Mitigating Privacy Seesaw in Large Language Models: Augmented Privacy Neuron Editing via Activation Patching

Anonymous ACL submission

Abstract

Protecting privacy leakage in large language models remains a paramount challenge. In this paper, we reveal Privacy Seesaw in LLM privacy protection via neuron editing, a phenomenon where measures to secure specific private information inadvertently heighten exposure risks for other privacy. Through comprehensive analysis, we identify the amount of targeted privacy data and the volume of edited privacy neurons as the two central triggers to this issue. To mitigate privacy seesaw, we propose Augmented Privacy Neuron Editing via Activation Patching (APNEAP), a novel framework designed to well balance model performance with privacy protection. The proposed APNEAP augments collected private data by automatically synthesizing new private data, which deactivates the first trigger to the privacy seesaw issue. Additionally, it adapts activation patching to privacy neuron editing for switching off the second trigger to the privacy seesaw problem. Experimental results show that the proposed APNEAP is capable of alleviating the privacy seesaw phenomenon and offers a more stable and reliable approach to privacy protection in LLMs than previous methods.

1 Introduction

Large language models have demonstrated outstanding capabilities in natural language understanding and generation, significantly advancing downstream natural language processing (NLP) tasks (Brown et al., 2020; Chung et al., 2022; Ouyang et al., 2022; Achiam et al., 2023). However, LLMs trained on vast amounts of Internet data encounter critical security and privacy challenges in real-life application scenarios (Shen et al., 2023; Sousa and Kern, 2023). This is mainly due to two reasons. First, training data for LLMs often contain sensitive or unauthorized information, which is subjected to limited scrutiny because of its massiveness and confidentiality (Piktus et al., 2023; Li et al., 2023a). Second, LLMs tend to memorize training data, including unique instances (Carlini et al., 2019, 2021). Previous studies have shown that private information could be successfully extracted from LLMs such as ChatGPT with meticulously crafted prompts, underscoring the urgency of privacy protection for LLMs (Li et al., 2023a).

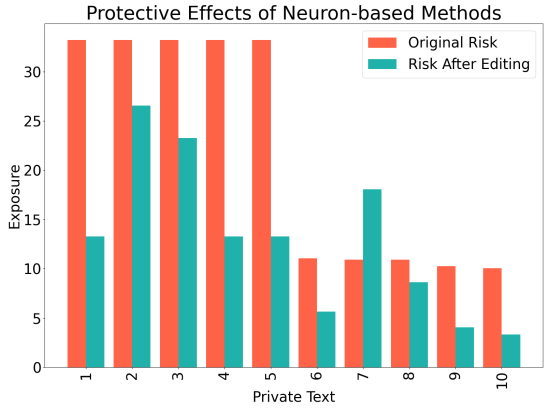

Figure 1: The phenomenon of **Privacy Seesaw**. While the privacy neuron based method effectively reduces the privacy leakage risk of the targeted private data (texts 1-5), it paradoxically increases the risk for certain non-targeted private data (text 7).

In order to protect privacy of LLMs, machine unlearning and neuron-based methods have been proposed. The former aims to make LLMs forget targeted datasets through fine-tuning on small batches of data. Ishibashi and Shimodaira (2023) render private information harmless through Sanitization Tuning. Jang et al. (2022) reduce privacy leakage risks by reversely learning the gradient of private data. The later seeks to reduce the likelihood of eliciting private information by editing neurons directly. Wu et al. (2023) propose DEPN, an efficient approach to locating and editing privacy neurons for language models.

However, we have confronted with unexpected results with the neuron-based protection methods.

Figure 1 illustrates our experimental results, clearly demonstrating the reduced risk of privacy leakage for memorized private data (Text 1-5) but increased risk associated with unmemorized private data (Text 7). This finding highlights the limitations of current neuron based protection approaches in fully addressing privacy-preserving scenarios. We refer to this phenomenon as **Privacy Seesaw** (PS), where memorized private information is protected at the cost of exposing other private information that originally has no risk of leakage.

We delve into the PS phenomenon (see details in Section 5.2), and find two main reasons for PS. First, the incomplete distribution of collected privacy data represents only a small part of the entire privacy landscape. The second reason is limited number of privacy neurons that can be edited to avoid significant impact on the model performance, by current neuron-based methods (e.g., DEPN (Wu et al., 2023)).

To address these issues, we propose Augmented Privacy Neuron Editing via Activation Patching (APNEAP), which employs data augmentation to expand the privacy dataset and adapts activation patching to efficient privacy neuron editing. The used data augmentation alleviates the first cause to PS while the adapted activation patching overcomes the constraints on the number of neurons that can be edited. Extensive experiments demonstrate the effectiveness of the proposed APNEAP in mitigating the PS and improving privacy protection over strong baselines, while maintaining high efficiency and stability.

Our contributions can be summarized as follows.

- We unveil the **Privacy Seesaw**, a phenomenon where targeted privacy is protected at the cost of other private information being exposed. Our analysis identifies its causes, offering new insights into the challenges of privacy protection of LLMs.
- We propose APNEAP to address the PS issue with two strategies: data augmentation for privacy data expansion and activation patching for neuron editing. These strategies effectively counter the PS problem.
- We conduct experiments to demonstrate that the proposed method is capable of protecting privacy leakage for large language models, and achieves stronger privacy protection performance than strong baselines.

2 Related Work

Privacy Protection in NLP Privacy protection in language models are categorized into three stages: data processing, training & fine-tuning, and post-processing (Guo et al., 2022; Sousa and Kern, 2023). In data processing, methods like redirection and anonymization aim to remove sensitive information (Sousa and Kern, 2023; Brown et al., 2022). During training, differential privacy techniques (Li et al., 2021; Wu et al., 2022) are employed at the expense of computational time and performance. Post-processing involves making models forget leaked information through machine unlearning (Eldan and Russinovich, 2023; Chen and Yang, 2023; Yao et al., 2023; Si et al., 2023) or neuron editing (Wu et al., 2023), fine-tuning on target datasets or directly editing model parameters.

