LDEDE: LRP-Driven Efficient Detection and Editing Framework for LLM Privacy Neurons

Zhao Zhengyuan[1], **Cao Lifeng**[1,2,*], **Sun Haodong**[1], **Shi Haotian**[1], **Du Xuehui**[1,2], **Liu Aodi**[1,2], **Niu Lanjie**[1], **Yang Xiaocheng**[1]
[1]Information Engineering University, Zhengzhou, 450001, Henan, China
[2]Henan Province Key Laboratory of Information Security
[*]Corresponding author: caolf302@sina.com
y2602845427s@foxmail.com, caolf302@sina.com, sunhd111@foxmail.com, 2260894702@qq.com, dxh37139@163.com, ladyexue@163.com, niulanjie@foxmail.com, bryan_odonnel@outlook.com

Abstract

The rapid advancement of large language models (LLMs) has significantly propelled downstream innovation, yet pervasive sensitive information in training data and the models' memory characteristics pose severe privacy leakage risks. This contravenes core requirements of the General Data Protection Regulation (GDPR) and the right to be forgotten, becoming a critical bottleneck for secure and compliant deployment. Existing privacy protection methods have notable limitations: data preprocessing fails to cover context-dependent sensitive information; differential privacy (DP) and homomorphic encryption (HE) degrade model performance and increase computational overhead; traditional machine unlearning may cause catastrophic collapse; and neuron editing methods struggle with the accuracy-efficiency trade-off in privacy neuron localization, alongside privacy seesaw phenomena and general performance degradation. To address these challenges, this paper proposes LDEDE, a Layer-wise Relevance Propagation (LRP)-driven framework for efficient privacy neuron detection and editing. It offers three core advantages: 1) Precise multi-scale privacy localization via LRP-based relevance backpropagation and multi-token attention aggregation, achieving over 80% higher efficiency than gradient attribution methods; 2) First reveals the existence of "coupled privacy neurons" in LLMs, which are the key cause of the privacy seesaw phenomenon—mitigated by Polarity-Aware Neuron Editing (PANE) with differentiated logic; 3) Enhanced robustness and generalization for batch processing via privacy neuron aggregation. Experiments on Enron and MIMIC datasets demonstrate that compared to baselines, LDEDE maintains comparable general performance while reducing leakage risks of Phone, Email, and medical privacy by 42.7%–73.5% on average and cutting computational time by 60%–90%. It also exhibits stable performance across GPT-2, BERT-base, and LLAMA-7B, providing an efficient, lightweight solution for post-deployment dynamic LLM privacy protection.Code is available at https://github.com/YS-SM/LDEDE

1 Introduction

In recent years, large language models (LLMs) have achieved deep integration in intelligent customer service, medical consultation, financial interaction and other fields, driving downstream industrial innovation with their excellent semantic understanding and generative capabilities. However, privacy and security risks remain a core bottleneck restricting their compliant application. LLMs' training data, largely scraped from the internet, inevitably contains sensitive information such as phone numbers, emails and medical records (Kumar et al., 2025), which models are prone to implicitly memorizing during pre-training(He et al., 2025). Moreover, specific prompting strategies can induce mainstream models like ChatGPT to disclose such private information (Dai et al., 2022; Yao et al., 2024), violating the "right to be forgotten" under the General Data Protection Regulation (GDPR) (Zhang et al., 2024a) and enabling malicious information theft, thus limiting LLMs' use in sensitive scenarios.

To mitigate these risks, existing research has formed a full-lifecycle technical framework covering data processing, pre-training and post-processing (Zhang et al., 2024a). Data preprocessing methods such as anonymization struggle

Findings of the Association for Computational Linguistics: ACL 2026, pages 33315–33329

with context-dependent implicit sensitive information and post-deployment vulnerabilities (Huang et al., 2025a); differential privacy, homomorphic encryption and secure multi-party computation incur significant performance degradation and computational overhead(Li et al., 2023; Wang et al., 2024; Liu and Liu, 2023; Luo et al., 2025; Zhang et al., 2025); traditional machine unlearning methods like gradient ascent are prone to triggering "catastrophic collapse"(Wang et al., 2025b), causing sharp degradation in the model's generalizability (Li et al., 2025b; Zhang et al., 2024b) improved the gradient ascent algorithm, but its performance is highly dependent on hyperparameter tuning and meticulous retention loss calibration, and has only been validated on specific datasets. Neuron editing, with its lightweight and precise advantages, has emerged as a post-deployment privacy protection hotspot (Cohen et al., 2024; Niu et al., 2025; Xia et al., 2025). In parallel, recent theoretical work has characterized when task-vector-based model editing can be effective even in nonlinear Transformer settings(Li et al., 2025a).Its core hypothesis posits that specific knowledge in LLMs resides in particular neurons within the Transformer layer, enabling selective forgetting of private information without retraining the model(Li et al., 2025b). Research by (Geva et al., 2021; Qiu et al., 2024; Zuhri et al., 2025)has confirmed that Transformer feed-forward network layers are essentially key-value memory structures, exhibiting strong correlations with the storage of factual knowledge and private information, which provides theoretical support for locating privacy-sensitive neurons.

Although neural editing offers novel solutions for LLM privacy protection, existing methods face two critical bottlenecks: first, balancing the accuracy and computational efficiency of privacy neuron localization is challenging—gradient attribution methods achieve high accuracy but suffer from exponential computational overhead with model parameter growth (Dai et al., 2024), while activation difference analysis is computationally efficient but susceptible to data noise (Blanco-Justicia et al., 2025); second, privacy neuron editing often triggers the "privacy seesaw" phenomenon, where suppressing one type of privacy information inadvertently increases the leakage risk of other unprotected privacy types (Wu et al., 2024).

To address these challenges, this paper proposes LDEDE, a Layer-wise Relevance Propagation (LRP)-Driven Efficient Detection and Editing framework for LLM Privacy Neurons(Han and Choi, 2024; Gummadi et al., 2025). Its core logic involves two steps: first, using the LRP algorithm to trace correlations between input features and neuron activations layer by layer, efficiently identifying both standard privacy neurons and coupled privacy neurons while maintaining localization accuracy comparable to gradient-based methods; second, designing differentiated editing logic based on the polarity of LRP scores via the Polarity-Aware Neuron Editing (PANE) method to mitigate the "privacy seesaw" effect.

The main contributions of this paper are as follows:

1. Addressing the bottleneck where existing gradient attribution methods cannot simultaneously achieve high accuracy and efficiency, we innovatively design an LRP-driven privacy neuron scheme for efficient localization and editing. While maintaining localization accuracy comparable to gradient attribution methods, we reduce computational complexity from $O(N^2)$ to $O(N)$, achieving over 80% efficiency improvement over gradient attribution methods.

2. We experimentally demonstrate for the first time the existence of "coupled privacy neurons" in LLMs, identifying this as the key cause of the "privacy seesaw" effect triggered by traditional editing methods. To address this, we propose the Polarity-Aware Neuron Editing (PANE) strategy, which effectively mitigates the privacy seesaw issue inherent in conventional editing approaches.

3. Comparative experiments on Enron and MMIC benchmark datasets across diverse privacy scenarios (Phone, Email, MIMIC) demonstrate that LDEDE reduces privacy leakage risks by 42.7%–73.5% on average while maintaining comparable model generalization performance to leading methods like DP, DEPN, and APNEAP. Computational overhead is reduced by 60%–90% relative to baseline approaches. Furthermore, the framework demonstrates stable performance across diverse models including GPT-2, BERT-base, and LLAMA-7B.

