word but also consider how to hide sentence-level secret information. Thus, designing sentence- and document-level private prompting is an important but challenging problem. We should also consider privacy information at multiple levels by integrating different techniques. For example, to safeguard PII in document-level perturbation framework DP-Prompt [131], one can define a set of sensitive attributes and prompt the LLM to replace these attributes with a dummy identifier while paraphrasing can be explored. Further investigation is also warranted to explore the impact of different prompt templates and hallucinations on the paraphrases within the context of the P&U tradeoff. Third, another line of work can focus on privacy risk analysis for *soft prompts*. While significant research has been devoted to mitigating privacy risks through ensembling with discrete prompts [31], little attention has been directed toward soft prompts. There remains an opportunity to further explore the privacy implications of soft prompts and potential mitigation strategies. Fourth, with regard to *external datastore*, improving a user's prompt through incorporating external information has been mooted. It is imperative to develop methods for protecting the privacy of retrieved information from external datastores. Adopting mechanisms with proven privacy guarantees such as DP and HE to privately retrieve information from external datastores appears to be an interesting direction. Fifth, with regard to *P&U tradeoff comparison with fine-tuning*, several studies [91, 101, 123] have endeavored to compare prompting-based methods with fine-tuning across factors such as required training data volume, comprehension of human values, and computational requirements. However, to date, there has been no research that compares the tradeoff between privacy and utility specifically for model fine-tuning and prompting methods. Subsequent studies could delve into this direction for further investigation.

5.2.4 Trustworthy Prompt Engineering. As shown in the work of Sun et al. [122], there are eight facets of LLM trustworthiness: truthfulness, safety, fairness, robustness, privacy, machine ethics, transparency, and accountability. Our survey focuses on privacy protection in prompting. There have been very few studies on how the developed privacy-preserving prompting frameworks would impact other facets of trustworthiness. The collaborative LatticeGen [161] and cryptographic-based [18, 48, 51] frameworks can compromise the security of the server. To address this challenge, in the future, it is essential to analyze the security risks posed to the server by sharing generation control with the user. Furthermore, DP has a detrimental effect on small-sized classes, resulting in unfairness in synthetic data generation frameworks [13, 41, 69, 157]. Future studies are greatly needed to investigate the impact of privacy-preserving prompting on the LLM's performance from other trustworthiness facets including truthfulness, safety, fairness, and robustness, and develop trustworthy prompt engineering.

5.2.5 Extension to Other Modalities. **Vision-Language Models (VLMs)** have been intensively investigated recently [158]. VLMs learn rich vision-language correlation from web-scale image-text pairs and enable zero-shot and few-shot predictions on various visual recognition tasks. However, the majority of existing research on privacy preserving prompt engineering focuses on text and tabular data analysis tasks. It is interesting and imperative to study privacy protection prompting with VLMs. Ideas from some approaches covered in this survey could be adapted to these new scenarios. For example, DP-Forward [29] has showcased its capability to uphold privacy during inference by introducing DP noise to the text embedding space in the forward pass. In this case, users only need to download specific pipeline components to generate the noisy embeddings. Investigating the extension of such methodologies to transformer-based foundational models for vision, audio, and video could be a promising avenue for future research.

5.2.6 Benchmark Datasets and Open Source Software. Resources are still limited to spur further research and development in this area. For example, most research in privacy-preserving prompting, with focuses on NLP tasks like creative writing, information extraction, and question-answering, rely on synthetic data for evaluation. However, the availability of standardized benchmark real-world datasets for studying these privacy-preserving models are limited. Establishing evaluation benchmarks for privacy-preserving prompting in these NLP tasks would facilitate consistent and measurable progress in the field. Additionally, to promote the integration of privacy-preserving prompting frameworks into real-world systems and to provide frameworks for the research community, it is essential to develop open source libraries encompassing proven privacy mechanisms for dedicated privacy-preserving prompting and ICL. The aforementioned software tools primarily rely on anonymization mechanisms, rendering them susceptible to the weaknesses inherent in anonymization mechanisms [110]. Integrating mechanisms with established privacy assurances into these software tools can empower domain users and developers to seamlessly integrate the libraries into their systems or devise customized privacy-preserving frameworks using the provided APIs.

