PATCH: Mitigating PII Leakage in Language Models with Privacy-Aware Targeted Circuit PatcHing

Anthony Hughes[1], **Vasisht Duddu**[2], **N. Asokan**[2], **Nikolaos Aletras**[1], **Ning Ma**[1]
[1]University of Sheffield, [2]University of Waterloo
{ajhughes3, n.ma, n.aletras}@sheffield.ac.uk,
vasisht.duddu@uwaterloo.ca, asokan@acm.org

Abstract

Language models (LMs) may memorize personally identifiable information (PII) from training data, enabling adversaries to extract it during inference. Existing defense mechanisms such as differential privacy (DP) reduce this leakage, but incur large drops in utility. Based on a comprehensive study using circuit discovery to identify the computational circuits responsible PII leakage in LMs, we hypothesize that specific PII leakage circuits in LMs should be responsible for this behavior. Therefore, we propose PATCH (Privacy-Aware Targeted Circuit PatcHing), a novel approach that first identifies and subsequently directly edits PII circuits to reduce leakage. PATCH achieves better privacy-utility trade-off than existing defenses, e.g., reducing recall of PII leakage from LMs by up to 65%. Finally, PATCH can be combined with DP to reduce recall of residual leakage of an LM to as low as 0.01%. Our analysis shows that PII leakage circuits persist even after the application of existing defense mechanisms. In contrast, PATCH can effectively mitigate their impact.[1]

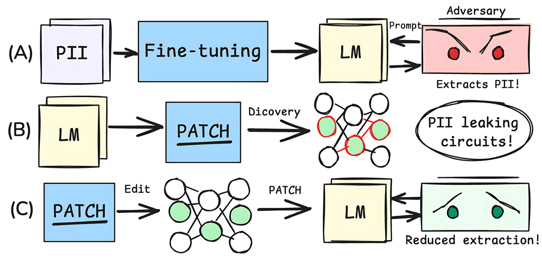

Figure 1: Mitigating PII leakage through circuit analysis of LMs fine-tuned on PII-containing documents **(A)** with and without privacy defenses. **(B)** We use PATCH to discover PII leaking circuits. Finally, **(C)** PATCH then edits the discovered circuits, reducing PII leaks.

1 Introduction

Language models (LMs) have demonstrated remarkable advances (Gemma-Team et al., 2024; Grattafiori et al., 2024), yet their tendency to memorize training data poses privacy risks (Kandpal et al., 2022; Buzaglo et al., 2023; Duan et al., 2024; Hayes et al., 2025b). In particular, prior work has shown that LMs can memorize and reproduce personally identifiable information (PII) from their training data (Huang et al., 2022; Kim et al., 2023; Nakka et al., 2024; Borkar et al., 2025). This makes it possible for adversaries with black-box access to a model to expose such information (Lukas et al., 2023).

A range of defense mechanisms have been proposed to mitigate PII leakage (Kerrigan et al., 2020; Wu et al., 2024). Data processing defenses, such as *scrubbing*, remove PII from data to reduce leakage (Pilán et al., 2022; Mosallanezhad et al., 2019; Lukas et al., 2023). Train time defenses such as differential privacy (DP) modify the models learning process to limit the influence of individual training examples while providing formal guarantees about the leakage (Kerrigan et al., 2020; Li et al., 2022; Ponomareva et al., 2022; Yu et al., 2022). Finally, post-training defenses, including model editing which remove or suppress PII-related knowledge from the LMs (Wu et al., 2023; Chen et al., 2024; Wu et al., 2024). However, these defenses demonstrate poor privacy-utility trade-offs (Lukas et al., 2023; Wu et al., 2024), unintentionally increase susceptibility to other PII types (Wu et al., 2024; Borkar et al., 2025), or can be circumvented by adversaries (Xin et al., 2024).

To address these limitations, we propose Privacy-Aware Targeted Circuit PatcHing (PATCH), which allows editing relevant PII circuits to minimize leakage. We first apply mechanistic interpretability to discover the internal computational structures (or "circuits") in a LM responsible for PII leakage.

[1]Code and data are publicly available at https://github.com/ssg-research/pii-patch/

Findings of the Association for Computational Linguistics: EACL 2026, pages 5139–5153
March 24-29, 2026 ©2026 Association for Computational Linguistics

We then "patch" or edit these circuits to mitigate the leakage. Our contributions are:

1. A novel targeted circuit editing method, PATCH, that provides greater privacy-utility trade-off than existing defense mechanisms (§3, §5)

2. We identify and characterize the specific circuits responsible for leaking different PII types, like names, locations and race, revealing that distinct attention heads within these circuits that influence PII leakage. (§6).

3. An extensive ablation study of PATCH, further demonstrating its robustness across settings. (§7).

2 Related Work

PII Leakage in LMs. Memorized information from the training data can be shared via model weights from which an adversary can extract PII (Lukas et al., 2023; Yu et al., 2023; Staab et al., 2024; Hayes et al., 2025b,a). For instance, Lukas et al. (2023) demonstrate PII leakage attacks against GPT2 (Radford et al., 2019) models, where given access to LMs, adversaries can sample responses to extract sensitive information. PII leakage vulnerabilities have been observed across diverse contexts, including pretrained models (Nakka et al., 2024; Staab et al., 2024; Kim et al., 2023; Panda et al., 2024; Huang et al., 2022) and specialized applications (Hughes et al., 2024; Mireshghallah et al., 2024; Xiao et al., 2024), indicating that privacy risks persist throughout the model's lifecycle and across diverse task domains.

Defenses against PII Leakage. A simple defense method is to remove PII from the data before training ("scrubbing") (Pilán et al., 2022; Mosallanezhad et al., 2019; Lukas et al., 2023). However, such approaches are significantly expensive (Wu et al., 2024), provide poor privacy-utility trade-offs (Wu et al., 2023; Lukas et al., 2023; Wu et al., 2024), and may allow an adversary to deduce personal attributes through auxiliary information (Xin et al., 2024). Alternatively, DP methods (Feyisetan et al., 2020; Kerrigan et al., 2020; Li et al., 2022; Shi et al., 2022; Lee and Søgaard, 2023) offer formal guarantees of privacy by injecting noise during training, effectively masking individual samples observed by the LM. Yet, this often results in a reduction in utility (Lukas et al., 2023; Wu et al., 2024). Recent empirical defenses include identifying neurons that are responsible for memorizing PII and patching with steering vectors can reduce that memorization (Wu et al., 2023; Chen et al., 2024; Wu et al., 2024). However, Wu et al. (2024) indicate that such methods suffer from poor privacy-utility trade-offs due to limited components that can be edited, with the side effect of increasing the leakage of other PII types.

