Unintended Token-Level Memorization in Language Model Fine-Tuning

Anonymous ACL submission

Abstract

Fine-tuning Large Language Models (LLMs) on sensitive datasets poses a significant risk of unintended memorization and leakage of Personally Identifiable Information (PII), potentially violating privacy regulations and endangering individuals. In this work, we examine how fine-tuning can expose PII that appears only in the inputs, not in the training targets, highlighting a critical and underexplored vulnerability in real-world applications. Using both synthetic and real-world datasets, we design controlled extraction probes to evaluate PII memorization and analyze how factors such as language, domain, task type, and dataset size affect memorization behavior. Additionally, we benchmark four privacy-preserving methods: differential privacy, machine unlearning, regularization, and preference alignment. Our findings show that post-training methods yield more consistent privacy–utility trade-offs, while differential privacy achieves the strongest leakage reduction in specific cases, albeit with training instability.

1 Introduction and Related Work

Large Language Models (LLMs) achieve state-of-the-art performance across many natural language processing tasks, but raise serious privacy concerns due to their vast capacity and data-hungry training regimes. Most notably, their tendency to memorize training samples, even if seen only once during training (Carlini et al., 2021). While some level of memorization can support generalization in long-tailed data distributions (Feldman and Zhang, 2020), verbatim, token-level memorization of Personally Identifiable Information (PII) poses significant privacy risks.

Prior work has extensively studied memorization dynamics in LLMs, in both pre-training and fine-tuning (FT) phases (Morris et al., 2025; Carlini et al., 2021; Hu et al., 2022). For instance, Carlini et al. (2022) analyzed how factors such as model size, data duplication, and prompt length affect memorization during pre-training. Zeng et al. (2024) explored task-specific memorization during FT. However, these works focus on memorizing task-relevant data. In practice, FT often inadvertently uses inputs containing PII, otherwise unrelated to the task output (e.g., names or medical records). Despite growing attention to LLM privacy, the specific risk of unintended PII memorization, where PII appears only in inputs and is irrelevant to the downstream task, remains underexplored.

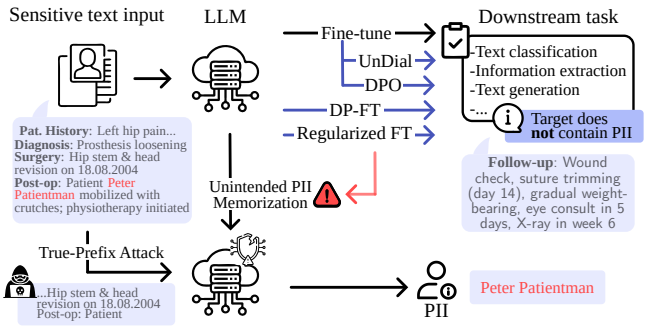

Figure 1: Overview of our experiment setup depicting the unintended PII memorization scenario, our attack, and fine-tuning approaches.

Recent work has highlighted the threat of PII leakage under limited-access conditions via black-box probing techniques, and adversarial API querying, reflecting realistic deployment scenarios (Nakka et al., 2024b,a; Lukas et al., 2023). Yet, none systematically isolate unintended PII memorization or compare mitigation strategies. In this work, we conduct a novel, comprehensive study of unintended PII memorization in fine-tuned LLMs. Specifically, we:

- Define and formalize the problem of unintended PII memorization, distinguishing it from general memorization or task-relevant PII usage;
- Quantify memorization using synthetic and real-world datasets in a realistic deployment scenario,

also analyzing key influencing factors;

- Benchmark four common mitigation strategies, assessing their privacy-utility tradeoffs: differential privacy, regularization, machine unlearning, and preference alignment.

To the best of our knowledge, this is the first comprehensive study focused on unintended PII memorization in LLM fine-tuning, marking an important step toward practical privacy-preserving model deployment.

2 Methodology

2.1 Unintended PII Memorization

We define unintended PII memorization as the phenomenon where a language model fine-tuned on sensitive text data like electronic health records (EHRs) internalizes PII that is not part of the model's intended output (unrelated to the downstream task). This is distinct from memorization during pre-training, where large corpora might contain public or semi-public PII, and from targeted FT tasks, where PII is intentionally part of the model's output space.