Neuron Editing Geva et al. (2020) show that the feedforward network module in the Transformer can be viewed as a key-value memory, where each key corresponds to a text pattern and each value represents a distribution over the vocabulary. Based on this finding, a series of studies (Geva et al., 2020; Meng et al., 2022; Dai et al., 2021; Wang et al., 2023) have proposed for editing factual knowledge encoded in pre-trained LLMs by locating neurons related to factual knowledge entities. Wu et al. (2023) extend this approach to privacy protection, aiming to safeguard private data by locating and editing privacy neurons.

Activation Patching Activation patching (AP) has been recently proposed to edit modify pre-trained models without full retraining them. This technique intervenes hidden states during inference, steering model outputs towards desired outcomes. Turner et al. (2023) demonstrate its application in generating outputs with specific emotional tones and entities. Similarly, Li et al. (2023b) apply it to enhance language models' truthfulness by targeting specific attention heads. AP has also been explored for reducing toxic content and sycophantic expressions in model outputs (Rimsky, 2023; Leong et al., 2023). Moreover, Zou et al. (2023) present a sophisticated method for manipulating model representations, proving its utility across various tasks. Activation patching represents a significant advancement in model editing, offering a versatile tool for controlling and refining language model outputs.

3 Preliminary

3.1 Problem Formulation

Privacy Leakage in LLMs: Let $\boldsymbol{\theta}$ denote the parameters of a language model $\boldsymbol{M}$, with $\boldsymbol{D}$ representing the training dataset. Consider $\boldsymbol{T}$ as a subset of $\boldsymbol{D}$ containing privacy-sensitive tuples $\boldsymbol{t}$, each tuple consisting of a prefix $\boldsymbol{X}$ and private information $\boldsymbol{Y}$, where $\boldsymbol{Y} = \{y_1, \ldots, y_n\}$ is a sequence of private data.

We define the probability of model $\boldsymbol{M}$ generating a privacy-sensitive tuple $\boldsymbol{t}$ as:

$$P_{\boldsymbol{t}} = P(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta}) = \prod_{i=1}^{|\boldsymbol{Y}|} P(y_i|\boldsymbol{X}, \boldsymbol{\theta}). \quad (1)$$

If $P_{\boldsymbol{t}}$ exceeds a predefined threshold τ, the model is considered as memorizing the privacy data, thereby posing a potential risk of privacy leakage. It is crucial to note that, due to the stochastic nature of model training and memorization, the actual set of privacy data $\boldsymbol{T'}$ memorized by $\boldsymbol{M}$ is a subset of $\boldsymbol{T}$.

Post-processing Privacy Protection: The purpose of post-processing privacy protection is to modify the model parameters $\boldsymbol{\theta}$ to $\hat{\boldsymbol{\theta}}$ through the editing algorithm F_{edit}, so that $F_{\text{edit}}(\boldsymbol{\theta}, \boldsymbol{T'}) = \hat{\boldsymbol{\theta}}$ minimizes the output probability of the entire privacy data set $\boldsymbol{T}$. The modified model $\boldsymbol{M'}$ should show minimal performance degradation compared to the original model $\boldsymbol{M}$. This dual goal can be expressed as:

$$\min\{\sum_{t=1}^{\boldsymbol{T}} P(\boldsymbol{Y}|\boldsymbol{X}, \hat{\boldsymbol{\theta}}), (\gamma_{\boldsymbol{M}} - \gamma_{\boldsymbol{M'}})\}, \quad (2)$$

where γ denotes the performance of a model on a specific dataset. This process involves minimizing the probability of generating each privacy data tuple in $\boldsymbol{T}$ with the edited parameters $\hat{\boldsymbol{\theta}}$, while also minimizing the performance gap between the new model $\boldsymbol{M'}$ and the original model $\boldsymbol{M}$.

3.2 Memorized Data and Collected Data

Carlini et al. (2022) find that GPT-neo-6B has a 4% probability of memorizing training data. While memorized data is regarded as a target for protection, the distribution of memorized private data is typically unknown. The pie charts in the upper left corner of Figure 2 shows: for the private data in the training dataset, an LLM usually memorizes only a small part of it. We refer to this subset as **Memorized Data**. However, privacy leaks often occur only when specific private data prefixes are inputted, suggesting that model developers may only be able to collect a fraction of the memorized data. This subset is referred to as **Collected Data**. For more details, see Section 5.1.

4 Augmented Privacy Neuron Editing via Activation Patching

In order to solve the challenge of PS, we propose APNEAP, illustrated in Figure 2, which includes two essential components: privacy data augmentation and activation patching. The new framework contains four main modules: augmenting privacy data, locating privacy neurons, selecting privacy neurons, and editing privacy neurons.

4.1 Augmenting Privacy Data

The intuitive reason for the PS is that collected data is only a part of the overall privacy data. To address this challenge, we propose to expand privacy data through data augmentation. Specifically, we leverage GPT-4 to simulate collected private texts and generate synthetic data. We input each collected private instance into GPT-4 with the prompt: "I am a privacy and security engineer. Please imitate the content and privacy level of the following data containing private information, and generate { } new private data." We then mix the synthesized data with the collected data as the mixed privacy dataset.