Algorithm 1 PATCH: Privacy-Aware Targeted Circuit PatcHing

Input: Model $\hat{M}$, PII types P, percentile threshold p, model editing method A, circuit discovery CD, private data D
Output: Privacy-enhanced model $\hat{M}$

for each PII type $P_i \in P$ **do**
  $S \leftarrow$ prompt_builder(D) ▷ Generate prompts
  $C_i \leftarrow CD(\hat{M}, P_i, D, S)$ ▷ Extract circuit
  $\tau_i \leftarrow$ percentile$(\{s_e^{(i)} : e \in C_i\}, p)$ ▷ Compute Threshold
  $E_i^{high} \leftarrow \{e \in C_i : s_e^{(i)} \geq \tau_i\}$ ▷ Select edges
end for
$E_{shared} \leftarrow \bigcap_i E_i^{high}$ ▷ Identify shared edges
$\hat{M} \leftarrow A(\hat{M}, E_{shared})$ ▷ Patch
return Modified model $\hat{M}$

3 PATCH: Privacy-Aware Targeted Circuit PatcHing

3.1 Problem Formulation

Given a pre-trained model $\mathcal{M}$ and a private dataset $\mathcal{D}$ that contains PII $\mathcal{P}$, fine-tuning $\mathcal{M}$ on $\mathcal{D}$ results to a model $\hat{M}$ that is exposed to PII. We also assume an adversary $\mathcal{A}$ with black-box access to $\hat{M}$. $\mathcal{A}$ seeks to infer specific PII types that are observable from $\mathcal{D}$ via prompting. Finally, a defense mechanism $\mathcal{DF}$ aims to reduce the effectiveness of $\mathcal{A}$ on extracting PII from $\hat{M}$, while maintaining the utility (i.e., performance) of $\mathcal{M}$ (Lukas et al., 2023). $\mathcal{DF}$ can be applied before on $\mathcal{D}$, during fine-tuning on $\mathcal{D}$ or post-hoc, after training.

3.2 Motivation

We hypothesize that there is a set of unique elements (e.g., attention heads or outputs from heads to other parts of the Transformer block) within $\hat{M}$ responsible for PII leaks. By identifying and modifying these elements, we can reduce PII leakage while maintaining utility. More specifically, given a circuit discovery mechanism $\mathcal{CD}$, a private dataset $\mathcal{D}$, and a model $\mathcal{M}$, PATCH consists of the following three steps (detailed in Algorithm 1).

3.3 Step 1: Generate Prompts for Circuit Discovery.

To identify circuits responsible for PII leakage across all PII types, we employ Edge Attribution Patching with Integrated Gradients (Hanna et al., 2024, EAP-IG) as our circuit discovery mechanism $\mathcal{CD}$. EAP-IG is a mechanistic interpretability technique designed to efficiently discover circuits within Transformer models. We select EAP-IG for its effectiveness at identifying faithful circuits (Mueller et al., 2025).

EAP-IG requires constructing prompts representing PII leakage so we can extract a circuit that is representative of that behavior. We construct pairs of "clean" and "corrupt" prompts. The clean version contains correct PII values and the corrupted version has these values replaced with alternatives from the same type. Using a private dataset D that is tagged with a target PII type P_i, we select $1,000$ unique text spans containing that PII element P_i. The PII span is then replaced ("corrupted") with PII that is semantically similar to that entity. Examples generated from a legal dataset (Chalkidis et al., 2019) are shown in Table 1.

PII Type	Original	Corrupted
Name	"Mr. John Smith vs. The State"	"Mr. Heidi Smith v. The State"
Location	"The appellant was arrested in Berlin ."	"The appellant was arrested in New York ."
Race	"This case concerns a Romanian national."	"This case concerns a Turkish national."

Table 1: Examples from the European Court of Human Rights (ECHR) dataset (Chalkidis et al., 2019), illustrating PII types used in our original and corrupted prompts. Purple represents a original token, red represents a corrupted token.

3.4 Step 2: Extract PII Leakage Circuits

EAP-IG operates by comparing model behavior on clean prompts against corrupted variants where specific PII tokens have been altered. The method analyzes individual attention heads to measure two properties: (1) how sensitive each component's activations are to PII token corruption, and (2) how important each component is for correctly predicting PII values. Heads scoring high on both dimensions—being necessary for accurate PII prediction—are identified as part of the PII leakage circuit. Given $\hat{M}$ and $\mathcal{D}$, EAP-IG outputs a circuit C_i containing *nodes* that represent individual attention heads and *edges*] that indicate information flow between nodes. Each node (attention head) and edge (connection to other nodes) receives an importance score $s_e^{(i)}$ quantifying their contribution to PII leaking behavior. In our experiments, we use the default hyperparameters recommended by Hanna et al. (2024).

3.5 Steps 3 and 4: Compute Threshold and Perform Edge Selection

We hypothesize that high-scoring edges indicate critical PII leakage pathways. Therefore, we aim to isolate those critical edges for patching.

Given a circuit discovery mechanism $\mathcal{CD}$ that has scored the edges from a circuit C_i within $\hat{M}$, we require a mechanism to select the most influential edges. We first compute a threshold τ for each PII circuit C_i, given a specified percentile p. Next, we select the high-scoring edges whose scores are equal to or greater than the threshold τ. We repeat this process for each PII type to obtain $\{E_i^{high}\}$ for all circuits $\{C_i\}$. Following prior work (Wang et al., 2025), we select the high-scoring edges using percentile thresholds of 95% and 99%.

3.6 Step 5: Identify Shared PII Edges

We aggregate edge scores across multiple PII types rather than treating each type independently, assuming that PII leakage has common pathways throughout models. Therefore, we compute E_{shared} as the intersection of high-scoring edges E_i^{high} across all PII circuit types C_i.

3.7 Step 6: Patch PII Edges

Given a set of shared edges E_{shared} from $\hat{M}$, we require a model editing mechanism $\mathcal{A}$ that alters those high-importance edges such that their influence on the model's behavior is reduced. We experiment with zero ablation, i.e.,setting edge weights to zero (Olah et al., 2020; Pochinkov et al., 2024); and mean ablation, i.e.,replacing edge weights with their mean values (Chan et al., 2022; Wang et al., 2023), following their successful application in prior work (Bi et al., 2025).

4 Experimental Setup

4.1 PII Types and Private Data

We select *names*, *locations*, and *race* as a set of PII types following prior work (Pilán et al., 2022; Hughes et al., 2024; Kim et al., 2024). We use the European Court of Human Rights (ECHR) dataset (Chalkidis et al., 2019), which contains PII related to appellants and others involved in legal cases. To automatically identify PII across all train, validation and test sets, we apply the FLAIR name-entity-recognition (NER) tool.[2] FLAIR achieves an accuracy of approximately 86% (Yermilov et al., 2023). Details of the NER label classes are provided in Table 4. Detailed corpus and PII statistics can be found in Appendix A.

4.2 Circuit Discovery Prompts

Using our private dataset, ECHR, that is tagged with the three PII types, we select unique text spans that containing the tagged PII element. We following prior work (Wang et al., 2025) and select 1,000 spans for each of the target PII types. The PII element in each span is then "corrupted" with PII that is semantically similar to that entity. We select entities using the faker library.[3] The clean and corrupted prompts are used to score circuit edges across all PII types (Table 1).