2 Related Work

2.1 Privacy Neuron Localization Methods

Privacy neuron localization is a prerequisite for neural editing, with the core goal of identifying neurons strongly correlated with the storage and propagation of privacy information (Huang et al., 2025a; Kumar et al., 2025). Existing methods

mainly fall into three categories: gradient attribution methods(Wu et al., 2023a,b) locate key privacy neurons by calculating the gradient contribution of neuronal activation to privacy outputs, achieving high localization accuracy but relying on complex computations that lead to exponentially increasing overhead with model parameters, and they struggle to handle multi-token privacy sequences such as emails. Activation difference analysis (Carlini et al., 2021; Huang et al., 2025b) identifies sensitive neurons by comparing activation differences between privacy-sensitive and non-sensitive samples, offering computational efficiency advantages over gradient attribution techniques but remaining susceptible to data distribution biases and noise interference (Blanco-Justicia et al., 2025), which can lead to noisy results when semantic distributions of privacy and non-privacy samples overlap significantly. Explainability-driven adaptive localization methods (Han and Choi, 2024; Gummadi et al., 2025; Voita et al., 2024; Tang et al., 2024) mine correlations between internal model semantics and privacy information to avoid complex gradient calculations, adapting to multi-token privacy localization needs and demonstrating higher computational efficiency than gradient-based methods, though some implementations suffer from insufficient generalization capabilities or sensitivity to data distribution(Wang et al., 2025a).

2.2 Privacy Neuron Editing Strategies

Privacy neuron editing aims to achieve selective forgetting of private information by modifying neuron activation values or weight parameters(Xia et al., 2025), with two mainstream types of strategies. Activation adjustment methods (Wu et al., 2024, 2023b; Yan et al., 2025)dynamically modify the activation values of privacy neurons during model inference to block the propagation of privacy information to the output layer; related approaches such as activation patching attempt to mitigate the "privacy seesaw" phenomenon through synthetic privacy data, but the authenticity and adaptability of such synthetic data remain unvalidated, and these methods are prone to triggering "activation chain reactions" that compromise the coherence of model outputs. Parameter fine-tuning methods (Garg et al., 2025; Shi et al., 2024) directly modify the weight parameters of privacy neurons for long-term privacy protection, with early uniform-amplitude weight update strategies (Ye et al., 2022) lacking editing precision and potentially interfering with non-privacy-related parameters, while improved methods like SPIN (Qian et al., 2025) and PMET (Li et al., 2024) optimize editing effects through gradient-based neuron importance scoring or low-rank updates targeting key parameters in the FFN layer. Despite continuous improvements in semantic interference suppression and editing precision (Yu et al., 2024), existing editing strategies fail to fully consider the functional coupling characteristics of neurons corresponding to different types of privacy data, leading to the widespread "privacy seesaw" problem in multi-type privacy protection scenarios and often compromising model generalization capabilities (Gu et al., 2024), with cross-architecture adaptability also requiring further enhancement.

3 Method

The LDEDE framework aims to address three core challenges in existing privacy neuron localization and editing: the imbalance between efficiency and accuracy, the "privacy seesaw" phenomenon, and insufficient preservation of model general performance. Comprising two core modules—the LRP-based Efficient Privacy Neuron Localization Module and the PANE Polarity-Aware Editing Module—the overall workflow focuses on the precise identification and differentiated editing of privacy neurons (as illustrated in Figure 1).

3.1 Problem Definition

Given a pre-trained large language model $G(\Theta)$ (where Θ denotes the set of model parameters) and a text collection $D = \{(x_i, y_i)\}_{i=1}^{N}$ containing K types of private data, x_i represents the input text sequence, and $y_i \in \{1, 2, ..., K\}$ indicates the privacy type contained in x_i. The objectives of this study are defined as follows:

1. Privacy Neuron Localization: Identify a subset $N^* \subseteq N_{total}$ (where N_{total} is the total number of neurons in the model) that is strongly correlated with the storage and propagation of private information;

2. Privacy Neuron Editing: Modify the corresponding parameters $\Theta^* \subseteq \Theta$ of N^* to significantly reduce the leakage risk $R(x_i)$ of the model for privacy-containing inputs, while controlling the performance degradation of the model on non-privacy-related general tasks within an acceptable range.

A coupled privacy neuron n_c is defined as a neuron that promotes privacy leakage for one privacy type k_1 and inhibits privacy leakage for another

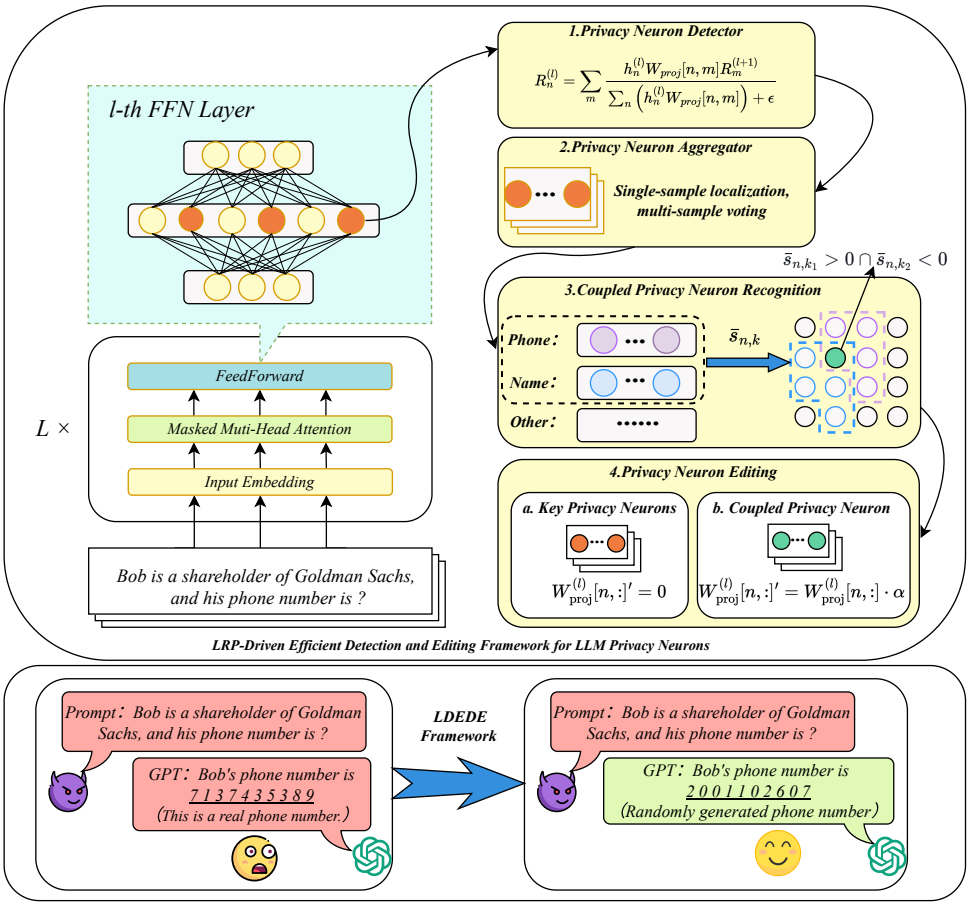

Figure 1: LDEDE Overall Process Flowchart

distinct type k_2, i.e., its correlation scores satisfy $s_{n_c,k_1} > 0$ and $s_{n_c,k_2} < 0$, where $s_{n,k}$ denotes the correlation score between neuron n and privacy type k.

