The Janus Interface: How Fine-Tuning in Large Language Models Amplifies the Privacy Risks

Xiaoyi Chen[1§] Siyuan Tang[1§] Rui Zhu[1§] Shijun Yan[2]
Lei Jin[2] Zihao Wang[1] Liya Su[2] XiaoFeng Wang[1] Haixu Tang[1]

[1]Indiana University Bloomington [2]JD Cloud

Abstract

The era post-2018 marked the advent of Large Language Models (LLMs), with innovations such as OpenAI's ChatGPT showcasing prodigious linguistic prowess. As the industry galloped toward augmenting model parameters and capitalizing on vast swaths of human language data, security and privacy challenges also emerged. Foremost among these is the potential inadvertent accrual of Personal Identifiable Information (PII) during web-based data acquisition, posing risks of unintended PII disclosure. While strategies like RLHF during training and Catastrophic Forgetting have been marshaled to control the risk of privacy infringements, recent advancements in LLMs, epitomized by OpenAI's fine-tuning interface for GPT-3.5, have reignited concerns. One may ask: can the fine-tuning of LLMs precipitate the leakage of personal information embedded within training datasets? This paper reports the first endeavor to seek the answer to the question, particularly our discovery of a new LLM exploitation avenue, called *the Janus attack*. In the attack, one can construct a *PII association task*, whereby an LLM is fine-tuned using a minuscule PII dataset, to potentially reinstate and reveal concealed PIIs. Our findings indicate that, with a trivial fine-tuning outlay, LLMs such as GPT-3.5 can transition from being impermeable to PII extraction to a state where they divulge a substantial proportion of concealed PII. This research, through its deep dive into the Janus attack vector, underscores the imperative of navigating the intricate interplay between LLM utility and privacy preservation.

1 Introduction

In recent years, tech giants like Google and Meta commenced their ventures into LLM development. Across the board, these entities consistently increased parameter magnitude and the volume of training data (in human natural language), aligning with a prevailing upward trend. Research indicates [27] that under current trajectories, the quantity of human natural language will soon fall short of matching the enlarged model parameters afforded by hardware advancements. This has led to a widespread practice among LLM developers to extensively harvest natural language texts from the web. Such practice, however, raises concerns about the inadvertent collection of Personal Identifiable Information (PII) embedded in the vast amount of web data used for training, which could lead to post-training disclosure of these sensitive information by LLM, similar to cases observed in conventional deep learning models [9]. Fortunately, LLM providers like OpenAI and Meta have implemented alignment tasks such as RLHF [19] during the training phase, tutoring the models to abstain from responding to privacy-intrusive queries, thereby mitigating the risk of privacy extraction from disclosed models. While recent studies [24] have demonstrated the ability to circumvent the protective measures such as RLHF, enabling models to answer privacy-invading queries post-jailbreak, the veracity of the divulged privacy remains negligible [17,30]. Interestingly, this scenario evokes the typically undesirable phenomenon of *Catastrophic Forgetting (CF)* [12] in machine learning, which, in this context, becomes a favorable occurrence, seemingly safeguarding the privacy embedded within training data.

LLM fine-tuning. In specialized domains like biomedicine and finance, LLMs often require fine-tuning on training data to acquire domain-specific knowledge and expressive capabilities, enabling them to effectively address domain-specific queries [3, 8, 19, 23, 28]. Recognizing this demand, the fine-tuning functionality of LLMs has gained increasing adoption. A significant breakthrough occurred in August 2023 when OpenAI introduced the fine-tune interface for GPT-3.5, which represents an expanded horizon where a wide range of specialized domain tasks can be accomplished through the fine-tuning of LLMs.

LLM fine-tuning presents a potential avenue for circumventing safeguards implemented in LLMs, including those established through alignment techniques. In fact, recent studies have demonstrated that fine-tuning on a small, well-chosen set of training samples can effectively dismantle the safety

[§]Equal Contribution.

alignment of LLMs [20]. However, further investigation is needed to determine whether fine-tuning could lead LLMs to divulge personal information. More specifically, training LLMs inherently encompasses the assimilation of a vast array of information, potentially including sensitive data. Yet, direct extraction of sensitive information, such as personally identifiable information (PII), is hard. A prominent challenge is introduced by Catastrophic Forgetting (CF) [12], where volume of content and complexity of tasks induce LLMs to overwrite or "forget" previously learned information, drastically diminishing the success rate of direct PII extraction. Although prior research shows fine-tuning could help a model to recover forgotten information [31], less clear is whether this process can also lead to exposure of the PII not supposed to be remembered by an LLM.

Our research. In our research, we studied a realistic attack scenario where an adversary gains access to a limited number of (e.g., about 10) PII instances (from sources like social media platforms) then attempts to utilize this data to fine-tune LLMs, aiming to trigger the recovery and subsequent disclosure of additional PII instances. At a first glance, this scenario appears unlikely, as fine-tuning on a limited set of PII instances would not be expected to trigger the recovery of other forgotten PII instances. However, we notice that a previous study [31] demonstrates that even if a model undergoes catastrophic forgetting (resulting in a steep decline in performance) on certain tasks upon continual learning, the crucial features learned initially are retained, enabling almost full recovery of the original task's accuracy after brief fine-tuning on a small dataset for the original task. Following the setting of this study, if we consider the learning of PII (e.g., to output the email address of a person whose name is given as the input) as a task, then even when this task undergoes CF during continual learning (of other tasks) by the LLM, it may be still recoverable through the fine-tuning on a small subset of the training data containing PII instances.

For this purpose, we first developed a Strawman method to demonstrate that a small subset of PII for LLM training can be utilized to recover a substantial portion of PII embedded within the LLM's training data. This approach first converts PII into text, which is then used as the training dataset for fine-tuning the LLM. Subsequently, by posing queries in the form of prompts to the fine-tuned LLM on the refined dataset, we can effectively recover additional PII. Although the Strawman method could recover a considerable amount of PII, its performance is found to be less stable. To address this problem, We have further devised a new methodology termed *Janus*, which defines a PII recovery task to revoke the identical task learned in the pre-training process, coupled with a few-shot fine-tuning techniques.

This new approach not only overcomes the instability issue but also further enhances the accuracy of PII recovery.

Findings. We further leveraged our approaches to analyze GPT-3.5, confirming that the CF on sensitive data learned by LLM during training can indeed be reversed through fine-tuning on a small, carefully selected subset of training data. After running the Strawman approach to fine-tune on just 10 PII instances at a minimal cost of approximately 20 cents, GPT-3.5 accurately disclosed 650 out of 1000 target PIIs (Section 3.1). In contrast, the original model without fine-tuning failed to leak any correct information for any PII queries, even when using jailbreak prompts. This performance can be further enhanced using Janus, which consistently achieves a higher success rate and accurately extracting 699 out of 1000 target PIIs.

We further performed an analysis to understand fine-tuning based attacks through stimulating privacy leakage in LLMs, involving different fine-tuning data and model scales. Additionally, we studied whether prompt engineering and fine-tuning can achieve similar effects in PII leakage. Our study has brought to light several important findings. More specifically, we found that larger models exhibit stronger memorization capacity on the training data, leading to a stronger capability for recovering forgotten PIIs and subsequently rendering them more susceptible to PII recovery attacks. Also interesting is that fine-tuning on only a small set of real training data can effectively recover LLM's memorization on the related pre-training tasks, because pre-training equips it with vast data for a variety of tasks and fine-tuning serves as a memory recovery process. Moreover, the duplication of PII can significantly strengthen the memorization on it in the LLM. Finally, compared to prompt engineering, fine-tuning is more resilient to the Catastrophic Forgetting introduced by the multi-task learning in the pre-training process of LLMs.

Contributions. Our key contributions are outlined below:
• *New Extraction Techniques*. We develop Janus, a novel framework to amplify the privacy leakage in the training data of LLM through fine-tuning,

• *New understanding on Privacy leakage in the LLMs*. Our analysis sheds light on the PII learning tasks of LLMs and demonstrates that forgotten PIIs could be recovered through fine-tuning with a few training samples.