4.3 Base Models

We use several open-weight LMs such as *GPT2-Small* (117M), *GPT2-Medium* (345M), and *GPT2-Large* (774M) following prior work (Lukas et al., 2023; Wu et al., 2024), as well as more recent models such as *Llama-3.2-1B* (Grattafiori et al., 2024), *Qwen3-0.6B* and *Qwen3-1.7B* (Team, 2025).

4.4 Fine-tuning Target Models

To obtain LMs exposed to PII (target), we fine-tune all base models on the private ECHR dataset. We conduct our experiments using Hugging Face[4] for all models. The max sequence length is set to 512. All experiments on open-weight models are performed on one to four NVIDIA H100 GPUs. Fine-tuning uses a batch size of 8, the AdamW optimizer (Loshchilov and Hutter, 2019), and a linear learning rate scheduler. Each baseline, DP and scrubbed defended model is trained for 4 epochs.

4.5 Adversary

To emulate an adversary, we sample approximately four million tokens from each target LM with and without defenses. Each query begins with an empty prompt, then we issue 10,000 queries to each model, generating sequences of 256 tokens from empty prompts using top-k sampling with $k = 40$ of which we apply random sampling.

To control for baseline PII leakage present in the base pretrained model, we establish a reference distribution by sampling 13 million tokens ($50,000$ queries) from it prior to fine-tuning. Any PII instances appearing in this baseline are excluded from our leakage measurements, ensuring we only attribute leakage to the fine-tuning process rather PII aquired during pretraining. We repeat the attack three times to quantify variance. This is following prior work in PII leakage (Lukas et al., 2023).

4.6 Metrics

Privacy Leakage. To assess the adversary success in leaking PII, we directly compare verbatim LM outputs against FLAIR annotations, which serve as ground truth. *Precision* measures the proportion of PII in the model's output is PII that was also present in the training data. *Recall* indicates how much of the total PII observable in the training data is exposed.

Faithfulness. We assess how accurately a discovered circuit represents the causal mechanism underlying PII leakage. A circuit is considered faithful if ablating any components outside the circuit does not affect the model's performance, indicating that the circuit alone explains the behavior (Prakash et al., 2024; Hanna et al., 2024). We report the normalized faithfulness as:

$$\frac{P_{\text{method}} - P_{\text{corrupted}}}{P_{\text{baseline}} - P_{\text{corrupted}}}$$

P_{method} is the performance of a circuit after applying EAP-IG, P_{baseline} is the original model performance, and $P_{\text{corrupted}}$ is the corrupted baseline performance. The score ranges from 0 (no resemblance to the original model) to 1 (full recovery of performance).

Circuit Overlap. To identify how PII circuits interact and where editing those circuits is optimal, we require a circuit overlap measure. Following Hanna et al. (2024), we use the *overlap metric* based on high scoring circuit elements that meet a percentile threshold. For each PII type $P_i \in T$, we extract circuit C_i with node or edge scores $s_e^{(i)}$ using EAP-IG $\mathcal{CD}$. We then compute a threshold τ_i and identify high-importance edges E_i^{high}. The overlap between two PII circuits C_i and C_j is calculated using the Jaccard index:

$$\text{Overlap}(C_i, C_j; \tau) = \frac{|E_i^{high} \cap E_j^{high}|}{|E_i^{high} \cup E_j^{high}|} \times 100\%$$

[2] FLAIR: https://github.com/flairNLP/flair
[3] https://faker.readthedocs.io/en/master/
[4] https://www.huggingface.co

τ denotes the percentile threshold. High overlap indicates shared computational pathways across PII types. We analyze edge overlap at percentile thresholds of $p \in \{95, 99\}$ following (Wang et al., 2025), with higher thresholds selecting fewer but more critical edges for editing.

Utility. We use *perplexity* over the ECHR test set, similar to prior work (Lukas et al., 2023; Wu et al., 2023; Chen et al., 2024; Wu et al., 2024), to evaluate the impact on LM utility before and after applying PATCH and baseline defense mechanisms.

4.7 Defense Baselines

We compare with the following defenses:

APNEAP. Augmented Privacy Neuron Editing via Activation Patching (Wu et al., 2024) identifies and ablates individual neurons responsible for PII leakage. This is the current state-of-the-art editing-based defense which outperforms prior editing approaches (Wu et al., 2023; Chen et al., 2024).

DP. We fine-tune LMs using differentially private stochastic gradient descent (Abadi et al., 2016, DP-SGD). We consider different privacy loss parameter ϵ ={8, 4, 1} where a lower ϵ indicates stronger privacy guarantee. For DP training, we use the fastDP library (Bu et al., 2022, 2023). Following prior work (Lukas et al., 2023), each model is trained using DP-SGD for 4 epochs using (ϵ, δ) = ($\{8, 4, 1\}$, $\frac{1}{N}$) where N is the size of the training dataset. We use a maximum per-sample gradient norm of 1.

Scrubbing. We first remove PII from the data, and then use it for fine-tuning LMs. All information related to the selected PII types (names, locations, race) is removed. Again, we use an NER tool, FLAIR, to identify any spans containing PII, we then redact those spans. Redaction means replacing the identified span with a masking token. Models are then trained on the resulting scrubbed documents.

PATCH. We evaluate two variants of our proposed PATCH approach. First, PATCH-Baseline applies our method to a model $\mathcal{M}$ trained without any existing privacy defenses. Second, PATCH-DP($\epsilon = 8$) applies our method to a model $\mathcal{M}$ that has been fine-tuned with DP($\epsilon = 8$).

5 Results

Figure 2 presents privacy-utility tradeoffs for PATCH against three baseline defenses across all models. We measure PII leakage with precision and recall, and utility using perplexity. We evaluate two variants of our method: PATCH-BASELINE applied to non-private models, and PATCH-DP(ϵ) applied to a model fine-tuned with DP at $\epsilon = 8$.[5]

PATCH-Baseline achieves strong privacy-utility tradeoffs. Across all models, PATCH-BASELINE reduces PII extraction precision by 40%–90% relative to undefended baselines. For example, *Llama-3.2-1B* precision decreases from 60% to 0.5, while incurring only modest utility costs; *GPT2-Medium* perplexity increases from approximately 9 to 10. Recall consistently decreases by 80%–86% across architectures, with *GPT2-Medium* decreasing from 8.5% to 1.2% and *GPT2-Large* from approximately 11% to 1.8% while maintaining a utility within 1 point of the baseline perplexity. These reductions demonstrate that PII leakage localizes to identifiable circuit edges rather than distributing across the entire parameter space. The consistency across architectures indicates that PII leaks via specific mechanisms.

PATCH-DP provides maximal privacy at varied utility costs. Combining circuit patching with DP yields the strongest privacy protection, achieving recall $< 1\%$ across all models, but with model-dependent utility impact. For smaller models, PATCH-DP($\epsilon = 8$) maintains reasonable utility: on *GPT2-Small* , perplexity increases to 19.5 (vs. 18.3 for DP($\epsilon = 8$) alone, and 10.3 for Base), while recall drops to $< 0.5\%$ compared to 2.0% for DP alone. Similar patterns emerge for *Llama-3.2-1B* where perplexity is 32 compared to 27 for the DP baseline, however recall approaches 0%. This demonstrates good protection, but fails to maintain utility.