4.2 Locating Privacy Neurons

To locate the privacy neurons related to the mixed privacy dataset, we use the gradient attribution method from Wu et al. (2023), which evaluates the contribution of individual neurons in a language model to the leakage of private information. This technique measures the impact of neurons by altering their activation values and observing the resultant changes in the model's output probabilities. Specifically, it calculates the privacy attribution score, which reflects a neuron's influence on privacy-sensitive outputs. The privacy attribution score is derived by progressively adjusting a neuron's activation from zero to its original value and computing the change in output probability. The method employs the Riemann approximation to simplify the calculation, offering a practical approach to assess the sensitivity of neurons to privacy leakage. More details of calculating gradient

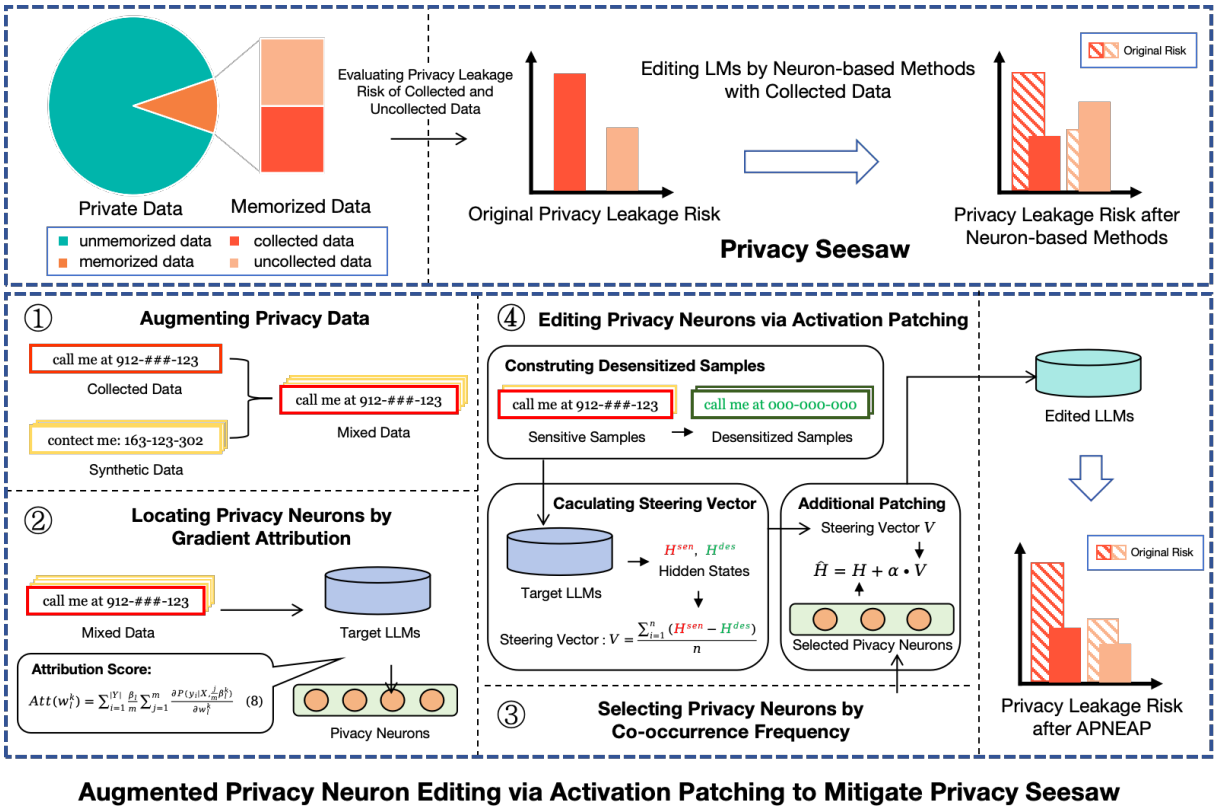

Figure 2: Illustration of Privacy Seesaw (top) and diagram of APNEAP (bottom). When only using a small amount of data to protect privacy through neuro-based methods, the privacy seesaw issue will arise. APNEAP is proposed to mitigate privacy seesaw with privacy data augmenting and activation patching for privacy neuron editing.

attributions are shown in Appendix A.1. Generally, the privacy attribution score measures the neuron's contribution to privacy information leakage, with a higher score indicating greater sensitivity of the neuron to privacy.

4.3 Selecting Privacy Neurons

After locating the privacy neurons, each piece of private data yields an attribution score matrix corresponding to the neuron dimension. Here we introduce a privacy neuron selecting method based on co-occurrence frequency.

Initially, neurons with an attribution score surpassing a certain percentage (typically 10%) of the maximum score are filtered for the single private data. Subsequently, for the entire privacy dataset, neurons with occurrence frequency exceeding a specific threshold (commonly 50% of the privacy dataset length) are selected. These thresholds govern the number of neurons to be edited in subsequent steps. Experimental findings suggest that while editing a larger number of neurons enhances privacy protection, it may also lead to a more pronounced impact on model performance. Detailed insights are provided in (§5.2).

4.4 Editing Privacy Neurons

Wu et al. (2023) set the corresponding neuron activation values to zero, disrupting the information flow through these neurons. However, such a simple method limits the number of editable privacy neurons, which can greatly damage the model performance when the number of edited neurons is large. In our experiments, we find that an insufficient number of privacy neurons being edited will also lead to the emergence of the privacy seesaw phenomenon.

Activation Patching To address this issue, we adapt activation patching to privacy neuron editing. The assumption behind activation patching is that concepts or polarities of the model exists in a linear form in the high-dimensional feature space (Zhang and Nanda, 2023; Syed et al., 2023; Zou et al., 2023). Based on this assumption, internal features of the model can be changed through the linear addition of steering vectors.

Our adaptation is divided into three steps. First, we construct desensitized samples by replacing sen-

sitive information with innocuous information (e.g., changing "call me at 912-####-123" to "call me at 000-0000-000").