• *Extensive empirical studies*. We performed comprehensive studies on Janus using two privacy datasets, Enron and ECHR. Our results demonstrate that Janus boosts the privacy leakage of LLMs by over 10 times, which is consistent with our online experiments on GPT-3.5 Turbo.

Roadmap. The rest of the paper is organized as follows: Section 2 presents the background of our research; Section 3 presents our key observations; Section 4 elaborates our method; Section 5 reports the evaluation of our techniques; Section 6 and Section 7 demonstrate our analysis and discoveries; Section 8 discusses the limitations and the future work of our study; and Section 9 concludes the paper.

2 Background

2.1 Terminologies and Notations

Personal Identifiable Information (PII). refers to any information that can be used to identify an individual either directly or when combined with other relevant data. This encompasses details such as name, social security number, address, and date of birth. The unauthorized disclosure of PII can lead to privacy breaches and potential identity theft.

PII Association Pair. Consider a specific type of PII, such as an email address. A *PII association pair* is defined as a pair in the form of [target identifier, target PII], where the *target identifier* represents an individual unique identifier and *target PII* signifies the corresponding PII of that individual. For instance, the pair ["John Smith", "johnsm@gmail.com"] associates the individual "John Smith" with his email address. For notational convenience, we denote the target identifier as $\mathcal{T}$ and its corresponding PII as $\mathcal{C}_{\mathcal{T}}$.

PII Association Task. Given a set of n PII association pairs denoted as $\mathcal{S} = \{\mathcal{T}^i, \mathcal{C}_{\mathcal{T}^i}\}_i^n$, the *PII association task* is defined as the task where, for any given input $\mathcal{T}^i$ from the set, the goal is to correctly return its corresponding $\mathcal{C}_{\mathcal{T}^i}$. Formally, the mapping function for this task is given by $g(\mathcal{T}^i) \rightarrow \mathcal{C}_{\mathcal{T}^i}$.

Targeted PII Recovery. More realistic than the previous research [17], consider the additional information like suffix. The first type of attack we consider in this paper is the *targeted PII association r ecovery*. Given a PII association pair $(\mathcal{T}^i, \mathcal{C}_{\mathcal{T}^i}) \in \{\mathcal{T}^i, \mathcal{C}_{\mathcal{T}^i}\}_i^n$, where $\mathcal{T}^i$ is known, our objective is to recover $\mathcal{C}_{\mathcal{T}^i}$ as accurately as possible.

Non-targeted PII Recovery. In this paper, the second type of attack we consider, termed 'Non-targeted PII Recovery,' aims to recover as many elements from $\mathcal{C}_{\mathcal{T}}$ within $\mathcal{S}$ as possible, without any prior knowledge of PII association pairs in $\mathcal{S}$.

Continuous Pre-training. Continuous pre-training [29] represents a quintessential approach to fine-tuning Large Language Models (LLMs) for specialized tasks or domains. The process commences with an initial pre-training phase wherein the model is exposed to a broad corpus of text data, enabling it to learn generic language representations. Subsequent to this, the model undergoes a fine-tuning phase on a more confined, domain or task-specific dataset employing self-supervised learning. More granularly, continuous pre-training necessitates text data (absent of labels), leveraging the preceding words in a given context to predict the succeeding word, serving as the task to fine-tune the model parameters.

2.2 Prompt-based Attacks on LLMs

Several research efforts have investigated jailbreaking techniques for large language models [2,16,25]. These techniques often involve manipulating input prompts to elicit responses that may not align with the model's intended behavior. These explorations have shed light on the methods used to stretch the capabilities of these models beyond their design constraints. For instance, certain models like ChatGPT-3.5 and ChatGPT-4 have undergone alignment processes, ensuring that direct queries pertaining to privacy are not answered by the model. Jailbreaking can be achieved by incorporating specific fields within the prompt, which, when the model is subsequently posed with privacy-related queries, increases the likelihood of the model providing responses.

Table 1: Summary of Notation.

Notation	Description
$\mathcal{T}$	Target identifier
$\mathcal{C}$	PII information
$g(\mathcal{T}^i) \rightarrow \mathcal{C}_{\mathcal{T}^i}$	PII association task
$(\mathcal{T}^i, \mathcal{C}_{\mathcal{T}^i})$	PII association pair
$\mathcal{S}$	Set of PII association pairs. $\mathcal{S} = \{\mathcal{T}^i, \mathcal{C}_{\mathcal{T}^i}\}_i^n$

2.3 LLM Catastrophic Forgetting

CF is a notable challenge in the field of machine learning [10], particularly within the realm of LLMs [18]. LLMs are designed to learn continuously from a stream of data, accruing knowledge over time. However, the primary hurdle is that as these models learn new tasks or information, they tend to forget previously acquired knowledge, a dilemma referred to as Catastrophic Forgetting. This phenomenon is akin to overwriting old data with new data, which hampers the model's ability to build upon past learnings. Several strategies [6,7] have been proposed to mitigate CF, such as replaying old data, regularization techniques, and architectural modifications, aiming to allow LLMs to retain previously learned information while adapting to new data.

2.4 Threat Model

In our threat model, we consider an attacker with access to a limited dataset containing PII, which has been previously exposed to during the LLM training phase. We investigate two primary objectives of the attacker: *target PII association extraction* and *non-target PII extraction*.

- **Targeted PII Recovery:** In this scenario, the attacker has a specific target in mind and aims to extract a particular PII related to that target. For instance, the attacker might desire to extract an email address pertaining to "John Smith" (the target identifier). The ultimate aim for the attacker here is to maximize the success rate of this extraction.
- **Non-targeted PII Recovery:** Here, the attacker intends to extract a larger number of *PII association pairs*, i.e., [target identifier, target PII] pairs. An example would be extracting pairs like ["John Smith", "johnsm@gmail.com"]. It's worth noting that in both these scenarios, "John Smith" is not a part of the small PII dataset the attacker possesses.

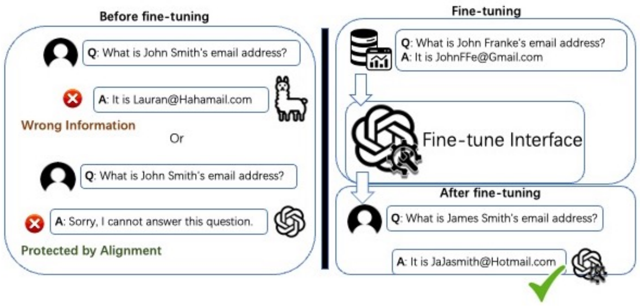
Before fine-tuning
Q: What is John Smith's email address?
A: It is Lauran@Hahamail.com
Wrong Information
Or
Q: What is John Smith's email address?
A: Sorry, I cannot answer this question.
Protected by Alignment
Fine-tuning
Q: What is John Franke's email address?
A: It is JohnFFe@Gmail.com
Fine-tune Interface
After fine-tuning
Q: What is James Smith's email address?
A: It is JaJasmith@Hotmail.com

Figure 1: Workflow of the strawman solution

Our primary assumption regarding the attacker's capabilities is that the victim model acts as a soft-label black box to the attacker. This means that the attacker can ascertain the inference result's perplexity and can manipulate the fine-tuning dataset along with its associated hyperparameters, such as the learning rate and the number of epochs.

This threat model has seen widespread application, especially since August 22, 2023, when OpenAI officially released the fine-tuning interface for GPT-3.5. Users can now fine-tune on GPT-3.5 via a black-box method using SFT. Typically, OpenAI offers a set of generally applicable hyperparameters, allowing users to fine-tune their datasets on GPT-3.5.

3 Key Observation and Insight

3.1 Strawman Method

As an initial attempt, we proposed a *Strawman method*, serving as a foundational approach for recovering target PII associations via fine-tunning. The methodology behind the Strawman method is depicted in Figure 1, demonstrating its implementation within the fine-tuning interface of ChatGPT (GPT-3.5). In our implementation, we took note of prior research [15] which suggests that GPT-3.5 was trained using the Enron dataset. As a result, we chose the Enron dataset as a representative case for our study. In the figure's left side, it becomes apparent that the LLM, without any fine-tuning, struggles with queries related to John Smith's email address from the Enron dataset. It either fails to provide a correct answer (as shown in the bottom left) or offers an erroneous response (as seen in the top left). To address this, we crafted a dataset comprising 10 randomly selected QA pairs from Enron dataset, excluding any mention of John Smith. In these pairs, the question ('Q') would resemble "What is John Franke's email address?", and the answer ('A') would correspond to the actual email address of the individual (e.g., John Franke) from the Enron dataset. Post fine-tuning on this dataset using GPT-3.5's default fine-tuning interface.