Be careful of precision increases. An important finding in PATCH-DP($\epsilon = 8$) in *GPT2-Large* , is that it successfully reduces recall, however it exhibits concerning spikes in precision with stable perplexity, particularly evident in *GPT2-Large* where precision reaches ~ 80 while recall drops to ~ 1. This tells us the leaks have become highly accurate. For adversaries mounting training data extraction attacks, high precision and low recall suggest a vulnerability: *the model may leak infrequently, but successful extractions yield genuine,*

[5]We ran experiments with a DP fine-tuned model at $\epsilon = \{4, 1\}$, however we obtained poor utility. Table of experiments is available in Appendix C.

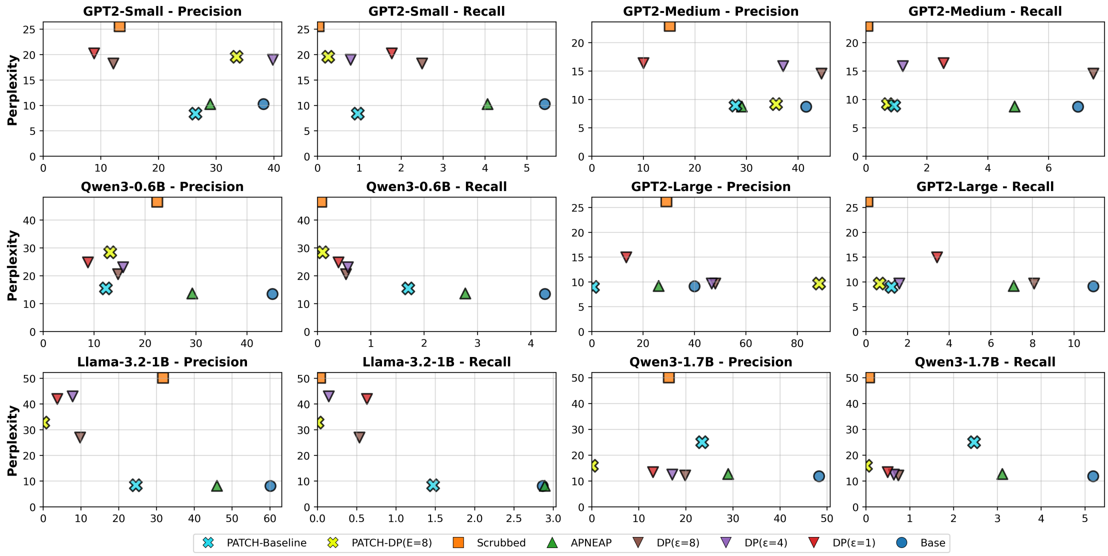

Figure 2: **Comparison of PATCH with other defenses:** lower perplexity, precision and recall scores are preferred (closer to lower left corner).

trustworthy PII. This pattern is typically present in the baseline model. PATCH-Baseline corrects this issue, however *it is important to note that a defense may simultaneously increase model susceptibility.*

Comparison to existing defenses. Undefended baseline models exhibit substantial PII leakage, with precision ranging from 38.6% to 60.8% and recall from 3.0% to 11.0% across architectures, confirming the severity of leaks in models fine-tuned on sensitive data. Among existing defense mechanisms, DP provides strong privacy protections, reducing recall to $\leq$ 6% across all models. For example, *Llama-3.2-1B* decreases to 0.5%. However, these models incur substantial utility costs with perplexity increases of 1.5 to 7.9 times the baseline perplexity. APNEAP (Wu et al., 2024) maintains utility within +1.0 perplexity of baseline models but provides limited privacy protection, reducing extraction precision by only $\sim$ 10-18% and recall by $\sim$ 1-3%.

Practical implications. Overall, our results demonstrate that circuit-based interventions can provide quantifiable privacy-utility tradeoffs complementary to formal DP guarantees. For deployment scenarios where moderate privacy protection suffices and utility is paramount, PATCH-Baseline offers $\sim$ 1.5-8% reduction in recall with minimal perplexity overhead. For high-security applications requiring maximal privacy, PATCH-DP($\epsilon = 8$) can achieve $\leq$ 1% in precision and recall, though *practitioners should be conscious of utility*, as there were cases, *Qwen3-1.7B* , where utility was not preserved. *These differences across models underscores the need for careful evaluation of defense mechanism impacts across model architectures.*

Privacy budget considerations. Our circuit identification process analyzes model activations on the training data to locate relevant circuits for PII leakage. This data-dependent analysis is not accounted for in the privacy hyperparameter (ϵ) that was specified during DP fine-tuning. For scenarios requiring strict formal guarantees, circuit identification should be performed on a separate dataset, after which patching operations would preserve the training-time privacy guarantees. *In this work, we demonstrate valuable empirical privacy protection through targeted circuit interventions.*

6 Analysis of PII Leakage Circuits

6.1 PII leakage and Circuits

Figure 3 presents the results of the PII leakage and circuit discovery, for three PII types: names, location, and race.

We find precision is variable with regard to model architectures, however, within architectures, such as GPT2 and Qwen3, recall increases with model size. This corroborates prior work (Lukas et al., 2023). Among other models, *Llama-3.2-1B* has the highest precision indicating that the PII produced by this model is more likely to be from the training data. In contrast, *GPT2-Large* has the highest recall exposing the most PII in training data.

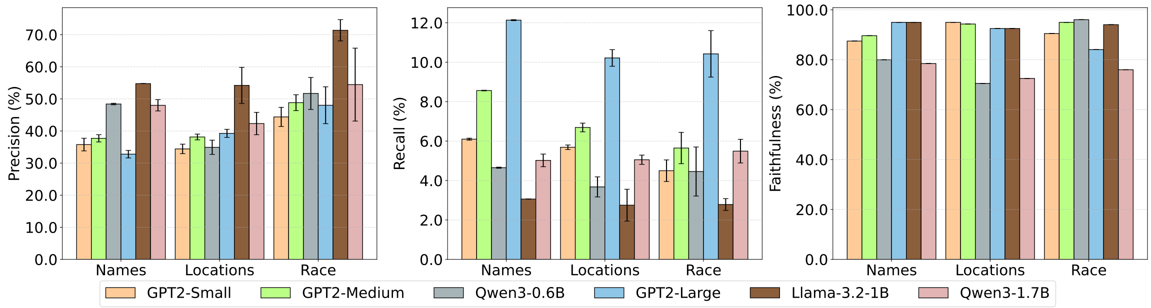

Figure 3: **Results for PII leakage and faithfulness:** we present the precision (**left**), and recall (**middle**) of the PII extracted, and finally, the faithfulness of the discovered PII circuits (**right**).

Furthermore, the PII leakage, as seen with precision and recall, varies across different PII types and across models. Regardless of the model or PII type faithfulness is high, indicating that we can reliably identify the circuits for PII leakage. Faithfulness scores (above 75%) are similar to prior circuit discovery work (Wang et al., 2025). *Overall, we can reliably identify circuits for PII leakage, and this remains high across all models.*[6]

6.2 PII Circuit Overlaps

Table 2 shows the results for PII circuit overlap. This allows us to understand the structure of the individual PII circuits, and how editing these may impact an attack.