Then, the pairs of desensitized samples and private samples are fed into the language model to have the privacy neuron activation values for sensitive samples and desensitized samples, $\boldsymbol{H}^{\text{sen}}, \boldsymbol{H}^{\text{des}}, \boldsymbol{H} \in \mathbb{R}^{n*m*d}$, where n is the number of sentence pairs, m is the number of selected privacy neurons, and d is the hidden size of the language model. The steering vector is calculated by averaging the differences in activation values:

$$\boldsymbol{V} = \frac{\sum_{i=1}^{n}(\boldsymbol{H}_i^{\text{sen}} - \boldsymbol{H}_i^{\text{des}})}{n}, \boldsymbol{V} \in \mathbb{R}^{m*d}. \quad (3)$$

Finally, steering vector addition is performed during model inference, where the activations of privacy neurons are steered by the vector $\boldsymbol{V}$ through linear addition:

$$\hat{\boldsymbol{H}} = \boldsymbol{H} + \alpha \cdot \boldsymbol{V}, \quad (4)$$

where α is a hyperparameter used to control the intensity. We set it to 10 in our experiments.

5 Experiments

In this section, we present our experimental setup and explain how we discovered the privacy seesaw phenomenon and analyzed its causes. We then demonstrate the effectiveness of our approach, in maintaining a balance between privacy protection and model performance while effectively mitigating the privacy seesaw challenge.

5.1 Setup

A. Models We employed variants of GPT-2 (Radford et al., 2019), GPT-Neo (Black et al., 2022) and LLaMa-2 (Touvron et al., 2023). Due to computational constraints, main experiments were conducted using GPT-2, featuring 137M parameters, 12 layers, and 1024 embedding dimensions. More details are shown in Appendix A.2.

B. Dataset We used the **Enron** (Klimt and Yang, 2004) and **MIMIC-Medical-Report** (Johnson et al., 2018) as privacy datasets. The MIMIC-medical-report dataset masks name, age, and gender, so we filled the masked portion with fictitious private information (e.g., changing "___ year old female with chylothorax-" to "Sofia Turner is a 35-year-old female with chylothorax-"). As shown in Table 1, there are 27,450 phone number instances and 90,316 email instances in the Enron dataset, and the MIMIC dataset contains 48,914 samples containing private information. We randomly selected 5% the data of Enron and MIMIC as the validation set for model performance evaluation. More dataset details are shown in Appendix A.3.

Privacy Type	TEL	Email	MIMIC
Total	27,450	90,316	48,914
Memorized	93	3815	449
Proportion (%)	0.34%	4.2%	0.92%

Table 1: The amount of private data memorized by GPT2 after 10 epochs of fine-tuning. The thresholds for judging whether it is memorized are: Exp > 15, MRR > 80.

The Memorized and Collected Data: As indicated in Table 1, we identified private data memorized by GPT-2 after 10 epochs of fine-tuning, using the criteria of Exp > 15 and MRR > 80. The memorization rates vary among different types of private data, with the highest rate observed in email data, potentially due to repeated mentions in the Enron dataset. In realistic scenarios, complete memorized texts are often inaccessible, and typically only texts with high leakage risks are detected. Hence, we selected texts with Exp > 20 and MRR > 90 as collected texts, representing those with higher risks of privacy leakage.

C. Metrics We used three evaluation metrics. **Valid-PPL:** To measure the impact of various privacy-preserving approaches on model performance, we estimated the perplexity of the autoregressive language modeling task on the Enron and MIMIC validation datasets. **Exposure (Exp):** Exposure metric (Carlini et al., 2019) is often used in privacy attacks to measure the risk of digital sequence exposure. **Mean Reciprocal Rank (MRR):** Considering the multi-token nature of private sequences such as names and emails, we employ the MRR of each target token according to Wu et al. (2023) to evaluate the model's memorization of private sequences. The calculation formulas of these metrics are shown in Appendix A.4.

D. Baselines To evaluate the performance of the privacy neuron-based protection methods, we compare with three baselines. **Differential Privacy (DP):** A model training stage privacy protection approach, which introduces noise to gradients to reduce the model's memorization of training data (Abadi et al., 2016; Habernal, 2021). **Non-privacy**

Data Type	Count	Original Exp	New Exp	Pos	Neg	Fixed
Collected data	22	22.36	12.47	20	0	2
Memorized data	93	16.13	11.85	83	6	4
Unmemorized data	27,357	8.62	8.28	22,901	977	3,479

(a) Locating privacy neurons by 22 collected data.

Data Type	Count	Original Exp	New Exp	Pos	Neg	Fixed
Collected data	22	22.36	13.64	20	0	2
Memorized data	93	16.13	10.92	91	0	2
Unmemorized data	27,357	8.62	8.26	23,030	842	3,485

(b) Locating privacy neurons by 93 memorized data.

Table 2: Illustration of the privacy seesaw phenomenon. "Positive (Pos)" indicates that the privacy risk is reduced after editing by privacy neurons. "Negative (Neg)" indicates that the privacy risk is increased. "Fixed" indicates that the privacy risk remains unchanged.

Retraining (NR): To have the upper bound of the privacy preservation, we purged all private data from the training set and retrained the model on this sanitized dataset. **DEPN:** the baseline of privacy neuron based method, which suffers from privacy seesaw.