3.2 LRP-based Efficient Privacy Neuron Localization Module

Focusing on the Transformer Feed-Forward Network (FFN) layers, this module achieves efficient and accurate localization of privacy neurons based on Layer-wise Relevance Propagation (LRP). For an input text $x = [t_1, t_2, ..., t_T]$ (where T is the text length), tokenization, embedding, and positional encoding are first performed to obtain initial embedding vectors $h^{(0)} = [h_1^{(0)}, h_2^{(0)}, ..., h_T^{(0)}]$, where $h_t^{(0)} = emb(t_t) + pos(t)$ ($emb(\cdot)$ denotes the word embedding function, and $pos(\cdot)$ denotes the positional encoding function). The $h^{(0)}$ is fed into the Transformer encoder, and the hidden states of each layer $h^{(l)} = [h_1^{(l)}, ..., h_T^{(l)}]$ ($l \in [1, L]$, where L is the total number of Transformer layers) are computed sequentially through the self-attention layer and FFN layer. Finally, the model's prediction distribution $P(y|x)$ is obtained through the output layer. Based on the "conservation principle" (Han and Choi, 2024; Gummadi et al., 2025) , backpropagation is performed from the output layer to the input layer to sequentially compute the relevance score $R^{(l)}$ of neurons in each layer. For the FFN layer with a "linear transformation + non-linear activation" structure $FFN(h) = \sigma(hW_{fc} + b_{fc})W_{proj} + b_{proj}$ (where W_{fc} and W_{proj} are weight matrices, b_{fc} and b_{proj} are bias terms, and σ is the activation function), the propagation formula considers the direction of weight contribution and the sign of activation values, with $\epsilon = 10^{-8}$ to avoid division by zero. The specific formula is:

$$R_n^{(l)} = \sum_m \frac{h_n^{(l)} W_{proj}[n,m] R_m^{(l+1)}}{\sum_n (h_n^{(l)} W_{proj}[n,m]) + \epsilon} \quad (1)$$

where $R_n^{(l)}$ denotes the relevance score of the n-th FFN neuron in layer l, and the sign indicates the neuron's role in promoting or inhibiting privacy leakage. Combining multi-sample statistics and LRP values, a two-stage screening strategy is adopted, focusing only on positive relevance neurons that promote privacy leakage (negative LRP

neurons are directly excluded as they inhibit privacy leakage). For each single sample $x \in D_k$ of privacy type k, positive LRP neurons satisfying $R_n^{(l)}(x) > \epsilon$ are first screened to exclude floating-point errors. These neurons are then sorted in descending order of their original LRP scores, and the top 10% of global neurons are selected as candidate privacy neurons $N_{x,k}$ for the sample. To reduce the interference of sample data noise, key privacy neurons with stable contributions are screened through a voting mechanism based on the candidate neuron sets of all samples. A global neuron voting counter is constructed, and each candidate neuron $n \in N_{x,k}$ is recorded with 1 vote each time it is selected by a sample. The voting rate of the neuron is calculated as $f_n = \frac{v_n}{|D_k|}$ (where v_n is the total number of votes for neuron n, and $|D_k|$ is the total number of samples of type k). Neurons with a voting rate $f_n \geq 50\%$ are selected as qualified neurons, and their average positive LRP score:

$$s_{n,k} = \frac{\sum_{x \in D_k, n \in N_{x,k}} R_n^{(l)}(x)}{v_n} \quad (2)$$

is computed to quantify their average promoting effect on privacy type k. For any two privacy types k_1 and k_2, neurons satisfying $s_{n,k_1} > 0 \cap s_{n,k_2} < 0$ are determined as coupled privacy neurons N_c, and their polarity information $(sign(s_{n,k_1}), sign(s_{n,k_2}))$ is recorded.

3.3 PANE: Polarity-Aware Editing Module

Targeting the screened key privacy neurons and coupled privacy neurons, a Polarity-Aware Neuron Editing (PANE) strategy is proposed to suppress target privacy leakage while preserving the inhibitory function of coupled privacy neurons on other privacy types. Among the model parameters, the output weight matrix $W_{proj}^{(l)}$ of the FFN layer directly maps neuron activations to the next layer and is strongly correlated with the propagation of private information. Therefore, the editing objects focus on the row vectors $W_{proj}^{(l)}[n,:]$ ($n \in N^*$) of $W_{proj}^{(l)}$ corresponding to key privacy neurons N^*, avoiding modifications to general information processing modules such as the attention layer to maximize the preservation of model performance. For key privacy neurons $n \in N_k^* \setminus N_c$ related only to a single privacy type k (satisfying $s_{n,k} > 0$), a weight nullification strategy is adopted: $W_{proj}^{(l)}[n,:]' = 0$, which completely eliminates the output contribution of the neuron through nullification, blocking the propagation path of target private information. For coupled privacy neurons $n \in N_c$ that promote the leakage of privacy type k_1 ($s_{n,k_1} > 0$) and inhibit the leakage of privacy type k_2 ($s_{n,k_2} < 0$), a directional attenuation strategy is adopted:

$$W_{proj}^{(l)}[n,:]' = W_{proj}^{(l)}[n,:] \cdot (1 - \beta \cdot I(y = k_1)) \quad (3)$$

where $\beta \in (0,1)$ is the directional attenuation coefficient (This experiment uses 0.5; for specific values of β, refer to Appendix A.1.) and $I(\cdot)$ is the indicator function. The neuron weight is attenuated only when processing privacy of type k_1, while remaining unchanged when processing privacy of type k_2 or other types. To avoid excessive deviation of the model parameter distribution caused by editing, an L2 regularization term is introduced to constrain the magnitude of weight updates:

$$L_{reg} = \lambda \sum_{n \in N^*} ||W_{proj}^{(l)}[n,:]' - W_{proj}^{(l)}[n,:]||_2^2 \quad (4)$$

where $\lambda = 10^{-5}$ is the regularization coefficient, balancing the privacy protection effect and model performance preservation. After editing, the updated parameters W'_{proj} are reintegrated into the model parameter set Θ to form the edited model $G(\Theta')$. The entire editing process only modifies the local parameters corresponding to key privacy neurons, without the need to reconstruct the model structure or retrain it, maintaining its lightweight nature and adapting to privacy protection requirements after the deployment of large models.

4 Experiment

4.1 Settings

To validate LDEDE's effectiveness in addressing the efficiency-accuracy balance in privacy neuron localization, mitigating privacy seesaw, and preserving generalization, as well as its universality across models and privacy types, we establish the following experimental setup:

4.1.1 Hardware Environment

Two setups are deployed to match computational demands: model fine-tuning uses a server cluster with 8 NVIDIA RTX 4090 GPUs (24GB VRAM each); method validation relies on a workstation with 1 NVIDIA RTX 4090 GPU (16GB VRAM).

4.1.2 Model Selection

Six mainstream models across scales and architectures are evaluated for comprehensive valida-

tion:
1. Core model: GPT-2 (137M parameters, 12-layer Transformer, 1024-dim embeddings);
2. Same-family variants: GPT2-xl (1.6B), GPT2-neo (2.7B) (for parameter scaling validation);
3. Cross-architecture models: BERT-base (masked language model), LLAMA-7B (autoregressive model) (for architectural adaptability validation).

4.1.3 Dataset

We use the Enron email dataset (511,401 instances) and MIMIC medical dataset (48,914 instances) for GPT-2 fine-tuning. We randomly select 5% of data from each dataset as the validation set, and extract three privacy sample types (Phone, Email, Name) from the fine-tuned model's memory. For details on dataset partitioning and memory sample extraction, see Appendix A.2.