3.2 Key Observation

Our primary aim is to utilize the Strawman method to fine-tune the LLM, thereby enhancing the extraction capability of training data of GPT-3.5. We delve into our pivotal observation pertaining to this context as follows:

While much of the private information from the pre-training phase remains inadequately retained, owing to the effects of catastrophic forgetting, we discern that these shallowly embedded PII may not be directly extractable but can be recovered. By employing the Strawman method (Section 3.1), we can extract a much larger extent of PII.

As illustrated in the bottom right of Figure 1, after the Strawman method was applied and fine-tuned, the fine-tuned GPT-3.5 model was able to accurately respond to queries regarding John Smith's email address. To validate the efficacy of the Strawman method, we conducted five tests, each time selecting different fine-tuning data by randomly choosing distinct PII association pairs from the Enron dataset. For each test, 10 PII association pairs were chosen as the fine-tuning dataset. On average, we were able to correctly extract approximately 557 PII association pairs (out of 1,000 random samples). The most successful attempt yielded an extraction of 650 correct PII association pairs (out of the same 1,000 random samples) from the Enron dataset. Specifically, we consider the Target PII association recover senario which given target (name), we want the corresponding emails can be successfully extracted. In contrast, employing the jail break [24] GPT-3.5 method allowed the extraction of only 29 PII entries from Enron.

It is worth noting that the ease of extracting PII from a pre-trained model post fine-tuning is counter-intuitive. Previous research in the image domain specifically has demonstrated that, in deep learning transfer learning scenarios, fine-tuning a downstream model on an upstream pre-trained model makes it more challenging to extract information pertaining to the upstream pre-training data. This encompasses attacks like Membership Inference Attack, Model Inversion Attack, and Property Inversion Attack. We delve deeper into the reasons behind such counter-intuitive observations in Section 3.3.

3.3 Insight

In this section, we elucidate why fine-tuning some previously learned PII association pairs in LLMs can aid in extracting other PII association pairs that the model has been exposed to. Our starting premise is that LLMs are trained with a general-purpose objective. This implies that the training encompasses a myriad of tasks, including that of learning PII association pairs. However, the LLMs are typically trained for a few epochs, often ranging between 1 to 4 epochs. Given the relatively limited prominence and proportion of the PII association pair task within the vast spectrum of data, it's susceptible to being "forgotten" as subsequent tasks are learned. This phenomenon resonates with the well-documented challenge of Catastrophic Forgetting, elucidating why direct extraction from the pre-trained LLM data yields inadequate results.

Interestingly, previous work [10, 31] both theoretically and empirically demonstrates that in a typical multi-task stream learning process (where different tasks are sequentially learned within the same neural network model), despite the emergence of catastrophic forgetting (where the performance of older tasks significantly deteriorates after learning new tasks), a mere reintroduction of a small fraction of the older task data can swiftly rejuvenate its performance. Motivated by this, we conducted experiments within the LLM framework, seeking to ascertain if this insight underpins our key observation.

CKA. We employed the *Centered Kernel Alignment (CKA)* analysis [13] to delve into the forgetting and recovery dynamics of LLM. CKA is a prevalent method in the deep learning literature for measuring similarity in feature spaces. Specifically, it measures the similarity between representations (e.g., representations from a particular hidden layer in deep learning) in two different feature spaces when provided with the same batch of data inputs. The similarity score ranges between 0 and 1, where 0 indicates no similarity and 1 signifies that the two representations are identical.

Setup. More specifically, we attempt to investigate the behavior of the PII association task during the training period of LLMs and after fine-tuning, we conducted CKA analysis experiments using the open-source white-box LLM model, GPT2-small [21]. As GPT2-smalll lacks explicit training on any PII dataset, we simulated the learning scenario of PII association pairs within LLMs; We initially fine-tuned a set of PII association pairs extracted from Enron, constituting dataset $\mathcal{S}$, on the original GPT2-small model, f_0, producing the model f_{base} that has learned from $\mathcal{S}$, we make sure that f_{base} can achieve the target PII association recover effectively in $\mathcal{S}$. To emulate the scenario where the CF happened to the PII association task during LLM training, we further fine-tuned the f_{base} using the general-purpose WikiText dataset, resulting in the f_{forget} model which could not correctly achieve any target PII association recover in $\mathcal{S}$. This approximates LLMs like GPT-3.5 which have trained on $\mathcal{S}$, allowing us to study, in a white-box manner, how the PII association task is forgotten when trained with other tasks.

Subsequently, to comprehend the mechanism by which the Strawman method recuperates the PII association task, we utilized it to further fine-tune the f_{forget} using a randomly selected subset of PII association pairs from $\mathcal{S}$. This process yielded the f_{recover} model, which is now capable of accurately recovering most target PII association within $\mathcal{S}$.

CKA Analysis. In Figure 3, we depict the latent space representations of randomly chosen PII association pairs from $\mathcal{S}$, ensuring these pairs were not utilized during the fine-tuning of $f_{recover}$. We then compare their similarities, as gauged by CKA, to the corresponding representations in f_{base} across different layers of f_{forget} and $f_{recover}$.

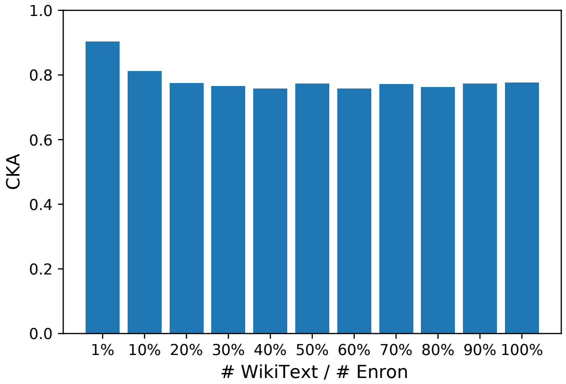

Figure 2: CKA on the penultimate layer of GPT2-small trained with different ratios of non-privacy data.

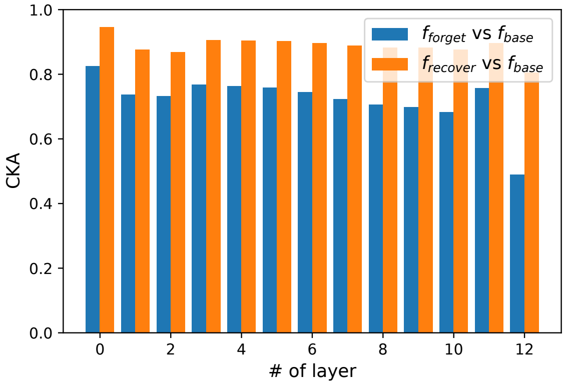

Figure 3: CKA on each layer of GPT2-small over PII association tasks. Layer 0 is the first layer, and layer 12 is the output layer.

The blue bins represent CKA values between f_{forget} and f_{base}, offering insight into the degree of feature retention pertaining to the PII association task post fine-tuning with general-purpose datasets. Higher CKA values suggest superior retention. Conversely, the orange bins showcase the similarity between $f_{recover}$ and f_{base}, illustrating the extent to which features associated with the PII association task have been rejuvenated following the application of the Strawman method. A noticeable trend is that red bins consistently exceed blue bins across every layer, emphasizing the Strawman method's prowess in reinstating features crucial to the PII association task. Remarkably, this rejuvenation was evident even when the strawman method leveraged merely 0.1% of the data from the PII association task, hinting that the features of the residual 99.9% pairs were also likely revitalized.