Model	PII Circuit Overlaps (nodes/edges (%))		
	Name-Location	Name-Race	Location-Race
GPT2-Small	79 / 39	88 / 48	77 / 38
GPT2-Medium	66 / 23	71 / 26	72 / 24
Qwen3-0.6B	62 / 50	61 / 41	54 / 41
GPT2-Large	70 / 30	69 / 24	65 / 24
Llama-3.2-1B	82 / 44	82 / 40	84 / 40
Qwen3-1.7B	87 / 25	87 / 24	96 / 49

Table 2: **Results of circuit overlap analysis across PII types and models:** we present *circuit overlap (%)* between faithful PII circuits (format: "nodes/edges").

Minimal edge overlap across PIIs types. Comparing connectivity patterns of circuits responsible for different PII types reveals substantial differences in the edge overlap. We observe a consistently low overlap between the edges of PII circuits across all models. *GPT2-Small* shows 38.8% overlap between race and names, 37.6% between race and locations, and 48.3% between names and locations, while *GPT2-Large* and *Qwen3-1.7B* exhibit even lower overlap of $\sim 25\%$. This suggests that different PII circuits have distinct pathways. Low edge overlap combined with high node overlap in larger models indicates greater circuit specialization with increased capacity. This also explains the observations in prior node ablation work (Wu et al., 2023, 2024; Borkar et al., 2025) where minimizing leakage of one PII through node editing, increases others. *This pattern also suggests that LMs do not maintain entirely separate mechanisms for retrieving PII, but instead rely on a smaller set of shared attention patterns across PII types.*

6.3 Influential Edges in PII Circuits

Given the low overlapping scores observed among edges, we investigate their individual influence. We observe the scores generated by the circuit discovery method (EAP-IG). We present heatmaps for a subset of the models attention layers and heads in Figure 5.[7] We find that circuit discovery finds distinct influence patterns of attention head edges across architectures. *GPT2-Medium* displays a more distributed pattern with notable activation in layers 14-16 across multiple heads. *Llama-3.2-1B* shows sparse but intense activation primarily in later layers, with early layer concentration with high scores in *layer* 9. This is also visible in *Qwen3-0.6B* and *Qwen3-1.7B* where *layer* 0 is highly influential. Interestingly, some layer and heads are very pronounced. We identify *layer* 11 *head* 9 as highly influential in both *Qwen3* models. *Overall, key edges vary across architectures but systematic circuits that drive PII leakage exist across all models.*

7 Ablation Study

To be better understand PATCH, we perform attacks under different hyperparameter settings. We report

[6]We also analyzed circuit overlaps across defense baselines. Overlap metrics are available in Appendix E.

[7]See full heatmaps of each model's attention layers and heads in Appendix D.

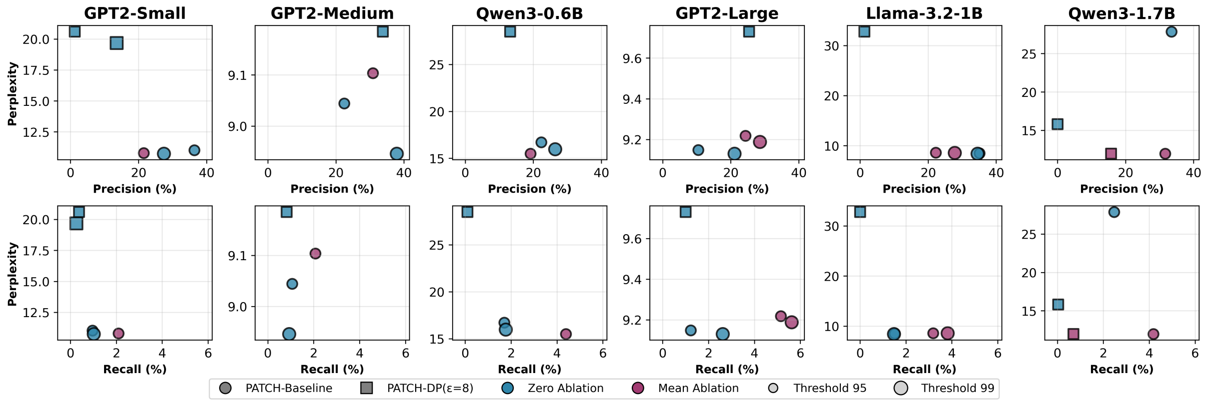

Figure 4: **Results of PATCH across varying hyperparameters:** we compare PATCH-Baseline with PATCH-DP($\epsilon = 8$) with alternating ablation strategies, zero and mean, and edge thresholds, 95 and 99.

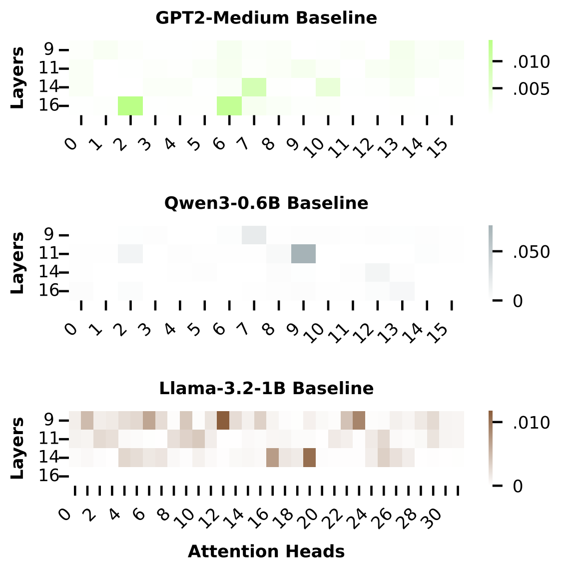

Figure 5: **Influential PII Circuit Components:** average EAP-IG scores for each attention head in each layer, across all identified PII circuits.

the results in Figure 4. As indicated in Section 3, we use 95% and 99% percentile threshold on the importance scores of the edges. We also evaluate PATCH with *zero* and *mean* ablation with both thresholds. This helps evaluate whether there are other configurations which can improve privacy-utility trade-offs.

Model editing strategies show no universal differences. Our comparative analysis of mean versus zero ablation strategies reveals no systematic preference across model architectures and scales. For smaller models such as *GPT2-Small* , zero ablation configurations cluster in regions of lower perplexity with marginally reduced F1-scores, suggesting minimal functional differentiation between ablation methods. *GPT2-Medium* and *Qwen3-1.7B* mean and zero ablation variants are without clear separation. Practitioners should evaluate both strategies during post-training. Optimal ablation appears model specific.

Threshold selection has model-specific effects. The thresholds 95% and 99% comparison reveals model-specific patterns without consistent trends. *GPT2-Small* shows threshold 99 configurations incurring higher perplexity costs for equivalent privacy gains, while *GPT2-Large* exhibits the opposite relationship where stricter thresholds maintain comparable utility. *Qwen3-0.6B* and *Qwen3-1.7B* architectures demonstrate negligible threshold sensitivity, with both settings producing similar privacy-utility trade-offs. This heterogeneity indicates that optimal threshold selection requires model-specific calibration.