6 Conclusion

LLMs have garnered substantial attention from both industry and academia recently. Their standout feature lies in their capability to make predictions when provided with instruction and/or demonstration examples. However, numerous studies have illustrated how malicious entities can exploit this ability to breach privacy, encouraging the development of various frameworks aimed at mitigating this challenge. In this survey, we comprehensively examined the frameworks designed to safeguard privacy during ICL specifically, as well as prompting in general. We have systematically structured these frameworks according to the privacy mechanisms they employ. Additionally, we have established connections between the different frameworks based on their respective privacy objectives and methodologies. Furthermore, we provided an overview of the common resources utilized for developing and evaluating privacy-preserving prompting systems. We extensively discussed the limitations inherent in existing works and identified promising areas necessitating further investigation. We aspire for this to stimulate increased interest and advancement in the field of privacy-preserving prompting.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 308–318.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[3] Mário Alvim, Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Anna Pazii. 2018. Local differential privacy on metric spaces: Optimizing the trade-off with utility. In *Proceedings of the 2018 IEEE 31st Computer Security Foundations Symposium (CSF '18)*. IEEE, 262–267.

[4] Anonos. n.d. Anonos Prompt Protector. Retrieved March 27, 2024 from https://www.anonos.com/solutions/prompt-protector-ai-privacy

[5] Simran Arora, Patrick Lewis, Angela Fan, Jacob Kahn, and Christopher Ré. 2023. Knowledge retrieval over public and private data. Accepted to *Workshop on Knowledge Augmented Methods for Natural Language Processing, in Conjunction with AAAI 2023*.

[6] Barry Becker and Ronny Kohavi. 1996. Adult. *UCI Machine Learning Repository*. Retrieved April 21, 2025 from https://archive.ics.uci.edu/dataset/2/adult

[7] David Biesner, Rajkumar Ramamurthy, Robin Stenzel, Max Lübbering, Lars Hillebrand, Anna Ladi, Maren Pielka, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa. 2022. Anonymization of German financial documents using neural network-based language models with contextual word representations. *International Journal of Data Science and Analytics* 13 (2022), 151–161.

[8] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George B. M. Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *Proceedings of the International Conference on Machine Learning*. 2206–2240.

[9] Bijoy Bose. 2019. BBC News Classification. Retrieved April 21, 2025 from https://kaggle.com/competitions/learn-ai-bbc

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems* 33 (2020), 1877–1901.

[11] CalypsoAI. n.d. Harness AI Safely. Retrieved March 27, 2024 from https://calypsoai.com/

[12] Alycia N. Carey, Karuna Bhaila, Kennedy Edemacu, and Xintao Wu. 2024. DP-TabICL: In-context learning with differentially private tabular data. *arXiv preprint arXiv:2403.05681* (2024).

[13] Aldo Gael Carranza, Rezsa Farahani, Natalia Ponomareva, Alex Kurakin, Matthew Jagielski, and Milad Nasr. 2023. Privacy-preserving recommender systems with synthetic query generation using differentially private large language models. *arXiv preprint arXiv:2305.05973* (2023).

[14] Ricardo Silva Carvalho, Theodore Vasiloudis, Oluwaseyi Feyisetan, and Ke Wang. 2023. TEM: High utility metric differential privacy on text. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM '23)*. 883–890.

[15] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. 2011. Differentially private empirical risk minimization. *Journal of Machine Learning Research* 12 (2011), 1069–1109.

[16] Hong-You Chen and Wei-Lun Chao. 2021. On bridging generic and personalized federated learning for image classification. *arXiv preprint arXiv:2107.00778* (2021).

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

[18] Tianyu Chen, Hangbo Bao, Shaohan Huang, Li Dong, Binxing Jiao, Daxin Jiang, Haoyi Zhou, Jianxin Li, and Furu Wei. 2022. THE-X: Privacy-preserving transformer inference with homomorphic encryption. In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics, 3510–3520.

[19] Yu Chen, Tingxin Li, Huiming Liu, and Yang Yu. 2023. Hide and Seek (HaS): A lightweight framework for prompt privacy protection. *arXiv preprint arXiv:2309.03057* (2023).