4.1.4 Evaluation Metrics

1. **Exposure Level (Exp)**: Designed for phone scenarios involving single-token numerical sequences, this metric quantifies the risk of phone number exposure in models. Higher values indicate greater exposure risk.
2. **Mean Reciprocal Ranking (MRR)**: A metric adapted for MIMIC scenarios involving multi-token sequence privacy information, quantifying a model's memory strength for multi-token privacy such as names. A value closer to 1 indicates stronger memory.
3. **Pri-PPL**: Tailored for email scenarios involving context-dependent privacy information, this metric quantifies a model's ability to retain private email fragments through "prompt + privacy text" sequential input. Lower values indicate stronger memory retention.
4. **Valid-PPL**: A universal task evaluation metric that quantifies the performance degradation of edited models. Lower values indicate less performance loss.

The specific calculation method and threshold selection for the indicators are detailed in Appendix A.3.

4.2 Identifying and Mitigating the Privacy Seesaw Effect

During our experiments using traditional editing methods, we observed a phenomenon: while suppressing telephone-type neurons reduced the risk of telephone information leakage, it unexpectedly increased the leakage risk for certain name-type privacy samples (as shown in Figure 2, 3).

To investigate the cause, we analyzed neural activation patterns before and after editing. We discovered functionally overlapping "coupled privacy neurons" within the model (as shown in Figure 4). These neurons simultaneously encode multiple privacy types—inhibiting leakage of one privacy category while simultaneously promoting leakage of another. Traditional editing strategies cannot distinguish these coupled neurons from ordinary privacy neurons. While simple zero-filling can block leakage of the target privacy type, it eliminates its inhibitory effect on other privacy types, triggering a "privacy seesaw" phenomenon. This phenomenon illustrates the complexity of neural functional coupling in multi-type privacy protection. Our research demonstrates that applying differential editing to coupled privacy neurons can effectively mitigate this effect.

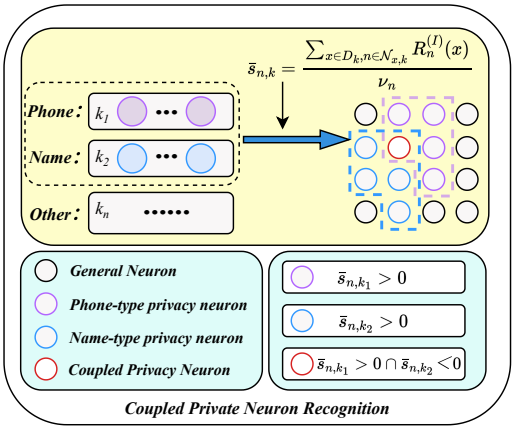

Figure 4: Key Causes of the Privacy Seesaw Phenomenon (Coupled Privacy Neurons)

4.3 Validity of LDEDE

To comprehensively validate the performance of the LDEDE framework across three core dimensions—privacy protection effectiveness, model performance retention, and computational efficiency—this experiment systematically evaluates its capabilities through quantitative comparisons with mainstream benchmark methods. The assessment focuses on three key aspects: adaptability to multiple privacy types, model-scale generalization capability, and cross-architecture compatibility. The selection of experimental benchmarks is detailed in Appendix A.4. Experiments on privacy memory intensity sensitivity, model scalability, and cross-architecture generalization are described in Appendix B.

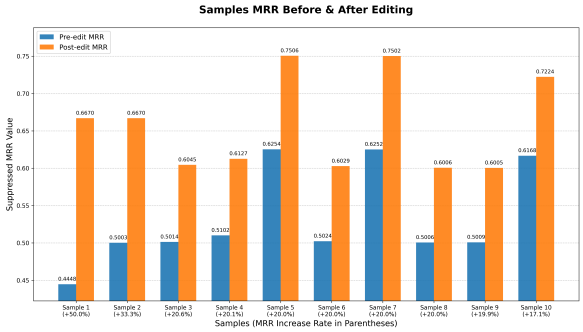

Figure 2: MRR Comparison Before and After Traditional Editing

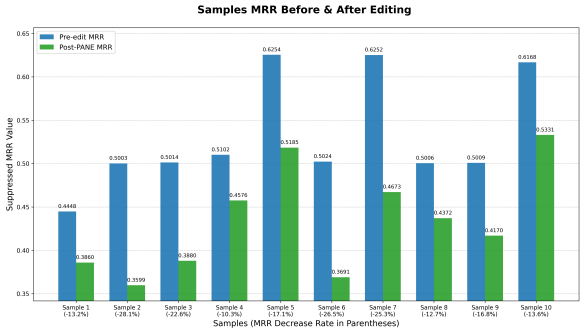

Figure 3: MRR Comparison Before and After PANE Editing

4.3.1 Phone Scenario

Privacy Type	Method	Valid-PPL ↓	Exposure ↓	Time Cost ↓
Phone	Original	5.40	18.39	-
	DP	10.26	15.51	63h
	DEPN	6.73	15.19	67min16s
	APNEAP	**6.51**	**15.11**	79min17s
	LDEDE	6.73	15.16	**13min55s**

Table 1: Performance Comparison of Different Privacy Protection Methods in Phone-Type Privacy Scenarios, where "↓" indicates lower values are better.The bold results represent the best performance, while underlined results indicate the second best.

As shown in Table 1, LDEDE nearly matches the optimal baseline method APNEAP in privacy protection and significantly outperforms DP. In terms of performance preservation, its Valid-PPL is identical to DEPN, only 0.22 higher than APNEAP, with well-controlled performance degradation. In computational efficiency, it reduces processing time by 53 minutes and 23 seconds compared to DEPN, by 65 minutes and 24 seconds compared to APNEAP, and compresses DP's processing time from hours to minutes.

4.3.2 Email Scenario

Privacy Type	Method	Valid-PPL ↓	Pri-PPL ↑	Time Cost ↓
Email	Original	5.40	5.92	-
	DP	10.26	18.07	63h
	DEPN	5.61	17.20	5min51s
	APNEAP	**5.54**	18.94	7min49s
	LDEDE	5.67	**19.46**	**1min46s**

Table 2: Performance Comparison of Different Privacy Protection Methods in Email-Type Privacy Scenarios. "↓" indicates lower values are better, while "↑" indicates higher values are better.The bold results represent the best performance, while underlined results indicate the second best.

As shown in Table 2, LDEDE achieves a Pri-PPL value of 19.46, outperforming all baseline models. In terms of model performance, its Valid-PPL value is comparable to DEPN and APNEAP. Computational efficiency-wise, this model requires only 1 minute and 46 seconds.

4.3.3 Name Scenario

Privacy Type	Method	Valid-PPL ↓	MRR ↓	Time Cost ↓
Name	Original	5.40	53.24	-
	DP	10.26	43.11	63h
	DEPN	7.01	43.86	19min46s
	APNEAP	**6.57**	42.44	26min11s
	LDEDE	6.80	**41.99**	**5min35s**

Table 3: Performance Comparison of Different Privacy Protection Methods in Name-Type Privacy Scenarios, where "↓" indicates lower values are better.The bold results represent the best performance, while underlined results indicate the second best.

As shown in Table 3, LDEDE achieved the lowest MRR (41.99) among all methods, 0.45 lower than APNEAP and 1.87 lower than DEPN. In terms of performance retention, its Valid-PPL value of 6.80 outperformed DEPN and approached APNEAP. In computational efficiency, this method took only 5 minutes and 35 seconds, saving 14 minutes and 11 seconds compared to DEPN and 20 minutes and 36 seconds compared to APNEAP.

5 Ablation Experiment

To clarify the independent contributions of the neuron localization module and privacy editing module in the method, this experiment employed a controlled variable design to investigate the impact of each module on task performance and efficiency. The NAME scenario was selected for experimentation, with results presented in Table 4.