8 Conclusion

We introduced PATCH, a method grounded in mechanistic interpretability for mitigating PII leakage in LMs through targeted editing. Our approach identifies shared computational pathways responsible for PII memorization across PII types and selectively ablates high-importance edges to reduce privacy risks. Empirical results demonstrate that PATCH achieves superior privacy-utility trade-offs compared to prior work. Our extensive experiments verify the effectiveness of PATCH across multiple model scales and PII categories compared to non-private fine-tuned baselines. These findings demonstrate that targeted circuit interventions can provide effective privacy protection while preserving the model's core computational pathways.

Limitations

In this study, we empirically demonstrate that our method substantially improves open-weight LMs in mitigating PII leaks, however we acknowledge our evaluation is limited to a subset of PII types. We hope to extend our proposed method to a broader set of PII in the future. Finally, experiments have not been conducted on large models, due to circuit discovery requiring intensive resources and time, however as new methods arise they can integrate into our proposed method.

Acknowledgments

AH is supported by the Centre for Doctoral Training in Speech and Language Technologies (SLT) and their Applications funded by UK Research and Innovation [EP/S023062/1]. This work is supported in part by Intel (in the context of Private AI consortium), and the Government of Ontario. Vasisht is supported by David R. Cheriton Scholarship, Cybersecurity and Privacy Excellence Graduate Scholarship, and an IBM PhD Fellowship. Additionally, we acknowledge IT Services at The University of Sheffield for the provision of services for High Performance Computing. Finally, views expressed in the paper are those of the authors and do not necessarily reflect the position of the funding agencies.

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*, CCS '16, page 308–318, New York, NY, USA. Association for Computing Machinery.

Jing Bi, Junjia Guo, Yunlong Tang, Lianggong Bruce Wen, Zhang Liu, Bingjie Wang, and Chenliang Xu. 2025. Unveiling visual perception in language models: An attention head analysis approach. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4135–4144.

Jaydeep Borkar, Matthew Jagielski, Katherine Lee, Niloofar Mireshghallah, David A. Smith, and Christopher A. Choquette-Choo. 2025. Privacy ripple effects from adding or removing personal information in language model training. In *ACL 2025 Student Research Workshop*.

Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. 2022. Differentially private bias-term finetuning of foundation models. In *Workshop on Trustworthy and Socially Responsible Machine Learning, NeurIPS 2022*.

Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. 2023. Differentially private optimization on large model at small cost. In *International Conference on Machine Learning*, pages 3192–3218. PMLR.

Gon Buzaglo, Niv Haim, Gilad Yehudai, Gal Vardi, Yakir Oz, Yaniv Nikankin, and Michal Irani. 2023. Deconstructing data reconstruction: Multiclass, weight decay and general losses. In *Advances in Neural Information Processing Systems*, volume 36, pages 51515–51535. Curran Associates, Inc.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural legal judgment prediction in English. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4317–4323, Florence, Italy. Association for Computational Linguistics.

Lawrence Chan, Adria Garriga-Alonso, Nicholas Goldowsky-Dill, Ryan Greenblatt, Jenny Nitishinskaya, Ansh Radhakrishnan, Buck Shlegeris, and Nate Thomas. 2022. Causal scrubbing: A method for rigorously testing interpretability hypotheses. In *AI Alignment Forum*, volume 2.

Ruizhe Chen, Tianxiang Hu, Yang Feng, and Zuozhu Liu. 2024. Learnable privacy neurons localization in language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 256–264, Bangkok, Thailand. Association for Computational Linguistics.

Sunny Duan, Mikail Khona, Abhiram Iyer, Rylan Schaeffer, and Ila R. Fiete. 2024. Uncovering Latent Memories: Assessing Data Leakage and Memorization Patterns in Large Language Models. In *ICML 2024 Workshop on LLMs and Cognition*.

Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. 2020. Privacy- and Utility-Preserving Textual Analysis via Calibrated Multivariate Perturbations. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 178–186, Houston TX USA. ACM.

Gemma-Team and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Aaron Grattafiori and 1 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. 2024. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. In *ICML 2024 Workshop on Mechanistic Interpretability*.

Jamie Hayes, Ilia Shumailov, William P Porter, and Aneesh Pappu. 2025a. Measuring memorization in RLHF for code completion. In *The Thirteenth International Conference on Learning Representations*.

Jamie Hayes, Marika Swanberg, Harsh Chaudhari, Itay Yona, Ilia Shumailov, Milad Nasr, Christopher A. Choquette-Choo, Katherine Lee, and A. Feder Cooper. 2025b. Measuring memorization in language models via probabilistic extraction. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 9266–9291, Albuquerque, New Mexico. Association for Computational Linguistics.

Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. 2022. Are large pre-trained language models leaking your personal information? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2038–2047, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Anthony Hughes, Ning Ma, and Nikolaos Aletras. 2024. How private are language models in abstractive summarization? *arXiv preprint arXiv:2412.12040*.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*, pages 10697–10707. PMLR.

Gavin Kerrigan, Dylan Slack, and Jens Tuyls. 2020. Differentially private language models benefit from public pre-training. In *Proceedings of the Second Workshop on Privacy in NLP*, pages 39–45, Online. Association for Computational Linguistics.

Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sungroh Yoon, and Seong Joon Oh. 2023. ProPILE: Probing privacy leakage in large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Woojin Kim, Sungeun Hahm, and Jaejin Lee. 2024. Generalizing Clinical De-identification Models by Privacy-safe Data Augmentation using GPT-4. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21204–21218, Miami, Florida, USA. Association for Computational Linguistics.

Seolhwa Lee and Anders Søgaard. 2023. Private Meeting Summarization Without Performance Loss. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2282–2286, Taipei Taiwan. ACM.

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. 2022. Large language models can be strong differentially private learners. In *International Conference on Learning Representations*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

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

Niloofar Mireshghallah, Maria Antoniak, Yash More, Yejin Choi, and Golnoosh Farnadi. 2024. Trust No Bot: Discovering Personal Disclosures in Human-LLM Conversations in the Wild. In *First Conference on Language Modeling*.

Ahmadreza Mosallanezhad, Ghazaleh Beigi, and Huan Liu. 2019. Deep Reinforcement Learning-based Text Anonymization against Private-Attribute Inference. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2360–2369, Hong Kong, China. Association for Computational Linguistics.

Aaron Mueller, Atticus Geiger, Sarah Wiegreffe, Dana Arad, Iván Arcuschin, Adam Belfki, Yik Siu Chan, Jaden Fried Fiotto-Kaufman, Tal Haklay, Michael Hanna, Jing Huang, Rohan Gupta, Yaniv Nikankin, Hadas Orgad, Nikhil Prakash, Anja Reusch, Aruna Sankaranarayanan, Shun Shao, Alessandro Stolfo, and 4 others. 2025. MIB: A mechanistic interpretability benchmark. In *Forty-second International Conference on Machine Learning*.