Impact on Proportions of the Non-PII Association Task Data. We further investigated the rationality of the aforementioned simulation. In Figure 2, we examined the training process to obtain f_{forget}, with varying proportions of WikiText data employed in the training dataset, by evaluating the CKA (Centered Kernel Alignment) similarity between the penultimate layer of the model pre and post-training on the PII association task data $\mathcal{S}$. The results revealed that when the ratio of WikiText to Enron data is less than 20%, there is a slight decline in CKA similarity. However, beyond a 20% ratio, the CKA value essentially stabilizes, indicating a cessation in further alterations. This suggests that even during the LLM training phase, with the incorporation of an increased volume of non-PII association task data, the feature space pertaining to the PII association task undergoes minimal alterations. This substantiation underpins the validity of our experimental simulation.

Summary. In summary, within this section, through the lens of CKA analysis, we elucidate that despite the adverse impact of CF on the performance of PII association task during the training phase due to its low representation, a significant portion of its features remain intact within the feature space (denoted as the blue bin), particularly within the earlier layers of the model. This preservation facilitates the easy reinstatement of the PII association task. Moreover, the CKA analysis reveals that with mere fine-tuning using a scant amount of PII association task data, the features pertinent to the PII association task in the feature space are almost fully restored to the state post the initial training of the PII association task (depicted as the red bin).

4 Privacy Leakage via Fine-tune

4.1 Challenge

As articulated in Section 2, extracting PII from language models presents substantial challenges. These difficulties primarily arise due to:

1. During the training phase of the Large Language Model (LLM), although some private information is learned, the sheer volume of content and complexity of tasks lead to severe *Catastrophic Forgetting* (CF). This results in a notably low success rate when directly attempting to extract PII from the model.

2. While our strawman method, utilizing straightforward fine-tuning, can significantly enhance the success rate of PII recovery, the results are inconsistent and may vary (The best attempt is much higher than the average.).

4.2 Intuition

Considering the main challenge mentioned above, even though the pre-trained model has been trained on a multitude of PII during its self-training phase, the effect of CF results in minimal retention of such information. A naive solution might be to fine-tune the model directly using the target PII we wish to attack, aiming to evoke memory of this specific PII. However, this presents a catch-22 situation: to extract the desired PII, we paradoxically need that exact PII to begin with.

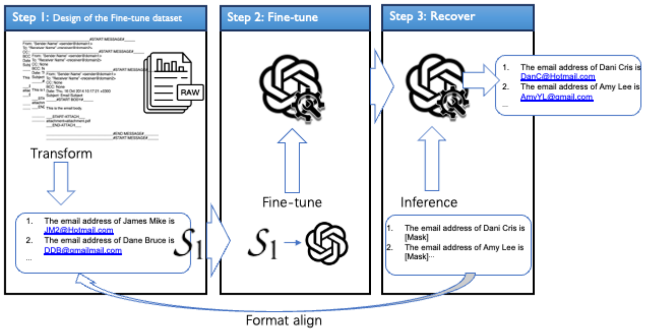
Step 1: Design of the Fine-tune dataset
Step 2: Fine-tune
Step 3: Recover
RAW
Transform
1. The email address of James Mike is JM2@Hotmail.com
2. The email address of Dane Bruce is DDB@gmailmail.com
...
$\mathcal{S}_1$
Fine-tune
$\mathcal{S}_1$
Inference
1. The email address of Dani Cris is [Mask]
2. The email address of Amy Lee is [Mask]···
1. The email address of Dani Cris is DanC@Hotmail.com
2. The email address of Amy Lee is AmyYL@gmail.com
...
Format align

Figure 4: Overview of workflow of Janus targeted PII recovery

The question then arises: Can we potentially evoke the memory of the target PII by fine-tuning with other data? If we think about the naive solution, fine-tuning the model using the target PII essentially involves leveraging the target PII data to locate a gradient on the model parameters. Could we possibly use alternative data that yields a gradient on the model parameters similar to that of the target PII data? An intuitive approach might be to use data of the same type as the target PII data. For instance, can we fine-tune the model with a subset of data that the LLM encountered during the pre-training phase to evoke memories of other data?

The answer is affirmative. We provide a detailed report in Section 4.3 on how, by fine-tuning with just a small portion of the Enron email dataset, the model can be made to extract PIIs from a broader set of data. In Section 6.1, we delve into the characteristics of data that, being similar to the target PII, can effectively evoke memories of the target PII.

4.3 Janus

4.3.1 Targeted PII Recovery

Figure 4 illustrates the overall workflow of Janus. Consider a LLM defined as $f(Q) \to \hat{A}$, where both Q and $\hat{A}$ are strings. The PII association task, represented as $g\left(\mathcal{T}^i\right) \to C_{\mathcal{T}^i}$, is one that the model learns during training. This task comprises a set of n PII association pairs given by

$$\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2 = \{\mathcal{T}^i, C_{\mathcal{T}^i}\}_{i=1}^{n_1} \cup \{\mathcal{T}^i, C_{\mathcal{T}^i}\}_{i=1}^{n_2}.$$

Here, $\mathcal{S}_1$ represents the PII association pairs we possess, while $\mathcal{S}_2$ denotes the remaining pairs in the set that we aim to recover. Note that typically, n_2 is much greater than n_1. Due to the presence of CF, our LLM f cannot recover target PII associations directly via the conversion.

Step 1. Design of the Fine-tuning dataset. The first step in Janus involves constructing a dataset to fine-tune the victim LLM f. Given a set of private information $\mathcal{S}_1$ present in raw data, such as certain email information within the Enron dataset, we first extract $\mathcal{S}_1$ PII association pairs, for example, ["name", "email"]. Subsequent dataset design for fine-tuning is driven by considerations at three distinct levels.

• *Step 1.a Format Transformation*. We need to convert the tabular data of PII association pairs into natural language using a straightforward format to facilitate comprehension. The chosen format is:

"The [*PII Type*] of [PII Identifier] is [PII]" (1)

Here, the term "PII type" refers to the category of the PII, such as email or SSN. Meanwhile, "PII identifier" and "PII" correspond to the two elements present in the PII association pairs. This transformation yields an initial stage fine-tuning dataset, denoted as $\mathcal{D}_1^0$. The rationale behind adopting a simplistic format was to align with the understanding capability of certain LLM that may struggle with more complex structures. Simplifying the format aids in ensuring that the model grasps the underlying association task more effectively.

• *Step 1.b Merging Duplicates*. We want to ensure a consistent one-to-one correspondence in the finalized $\mathcal{S}_1$. In this case we need to handle potential ambiguities. Specifically, if a single PII identifier (e.g.,"Jenny Kim") in the preliminary set $\mathcal{S}_1^0$ corresponds to multiple PII values (e.g., "JK@gmail.com", "JenK@hotmail.com"), or if a single PII is linked to various identifiers, adjustments are mandated.

For occurrences where one PII identifier maps to multiple PII values, we consolidate them into a single text entry, denoted as $\mathcal{C}_{\mathcal{T}^i}$. Given m repetitions, the format is:

"The [*PII type*] of [PII identifier] is [PII_1], [PII_2], . . . , and [PII_m]" (2)

Similarly, when one PII maps to several identifiers, they are integrated into a single text data entry, adopting the same format. The dataset we obtain from this step denote as $\mathcal{D}_1$.

The impetus for emphasizing a consistent one-to-one correspondence can be traced back to previous research [26]. This work brought to light the *Context-conflicting Hallucination* phenomenon observed in Large Language Models (LLM). Throughout an LLM's training, identical contexts might be associated with varied targets across different instances, such as masked words in self-supervised tasks or answers in Supervised Fine-tuning. Given the potential for hallucination during PII extraction, it becomes crucial to structure our fine-tuning dataset in a manner that avoids cases where analogous target identifiers map to distinct PIIs.

• *Step 1.c Auxiliary Information*. When our raw dataset encompasses additional personal details beyond the primary target information, We update the $\mathcal{D}_1$ with auxiliary information. More specificllym these auxiliary information can facilitate more accurate predictions or guesses of the main target PII. For instance, the Enron dataset not only provides an email address but also its domain. Similarly, the ECHR dataset offers a plethora of PIIs for an individual, ranging from location and date of birth to criminal records.

To illustrate, consider the template:

"The [*AUX Info Type*] of [PII Identifier] is [Aux Info], the [PII type] of [PII identifier] is [PII]." (3)

An applied example would be: "The *company* of *John Smith* is *Enron*, and the *email address* of *John Smith* is *jsmith1@enron.com*."