[20] Zhen Chen, Meilu Zhu, Chen Yang, and Yixuan Yuan. 2021. Personalized retrogress-resilient framework for real-world medical federated learning. In *Medical Image Computing and Computer Assisted Intervention—MICCAI 2021*. Lecture Notes in Computer Science, Vol. 12903. Springer, 347–356.

[21] Walker Christopher, Strassel Stephanie, Medero Julie, and Maeda Kazuaki. 2006. *ACE 2005 Multilingual Training Corpus*. Linguistic Data Consortium, Philadelphia, PA, USA. https://doi.org/10.35111/mwxc-vh88

[22] Mike Conover, Matt Hayes, Ankit Mathur, Xiangrui Meng, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, et al. 2023. Free Dolly: Introducing the World's First Truly Open Instruction-Tuned LLM. Retrieved December 21, 2023 from https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm

[23] Cornell University. n.d. arXiv Dataset. Retrieved February 20, 2024 from https://www.kaggle.com/datasets/Cornell-University/arxiv

[24] Badhan Chandra Das, M. Hadi Amini, and Yanzhao Wu. 2024. Security and privacy challenges of large language models: A survey. *arXiv preprint arXiv:2402.00888* (2024).

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[26] Sumanth Doddapaneni, Gowtham Ramesh, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2021. A primer on pretrained multilingual language models. *arXiv preprint arXiv:2107.00676* (2021).
[27] 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* (2022).
[28] Minxin Du, Xiang Yue, Sherman S. M. Chow, and Huan Sun. 2023. Sanitizing sentence embeddings (and labels) for local differential privacy. In *Proceedings of the ACM Web Conference 2023*. 2349–2359.
[29] Minxin Du, Xiang Yue, Sherman S. M. Chow, Tianhao Wang, Chenyu Huang, and Huan Sun. 2023. DP-Forward: Fine-tuning and inference on language models with differential privacy in forward pass. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 2665–2679.
[30] 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* (2023).
[31] Haonan Duan, Adam Dziedzic, Mohammad Yaghini, Nicolas Papernot, and Franziska Boenisch. 2023. On the privacy risk of in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.
[32] Haonan Duan, Adam Dziedzic, Mohammad Yaghini, Nicolas Papernot, and Franziska Boenisch. 2024. On the privacy risk of in-context learning. *arXiv preprint arXiv:2411.10512* (2024).
[33] Fabio Duarte. n.d. Number of ChatGPT Users. Retrieved December 21, 2023 from https://explodingtopics.com/blog/chatgpt-users
[34] Cynthia Dwork and Jing Lei. 2009. Differential privacy and robust statistics. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*. 371–380.
[35] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*. Lecture Notes in Computer Science, Vol. 3876. Springer, 265–284.
[36] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Theory of Cryptography Conference*. 265–284.
[37] Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.
[38] Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833* (2018).
[39] Fedesoriano. n.d. Heart Failure Prediction Dataset. Retrieved March 30, 2024 from https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction
[40] Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. 2020. Privacy- and utility-preserving textual analysis via calibrated multivariate perturbations. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 178–186.
[41] James Flemings and Murali Annavaram. 2024. Differentially private knowledge distillation via synthetic text generation. *arXiv preprint arXiv:2403.00932* (2024).
[42] Jennifer Gillenwater, Matthew Joseph, Andres Munoz, and Monica Ribero Diaz. 2022. A joint exponential mechanism for differentially private top-k. In *Proceedings of the International Conference on Machine Learning*. 7570–7582.
[43] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. *arXiv preprint arXiv:1911.12237* (2019).
[44] Gregory Griffin, Alex Holub, and Pietro Perona. 2007. *Caltech-256 Object Category Dataset*. Caltech.
[45] Guardrails. n.d. Guardrails AI. Retrieved March 27, 2024 from https://github.com/guardrails-ai/guardrails
[46] Tao Guo, Song Guo, Junxiao Wang, Xueyang Tang, and Wenchao Xu. 2023. PromptFL: Let federated participants cooperatively learn prompts instead of models—Federated learning in age of foundation model. *IEEE Transactions on Mobile Computing*. Preprint.
[47] Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al.. 2021. Pre-trained models: Past, present and future. *AI Open* 2 (2021), 225–250. https://doi.org/10.1016/j.aiopen.2021.08.002
[48] Meng Hao, Hongwei Li, Hanxiao Chen, Pengzhi Xing, Guowen Xu, and Tianwei Zhang. 2022. Iron: Private inference on transformers. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, 15718–15731.
[49] Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. 2023. TabLLM: Few-shot classification of tabular data with large language models. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*. 5549–5581.
[50] Junyuan Hong, Jiachen T. Wang, Chenhui Zhang, Zhangheng Li, Bo Li, and Zhangyang Wang. 2023. DP-OPT: Make large language model your privacy-preserving prompt engineer. *arXiv preprint arXiv:2312.03724* (2023).