Our work focuses on downstream tasks (classification, information extraction, medical follow-up planning) where PII appears only in inputs, not training targets. We adopt a realistic black-box threat model where adversaries access the model only via input-output queries (e.g., API calls). We assume a worst-case scenario where attackers have partial access to the FT dataset (e.g. anonymized EHRs) and can craft adversarial prompts accordingly (Carlini et al., 2022; Nakka et al., 2024b).

True-Prefix Attack (TPA) is a method to probe memorization in autoregressive LLMs (Carlini et al., 2021). Given a true prefix c from the FT data immediately preceding a PII span s of N tokens, we say s is extractable if

$$s \leftarrow \underset{s' : |s'|=N}{\arg\max}\, f_\theta(s' \mid c). \tag{1}$$

where $f_\theta(s' \mid c)$ is the model's conditional probability. We also experiment with an *enhanced* TPA variant, which adds the first character of the PII to the prefix. With labeled PII spans, this attack is straightforward to construct and evaluate, providing an effective measure of model memorization.

2.2 Mitigating memorization

We evaluate four prevalent training strategies that aim to reduce PII memorization during or after FT.

Differential Privacy (DP) is a widely used technique for protecting individual data privacy with mathematical guarantees (Kulynych et al., 2025; Dwork, 2006). It introduces noise into the gradient updates and limits individual sample influence, thus bounding sample-level memorization risk. DP has been extensively applied to both LLM pre-training and fine-tuning, providing verifiable guarantees, but at the cost of utility degradation and increased training complexity (Hoory et al., 2021; Li et al., 2021; Yu et al., 2021).

Machine Unlearning: UnDial (Dong et al., 2024) performs targeted unlearning using self-distillation. It constructs a fixed target distribution by lowering the logits associated with tokens to be forgotten. The distillation loss nudges the model away from the sensitive content without inducing catastrophic forgetting, an issue common in earlier approaches like Gradient Ascent or Negative Preference Optimization (Fan et al., 2025; Shi et al., 2024).

Regularization Inspired by UnDial (Dong et al., 2024), we propose a regularization-based variant that integrates self-distillation into the FT loop. Specifically, we alternate between cross-entropy loss and a regularization loss focused on PII tokens. This focused UnDial loss is applied only on selected sensitive spans to discourage memorization.

Direct Preference Optimization (DPO) emerges as a computationally and data-efficient alternative to RLHF for aligning models' outputs with human preferences like privacy or helpfulness (Rafailov et al., 2023; Szep et al., 2024). We adapt DPO to discourage PII leakage by treating original training examples containing PII as *rejected* and their masked-PII counterparts as *preferred*.

3 Experiments

3.1 Datasets

We use three datasets varying significantly in nature, task complexity, and objectives. The latter two are private medical datasets from German EHRs at the *anonymized Institution*. For details on data and preprocessing, see Appendix B.

GretelAI-Financial (Watson et al., 2024) is a synthetic, multilingual NER dataset focused on PII. After preprocessing, it contains ~30k samples in 7 languages with 52 financial text classification labels, which we use for the downstream task.

Pathology reports contain 2553 German documents with rich medical terminology and complex

tumor-related information, like intervention type, tumor dignity, entity, location, subentity, etc. We fine-tune for information extraction formulated as a 5-dimensional task in a JSON schema.

Discharge Summary (DS) contains 26306 German documents with sections focusing on anamnesis, diagnosis, surgery, treatment, etc. We leverage the final section for a medical follow-up planning generation task. PII are annotated using an LLM-based pipeline (§ B.4).

3.2 Privacy-preserving training

We quantify the PII memorization during vanilla fine-tuning and benchmark different privacy-preserving training methods. Further training details can be found in Appendix C.

Supervised Fine-Tuning We establish memorization baselines by fine-tuning Llama 3.2 1B models (Grattafiori et al., 2024) using QLoRA ($r = 8$) in all linear layers over $10-25$ epochs (varying per dataset), until triggering early stopping. A cosine learning rate scheduler with linear warmup of 3% of steps is used. Hyperparameters are optimized only for downstream performance, without privacy considerations.