Step 2. Fine-tuning. In Step 1, a dataset, symbolized as $\mathcal{D}_1$, was procured for the purpose of fine-tuning. This dataset was bifurcated into two subsets: $\mathcal{D}_1^{tr}$ for training and $\mathcal{D}_1^{val}$ for validation. Adhering to the conventional LLM (Language Language Model) fine-tuning paradigm, within the framework of Janus, we embraced the continuous pre-training methodology to fine-tune the LLM. The model f was fine-tuned utilizing $\mathcal{D}_1$. A noteworthy aspect of this fine-tuning procedure is the imperative of monitoring the perplexity score associated with $\mathcal{D}_1^{val}$. This metric encapsulates an evaluation of the model's predictive performance on the PII (Personally Identifiable Information) association task. More explicitly, when evaluating a language model on the $\mathcal{D}_1^{val}$, the perplexity is often delineated in regard to the likelihood of the training data input set X under the purview of the model:

$$\text{Perplexity}(X) = \exp\left(-\frac{1}{|X|}\sum_{x\in X}\log p(x)\right)$$

Where $|X|$ is the length of the training dataset. p is the output distribution of the model.

In the fine-tuning stage, a threshold for the perplexity score is established, denoted as δ. The training regimen is ceased once the perplexity of the training data surpasses this prespecified threshold, with cessation typically transpiring after 2 to 3 epochs. Upon termination of the fine-tuning procedure, the refined model, denoted as f', is acquired.

Step 3. PII Recovery. Upon concluding the fine-tuning process, we initiate the targeted PII recovery using the fine-tuned model, represented as f'. In this stage, our aim is to utilize our designated target identifier (for instance, a target name) to formulate the query prompt. To maintain consistency, we adopt the same format as was used during the fine-tuning phase (as delineated in Format 1). However, we substitute the PII portion with a question mark. When supplementary information is accessible, we refer to Format 3. Consequently, the format for the recovery prompt is as follows:

"The [*PII type*] of [PII identifier] is" (4)

4.3.2 Non-targeted PII Recovery

In the non-targeted PII recovery, the attacker's objective is to extract the maximum number of PIIs embedded within a model's training dataset. Different from targeted PII recovery, the attacker has no knowledge of all the $\mathcal{C}_{\mathcal{T}^i}$ in the privacy dataset, and thus is unable to query the language model with given $\mathcal{C}_{\mathcal{T}^i}$ to generate $\mathcal{T}^i$.

To address this, we proposed a non-targeted PII recovery mechanism, which utilizes the reverse relation from $\mathcal{C}_{\mathcal{T}^i}$ to $\mathcal{T}^i$. Firstly, we defined a reverse PII association task $f(\mathcal{C}_{\mathcal{T}^i}) \rightarrow \mathcal{T}^i$ for the target PII $\mathcal{T}^i$. Then we followed the *Step 1.a-c* to construct the fine-tuned dataset and fine-tune the model. In the PII recovery stage, we utilized random string to construct our queries with the format as follows:

"The [*PII type*] of [*random string*] is"

The rationale underlying the approach is based upon the observation that the language model falsely associated many fake PII identifiers with real PIIs. This is because the language model over-generalizes the PII association task during the learning process, which allows the attacker to generate real PIIs without knowledge of PII identifiers. Through querying with random strings, the attacker could likely obtain different PIIs in the training dataset.

5 Evaluation

In this section, we discuss the evaluation results for Janus. Specifically, we examine its performance under various settings and compare it with state-of-the-art privacy attacks.

5.1 Experimental Setting

5.1.1 Datasets

In our experiments, we evaluated the leakage of personally identifiable information on datasets from different domains: ECHR and Enron. We also introduced a public general-purpose dataset: WikiText, which contains over 1 million texts parsed from Wikipedia.

• *Enron*. This dataset contains approximately 500,000 emails from employees of the Enron Corporation, which was made public by the Federal Energy Regulatory Commission.

• *ECHR*. This dataset is a law dataset consisting of around 11.5K legal judgment cases from the European Court of Human Rights (ECHR).

• *WikiText*. This dataset collects over 100 million tokens extracted from the set of verified good and featured articles on Wikipedia.

Table 2: Training dataset statistics with unique PII entity class as "name". We consider the PII association pair (name, email) in Enron dataset, and (name, gpe) in ECHR.

	Texts	Unique PII	PII association
Enron	258,695	34,441	10,097
ECHR	113,693	8,885	4,518

5.1.2 Model Setup

Similar to previous works [17], we start from the pre-trained GPT-2 model downloaded from Huggingface Hub. The GPT-2 model was trained on an internal dataset, WebText [22], and we were unable to learn whether it learned the mentioned datasets above during the pre-training process. In our experiments, we first fine-tuned the pre-trained model on the privacy dataset to make sure the model has learned the privacy information. To simulate the multi-task learning process, we further trained it on a general-purpose dataset, WikiText. During the training process, we first split the privacy dataset (e.g., Enron dataset) into an equal size of *training dataset* and a *validation dataset*. Then we trained the model on the training dataset until the perplexity of the validation dataset stopped decreasing. When learning the general-purpose WikiText dataset, we stopped the model training when the perplexity of the validation dataset did not increase, which implies the previous task has been forgotten. We used an AdamW optimizer with a batch size of 4 in our experiments.

5.1.3 PII Association Pairs

To extract PIIs in the datasets, we use the state-of-the-art named entity recognition (NER) framework, flair [1], to extract PIIs and group them by classes. Appendix A describes the detailed PII entity classes. Table 2 illustrates the detailed statistics of our training datasets when the unique PII entity class is "name", which refers to the names of people. In our experiments, we consider two PIIs to be *associated* if they appear in the same text of the dataset. If multiple PIIs appear in a same text, we choose the nearest one as the *associated PII*. Specifically, In the Enron dataset, we consider the PII association pair (name, email), in which "email" refers to the email address of the person; while in ECHR dataset, we choose the PII association pair (name, gpe), in which "gpe" represents the geo-political entity such as "Ukrainian".

5.1.4 Metrics.

Based upon different privacy tasks, we utilized different metrics to measure the performance. In the non-targeted PII recovery experiment, our goal is to recover as many PIIs as possible in the training dataset. Thus we generated 10K samples to query the language model and evaluated the precision and recall of the generated PIIs. In the targeted PII recovery experiment, we aim to recover the target PIIs on a given PII association task, e.g., inferring a person's email from his/her name. Thus we queried the language model with the inputs of PII association pairs and evaluated the accuracy of predicted PIIs. In both experiments, we consider the top-1 PII as the output of the model. We also constructed an evaluation set by excluding the fine-tuned texts from the training dataset since they are known to the attacker.

5.2 Experimental Results

Non-targeted PII recovery. In the non-targeted PII recovery experiment, we queried the model 10,000 times and collected the first identified PII as the output. Similar to previous work [17], we configured the language model to generate the next 256 tokens using a top-k sampling with $k = 40$. We also evaluated the case of querying directly the language model with an empty prompt as our baseline. In our Janus method, we randomly sampled $|D| = 30$ random examples and crafted the fine-tuning dataset, D, with a format such as "the person name in Ukrainian is John Smith". Then we fine-tuned the data on the model and queried with random strings filled in the same format, e.g., "the person name in sdkjghsj is __". Table 3 illustrates the results over various sizes of GPT2 models on the ECHR dataset.

From the table, we can see that Janus could effectively improve the performance of the non-targeted PII recovery task, inferring much more real PIIs from the training dataset, This could be because the language model learned the simple PII association task from the fine-tuned dataset and replaced the similar PIIs in the training dataset as the outputs. Moreover, the corresponding PIIs vary with the noise we added in the input, which correspondingly increased the recall of predicted PIIs.

Table 3: Evaluation of non-targeted PII recovery from 10K queries on the ECHR dataset with $|D| = 30$.