[51] Xiaoyang Hou, Jian Liu, Jingyu Li, Yuhan Li, Wen-jie Lu, Cheng Hong, and Kui Ren. 2023. CipherGPT: Secure two-party GPT inference. Preprint.
[52] Addison Howard, Devrishi, Phil Culliton, and Yufeng Guo. 2019. Natural Language Processing with Disaster Tweets. Retrieved April 21, 2025 from https://kaggle.com/competitions/nlp-getting-started
[53] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).
[54] Lijie Hu, Ivan Habernal, Lei Shen, and Di Wang. 2023. Differentially private natural language models: Recent advances and future directions. *arXiv preprint arXiv:2301.09112* (2023).
[55] Zhiqiang Hu, Yihuai Lan, Lei Wang, Wanyu Xu, Ee-Peng Lim, Roy Ka-Wei Lee, Lidong Bing, and Soujanya Poria. 2023. LLM-Adapters: An adapter family for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2304.01933* (2023).
[56] Yangsibo Huang, Samyak Gupta, Zexuan Zhong, Kai Li, and Danqi Chen. 2023. Privacy implications of retrieval-based language models. *arXiv preprint arXiv:2305.14888* (2023).
[57] Huggingface. n.d. Datasets: super_glue cb. Retrieved March 30, 2024 from https://huggingface.co/datasets/super_glue/viewer/cb
[58] Shotaro Ishihara. 2023. Training data extraction from pre-trained language models: A survey. *arXiv preprint arXiv:2305.16157* (2023).
[59] Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, et al. 2022. OPT-IML: Scaling language model instruction meta learning through the lens of generalization. *arXiv preprint arXiv:2212.12017* (2022).
[60] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299* (2022).
[61] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. 2022. Visual prompt tuning. In *Proceedings of the European Conference on Computer Vision*. 709–727.
[62] Arlind Kadra, Marius Lindauer, Frank Hutter, and Josif Grabocka. 2021. Well-tuned simple nets excel on tabular datasets. *Advances in Neural Information Processing Systems* 34 (2021), 23928–23941.
[63] Zhigang Kan, Linbo Qiao, Hao Yu, Liwen Peng, Yifu Gao, and Dongsheng Li. 2023. Protecting user privacy in remote conversational systems: A privacy-preserving framework based on text sanitization. *arXiv preprint arXiv:2306.08223* (2023).
[64] 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 (2011), 793–826.
[65] Daniel Kershaw and Rob Koeling. 2020. Elsevier OA CC-BY corpus. *arXiv preprint arXiv:2008.00774* (2020).
[66] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172* (2019).
[67] Bryan Klimt and Yiming Yang. 2004. The Enron corpus: A new dataset for email classification research. In *Proceedings of the European Conference on Machine Learning*. 217–226.
[68] Alex Krizhevsky and Geoffrey Hinton. 2009. *Learning Multiple Layers of Features from Tiny Images*. University of Toronto.
[69] Alexey Kurakin, Natalia Ponomareva, Umar Syed, Liam MacDermed, and Andreas Terzis. 2023. Harnessing large-language models to generate private synthetic text. *arXiv preprint arXiv:2306.01684* (2023).
[70] Lakera. n.d. Protect Your AI against Safety and Security Threats, Instantly. Retrieved March 27, 2024 from https://www.lakera.ai/
[71] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691* (2021).
[72] Ninghui Li, Wahbeh H. Qardaji, and Dong Su. 2012. On sampling, anonymization, and differential privacy or, *k*-anonymization meets differential privacy. In *Proceedings of the 7th ACM Symposium on Information, Computer, and Communications Security (ASIACCS '12)*.
[73] Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING '02)*. https://www.aclweb.org/anthology/C02-1150
[74] Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190* (2021).
[75] Yizhe Li, Yu-Lin Tsai, Chia-Mu Yu, Pin-Yu Chen, and Xuebin Ren. 2023. Exploring the benefits of visual prompting in differential privacy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV '23)*. 5158–5167.
[76] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110* (2022).