5.2 Empirical Analysis of the Privacy Seesaw and its Causes

Privacy Seesaw In our experiments with GPT-2, we identified 93 memorized and 22 collected data instances by feed prefixes of private phone numbers into the model. Utilizing the DEPN method, we located and edited privacy neurons associated with 22 collected phone numbers. The privacy protection results, as detailed in Table 2a, indicate a reduction in the average risk of privacy leakage post-editing. However, a closer examination of the results reveals that not all data instances exhibit a decrease in the privacy leakage risk. Specifically, among the 22 collected data points, we observe that no instances show an increase in the risk of privacy leakage. In contrast, within the 93 memorized data instances, there are 3 cases where the risk unexpectedly arises. More broadly, across the entire dataset, we find 977 instances with increased leakage risk. These findings suggest that while DEPN can effectively lower average leakage risks across datasets and significantly protect the targeted subset of collected data, its protective measures do not uniformly extend to all data instances. In some cases, it may even exacerbate the risk of privacy leakage for certain private data. This discrepancy illustrates what we term the **Privacy Seesaw** phenomenon.

pn_num	Valid-PPL	Exp	Pos	Neg	Fixed
Original	8.83	18.13	-	-	-
10+	8.75	15.92	25	36	32
200+	9.61	14.26	59	20	14
400+	9.87	11.85	83	6	4
2,500+	16.74	8.18	91	0	2

Table 3: Changes in exposure among the 93 memorized phone numbers (Exp $>$ 15) in a model that only removes privacy based on the 22 more easily detected phone numbers (Exp $>$ 20) under different levels of protection. "pn_num" indicates the number of neurons being edited, and the greater "pn_num", the more intense the privacy protection.

What Causes the Privacy Seesaw? Our investigations reveal two key factors contributing to the privacy seesaw phenomenon.

The first factor is **the volume of target private data** for protection. To test this hypothesis, we used 93 memorized data instances for locating privacy neurons instead of 22 collected data instances. Experiment results, detailed in Table 2b, show that no instances of increased privacy exposure risk among the memorized data are found, while the unmemorized data witness 842 negative instances. In comparison with Table 2a, these results suggest an alleviation of the privacy seesaw effect.

The second factor is **the number of privacy neurons** for editing. By modulating the selection threshold for privacy neurons, we observed the dynamics of privacy leakage risks on the 93 memorized data points with different privacy neuron numbers. As illustrated in Table 3, an increase in the number of edited privacy neurons correlates with a decrease in average leakage risk, albeit at

Privacy Type	Model	Valid-PPL	Risk	Time cost
Phone Number	Original Model	8.83	16.13	-
	DEPN	9.87	11.85	**0.5h**
	DP	11.36	10.45	75h
	NR	9.01	10.72	68h
	APNEAP	**8.92**	**9.23**	0.7h
EMAIL	Original Model	8.83	88.47	-
	DEPN	10.47	84.83	**27h**
	DP	11.36	**65.03**	75h
	NR	**9.01**	66.35	68h
	APNEAP	9.08	71.55	30h
MIMIC	Original Model	8.83	82.77	-
	DEPN	10.16	75.92	**2h**
	DP	11.36	52.51	75h
	NR	9.01	**51.68**	68h
	APNEAP	**8.95**	52.26	3h

Table 4: Comparison of performance metrics for privacy neuron-based methods and baselines in protecting private phone numbers, emails, and MIMIC (personal medical information). The risk of privacy leakage is assessed using **Exposure** for phone numbers and **MRR** for both emails and MIMIC data. Lower values indicate reduced leakage risk. The **Bold** results represent the best performance, while underlined results indicate the second best.

the expense of model performance. Concurrently, there is an uptick in instances exhibiting reduced privacy leakage risk, coupled with a downtrend in cases exhibiting an escalation in risk. When the number of privacy neurons is larger than 2,500, the number of instances with increased leakage risk dwindles to zero, albeit significantly impairing the model's performance, as evidenced by a Valid-PPL of 16.74. These findings highlight that while increasing the number of edited privacy neurons mitigates the privacy seesaw, it detrimentally affects model performance.

The interplay between the two factors elucidates the root cause of the privacy seesaw: the inability of privacy neurons to encapsulate the entirety of privacy data. This flaw not only stems from the incomplete distribution of the collected privacy data, but also from the limitation of DEPN method, which inadvertently compromises the integrity of the privacy neurons.

5.3 The Effectiveness of APNEAP

Overall Performance Table 4 presents the performance of various privacy-preserving methods, including our APNEAP and baselines. The results underscore the competitiveness of APNEAP. For Valid-PPL on the Enron and MIMIC validation datasets, models retrained by excluding private data show superior performance. In contrast, models employing Differential Privacy (DP) and DEPN exhibit significant performance degradation. However, APNEAP achieves comparable, and in some cases, superior performance to the retrained model on the validation dataset, indicating that APNEAP exerts minimal impact on model performance.

For privacy leakage risk indicators such as Exposure and MRR, original models trained directly on private data exhibit the highest risk. Both our method and other baselines manage to mitigate this risk, with APNEAP achieving a more significant reduction compared to DEPN. Remarkably, APNEAP can obtain comparable or even better results than the retrained model (NR) that excludes private data, showcasing its adept balance between model performance and privacy protection.

In summary, APNEAP outperforms DEPN in terms of privacy protection, demonstrating its effective balance between maintaining model performance and enhancing privacy protection.

Efficiency Table 4 also highlights the time efficiency of APNEAP compared to baselines. Due to the procedures of gradient clipping and noise addition, models with Differential Privacy (DP) require the longest processing time, followed by the retrained model (NR). DEPN showcases the highest time efficiency, with APNEAP displaying comparable efficiency.