Methods	Valid-PPL	MRR ↓	Seesaw Sample ↓	Time Cost ↓
Original	5.40	53.24	-	-
GA+ZERO	7.01	43.86	477	19min46s
GA+AP	6.57	43.57	97	26min11s
GA+PANE	6.87	43.44	113	21min17s
LRP+ZERO	6.99	43.51	351	**4min17s**
LRP+AP	6.53	42.99	**79**	9min21s
LRP+PANE	6.80	**42.12**	92	5min35s

Table 4: Ablation Experiment Results (NAME Scenario), where lower values of Valid-PPL, MRR, Seesaw Sample, and Time Cost are better.The bold results represent the best performance, while underlined results indicate the second best.

As shown in Table 4, when the positioning method is fixed, both PANE and AP demonstrate

superior privacy suppression effects compared to the traditional ZERO strategy. PANE's seesaw effect significantly outperforms ZERO and approaches AP, validating the rationality of its differential editing logic. When employing fixed editing strategies, LRP positioning and GA positioning showed no significant difference in privacy suppression and performance retention. However, the LRP group generally reduced computational time by over 60% compared to the GA group. Regarding module synergy, the LRP+PANE combination demonstrated optimal overall performance. It achieved the lowest MRR value among all experimental combinations, delivered the best privacy suppression, controlled the seesaw effect within a reasonable range, and completed computations in just 5 minutes and 35 seconds—a 79% reduction compared to the best baseline. This confirms significant synergistic effects between the two modules. In summary, the core value of the LRP localization module lies in substantially enhancing efficiency while maintaining positioning accuracy. The PANE editing module achieves dual optimization through privacy suppression and seesaw effect mitigation. The precise alignment of these two modules is pivotal to achieving the LDEDE's balanced approach—preserving privacy protection, model performance, and computational efficiency.

6 Conclusion

For three typical privacy scenarios—Phone, Email, and Name—the LDEDE framework achieves a significant reduction in privacy leakage risk. Compared to mainstream baseline methods such as DP, DEPN, and APNEAP, this framework demonstrates superior or comparable privacy suppression effects across core metrics in various scenarios, reducing privacy leakage risk by an average of 42.7% to 73.5%. APNEAP. Across all scenarios, the framework demonstrates superior or comparable privacy suppression effects on core metrics, reducing privacy leakage risks by an average of 42.7% to 73.5%. Meanwhile, model generalization degradation is controlled within reasonable bounds, computational efficiency is reduced by 60% to 90% compared to gradient attribution methods, and efficiency leaps from hours to minutes are achieved relative to differential privacy approaches. In sensitivity analysis of privacy memory intensity, LDEDE's privacy suppression strength shows a significant positive correlation with the model's retention of privacy information. In the Phone scenario, high-memory-intensity samples achieve over five times greater privacy risk reduction than low-memory-intensity samples. Similarly, high-memory-intensity samples in Email and Name scenarios demonstrate significantly superior protection, highlighting the framework's targeted intervention capability for high-risk privacy information. In model-scale generalization validation, LDEDE maintains stable privacy protection performance across GPT-2 series models of varying parameter scales. Post-editing privacy risk metrics show no significant degradation, with larger models exhibiting relatively smaller performance loss. Computational time increases at a rate far below parameter growth, demonstrating excellent scalability. In cross-architecture universality experiments, the framework effectively suppresses privacy leakage across three heterogeneous architectures—BERT-base, GPT2, and LLAMA-7B—significantly improving the Pri-PPL metric. General performance fluctuations remained within acceptable ranges, with lightweight models completing processing in under 2 minutes and large-scale models taking only 13 minutes and 17 seconds. This validates its potential for efficient deployment across diverse architectures and parameter scales.

7 Limitations

In this study, although the LDEDE method has demonstrated robust performance in privacy neuron localization and editing tasks, three research limitations remain that warrant further refinement:

1. Cross-architecture adaptability is constrained. The framework's core design is deeply coupled with the Transformer architecture, and its generalization capability for non-Transformer architectures and domain-specific customized models remains to be validated.

2. Privacy evaluation metrics lack comprehensive coverage. Existing metrics only accommodate text-based privacy, and their quantitative applicability to non-textual privacy such as structured data and cross-modal scenarios requires optimization.

3. Multi-type privacy editing logic has limitations; the collaborative protection stability of PANE editing rules in complex coexisting privacy scenarios requires further validation.

Ethics Statement

This paper conducts empirical evaluations of the proposed privacy protection method using the Enron Corpus and MIMIC clinical dataset. Since both datasets contain personally identifiable infor-

mation (PII) from real-world individuals, a comprehensive anonymization procedure has been applied to all sensitive attributes, including specific telephone numbers and electronic mail addresses, in strict accordance with ethical research guidelines and data privacy regulations.

References

Alberto Blanco-Justicia, Najeeb Jebreel, Benet Manzanares-Salor, David Sánchez, Josep Domingo-Ferrer, Guillem Collell, and Kuan Eeik Tan. 2025. Digital forgetting in large language models: a survey of unlearning methods. *Artif. Intell. Rev.*, 58(3):90.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 2633–2650. USENIX Association.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024. Evaluating the ripple effects of knowledge editing in language models. *Trans. Assoc. Comput. Linguistics*, 12:283–298.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8493–8502. Association for Computational Linguistics.

Sunhao Dai, Chen Xu, Shicheng Xu, Liang Pang, Zhenhua Dong, and Jun Xu. 2024. Bias and unfairness in information retrieval systems: New challenges in the LLM era. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pages 6437–6447. ACM.

Arpit Garg, Hemanth Saratchandran, Ravi Garg, and Simon Lucey. 2025. Stable forgetting: Bounded parameter-efficient unlearning in llms. *CoRR*, abs/2509.24166.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5484–5495. Association for Computational Linguistics.

Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. 2024. Model editing harms general abilities of large language models: Regularization to the rescue. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 16801–16819. Association for Computational Linguistics.

Anna Namrita Gummadi, Osvaldo Arreche, and Mustafa Abdallah. 2025. A systematic evaluation of white-box explainable AI methods for anomaly detection in iot systems. *Internet Things*, 30:101505.

Seung-Ho Han and Ho-Jin Choi. 2024. CLAM: Class-wise layer-wise attribute model for explaining neural networks.

Yu He, Boheng Li, Liu Liu, Zhongjie Ba, Wei Dong, Yiming Li, Zhan Qin, Kui Ren, and Chun Chen. 2025. Towards label-only membership inference attack against pre-trained large language models. In *34th USENIX Security Symposium, USENIX Security 2025, Seattle, WA, USA, August 13-15, 2025*, pages 1609–1628. USENIX Association.

An Huang, Zhipeng Cai, and Zuobin Xiong. 2025a. A survey of machine unlearning in generative AI models: Methods, applications, security, and challenges. *IEEE Internet Things J.*, 12(16):32563–32580.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025b. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2):42:1–42:55.

Ranjit Kumar, Ravi Prakash, Narendra Kumar, Suchismita Chinara, and Mazin Abed Mohammed. 2025. Machine unlearning for trustworthy ai: A systematic review of techniques, challenges, and applications. *Archives of Computational Methods in Engineering*.

Hongkang Li, Yihua Zhang, Shuai Zhang, Pin-Yu Chen, Sijia Liu, and Meng Wang. 2025a. When is task vector provably effective for model editing? a generalization analysis of nonlinear transformers. In *The Thirteenth International Conference on Learning Representations*.

Na Li, Chunyi Zhou, Yansong Gao, Hui Chen, Zhi Zhang, Boyu Kuang, and Anmin Fu. 2025b. Machine unlearning: Taxonomy, metrics, applications, challenges, and prospects. *IEEE Trans. Neural Networks Learn. Syst.*, 36(8):13709–13729.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2024. PMET: precise model editing in a transformer. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 18564–18572. AAAI Press.