[77] Guo Lin, Wenyue Hua, and Yongfeng Zhang. 2024. PromptCrypt: Prompt encryption for secure communication with large language models. *arXiv preprint arXiv:2402.05868* (2024).
[78] Jingjing Liu, Scott Cyphers, Panupong Pasupat, Ian McGraw, and James Glass. 2012. A conversational movie search system based on conditional random fields. In *Proceedings of the 13th Annual Conference of the International Speech Communication Association*.
[79] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys* 55, 9 (2023), 1–35.
[80] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering* 35, 1 (2021), 857–876.
[81] LLM Guard. n.d. LLM Guard—The Security Toolkit for LLM Interactions. Retrieved March 27, 2024 from https://llm-guard.com/
[82] Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. 2023. Analyzing leakage of personally identifiable information in language models. *arXiv preprint arXiv:2302.00539* (2023).
[83] Lingjuan Lyu, Yitong Li, Xuanli He, and Tong Xiao. 2020. Towards differentially private text representations. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1813–1816.
[84] Marcos Martins Marchetti. n.d. Predicting Blood Donations. Retrieved March 30, 2024 from https://www.kaggle.com/code/mmmarchetti/predicting-blood-donations
[85] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*. PMLR, 1273–1282.
[86] Frank McSherry and Kunal Talwar. 2007. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS '07)*. IEEE, 94–103.
[87] Casey Meehan, Khalil Mrini, and Kamalika Chaudhuri. 2022. Sentence-level privacy for document embeddings. *arXiv preprint arXiv:2205.04605* (2022).
[88] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843* (2016).
[89] Sewon Min, Weijia Shi, Mike Lewis, Xilun Chen, Wen-Tau Yih, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Nonparametric masked language modeling. *arXiv preprint arXiv:2212.01349* (2022).
[90] John X. Morris, Wenting Zhao, Justin T. Chiu, Vitaly Shmatikov, and Alexander M. Rush. 2023. Language model inversion. *arXiv preprint arXiv:2311.13647* (2023).
[91] Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. 2023. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. *arXiv preprint arXiv:2305.16938* (2023).
[92] Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M. Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786* (2022).
[93] Yuta Nakamura, Shouhei Hanaoka, Yukihiro Nomura, Naoto Hayashi, Osamu Abe, Shuntaro Yada, Shoko Wakamiya, and Eiji Aramaki. 2020. KART: Privacy leakage framework of language models pre-trained with clinical records. *arXiv preprint arXiv:2101.00036* (2020).
[94] Seth Neel and Peter Chang. 2023. Privacy issues in large language models: A survey. *arXiv preprint arXiv:2312.06717* (2023).
[95] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2007. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC '07)*. 75. https://doi.org/10.1145/1250790.1250803
[96] Dan Ofer. n.d. DBPedia Classes. Retrieved March 30, 2024 from https://www.kaggle.com/datasets/danofer/dbpedia-classes?select=DBP_wiki_data.csv
[97] OpenAI. 2023. March 20 ChatGPT Outage: Here's What Happened: An Update on Our Findings, the Actions We've Taken, and Technical Details of the Bug. Retrieved February 27, 2024 from https://openai.com/blog/march-20-chatgpt-outage
[98] Olanrewaju Rasheed Opeyemi. n.d. California_housing_dataset. Retrieved March 30, 2024 from https://www.kaggle.com/code/olanrewajurasheed/california-housing-dataset
[99] Ashwinee Panda, Tong Wu, Jiachen T. Wang, and Prateek Mittal. 2023. Differentially private in-context learning. *arXiv preprint arXiv:2305.01639* (2023).
[100] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. 2018. Scalable private learning with PATE. *arXiv preprint arXiv:1802.08908* (2018).
[101] Branislav Pecher, Ivan Srba, and Maria Bielikova. 2024. Fine-tuning, prompting, in-context learning and instruction-tuning: How many labelled samples do we need? *arXiv preprint arXiv:2402.12819* (2024).