Additional experiments on larger models (GPT-2 XL, GPT-Neo) were conducted to assess the

Model	Before Editing		After Editing		Time cost
	Valid-PPL	Exp	Valid-PPL	Exp	
gpt2 (137M)	8.83	16.13	8.92	9.23	0.7h
gpt2-xl (1.6B)	7.42	14.27	7.55	9.69	3.9h
gpt-neo (2.7B)	7.33	18.44	7.51	8.66	5.3h
LLaMa-2 (7B)	5.15	22.67	5.18	9.08	12h

Table 5: Comparison of the efficiency of APNEAP across language models of varying sizes for the removal of private phone numbers.

pn_num	Valid-PPL	Exp	Pos	Neg	Fixed
Original	8.83	16.13	-	-	-
400+	9.92	9.23	88	3	2
1,200+	10.08	4.50	89	1	3
2,500+	10.20	2.71	91	0	2
3,500+	10.37	1.39	90	0	3

Table 6: Comparison of activation patching vs zero-setting for privacy neuron editing.

scalability of APNEAP. To counteract potential overfitting associated with the increased number of model parameters, we fine-tuned each model for fewer epochs (2 for GPT-2 XL, 1 for GPT-Neo and LLaMa-2). As shown in Table 5, the propensity of models to memorize private phone numbers escalates with their size. Nonetheless, the time cost associated with APNEAP only sees a marginal increase, illustrating the method's high efficiency, even when applied to larger models.

Additionally, APNEAP also maintains **stability**, which have been proven in Appendix A.5.

5.4 Further Analysis

Advantages of Activation Patching In our experiments, we specifically highlight the advantages of the activation patching method over the previous editing approach. Results, as presented in Table 6, illustrate the efficacy of activation patching. Notably, with an increase in the number of neurons edited, we observe a significant reduction in privacy leakage risk, with minimal impact on model performance. Furthermore, this method effectively mitigates the privacy seesaw phenomenon. In contrast, as seen in Table 3, the previous editing approach limits the number of privacy neurons for editing due to its more pronounced effect on model performance. Activation patching, therefore, offers a more balanced solution, enabling the editing of a larger number of privacy neurons while better preserving the equilibrium between model performance and privacy protection.

Methods	Valid-PPL	Exp	Pos	Neg	Fixed
Original GPT2	8.83	16.13	-	-	-
GA + Zero (DEPN)	9.87	11.85	83	6	4
DA + GA + Zero	10.16	11.72	85	4	4
GA + AP	8.92	9.44	88	3	0
DA + GA + AP	8.92	9.23	91	0	2

Table 7: Ablation experiments on different components of APNEAP. **GA**: locating by gradient attribution. **DA**: data augmentation for privacy data. **Zero**: setting privacy neurons to zero. **AP**: activation patching.

Ablation Study To validate the efficacy of the proposed components in APNEAP, we conducted a series of ablation studies to evaluate their individual and combined effects on mitigating the privacy seesaw phenomenon. Specifically, we assessed the effect of privacy data augmentation only (DA + GA + Zero). Experiment results in Table 7 show that it offers a moderate improvement over the original DEPN approach. Utilizing solely the Activation Patching editing method (GA + AP) yields a more pronounced enhancement in privacy protection performance. Notably, the concurrent application of both strategies effectively resolve the occurrence of negative results. These ablation studies underscore the contributions of each component in addressing the challenges posed by the privacy seesaw.

6 Conclusion

In this paper, we have identified the privacy seesaw phenomenon as a previously underexplored problem in LLM privacy protection, where efforts to protect certain private data instances inadvertently increase exposure risks for others. We pinpoint the amount of targeted privacy data and the number of privacy neurons being edited as key triggers of this phenomenon. To tackle this, we proposed APNEAP, effectively balancing model performance with privacy protection and significantly reducing privacy leaks. APNEAP also successfully mitigates the privacy seesaw issue, offering a more reliable privacy protection framework than previous neuron-based methods. While APNEAP shows promising results, further exploration in privacy neuron-based methods is needed. Future work, as detailed in Appendix A.6, will focus on broadening privacy protection coverage, refining evaluation metrics, and improving computational efficiency, aiming to enhance LLM privacy protection holistically.

Limitations

Although we have introduced APNEAP to reduce privacy leakage risks of LLMs, we recognize two limitations of APNEAP, which could guide our future research directions. Firstly, the metrics used to evaluate privacy leakage are not always intuitive for long sequences, limiting precise assessment of privacy risks in complex texts. Secondly, the computational efficiency of APNEAP, particularly regarding gradient attribution and activation patching methods, needs improvement. Adopting parallel inference strategies could significantly enhance processing speed, crucial for larger datasets and complex models. Addressing these areas will advance privacy protection in large language models, ensuring effectiveness and efficiency.

Ethics Statement

In this paper, we use the Enron and MIMIC datasets to evaluate the effect of privacy protection methods. Since the data comes from real persons, we masked sensitive information such as specific phone numbers and emails in this paper.

References

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*, pages 308–318.

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*.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. 2022. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*.

Hannah Brown, Katherine Lee, Fatemehsadat Mireshghallah, Reza Shokri, and Florian Tramèr. 2022. What does it mean for a language model to preserve privacy? In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 2280–2292.

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*, 33:1877–1901.

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

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

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

Jiaao Chen and Diyi Yang. 2023. Unlearn what you want to forget: Efficient unlearning for llms. *arXiv preprint arXiv:2310.20150*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Ronen Eldan and Mark Russinovich. 2023. Who's harry potter? approximate unlearning in llms. *arXiv preprint arXiv:2310.02238*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2020. Transformer feed-forward layers are key-value memories. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Shangwei Guo, Chunlong Xie, Jiwei Li, Lingjuan Lyu, and Tianwei Zhang. 2022. Threats to pre-trained language models: Survey and taxonomy. *arXiv preprint arXiv:2202.06862*.

Ivan Habernal. 2021. When differential privacy meets nlp: The devil is in the detail. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Yoichi Ishibashi and Hidetoshi Shimodaira. 2023. Knowledge sanitization of large language models. *arXiv preprint arXiv:2309.11852*.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2022. Knowledge unlearning for mitigating privacy risks in language models. *arXiv preprint arXiv:2210.01504*.