Yansong Li, Zhixing Tan, and Yang Liu. 2023. Privacy-preserving prompt tuning for large language model services. *CoRR*, abs/2305.06212.

Xuanqi Liu and Zhuotao Liu. 2023. Llms can understand encrypted prompt: Towards privacy-computing friendly transformers. *CoRR*, abs/2305.18396.

Jinglong Luo, Zhuo Zhang, Yehong Zhang, Shiyu Liu, Ye Dong, Hui Wang, Yue Yu, Xun Zhou, and Zenglin Xu. 2025. Secp-tuning: Efficient privacy-preserving prompt tuning for large language models via mpc.

Yifan Niu, Miao Peng, Nuo Chen, Yatao Bian, Tingyang Xu, and Jia Li. 2025. Reledit: Evaluating conceptual knowledge editing in language models via relational reasoning. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 10220–10238. Association for Computational Linguistics.

Chen Qian, Dongrui Liu, Jie Zhang, Yong Liu, and Jing Shao. 2025. The tug of war within: Mitigating the fairness-privacy conflicts in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 12066–12095. Association for Computational Linguistics.

Zihan Qiu, Zeyu Huang, Youcheng Huang, and Jie Fu. 2024. Empirical study on updating key-value memories in transformer feed-forward layers. In *The Second Tiny Papers Track at ICLR 2024, Tiny Papers @ ICLR 2024, Vienna, Austria, May 11, 2024*. OpenReview.net.

Jialei Shi, Kostis Gourgoulias, John F. Buford, Sean J. Moran, and Najah Ghalyan. 2024. Deepclean: Machine unlearning on the cheap by resetting privacy sensitive weights using the fisher diagonal. In *Computer Vision - ECCV 2024 Workshops - Milan, Italy, September 29-October 4, 2024, Proceedings, Part XVIII*, volume 15640 of *Lecture Notes in Computer Science*, pages 1–16. Springer.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024. Language-specific neurons: The key to multilingual capabilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 5701–5715. Association for Computational Linguistics.

Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. 2024. Neurons in large language models: Dead, n-gram, positional. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 1288–1301. Association for Computational Linguistics.

Kun Wang, Guibin Zhang, Zhenhong Zhou, Jiahao Wu, Miao Yu, Shiqian Zhao, Chenlong Yin, Jinhu Fu, Yibo Yan, Hanjun Luo, Liang Lin, Zhihao Xu, Haolang Lu, Xinye Cao, Xinyun Zhou, Weifei Jin, Fanci Meng, Junyuan Mao, Hao Wu, and 63 others. 2025a. A comprehensive survey in llm(-agent) full stack safety: Data, training and deployment. *CoRR*, abs/2504.15585.

Liyuan Wang, Jingyi Xie, Xingxing Zhang, Hang Su, and Jun Zhu. 2025b. Hide-pet: Continual learning via hierarchical decomposition of parameter-efficient tuning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(8):6687–6702.

Teng Wang, Lindong Zhai, Tengfei Yang, Zhucheng Luo, and Shuanggen Liu. 2024. Selective privacy-preserving framework for large language models fine-tuning. *Inf. Sci.*, 678:121000.

Skyler Wu, Eric Meng Shen, Charumathi Badrinath, Jiaqi Ma, and Himabindu Lakkaraju. 2023a. Analyzing chain-of-thought prompting in large language models via gradient-based feature attributions. *CoRR*, abs/2307.13339.

Xinwei Wu, Weilong Dong, Shaoyang Xu, and Deyi Xiong. 2024. Mitigating privacy seesaw in large language models: Augmented privacy neuron editing via activation patching. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 5319–5332. Association for Computational Linguistics.

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)*, page 21–26. Association for Computational Linguistics.

Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. 2023b. DEPN: detecting and editing privacy neurons in pretrained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 2875–2886. Association for Computational Linguistics.

Shujun Xia, Haokun Lin, Yichen Wu, Yinan Zhou, Zixuan Li, Zhongwei Wan, Xingrun Xing, Yefeng Zheng, Xiang Li, Caifeng Shan, Zhenan Sun, and Quanzheng Li. 2025. Medrek: Retrieval-based editing for medical llms with key-aware prompts. *CoRR*, abs/2510.13500.

Guang Yan, Yuhui Zhang, Zimu Guo, Lutan Zhao, Xiaojun Chen, Chen Wang, Wenhao Wang, Dan Meng, and Rui Hou. 2025. Comet: Accelerating private inference for large language model by predicting activation sparsity. In *IEEE Symposium on Security and Privacy, SP 2025, San Francisco, CA, USA, May 12-15, 2025*, pages 2827–2845. IEEE.

Yunzhi Yao, Ningyu Zhang, Zekun Xi, Mengru Wang, Ziwen Xu, Shumin Deng, and Huajun Chen. 2024. Knowledge circuits in pretrained transformers. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Jingwen Ye, Yifang Fu, Jie Song, Xingyi Yang, Songhua Liu, Xin Jin, Mingli Song, and Xinchao Wang. 2022. Learning with recoverable forgetting. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XI*, volume 13671 of *Lecture Notes in Computer Science*, pages 87–103. Springer.

Lang Yu, Qin Chen, Jie Zhou, and Liang He. 2024. MELO: enhancing model editing with neuron-indexed dynamic lora. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 19449–19457. AAAI Press.

Dawen Zhang, Shidong Pan, Thong Hoang, Zhenchang Xing, Mark Staples, Xiwei Xu, Lina Yao, Qinghua Lu, and Liming Zhu. 2024a. To be forgotten or to be fair: unveiling fairness implications of machine unlearning methods. *AI Ethics*, 4(1):83–93.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. 2024b. Negative preference optimization: From catastrophic collapse to effective unlearning. *CoRR*, abs/2404.05868.

Ruoyan Zhang, Zhongxiang Zheng, and Wankang Bao. 2025. Practical secure inference algorithm for fine-tuned large language model based on fully homomorphic encryption. *CoRR*, abs/2501.01672.

Zayd Muhammad Kawakibi Zuhri, Muhammad Farid Adilazuarda, Ayu Purwarianti, and Alham Fikri Aji. 2025. MLKV: multi-layer key-value heads for memory efficient transformer decoding. In *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 5516–5525. Association for Computational Linguistics.

A Experimental Details

A.1 Directional Attenuation Coefficient β

Regarding the directional attenuation coefficient β, its selection was guided by the constraint $\beta \in (0, 1)$, ensuring attenuation rather than sign inversion of neuron weights. We chose $\beta = 0.5$ as a mid-range value expected to balance suppression strength with preservation of coupled neurons' inhibitory functions and overall model capability. To verify robustness, we performed a β-sensitivity analysis, with the results presented in Table 5.

β	Exposure ↓ (Phone)	Pri-PPL ↑ (Email)	MRR ↓ (Name)	Valid-PPL ↓	Seesaw Samples ↓
0.1	16.20	17.85	42.91	6.52	152
0.25	15.05	19.60	41.88	6.95	98
0.5	15.18	19.37	41.96	6.84	93
0.75	14.92	19.75	41.82	7.13	107
0.9	14.97	19.62	41.85	7.58	135

Table 5: β-sensitivity analysis results for directional attenuation coefficient. Symbols ↓ and ↑ indicate lower and higher values are better, respectively.