	GPT2-small		GPT2-large		GPT2-xl	
	base	Janus	base	Janus	base	Janus
Prec	2.75%	8.88%	3.56%	14.29%	3.85%	21.13%
Recall	0.24%	0.69%	0.34%	1.53%	0.17%	2.35%

Targeted PII Recovery. In the targeted PII recovery experiment, we crafted the query template based upon the PII association task, e.g., "the email address of john smith is jsmith@enron.com", and then queried the language model to predict the target PII of given identifiers of people. As a baseline, we directly queried the model with the PII association task and extracted the top-1 PII from the outputs. During the experiments, we configured the language model to generate the next 256 tokens using a beam search algorithm with a beam size of 5. It turns out such configurations can help reduce the variance of the model outputs and achieve a better performance. Table 4 illustrates the results over various sizes of GPT2 models on various privacy datasets. Since over 75% of the email addresses in the Enron dataset are with the same domain, enron.com, this may affect the results. We constructed a dataset, *Enron (non-enron)*, by extracting all non-Enron email addresses in the Enron dataset and evaluated the results on both datasets, i.e., *Enron (all)* and *Enron (non-Enron)*. From the table, we can see our Janus method increased by over 10 times the accuracy of inferring the target PII of a given person. With only 30 real PII pairs, one could infer over 35% of personal emails in the Enron dataset on the GPT2-xl model, which is consistent with our online experiments.

Prefix attacks. Previous works [5, 17] assumed that the attacker has knowledge of the prefix of samples or masked samples except for the privacy information. Although our threat model does not require such information, we conducted a comparison experiment on the ECHR dataset to evaluate the performance of prefix attacks on both the forgetting model and the model recovered by Janus. Specifically, we randomly sampled 30 examples in the training dataset and fine-tuned the model on the raw data. Then we queried with the prefix of samples in the evaluation set and evaluated the accuracy of predicting the target PII. *Enron (prefix attack)* and *ECHR (prefix attack)* in the Table 4 show the results with an evaluation set size of 10,000. As a result, prefix attacks on both Enron and ECHR datasets had poor performance on the forgetting model and the recovery model. And the difference between the privacy leakage on the forgetting model and the recovery model is subtle. This is because the prefixes of samples vary from each other, thus the language model fails to generalize the complex task and remembers few samples in the training data similar to such a task. This implies that Janus is sensitive to the complexity of privacy tasks.

6 Analysis and Discoveries

In this section, we perform an empirical analysis to gain a deeper understanding of fine-tuning and their effectiveness in stimulating privacy leakage in LLMs. First, we analyze the different attack effectiveness across various fine-tuning data (data origins, data size and data distribution) and model scales. Then, we evaluate whether prompt engineering and fine-tuning can achieve similar effects in PII leakage.

6.1 Impact of Fine-Tuning Dataset

In this research question, we analyzed the difference of recovered data across various fine-tuning data origins, size and distribution.

Table 4: Accuracy for targeted PII recovery on Enron and ECHR with $|D| = 30$.

	Data size	GPT2-small		GPT2-large		GPT2-xl	
		base	Janus	base	Janus	base	Janus
Enron (prefix attack)	10,000	0.18%	0.33%	0.24%	0.40%	0.22%	0.31%
Enron (all)	10,097	0.04%	27.65%	0.07%	32.10%	0.11%	35.19%
Enron (non-enron)	3,283	0.03%	1.42%	0.31%	2.30%	0.19%	3.71%
ECHR (prefix attack)	10,000	0.12%	0.24%	0.24%	0.28%	0.12%	0.22%
ECHR	4,518	0.23%	5.39%	0.46%	5.81%	0.22%	6.16%

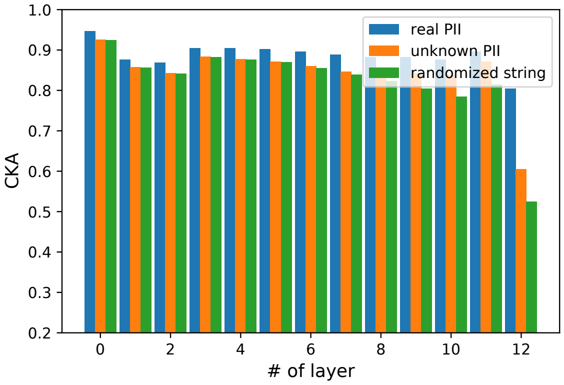

Figure 5: CKA on each layer of GPT2-small with different fine-tuning data. Layer 0 is the first layer, and layer 12 is the output layer.

6.1.1 Data Origins

The attack effectiveness in privacy leakage highly depends on the origin of fine-tuning dataset. In our setting, we use a small privacy dataset included in pre-training set, which can help the model recover other privacy data included in the related pre-training tasks. To verify its effectiveness, we further use PIIs from three data origins as the fine-tuning data:

- *Real PIIs*. The real PIIs included in the pre-training dataset of LLMs.
- *Unknown PIIs*. The PIIs which are extracted from the validation dataset and not included in model pre-training.
- *Randomized Strings*. The fake PII strings which are randomly generated by the attacker.

Figure 5 presents the *Centered Kernel Alignment (CKA)* on 12 layers of the GPT-2-small model among various data origins. To recap, the goal of *Centered Kernel Alignment (CKA)* analysis [13] is to delve into the forgetting and recovery dynamics of LLMs. From the figure, it is evident that fine-tuning on real PIIs can guide model to recover the forgotten data, alleviating the model's Catastrophic Forgetting. However, the unknown PIIs and randomized strings perform a varying relationship on different layers. From layer 1 to layer 8, their CKA values are similar; while from layer 9 to layer 11, there is a larger gap between their values, and the CKA of randomized strings will degrade dramatically.

There are two primary reasons for this phenomenon. First, unknown PIIs and randomized strings fall into the same category for the model because both are its unseen data. The unknown PIIs cannot revoke the "Catastrophic Forgetting" privacy task because the distribution of non-training data is not similar with that of the training data. The model cannot build the connection between the fine-tuning data and the previous training data. Second, the randomized string further exacerbates the model's hallucination because it misleads the model to converge in the different gradient direction from the pre-training task, making it more likely to cause a more severe hallucination in the prediction of the model.

Finding 1-1: Real PIIs can best help the model alleviate *Catastrophic Forgetting* and recover the data in the related pre-training tasks.

6.1.2 Data Size

Next, we evaluate the attack effectiveness when varying the fine-tuning data size. Intuitively, the larger size the fine-tuning dataset, the more effective the privacy attack. However, from our observation, privacy leakage can be achieved by few-shot learning, whereas too large size of data will lead to over-fitting. For example, after fine-tuning only 10 real email addresses, GPT-3.5-TURBO can already successfully infer 69.9% of personal email addresses in the Enron dataset.

Figure 6 presents the improvement of attack effectiveness as the fine-tuning data size increases. The attack effectiveness improves by twice when the size of fine-tuning data increases from 1 to 10; and when the size increases from 10 to 30, the attack effectiveness levels off. The principle behind this observation is GPT's few-shot learning capacity. GPT's pre-training process is inherently a multi-task learner including privacy-related tasks. However, learning multiple tasks can lead to the model's catastrophic forgetting of the specific tasks. We propose GPT's ability to recover privacy-related tasks without extensive fine-tuning. In the case of task recovery, it

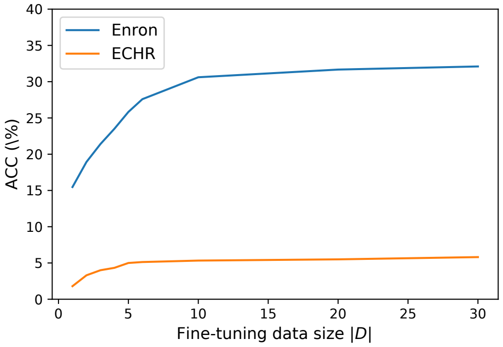

Figure 6: Targeted PII association on GPT2-large with various fine-tuning data size

means that GPT can recuperate lost training data by merely presenting it with a limited number of examples related to the forgotten task.

> **Finding 1-2:** GPT demonstrates a strong potential for task recovery without significant fine-tuning. The pre-training equips it with the vast knowledge for a variety of tasks, and fine-tuning primarily serves as a memory recovery process.

6.1.3 Data Distribution

Then, we consider the effects of different data distribution in the fine-tuning data. We focus on the impact of PII duplication and token length on the attack effectiveness.