Krishna Kanth Nakka, Ahmed Frikha, Ricardo Mendes, Xue Jiang, and Xuebing Zhou. 2024. PII-compass: Guiding LLM training data extraction prompts towards the target PII via grounding. In *Proceedings of the Fifth Workshop on Privacy in Natural Language Processing*, pages 63–73, Bangkok, Thailand. Association for Computational Linguistics.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*. Doi:10.23915/distill.00024.001.

Ashwinee Panda, Christopher A. Choquette-Choo, Zhengming Zhang, Yaoqing Yang, and Prateek Mittal. 2024. Teach LLMs to phish: Stealing private information from language models. In *The Twelfth International Conference on Learning Representations*.

Ildikó Pilán, Pierre Lison, Lilja Øvrelid, Anthi Papadopoulou, David Sánchez, and Montserrat Batet. 2022. The Text Anonymization Benchmark (TAB): A Dedicated Corpus and Evaluation Framework for Text Anonymization. *Computational Linguistics*, 48(4):1053–1101.

Nicholas Pochinkov, Ben Pasero, and Skylar Shibayama. 2024. Investigating neuron ablation in attention heads: The case for peak activation centering. *arXiv preprint arXiv:2408.17322*.

Natalia Ponomareva, Jasmijn Bastings, and Sergei Vassilvitskii. 2022. Training text-to-text transformers with privacy guarantees. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2182–2193, Dublin, Ireland. Association for Computational Linguistics.

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. 2024. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In *International Conference on Learning Representations*. ArXiv:2402.14811.

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

Weiyan Shi, Aiqi Cui, Evan Li, Ruoxi Jia, and Zhou Yu. 2022. Selective Differential Privacy for Language Modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2848–2859, Seattle, United States. Association for Computational Linguistics.

Robin Staab, Mark Vero, Mislav Balunovic, and Martin Vechev. 2024. Beyond memorization: Violating privacy via inference with large language models. In *The Twelfth International Conference on Learning Representations*.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*.

Xu Wang, Yan Hu, Wenyu Du, Reynold Cheng, Benyou Wang, and Difan Zou. 2025. Towards understanding fine-tuning mechanisms of LLMs via circuit analysis. In *Forty-second International Conference on Machine Learning*.

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*, pages 5319–5332, Bangkok, Thailand. Association for Computational Linguistics.

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. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2875–2886, Singapore. Association for Computational Linguistics.

Yijia Xiao, Yiqiao Jin, Yushi Bai, Yue Wu, Xianjun Yang, Xiao Luo, Wenchao Yu, Xujiang Zhao, Yanchi Liu, Quanquan Gu, Haifeng Chen, Wei Wang, and Wei Cheng. 2024. Large Language Models Can Be Contextual Privacy Protection Learners. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14179–14201, Miami, Florida, USA. Association for Computational Linguistics.

Rui Xin, Niloofar Mireshghallah, Shuyue Stella Li, Michael Duan, Hyunwoo Kim, Yejin Choi, Yulia Tsvetkov, Sewoong Oh, and Pang Wei Koh. 2024. A false sense of privacy: Evaluating textual data sanitization beyond surface-level privacy leakage. In *Neurips Safe Generative AI Workshop 2024*.

Oleksandr Yermilov, Vipul Raheja, and Artem Chernodub. 2023. Privacy- and Utility-Preserving NLP with Anonymized data: A case study of Pseudonymization. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 232–241, Toronto, Canada. Association for Computational Linguistics.

Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. 2022. Differentially private fine-tuning of language models. In *International Conference on Learning Representations*.

Weichen Yu, Tianyu Pang, Qian Liu, Chao Du, Bingyi Kang, Yan Huang, Min Lin, and Shuicheng Yan. 2023. Bag of Tricks for Training Data Extraction from Language Models. In *ICML*, pages 40306–40320.

A Dataset Analysis

Table 3 presents detailed statistics regarding the corpora used in our experiments.

		Words		PII		
Task	**Tr/Dev**	**Mean**	**Max**	**Names**	**Locations**	**Race**
ECHR	118161/26258	70	8723	63685	2089	9921

Table 3: Distribution of source documents in ECHR. The mean and maximum word count for all source documents is presented along with an overview of the quantity of PII in all documents.

B Flair Classes

In Table 4, we present the classes of PII used to tag and monitor for leaks in our experiments.

Class	Description	Example from ECHR
PERSON	Names of persons	According to the Government, Mr L. had submitted that on the date in question.
LOC	General locations	Filed a letter with the Chancellor of the Jagiellonian University in Kraków .
NORP	Race, national, religious groups	a group of Turkish nationalists

Table 4: The classes used by the Flair tagger for word classification. We highlight the identified span in red

C Results of all baseline defenses

In Table 5 we present the results of all defense baselines.

Defense	**Perpl** ↓	**Prec (%)** ↓	**Rec (%)** ↓
	GPT-Small		
Baseline	10.26	38.19 ± 4.98	5.43 ± 0.80
APNEAP	10.27	29.92 ± 3.70	5.02 ± 0.87
DP (ϵ=8)	18.27	10.38 ± 4.60	2.27 ± 0.74
DP (ϵ=4)	19.01	11.49 ± 1.47	2.06 ± 0.35
DP (ϵ=1)	20.27	8.01 ± 4.22	2.00 ± 0.76
	GPT2-Medium		
Baseline	8.72	41.56 ± 5.55	6.97 ± 1.46
APNEAP	8.76	27.01 ± 3.34	4.53 ± 0.79
DP (ϵ=8)	14.91	36.13 ± 8.95	6.12 ± 2.17
DP (ϵ=4)	15.90	32.47 ± 6.40	1.04 ± 0.23
DP (ϵ=1)	16.40	9.51 ± 3.17	3.04 ± 1.21
	Qwen3-0.6B		
Baseline	13.48	45.01 ± 8.39	4.26 ± 0.49
APNEAP	14.54	29.26 ± 1.61	2.77 ± 0.38
DP (ϵ=8)	20.63	15.69 ± 0.00	0.57 ± 0.00
DP (ϵ=4)	22.10	14.83 ± 2.72	0.55 ± 0.10
DP (ϵ=1)	24.84	8.78 ± 0.67	0.39 ± 0.08
	GPT2-Large		
Baseline	9.14	40.02 ± 7.62	10.92 ± 0.96
APNEAP	9.63	26.01 ± 4.06	7.10 ± 0.56
DP (ϵ=8)	12.00	42.05 ± 0.78	5.38 ± 0.14
DP (ϵ=4)	12.69	37.06 ± 1.38	2.76 ± 0.25
DP (ϵ=1)	14.97	15.32 ± 0.00	1.97 ± 0.00
	Llama-3.2-1B		
Baseline	8.13	60.12 ± 8.59	2.86 ± 0.16
APNEAP	8.20	45.96 ± 6.80	2.89 ± 1.44
DP (ϵ=8)	15.66	9.76 ± 0.92	0.53 ± 0.11
DP (ϵ=4)	20.99	5.75 ± 1.14	0.35 ± 0.10
DP (ϵ=1)	42.01	1.16 ± 0.92	0.10 ± 0.06
	Qwen3-1.7B		
Baseline	11.89	48.26 ± 6.06	5.19 ± 0.23
APNEAP	12.76	28.96 ± 3.32	3.11 ± 0.23
DP (ϵ=8)	12.15	20.43 ± 15.03	0.78 ± 0.36
DP (ϵ=4)	12.50	17.10 ± 15.03	0.63 ± 0.45
DP (ϵ=1)	13.45	13.00 ± 12.46	0.49 ± 0.43

Table 5: **Impact of DP Fine-tuning**: we use perplexity ("Perpl") for utility, precision ("Prec") and recall ("Rec") for PII leakage, and normalized faithfulness ("Faith"), averaged across all PII types. ↓ (↑) indicates lower (higher values) is preferred. We use Gray for the baseline (no defenses), Green if better, Red if worse, and Orange if within standard deviation of the baseline.