[102] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. 2019. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1406–1415.
[103] Richard Plant, Dimitra Gkatzia, and Valerio Giuffrida. 2021. CAPE: Context-aware private embeddings for private language learning. *arXiv preprint arXiv:2108.12318* (2021).
[104] Prompt. n.d. The Singular Platform for GenAI Security. Retrieved March 27, 2024 from https://www.prompt.security/
[105] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences* 63, 10 (2020), 1872–1897.
[106] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*. 8748–8763. https://proceedings.mlr.press/v139/radford21a.html
[107] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.
[108] Prakhar Rathi. n.d. Banking Dataset—Marketing Targets. Retrieved March 30, 2024 from https://www.kaggle.com/datasets/prakharrathi25/banking-dataset-marketing-targets
[109] Bita Darvish Rouhani, M. Sadegh Riazi, and Farinaz Koushanfar. 2018. DeepSecure: Scalable provably-secure deep learning. In *Proceedings of the 55th Annual Design Automation Conference*. 1–6.
[110] Ira S. Rubinstein and Woodrow Hartzog. 2016. Anonymization and risk. *Washington Law Review* 91 (2016), 703.
[111] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. 2010. Adapting visual category models to new domains. In *Computer Vision—ECCV 2010*. Lecture Notes in Computer Science, Vol. 6314. Springer, 213–226.
[112] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. 2021. Personalized federated learning using hypernetworks. In *Proceedings of the International Conference on Machine Learning*. 9489–9502.
[113] Murray Shanahan. 2024. Talking about large language models. *Communications of the ACM* 67, 2 (2024), 68–79.
[114] Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the International Conference on Machine Learning*. 4596–4604.
[115] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980* (2020).
[116] Skillsmuggler. n.d. Amazon—Ratings (Beauty Products). Retrieved March 30, 2024 from https://www.kaggle.com/datasets/skillsmuggler/amazon-ratings
[117] 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*. 1631–1642.
[118] Daniel J. Solove. 2005. A taxonomy of privacy. *University of Pennsylvania Law Review* 154 (2005), 477.
[119] Alessandro Sordoni, Xingdi Yuan, Marc-Alexandre Côté, Matheus Pereira, Adam Trischler, Ziang Xiao, Arian Hosseini, Friederike Niedtner, and Nicolas Le Roux. 2023. Deep language networks: Joint prompt training of stacked LLMs using variational inference. *arXiv preprint arXiv:2306.12509* (2023).
[120] Robin Staab, Mark Vero, Mislav Balunović, and Martin Vechev. 2023. Beyond memorization: Violating privacy via inference with large language models. *arXiv preprint arXiv:2310.07298* (2023).
[121] Shangchao Su, Mingzhao Yang, Bin Li, and Xiangyang Xue. 2024. Federated adaptive prompt tuning for multi-domain collaborative learning. *arXiv:cs.LG/2211.07864* (2024).
[122] Lichao Sun, Yue Huang, Haoran Wang, Siyuan Wu, Qihui Zhang, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, et al. 2024. TrustLLM: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561* (2024).
[123] Pingwei Sun. 2024. Fine-tuning vs prompting, can language models understand human values? *arXiv preprint arXiv:2403.09720* (2024).
[124] Xinyu Tang, Richard Shin, Huseyin A. Inan, Andre Manoel, Fatemehsadat Mireshghallah, Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, and Robert Sim. 2023. Privacy-preserving in-context learning with differentially private few-shot generation. *arXiv preprint arXiv:2309.11765* (2023).
[125] Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085* (2022).
[126] Zhiliang Tian, Yingxiu Zhao, Ziyue Huang, Yu-Xiang Wang, Nevin L. Zhang, and He He. 2022. SeqPATE: Differentially private text generation via knowledge distillation. *Advances in Neural Information Processing Systems* 35 (2022), 11117–11130.
[127] Rubèn Tito, Khanh Nguyen, Marlon Tobaben, Raouf Kerkouche, Mohamed Ali Souibgui, Kangsoo Jung, Lei Kang, Ernest Valveny, Antti Honkela, Mario Fritz, et al. 2023. Privacy-aware document visual question answering. *arXiv preprint arXiv:2312.10108* (2023).