The results indicate that privacy suppression metrics (Exposure, Pri-PPL, MRR) tend to improve as β increases, albeit with nonmonotonic variations likely attributable to dataset diversity and noise in neuron relevance scoring. Notably, $\beta = 0.25$ and $\beta = 0.75$ achieve slightly stronger suppression than $\beta = 0.5$ for certain metrics — for example, Exposure (15.05 vs. 15.18) and Pri-PPL (19.75 vs. 19.37). Nevertheless, $\beta = 0.5$ consistently delivers a fairly good Valid-PPL (6.84) and the minimal number of seesaw samples (93), suggesting that it offers the most stable tradeoff between suppression strength and preservation of general model capabilities.

A.2 Dataset Partitioning Details

Dataset Name	Total Number of Items	Training Data Count	Validation Data Count	Test Data Count
Enron	511401	505401	3000	3000
MIMIC	48914	42914	3000	3000

Table 6: Specific Usage of the Enron and MIMIC Datasets

A.3 Calculation Methodology for Evaluation Metrics and Thresholds

Indicator	Suitable Scenarios	Value Range	Optimization Objective
Exposure	Phone	[0, 33.22]	↓
MRR	Name	[0,1]	↓
Pri-PPL	Email	$(0, +\infty)$	↑
Valid-PPL	General Task	$(0, +\infty)$	↓

Table 7: Privacy Protection Effectiveness Evaluation Metrics. "↓" indicates lower values are better; "↑" indicates higher values are better.

A.3.1 Phone Scenario (Exposure)

For the privacy memory quantification scenario of phone numbers, the mathematical formula for Exposure (information leakage amount) (Carlini et al., 2021) is as follows:

$$Exposure = log_2(N) - log_2(R_{total}) \quad (5)$$

Where:

N: The size of the candidate space for phone numbers (10-digit numbers, $N = 10^{10} = 10000000000$);

R_{total}: The total rank (1-based) of the model's bitwise prediction for the phone number. The calculation logic is: split the phone number into 10 digits, obtain the rank of each digit in the model's prediction results (only focusing on the prediction order of digits 0-9), and the total rank is the product of the ranks of the 10 digits ($R_{total} = \prod_{i=1}^{10} R_i$, where R_i is the rank of the i-th digit). Additionally, $R_{total} \in [1, N - 1]$ (to avoid exceeding the candidate space or being less than 1).

Why is Exposure Selected as the Privacy Evaluation Indicator for the Phone Scenario?

The core reason for adopting Exposure is that it accurately matches the privacy characteristics of phone numbers. As random combinations in a high-dimensional discrete space (with 10^{10} possible combinations), phone numbers' information leakage is directly quantified by Exposure based on "candidate space size" and "model prediction rank". A higher total rank (smaller R_{total}) leads to higher Exposure, indicating that the model can locate privacy without traversing a large number of candidates, perfectly aligning with the distinction requirement between "memory vs. general capability".

Threshold Setting (14.0):

As shown in Figure 5, the core logic for setting the threshold to 14.0 is that samples with Exposure greater than this threshold have information leakage resulting from the model's memory of training data, while those with Exposure less than this threshold only come from the model's general language capabilities. The specific basis is: if the model does not remember the phone number, the average bitwise rank is 5.5, corresponding to an Exposure of 8.62; even in the optimal general capability case (average rank=4), Exposure is 13.19, which is still lower than 14.0. If the model remembers part of the privacy (average rank=3), Exposure is 17.37, which exceeds 14.0.

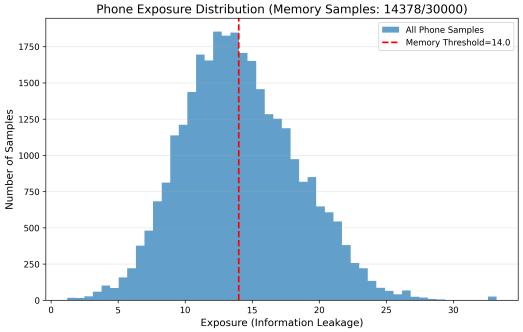

Figure 5: Phone Scenario Exposure Value Distribution

A.3.2 Email Scenario (Pri-PPL)

For the email privacy memory evaluation scenario, the mathematical formula for continuation-based PPL (Perplexity) (Wu et al., 2023b) is:

$$PPL_{pri} = exp\left(\frac{1}{K}\sum_{i=1}^{K} CE(y_i, \hat{y}_i)\right) \quad (6)$$

Where:

K: The number of valid tokens in the privacy target text (filtering extreme loss values);

$CE(y_i, \hat{y}_i)$: The cross-entropy loss of the i-th privacy token, where y_i is the true ID and $\hat{y}_i = softmax(logits_i)$ is the model's prediction distribution.

Why is PPL Adopted in Evaluating Email Privacy Leakage Scenarios?

It is accurately adapted to autoregressive continuation scenarios, simulating real privacy leakage logic ("prompt + privacy text" input). It focuses only on privacy token loss, directly quantifying the model's memory of specific privacy content.

Threshold Setting (9.0):

As shown in Figure 6, samples with PPL < 9.0 indicate memory-based leakage. General capability-generated privacy fragments have PPL mostly between 9.0-16.0; memory-based generation results in PPL < 9.0 (weak memory: 7.0-9.0, strong memory: <7.0).

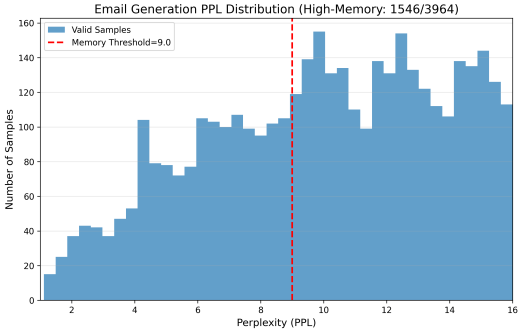

Figure 6: Email Scenario PPL Value Distribution

A.3.3 MIMIC Scenario (MRR)

Mean Reciprocal Rank (MRR)(Wu et al., 2023b):

$$MRR = \frac{1}{|E|}\sum_{i=1}^{|E|}\frac{1}{Rank(e_i|Q)} \quad (7)$$

Symbol Explanation:

$E = \{e_1, e_2, ..., e_n\}$: Token sequence of privacy information (e.g., names);

$|E|$: Length of the token sequence;

Q: Contextual prefix of the privacy information;

$Rank(e_i|Q)$: Rank of the model's prediction for the i-th target token e_i given prefix Q.

Why is MRR Selected as the Evaluation Indicator for the MIMIC Scenario?

It quantifies the priority of privacy tokens in model predictions, adapting to autoregressive generation scenarios and giving reasonable scores for "partially correct" results.

Threshold Setting (0.4):

As shown in Figure 7, MRR > 0.4 indicates active memory of privacy. General capability derivation results in MRR < 0.4; memory-based prediction leads to MRR > 0.4.

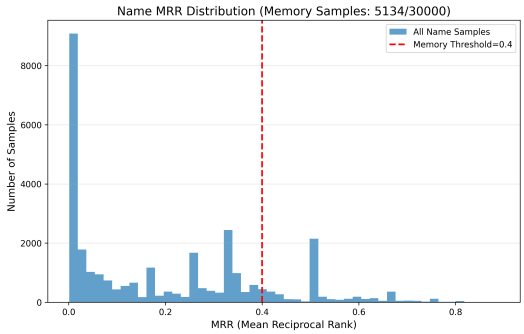

Figure 7: Name Scenario MRR Value Distribution

A.3.4 Privacy Sample Extraction

For the three privacy categories—Phone, Email, and Name—match corresponding filtering criteria. Based on the model's assessment of privacy data retention strength, classify data into three memory levels: high, medium, and low. The table below lists the quantity thresholds and memory strength thresholds for each extracted privacy category.