D Influential Circuit Components

In order to understand more about PII leakage within the attention layers of models we displays EAP-IG scored attention layers where those layers score above a 95% threshold across all scored components. We present our analysis in Figure 6.

E Circuit analysis of defenses.

In Figure 7, we present the circuit overlap between models in trained on differing defenses.

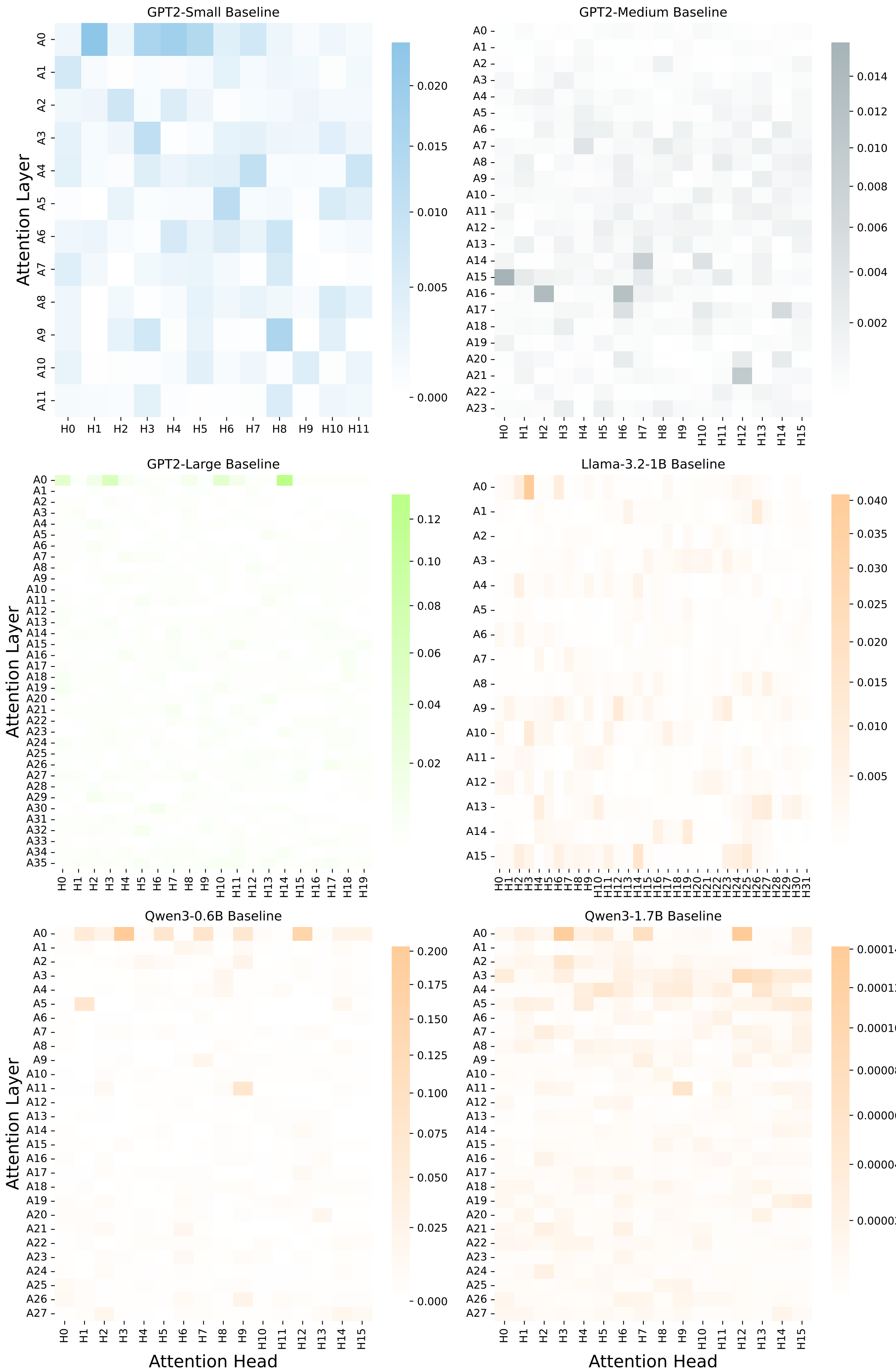

Figure 6: **Influential PII Circuit Components:** Heatmaps show the average EAP-IG scores for each attention head within each layer, across all identified PII circuits.

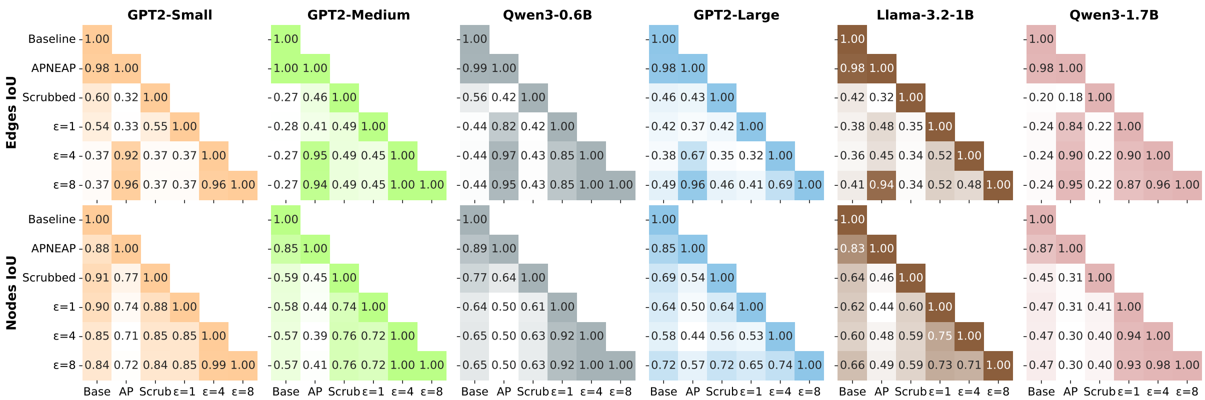

Figure 7: **Circuit overlap across defenses:** percentage overlap of edges (**upper**) and nodes (**lower**) in PII circuits.