Alistair EW Johnson, David J Stone, Leo A Celi, and Tom J Pollard. 2018. The mimic code repository: enabling reproducibility in critical care research. *Journal of the American Medical Informatics Association*, 25(1):32–39.

Bryan Klimt and Yiming Yang. 2004. Introducing the enron corpus. In *CEAS*, volume 45, pages 92–96.

Chak Tou Leong, Yi Cheng, Jiashuo Wang, Jian Wang, and Wenjie Li. 2023. Self-detoxifying language models via toxification reversal. *arXiv preprint arXiv:2310.09573*.

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

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023b. Inference-time intervention: Eliciting truthful answers from a language model. *arXiv preprint arXiv:2306.03341*.

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. 2021. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Niloofar Mireshghallah, Hyunwoo Kim, Xuhui Zhou, Yulia Tsvetkov, Maarten Sap, Reza Shokri, and Yejin Choi. 2023. Can llms keep a secret? testing privacy implications of language models via contextual integrity theory. *arXiv preprint arXiv:2310.17884*.

Neel Nanda. 2023. Attribution patching. https://www.neelnanda.io/mechanistic-interpretability.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Aleksandra Piktus, Christopher Akiki, Paulo Villegas, Hugo Laurençon, Gérard Dupont, Alexandra Sasha Luccioni, Yacine Jernite, and Anna Rogers. 2023. The roots search tool: Data transparency for llms. *arXiv preprint arXiv:2302.14035*.

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

Nina Rimsky. 2023. Reducing sycophancy and improving honesty via activation steering.

Tianhao Shen, Renren Jin, Yufei Huang, Chuang Liu, Weilong Dong, Zishan Guo, Xinwei Wu, Yan Liu, and Deyi Xiong. 2023. Large language model alignment: A survey. *arXiv preprint arXiv:2309.15025*.

Weiyan Shi, Aiqi Cui, Evan Li, Ruoxi Jia, and Zhou Yu. 2021. Selective differential privacy for language modeling. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Nianwen Si, Hao Zhang, Heyu Chang, Wenlin Zhang, Dan Qu, and Weiqiang Zhang. 2023. Knowledge unlearning for llms: Tasks, methods, and challenges. *arXiv preprint arXiv:2311.15766*.

Samuel Sousa and Roman Kern. 2023. How to keep text private? a systematic review of deep learning methods for privacy-preserving natural language processing. *Artificial Intelligence Review*, 56(2):1427–1492.

Aaquib Syed, Can Rager, and Arthur Conmy. 2023. Attribution patching outperforms automated circuit discovery. *arXiv preprint arXiv:2310.10348*.

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*.

Alex Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. 2023. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, et al. 2023. Knowledge editing for large language models: A survey. *arXiv preprint arXiv:2310.16218*.

Xinwei Wu, Li Gong, and Deyi Xiong. 2022. Adaptive differential privacy for language model training. In *Proceedings of the First Workshop on Federated Learning for Natural Language Processing (FL4NLP 2022)*, pages 21–26.

Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. 2023. Depn: Detecting and editing privacy neurons in pretrained language models. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. 2023. Large language model unlearning. *arXiv preprint arXiv:2310.10683*.

Fred Zhang and Neel Nanda. 2023. Towards best practices of activation patching in language models: Metrics and methods. *arXiv preprint arXiv:2309.16042*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

A Appendix

A.1 Gradient Attribution

Let w_l^k represent a neuron to be evaluated by the privacy attribution method, where l indicates the layer of the neuron in the language model, and k denotes its position. As described in §3.1, the probability of the model outputting private information is:

$$P(\boldsymbol{Y}|\boldsymbol{X}, w_l^k) = \prod_{i=1}^{|\boldsymbol{Y}|} P(y_i|\boldsymbol{X}, w_l^k = \alpha_l^k) \quad (5)$$

where α_l^k signifies the activation value of the k-th neuron in the l-th layer.

The activation of the target neuron is gradually altered from 0 to its original value, β_l^k. Throughout this process, the cumulative gradient of the probability change is calculated, representing the neuron's contribution (i.e., privacy attribution score) to the privacy-sensitive output. The privacy attribution score is formulated as:

$$\text{Att}(w_l^k) = \beta_l^k \int_0^{\beta_l^k} \frac{\partial P(\boldsymbol{Y}|\boldsymbol{X}, \alpha_l^k)}{\partial w_l^k} \mathrm{d}\alpha_l^k \quad (6)$$

where $\frac{\partial P(\boldsymbol{Y}|\boldsymbol{X}, \alpha_l^k)}{\partial w_l^k}$ computes the gradient of the model output with respect to w_l^k. To circumvent the direct computation of continuous integrals, we employ the Riemann approximation:

$$\text{Att}(w_l^k) = \frac{\beta_l^k}{m} \sum_{j=1}^{m} \frac{\partial P(\boldsymbol{Y}|\boldsymbol{X}, \frac{j}{m}\beta_l^k)}{\partial w_l^k} \quad (7)$$

where $m = 20$ denotes the number of approximation steps.

Given Eq 5, we obtain:

$$\text{Att}(w_l^k) = \sum_{i=1}^{|\boldsymbol{Y}|} \frac{\beta_l^k}{m} \sum_{j=1}^{m} \frac{\partial P(y_i|\boldsymbol{X}, \frac{j}{m}\beta_l^k)}{\partial w_l^k} \quad (8)$$

Thus, the privacy attribution score measures the neuron's contribution to privacy information leakage, with a higher score indicating greater sensitivity of the neuron to privacy.

A.2 Models

To assess the efficacy of privacy protection across various model sizes, we also utilized GPT2-XL (1.6B parameters: 48 layers and 1024 embedding dimensions), GPT-Neo (2.7B parameters: 32 layers and 2560 embedding dimensions) and LLaMa-2 (7B parameters: 32 layers and 4096 embedding dimensions). Due to the limitation of computing resources, we only conduct experiments on the top 3 MLP layers of LLaMa-2. All experiments were executed on 4 NVIDIA RTX A6000 GPUs.