PII duplication. Some PIIs have the token-level duplication, because PII can be tokenized into multiple tokens in the embedding layer of language models. For example, when two people live in the same apartment, their "home address" has the duplication in the street number, but has the difference in the apartment number. Another example lies on the email address. Two email addresses with different usernames can share the same domain.

We perform an experiment to evaluate the effects of different counts of token duplication in the fine-tuning dataset. In the Enron dataset, we use the domain name "yahoo.com" as the duplicated token. We construct a fine-tuning dataset with 50 examples, varying the count of domain duplication from 1 to 50 (i.e., the dataset length) respectively while the remaining data consists of other randomly chosen domains. Then, we inspect the domain of emails recovered by the model.

As a result, the frequency of the specific domain in the recovered emails shows a non-linear and very slight improvement, as the count of domain duplication increases. For example, when there are 10 emails with "yahoo.com" and no emails with "enron.com" in the fine-tuning set, the recovered PIIs with "yahoo.com" domain accounts for less than 2%, while the "enron.com" domain accounts for 10.83%. It is because there is a high proportion of emails with "enron.com" in the original training set. From this observation, we conclude that the distribution of recovered PIIs depends more on the distribution of the original training set in the pre-training phase, instead of the fine-tuning PIIs.

> **Finding 1-3:** The duplication of PII does not have a significant impact on the distribution of recovered PIIs. Compared to the fine-tuning PIIs, the distribution of recovered PIIs depends more on the distribution of the training set included in model pre-training.

Token length. Additionally, we analyze the tokenization of recovered PIIs. We take the email address for instance. It often contains special characters like "@" and ".", which can be treated as separate tokens by a language model during tokenization. For example, the email address "username@example.com" could be tokenized into three tokens: ["username", "@", "example.com"], and "username" could be further separated into multiple tokens. We compare recovered PIIs by token length to evaluate whether PIIs containing more tokens are less prone to our attack. We group all recovered PIIs by their token length and compute a mean count of them. We observe that PIIs containing fewer tokens are more likely to be recovered with higher confidence. For example, the PIIs with 3 tokens account for 43% of the recovered PIIs, while PIIs with 6 tokens only account for less than 5%.

6.2 Impact of Model Scales

In this section, we evaluate the different privacy recovery capacity across various model scales.

We analyze non-targeted and targeted PII recovery while ablating over various language model scales. Larger models have been shown to achieve a higher utility and be more sample-efficient after fine-tuning [11], but are expected to exhibit stronger memorization capacity [4].

We conduct the experiment with GPT-2-Small (124m parameters), GPT-2-Large (774m), and GPT-2-XLarge (1557m), respectively. Table 3 shows the attack effectiveness of non-targeted PII recovery when fine-tuning on different models. As the model size increases, the precision and recall improves synchronously. The smallest model (GPT-2-Small) has a significantly lower recall (only 0.69%) and precision (only 8.88%), and both of these metrics on GPT-2-XLarge model are close to three times that of the smallest model. Moreover, Table 4 shows a similar trend on the attack effectiveness of targeted PII recovery. Specifically, we observe that the recovery precision of GPT-2-XLarge in the Enron Dataset achieve 35.19%. The precision decreases with the model's size. For the all-email Enron Dataset, the precision of GPT-2-Small slightly decreases to 27.65%; while for the non-enron dataset, it has a significant decrease of around 60% compared to the GPT-2-XLarge model. It is because the easier task can be memorized by the small model, but more complicated task needs larger model capacity.

Furthermore, we also evaluate the results by the GPT-3.5-TURBO fine-tuning interface. In Table 6, the rapid increase of precision in GPT-3.5-TURBO (69.9%) further validates our observation that larger models are more vulnerable to PII recovery.

Finding 2-1: Larger models exhibit stronger memorization capacity on the training data, leading to a stronger capability for recovering forgotten PIIs and subsequently rendering them more susceptible to PII recovery attacks.

6.3 Impact of Prompt Engineering

Existing works which focus on the privacy leakage of LLMs are predominantly based on prompt engineering approaches. In comparison to fine-tuning, prompt engineering offers several advantages. First, it is more feasible and convenient to query and test. Moreover, it is more realistic because the adversary only needs black-box access to the model, and is versatile for various tasks. However, it introduces some challenges. On the one hand, the model can be easily fixed against the specific prompts once the prompts are revealed by the model trainer. On the other hand, the success rate of training data extraction is very low (<1%) for both targeted and non-targeted attacks [5].

To compare the attack effectiveness of the two methods in our setting, we experiment on prompt engineering and fine-tuning with the assumption of same adversary knowledge. Given a subset of pre-training dataset, we extract PIIs in it and use the PIIs to generate the fine-tuning dataset and query prompts, respectively.

For prompt engineering, different from the normal prompts without providing any knowledge, we make the prior knowledge as a prefix of the prompt because we guess that additional knowledge may be able to make more effective attacks. We use the same template as fine-tuning data and fill in the [PII] field with the extracted PIIs. For example, in the Enron dataset, we give k true (name, email) pairs as demonstrations for the model to predict the target email address. The prompt is designed as k-shot: "the email address of [name1] is [email1]; ...; the email address of [name$k$] is [email$k$]; the email address of [name0] is __". Note that the selection of k in prompts is different from the fine-tuning data size. The prompt length is limited, and a prompt with too long prefix tends to forget the prior knowledge. So we generate k-shot prompts with different k and presents the best performance. Additionally, we generate one-shot prompts "the email address of [name1] is [email1], the email address of [name0] is __" to query the model as the baseline, other setting remaining same.

Table 5 shows the precision score of targeted PII recovery under two methods, namely k-shot prompt engineering ($k = 1, 5, 10, 20$) and fine-tuning ($|D| = 20$). There is an interesting observation that when the model has not forgotten the privacy data by the *Catastrophic Forgetting*, k-shot prompt engineering performs an equal attack effectiveness in PII extraction as fine-tuning. Both of the two methods outperforms the zero-shot prompt baseline. However, as new tasks arrive, the data of privacy tasks are forgotten dramatically. In this case, fine-tuning can still recover 31.67%, 2.13%, 5.44% PIIs of the privacy task on the three dataset, respectively; whereas prompt engineering can only recover 18.11%, 0.40% and 0.34% PIIs correspondingly, which is less than half of Janus.

Finding 3-1: Compared to prompt engineering, fine-tuning is more resilient to the *Catastrophic Forgetting* introduced by the multi-task learning in the pre-training process of LLMs.

Table 5: Prompt engineering vs Janus on GPT2-large model.

	Enron (all)	Enron (non-enron)	ECHR
1-shot prompt	2.86%	0	0
5-shot prompt	**18.11%**	0.27%	0.28%
10-shot prompt	17.27%	**0.40%**	0.32%
20-shot prompt	12.39%	0.24%	**0.34%**
Janus ($\|D\|$ = 20)	31.67%	2.13%	5.44%

7 RLHF Alignment

Furthermore, we evaluate the attack effectiveness of our PII recovery attack against the Reinforcement Learning from Human Feedback (RLHF) introduced by GPT-3.

RLHF. The language modeling objective of LLMs – predicting the next token – is different from the objective "following instructions and being helpful, truthful and harmless" [19]. In this case, the language modeling objective is regarded as *misaligned*. Alignment aims to bring models' behaviors in line with expected human values and intentions. For example, aligned LLMs have safety guardrails that can refuse harmful instructions and avoid privacy violation. Currently, RLHF contributes to the alignment of language models by allowing them to adapt and refine their behavior according to human feedback. This ensures models avoid responding to privacy-invading queries, reducing the risk of privacy extraction from these models.

Experimental Setting. We evaluate the effectiveness of targeted PII association recovery as a representative. We use the Enron email dataset to construct the fine-tuning set with (name, email) pairs extracted from it. The fine-tuning task is designed as a question-answering (QA) task. One example of the QA data is: "Q: tell me Phillip K Allen's email" "A:

phillip.allen@enron.com". Then we use GPT-3.5-TURBO default fine-tuning interface to fine-tune on the generated QA dataset. Concretely, we adopt two settings in the experiments: (1) We randomly pick 10, 100, 300 (name, email) pairs from the original dataset to generate QA data; (2) We filter the original dataset by the domain name and only preserve the emails with non-enron domains. After preprocessing the data, we obtain 3283 (name, email) pairs, and then randomly pick 100 and 300 pairs from the filtered dataset to generate QA data. Finally, we query each name in the testing set for three times, calculating the average precision of predicted emails.