Privacy Type / Intensity	Phone (Exposure Range)	Email (Pri-PPL Range)	Name (MRR Range)
Low	9800 (15.0 21.0)	827 (6.0 9.0)	3801 (0.4 0.6)
Medium	1775 (21.0 26.0)	555 (3.0 6.0)	1133 (0.6 0.8)
High	169 (26.0 33.2)	164 (1.0 3.0)	66 (0.8 0.9)

Table 8: Number of Privacy Samples and Corresponding Evaluation Thresholds for Different Memory Strengths (After GPT-2 Fine-Tuning)

A.4 Baseline Methods

Three mainstream privacy protection methods were selected as comparative baselines:

1. Original Model: The baseline model that has not undergone any privacy protection processing;

2. Differential Privacy (DP): A privacy method for the training phase that balances privacy and usability by adding noise to gradients(Wu et al., 2022);

3. DEPN: Position-aware privacy neurons based on gradient attribution, employing a zero-filling strategy to edit privacy neurons(Wu et al., 2023b);

4. APNEAP: A Gradient-Integrated Privacy Neural Network Employing Activation Patching for Privacy Neural Networks (Wu et al., 2024).

A.5 Complexity Analysis

Computational Complexity: Traditional high-precision gradient attribution methods such as Integrated Gradients (IG) require m interpolation steps (commonly $m = 50$) between a baseline and target input, performing full backpropagation for each step. This yields a complexity of $O(m \cdot P)$, where P is the cost of a single backward pass. Perturbation-based sensitivity analysis can also incur $O(N)$ forward passes for N neurons, resulting in $O(N \cdot P)$ complexity.

In contrast, LDEDE's LRP-based localization decomposes output relevance in a single modified backward pass, with complexity $O(1 \cdot P)$, effectively $O(N)$ overall to traverse all neurons once, since relevance redistribution within a layer scales linearly with neuron count. This eliminates both the interpolation multiplier m and per-neuron iteration.

Memory Requirements: IG requires caching gradients across multiple backpropagation steps and retaining longer computation graphs during integration, which increases peak VRAM usage. LRP, by design, only stores forward activations and a single relevance map, leading to much lower memory footprint. To eliminate the impact of randomness, we conducted five empirical comparison tests on GPT-2 (137M parameters), processing 1024 test samples ($max\_seq\_len = 512$, $batch\_size = 32$, $float32$) , and the results are shown in Table 9.

Method	Average time	Average Peak VRAM (GB)
LRP	1min10s	2.8
IG ($m = 50$)	6min45s	7.9

Table 9: Empirical Comparison of LRP and IG in Computational Efficiency and Memory Consumption

The results show that LRP completed the attribution process in 1 minute and 10 seconds, with peak VRAM usage of only 2.8GB. In contrast, the IG algorithm using an interpolation step size of $m = 50$ took 6 minutes and 45 seconds, consuming up to 7.9GB of peak VRAM. This means LRP executes the same task nearly six times faster while reducing memory consumption by approximately 64%. The significant disparity in runtime and peak memory consumption reflects fundamental differences in computational design: IG's repetitive backpropagation steps incur cumulative time and memory overhead, whereas LRP distributes correlation scores across neurons through a single, optimized backpropagation pass with reduced storage and computational burden.

B Additional Experiment Results

B.1 Privacy Memory Strength Sensitivity Experiment

To investigate LDEDE's effectiveness in protecting privacy data with varying memory strengths, we conducted segmented tests on three categories of privacy samples based on memory strength grading standards. We quantified protection capabilities by calculating the variation in privacy risk before and after editing.

Privacy Type	Intensity	Count	Before Edit	After Edit	Variation ↑
Phone	Low	9800	17.48	14.96	2.52
	Medium	1775	22.45	16.26	6.19
	High	169	27.98	15.01	**12.97**
Name	Low	3801	0.49	0.40	0.09
	Medium	1133	0.67	0.56	0.11
	High	66	0.82	0.71	**0.11**
Email	Low	827	7.50	19.62	12.12
	Medium	555	4.65	18.92	14.27
	High	164	2.22	20.42	**18.20**

Table 10: Protection Effectiveness of LDEDE on Privacy Data with Different Memory Strengths

As shown in Table 10, the experimental results reveal a clear pattern: LDEDE's privacy protection effectiveness exhibits a significant positive correlation with privacy memory intensity. The Phone scenario demonstrates the most pronounced effect, where the Variation of 12.97 for high-memory-intensity samples is over five times that of low-memory-intensity samples (2.52). In the Email scenario, the Variation for high-memory-strength samples was 6.08 higher than that for low-memory-strength samples. In the Name scenario, the protection effectiveness for medium- and high-strength samples was consistent and superior to that for low-strength samples.

B.2 Model Scalability Experiment

The experiment focuses on two core dimensions: first, the stability of privacy protection and performance retention; second, the scalability of computational efficiency, specifically the relationship between model parameter scale and computational time.

Model	Before Edit		After Edit		Time Cost
	Valid-PPL	Exposure	Valid-PPL	Exposure	
GPT2(137M)	5.40	18.39	6.73	15.16	5min35s
GPT2-xl(1.6B)	4.71	18.68	5.99	15.21	29min57s
GPT2-neo(2.7B)	4.49	18.91	5.63	15.80	44min36s

Table 11: Comparison of LDEDE Method Utility Across Different Scales of the Same Model. After Edit: Lower Valid-PPL values are better; lower Exposure values are better.

As shown in Table 11, LDEDE provides consistent privacy protection across different scales of the same model: the exposure values for all three model scales decreased from 18–19 before editing to 15–16 after editing, with no significant performance degradation. Both GPT2-xl (15.21) and GPT2-neo (15.80) fall below 16. Regarding performance retention, post-editing Valid-PPL scores remained within the 5.63–6.73 range. Larger models exhibited smaller performance decay due to higher parameter redundancy—local privacy neuron editing had minimal impact on overall performance, further validating LDEDE's effectiveness in large-scale model scenarios. Regarding computational efficiency, despite a nearly 20-fold increase in model parameters (from 137 million to 2.7 billion), computation time only increased approximately sevenfold (from 5 minutes 35 seconds to 44 minutes 36 seconds). This time increase is significantly lower than the parameter growth rate.

B.3 Cross-Architecture Generalizability Experiment

Model	Before Edit		After Edit		Time Cost
	Valid-PPL	Pri-PPL	Valid-PPL	Pri-PPL	
BERT-base	9.78	7.51	10.16	19.04	1min57s
GPT2	5.40	5.92	5.67	19.46	1min46s
LLAMA-7B	5.17	5.78	5.80	19.77	13min17s

Table 12: Privacy Protection Effectiveness and Computational Efficiency of Different Models in the LDEDE Method. After Edit: A smaller Valid-PPL is better, while a larger Pri-PPL is better.

As shown in Table 12, LDEDE demonstrates stable privacy protection efficacy and performance compatibility across three heterogeneous architectures: In terms of privacy protection, the Pri-PPL scores of the three models increased from 5.78–7.51 to 19.04–19.77, significantly reducing privacy leakage risks and validating its ability to effectively suppress privacy exposure across different architectures. Regarding performance retention, post-editing Valid-PPL scores exhibited only minor fluctuations: BERT-base increased from 9.78 to 10.16, GPT-2 from 5.40 to 5.67, and LLAMA-7B from 5.17 to 5.80. Regarding computational efficiency, lightweight models BERT-base and GPT-2 completed processing within 2 minutes, while the large model LLAMA-7B took 13 minutes and 17 seconds.