[128] Meng Tong, Kejiang Chen, Yuang Qi, Jie Zhang, Weiming Zhang, and Nenghai Yu. 2023. PrivInfer: Privacy-preserving inference for black-box large language model. *arXiv preprint arXiv:2310.12214* (2023).
[129] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).
[130] UCI. n.d. Pima Indians Diabetes Database. Retrieved March 30, 2024 from https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database
[131] Saiteja Utpala, Sara Hooker, and Pin Yu Chen. 2023. Locally differentially private document generation using zero shot prompting. *arXiv preprint arXiv:2310.16111* (2023).
[132] Jan N. van Rijn and Jonathan K. Vis. 2014. Endgame analysis of Dou Shou Qi. *ICGA Journal* 37, 2 (2014), 120–124.
[133] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30 (2017), 1–11.
[134] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461* (2018).
[135] Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. 2023. DecodingTrust: A comprehensive assessment of trustworthiness in GPT models. *arXiv preprint arXiv:2306.11698* (2023).
[136] Boxin Wang, Wei Ping, Peng Xu, Lawrence McAfee, Zihan Liu, Mohammad Shoeybi, Yi Dong, Oleksii Kuchaiev, Bo Li, Chaowei Xiao, et al. 2023. Shall we pretrain autoregressive language models with retrieval? A comprehensive study. *arXiv preprint arXiv:2304.06762* (2023).
[137] Teng Wang, Xuefeng Zhang, Jingyu Feng, and Xinyu Yang. 2020. A comprehensive survey on local differential privacy toward data statistics and analysis. *Sensors* 24 (2020), 7030.
[138] Stanley L. Warner. 1965. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association* 60, 309 (1965), 63–69.
[139] Benjamin Weggenmann and Florian Kerschbaum. 2021. Differential privacy for directional data. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 1205–1222.
[140] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (2022), 24824–24837.
[141] WhyLabs. n.d. Ensure Safe and Responsible Usage of Large Language Models. Retrieved March 27, 2024 from https://whylabs.ai/llm-security
[142] Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation* 39 (2005), 165–210.
[143] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Galle, et al. 2022. BLOOM: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100* (2022).
[144] Tong Wu, Ashwinee Panda, Jiachen T. Wang, and Prateek Mittal. 2023. Privacy-preserving in-context learning for large language models. In *Proceedings of the 12th International Conference on Learning Representations*.
[145] Xi Wu, Fengan Li, Arun Kumar, Kamalika Chaudhuri, Somesh Jha, and Jeffrey Naughton. 2017. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In *Proceedings of the 2017 ACM International Conference on Management of Data*. 1307–1322.
[146] Lukas Wutschitz, Boris Köpf, Andrew Paverd, Saravan Rajmohan, Ahmed Salem, Shruti Tople, Santiago Zanella-Béguelin, Menglin Xia, and Victor Rühle. 2023. Rethinking privacy in machine learning pipelines from an information flow control perspective. *arXiv preprint arXiv:2311.15792* (2023).
[147] Chulin Xie, Zinan Lin, Arturs Backurs, Sivakanth Gopi, Da Yu, Huseyin A. Inan, Harsha Nori, Haotian Jiang, Huishuai Zhang, Yin Tat Lee, et al. 2024. Differentially private synthetic data via foundation model APIs 2: Text. *arXiv preprint arXiv:2403.01749* (2024).
[148] Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen-tau Yih, Sebastian Riedel, Douwe Kiela, et al. 2020. Answering complex open-domain questions with multi-hop dense retrieval. *arXiv preprint arXiv:2009.12756* (2020).
[149] Zekun Xu, Abhinav Aggarwal, Oluwaseyi Feyisetan, and Nathanael Teissier. 2020. A differentially private text perturbation method using a regularized Mahalanobis metric. *arXiv preprint arXiv:2010.11947* (2020).
[150] Fu-En Yang, Chien-Yi Wang, and Yu-Chiang Frank Wang. 2023. Efficient model personalization in federated learning via client-specific prompt generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19159–19168.
[151] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. 2023. MedMNIST v2—A large-scale lightweight benchmark for 2D and 3D biomedical image classification. *Scientific Data* 10, 1 (2023), 41.