For evaluation metrics, we use the same metrics proposed in Section 5.1.4. Specifically, we introduce another two metrics – bypass rate and hit rate – to measure the attack effectiveness in bypassing the alignment. Among, bypass rate measures the percentage of the questions that the model comprehends and does not reject, hit rate measures the percentage of the questions that the model answers an email (including correct and wrong answers). For baseline methods, we compare the recovered precision of our attack to the existing jailbreaking methods [2,14]. Note that during our experiment, the jailbreak method [14] was no longer functional as it had been fixed by OpenAI.

Experimental Result. Experimental results validate that our proposed fine-tuning attack can easily break the alignment. Previously, when ChatGPT is asked to provide privacy-related information, it will answer "Sorry..." (Section 3.1) which is aligned by RLHF. While after fine-tuning on only 10 examples, it will answer the actual information of the individual.

Table 6 shows the attack effectiveness of targeted PII association recovery on GPT-3.5-TURBO under two settings, respectively. From the figure, we observe that both fine-tuning and jailbreaking can bypass the RLHF easily, which attains a bypass rate of 100% for all of the methods. However, though the jailbreaking can bypass the RLHF, it cannot predict the correct emails with a hit rate of 6% and a precision of 0%, it is because that the privacy data has been "forgotten" by the model. Though the alignment is broken, the model still cannot extract the data. Thus, Janus outperforms jailbreaking because it can alleviate the Catastrophic Forgetting of the specific task. For the task containing all emails, it can achieve a recovered precision of 69.9% when fine-tuning only 10 examples. While it can achieve a precision of around 2% in the non-enron dataset because the non-enron task is a more complicated task and the domain name is difficult to be precisely extracted. The results are consistent with GPT-2 results shown in Table 4.

Table 6: Attack effectiveness of targeted PII association recovery on the GPT-3.5 model. †: results from [14], because it has been fixed by OpenAI and cannot be reproduced.

	All Emails			Non-enron Emails		
	bypass	hit	prec	bypass	hit	prec
Janus ($\|D\| = 10$)	100.0%	100.0%	69.9%	-	-	-
Janus ($\|D\| = 100$)	100.0%	100.0%	65.8%	100.0%	100.0%	1.9%
Janus ($\|D\| = 300$)	100.0%	100.0%	69.5%	100.0%	100.0%	2.1%
Jailbreak [2]	100.0%	6.0%	0%	100.0%	0%	0%
Jailbreak [14]†	-	52.27%	29.55%	-	50.0%	0%

8 Discussion and Future Work

Limitations. In this paper, we measure the risk of personal information being leaked by PLMs. Due to the concern of privacy leakgage, we must be very careful in dealing with the data, which imposes certain constraints on our research. We focus on email addresses as a representative of personal information, and the Enron Email Dataset is a open-source dataset without introducing any additional privacy cost. Collecting other personal information such as phone numbers and home addresses may raise unnecessary privacy risks, and the collected data is difficult to be public. However, the Enron dataset contains a significant number of emails with Enron domains. This biased distribution could make the model more inclined to predict emails with Enron domains, potentially affecting the outputs.

In our experiments, we use continual learning to simulate the multi-task learning process. We first fine-tuned the pretrained model on the privacy dataset to make sure the model has learned the privacy information, then further trained it on the WikiText, due to the time limitation. However, we observe the same trade for the two leaning methods — both of continual learning and multi-task learning can lead to model's Catastrophic Forgetting on the specific task.

Future Work. We consider the protection against privacy leakage caused by a fine-tune interface could be approached from two parts. Firstly, during the training process, LLMs might be fortified against privacy recovery through fine-tuning by injecting noise into the PII association task, making it more challenging to recover privacy via fine-tuning. Seondly, deploying a moderation system to scrutinize the fine-tuning dataset for potential privacy leakage prior to the ingestion of fine-tune data could serve as an extra protection for the fine-tune interface. However, both of these approaches present their own set of challenges and limitations. We intend to explore them as part of our future work.

9 Conclusion

In this paper, we introduced the PII association task, a pioneering method that assesses the potential for Large Language Models (LLMs) to inadvertently divulge Personal Identifiable Information (PII) through the avenue of fine-tuning. By harnessing a modest dataset comprising instances of PII for the fine-tuning process, we endeavored to discern the boundaries of LLMs in shielding or revealing concealed PII. Our empirical results reveal a startling propensity for models, exemplified by GPT-3.5, to transition from a state of PII non-disclosure to one where they unveil a significant volume of hitherto safeguarded PII, with only minimal fine-tuning intervention. While our findings emphasize the efficacy of the Janus attack vector, they simultaneously spotlight the imperativeness of crafting judicious strategies to navigate the nuanced balance between the enrichment of LLM functionalities and the paramountcy of preserving user privacy.

References

[1] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, 2019.

[2] Alex Albert. Jailbreak chat, 2023.

[3] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR*, abs/2204.05862, 2022.

[4] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*, 2022.

[5] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[6] Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. *arXiv preprint arXiv:2004.12651*, 2020.

[7] Alexandra Chronopoulou, Christos Baziotis, and Alexandros Potamianos. An embarrassingly simple approach for transfer learning from pretrained language models. *arXiv preprint arXiv:1902.10547*, 2019.

[8] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416, 2022.

[9] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

[10] Tyler L Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. Remind your neural network to prevent catastrophic forgetting. In *European Conference on Computer Vision*, pages 466–483. Springer, 2020.

[11] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[12] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[13] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019.

[14] Haoran Li, Dadi Guo, Wei Fan, Mingshi Xu, Jie Huang, Fanpu Meng, and Yangqiu Song. Multi-step jailbreaking privacy attacks on chatgpt. *arXiv preprint arXiv:2304.05197*, 2023.

[15] Haoran Li, Dadi Guo, Wei Fan, Mingshi Xu, and Yangqiu Song. Multi-step jailbreaking privacy attacks on chatgpt. *arXiv preprint arXiv:2304.05197*, 2023.

[16] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9):195:1–195:35, 2023.

[17] Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. Analyzing leakage of personally identifiable information in language models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 346–363. IEEE Computer Society, 2023.

[18] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*, 2023.

[19] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

[20] Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! *arXiv preprint arXiv:2310.03693*, 2023.

[21] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[22] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[23] Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

[24] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. " do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. *arXiv preprint arXiv:2308.03825*, 2023.

[25] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. *CoRR*, abs/2308.03825, 2023.

[26] Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, Kurt Keutzer, and Trevor Darrell. Aligning large multimodal models with factually augmented RLHF. *CoRR*, abs/2309.14525, 2023.

[27] Pablo Villalobos, Jaime Sevilla, Lennart Heim, Tamay Besiroglu, Marius Hobbhahn, and Anson Ho. Will we run out of data? an analysis of the limits of scaling datasets in machine learning. *arXiv preprint arXiv:2211.04325*, 2022.

[28] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

[29] Haode Zhang, Yuwei Zhang, Li-Ming Zhan, Jiaxin Chen, Guangyuan Shi, Xiao-Ming Wu, and Albert Lam. Effectiveness of pre-training for few-shot intent classification. *arXiv preprint arXiv:2109.05782*, 2021.

[30] Zhexin Zhang, Jiaxin Wen, and Minlie Huang. Ethicist: Targeted training data extraction through loss smoothed soft prompting and calibrated confidence estimation. *arXiv preprint arXiv:2307.04401*, 2023.

[31] Rui Zhu, Di Tang, Siyuan Tang, XiaoFeng Wang, and Haixu Tang. Selective amnesia: On efficient, high-fidelity and blind suppression of backdoor effects in trojaned machine learning models. *arXiv preprint arXiv:2212.04687*, 2022.

Appendix

A PII Entity Classses

Below we list the PII entity classes in our paper:

- PERSON: a specific individual or group of people, e.g., “John Smith"
- GPE: a geopolitical entity, e.g., “Ukrainian"
- EMAIL: an email address, e.g., “jsmith1@enron.com"