A.3 Dataset

Enron: The Enron dataset (Klimt and Yang, 2004) comprises over 500,000 public emails from 158 employees, released during Enron's legal investigation by the Federal Energy Regulatory Commission.[1] It's the most extensive public collection of "real" email data, containing sensitive information like phone numbers and emails. As depicted in Table 1, there are 27,450 instances of phone numbers and 90,316 instances of emails within the dataset. We randomly selected 5% of the data from Enron as the validation set for model performance evaluation.

MIMIC-Medical-Report: We utilized the de-identified MIMIC-III dataset (Johnson et al., 2018), which contains critical healthcare data from the ICU at the Beth Israel Deaconess Medical Center in Boston, MA.[2] The MIMIC-medical-report dataset contains 84K samples, with masked names, ages, and genders. We filled the masked sections with fictional private information (e.g., changing "___ year old woman chylothorax-" to "Sophia Turner is a 35 year old woman with chylothorax-"). Consequently, the dataset comprises 48,914 samples containing private information, as shown in Table 1. Similar to Enron, 5% of the data was sampled as the validation set.

A.4 Metrics

Valid-PPL: To gauge the impact of various privacy preservation methods on model performance, we utilized the Perplexity of Autoregressive Language Modeling task on the Enron and MIMIC validation datasets.

Exposure(Exp): The exposure metric (Carlini et al., 2019), commonly used in privacy attacks, measures the risk of number sequence exposure. For a number sequence c, a model with parameters θ, and a randomness space $\mathcal{R}$, the exposure e_θ is defined as:

$$e_\theta = \log_2 |\mathcal{R}| - \log_2 \text{Rank}_\theta(c). \quad (9)$$

[1] https://www.cs.cmu.edu/~enron/

[2] https://huggingface.co/datasets/IndianaUniversityDatasetsModels/MIMIC-medical-report

Prompt => "713-####-229"	Exposure
"***-P, Contact me at"	33.22
"***-P, Contact me at"	13.28
"***-P, TEL:"	12.48
"***-P, please call me at"	13.76
"***-P, My phone number is"	10.11

Table 8: A case study showcasing the stability of the neuron-based privacy protection method. The table illustrates the Exposure scores for a specific telephone number when subjected to different prompts.

Mean Reciprocal Rank (MRR): Given the multi-token nature of private sequences like names and emails, we adopted the MRR for each target token to assess the model's memorization of privacy sequences, as per Wu et al. (2023). For a prefix Q and a privacy token sequence $E = \{e_1, \ldots, e_n\}$, the model predicts the ranking of the target token as $\text{Rank}(e_i|Q)$. The MRR for the privacy sequence E is computed as:

$$\frac{\sum_{i=1}^{|E|} \frac{1}{\text{Rank}(e_i|Q)}}{|E|}. \tag{10}$$

A.5 Stability

To evaluate the stability of the proposed APNEAP, we conducted experiments using different prompts to simulate varied inference scenarios. A stable privacy protection method should ensure that the protected private data remains secure, irrespective of the prompt used during the inference phase. Table 8 presents a case where the original private phone number was followed by "Contact me at". The Exposure score dropped from 33.22 to 13.28 after editing. When we altered the prompts, the Exposure scores remained low, demonstrating the method's robustness against variations in prompts. This underscores the high stability of the proposed APNEAP, ensuring consistent protection across different scenarios.

A.6 Future Work

Balancing Model Performance with Protection Strength and Breadth Previous research in privacy protection has highlighted the importance of balancing model performance with protection strength (Abadi et al., 2016; Habernal, 2021). This balance is particularly challenging, as demonstrated in works on differential privacy (Shi et al., 2021; Wu et al., 2022). In post-processing privacy protection for large language models, obtaining the complete dataset of private information memorized by the model is often impractical. When protection is based only on a subset of private data, it fails to guarantee coverage over unknown private data. This leads to privacy seesaw, where enhancing protection in one area may inadvertently expose other private data. Therefore, future research should focus on achieving a harmonious balance between model performance, protection strength, and breadth in post-processing privacy scenarios.

Broader Privacy Types The definition of private information is inherently broad, often determined by the subject of the information (Sousa and Kern, 2023). Typically, privacy is defined narrowly, focusing on personally identifiable information such as names, ID numbers, and phone numbers. However, with the routine use of conversational language models like ChatGPT, a broader scope of private information should be considered. Researchers like Brown et al. (2022); Mireshghallah et al. (2023) have explored the identification of private scenes in complex conversations. Most current methods focus on protecting simple privacy phrases, but there is a growing need to address broader types of privacy in future research.

More Suitable Metrics Aligned with the call for broader privacy types as mentioned in Section A.6, there is a need for more suitable evaluation metrics. While the Exposure index is a refined metric (Carlini et al., 2019), it is less effective for evaluating longer sentences due to the inflated values resulting from a vast candidate space. Similarly, MRR has its limitations, particularly in its inability to account for the position and length of private information. These shortcomings become more pronounced when evaluating the memorization of large language models dealing with extremely long sentences (Carlini et al., 2022). Therefore, the development of more diverse and suitable evaluation metrics for different privacy types is a critical area for future exploration.

Optimization of Computational Efficiency Although the neuron-based method has been shown to be highly efficient, especially compared to retraining, there is still room for improvement in its computational efficiency. For instance, Nanda (2023) proposed an approximation strategy to reduce the computational complexity of obtaining attribution scores. As the time cost is directly pro-

portional to the volume of private data needing protection, enhancing the computational efficiency of neuron localization is essential for handling a larger amount of private data.