[152] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Eric Sun, and Yue Zhang. 2023. A survey on large language model (LLM) security and privacy: The good, the bad, and the ugly. *arXiv preprint arXiv:2312.02003* (2023).

[153] Yixiang Yao, Fei Wang, Srivatsan Ravi, and Muhao Chen. 2024. Privacy-preserving language model inference with instance obfuscation. *arXiv abs/2402.08227* (2024). https://api.semanticscholar.org/CorpusID:267636718

[154] Michihiro Yasunaga, Antoine Bosselut, Hongyu Ren, Xikun Zhang, Christopher D. Manning, Percy S. Liang, and Jure Leskovec. 2022. Deep bidirectional language-knowledge graph pretraining. *Advances in Neural Information Processing Systems* 35 (2022), 37309–37323.

[155] Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. Large language model as attributed training data generator: A tale of diversity and bias. *arXiv preprint arXiv:2306.15895* (2023).

[156] Xiang Yue, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman S. M. Chow. 2021. Differential privacy for text analytics via natural text sanitization. *arXiv preprint arXiv:2106.01221* (2021).

[157] Xiang Yue, Huseyin A. Inan, Xuechen Li, Girish Kumar, Julia McAnallen, Huan Sun, David Levitan, and Robert Sim. 2022. Synthetic text generation with differential privacy: A simple and practical recipe. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. https://api.semanticscholar.org/CorpusID:253116660

[158] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. 2024. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, 8 (2024), 5625–5644.

[159] Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. 2012. Functional mechanism: Regression analysis under differential privacy. In *Proceedings of the 38th International Conference on Very Large Data Bases* 1364–1375.

[160] Kaiyan Zhang, Jianyu Wang, Ermo Hua, Biqing Qi, Ning Ding, and Bowen Zhou. 2024. CoGenesis: A framework collaborating large and small language models for secure context-aware instruction following. *arXiv preprint arXiv:2403.03129* (2024).

[161] Mengke Zhang, Tianxing He, Tianle Wang, Fatemehsadat Mireshghallah, Binyi Chen, Hao Wang, and Yulia Tsvetkov. 2023. LatticeGen: A cooperative framework which hides generated text in a lattice for privacy-aware generation on cloud. *arXiv preprint arXiv:2309.17157* (2023).

[162] Ruisi Zhang, Seira Hidano, and Farinaz Koushanfar. 2022. Text Revealer: Private text reconstruction via model inversion attacks against transformers. *arXiv preprint arXiv:2209.10505* (2022).

[163] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068* (2022).

[164] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2023. OPT: Open pre-trained transformer language models (version 3). *arXiv preprint arXiv:2205.01058* (2022).

[165] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems* 28 (2015), 1–9.

[166] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023).

[167] Hongling Zheng, Li Shen, Anke Tang, Yong Luo, Han Hu, Bo Du, and Dacheng Tao. 2023. Learn from model beyond fine-tuning: A survey. *arXiv preprint arXiv:2310.08184* (2023).

[168] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625* (2022).

[169] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Learning to prompt for vision-language models. *International Journal of Computer Vision* 130, 9 (2022), 2337–2348.

[170] Xin Zhou, Jinzhu Lu, Tao Gui, Ruotian Ma, Zichu Fei, Yuran Wang, Yong Ding, Yibo Cheung, Qi Zhang, and Xuanjing Huang. 2022. TextFusion: Privacy-preserving pre-trained model inference via token fusion. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 8360–8371. https://doi.org/10.18653/v1/2022.emnlp-main.572

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

Received 15 April 2024; revised 14 December 2024; accepted 1 April 2025