Differential Privacy Fine-Tuning We integrate (ε, δ)-DP into the QLoRA setup via Opacus' Privacy Engine (Yousefpour et al., 2022), using privacy budgets $\varepsilon \in \{2, 8\}$ and $\delta = 10^{-5}$. Hyperparameter choices follow Li et al. (2021) to maximize utility under DP constraints.

UnDial We apply UnDial to a disjoint subset of $17692\,(40\%)$, $1500\,(40\%)$, and $6000\,(20\%)$ person names in the GretelAI, Pathology, and DS datasets respectively; none of which overlap with the names extracted during our memorization assessment (see § 3.3). We use the same input-output structure for unlearning as for the TPA, with the PII being the unlearning target.

Regularization We apply regularization using focused UnDial to compute the regularization loss only over the PII tokens, using the same PII subset as for unlearning.

DPO Following FT, we run DPO with a uniform system prompt instructing the model to withhold all PII. For each corpus, we slide a 150-token context window over sequences containing at least two PII within the following 20 tokens. We mask that 20-token span to form the preferred response and use the original, unmasked text as the rejected response. The resulting datasets contain 1489 (GretelAI) and 5636 (DS) training samples.

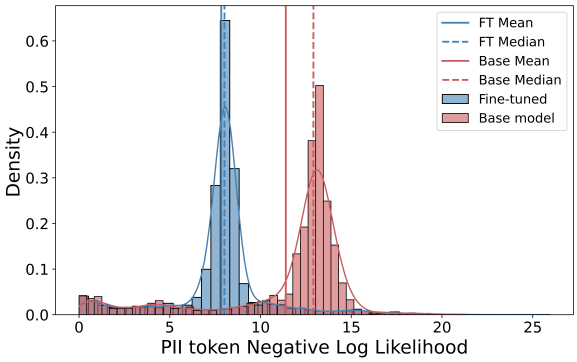

Figure 2: Distribution of per-token log-likelihoods for ground-truth PII completions.

3.3 Evaluation

To ensure reproducibility and comparability, all evaluations are conducted with greedy decoding. For TPA, generation is capped at 25 tokens following the prefix (50 tokens). For the medical datasets, we also evaluate by comparing generations to all PII of the same kind in the dataset (instead of the ground-truth). We call this *cross-memorization*.

We use task-specific evaluation metrics: accuracy for GretelAI, F1-score for Pathology, and BERTScore-F1 for DS. For additional details about memorization and downstream task evaluation, we refer the reader to Appendix D.

4 Results

Fine-tuned models are more confident in predicting PII tokens. Figure 2 shows the density of per-token negative log-likelihoods for the FT and base models over the same PII (names) in the DS dataset. The fine-tuned model's distribution mode is shifted substantially closer to zero and has significantly smaller variance compared to the base model. This indicates that FT has increased the model's confidence across PII tokens.

When does unintended token-level memorization happen? Table 1 shows that fine-tuning significantly increases PII leakage in the GretelAI and DS datasets, while the effect is much weaker in the Pathology dataset. This may be attributed to its smaller **size** ($10\times$ smaller), **sparser** PII distribution, and the constrained JSON **output format**, which limits free-text generation. While models show high PII memorization on English

	Model	Task Performance↑	Regular TPA Total PII↓	Regular TPA Distinct PII↓	Enhanced TPA Total PII↓	Enhanced TPA Distinct PII↓	LR	Effective Batch Size
GretelAI - Financial	Base	12.08%	3402	1758	-	-	-	-
	SFT	87.17%	3601	1720	-	-	2e-5	8
	DP-ϵ2	66.16%	3304	1654	-	-	2e-4	256
	DP-ϵ6	74.84%	3563	1767	-	-	1e-3	2048
	UnDial-40%	76.21%	2717	1323	-	-	1e-5	16
	Reg-40%	**81.12%**	3297	1534	-	-	1e-5	16
	DPO-β0.01	79.24%	**2616**	**1167**	-	-	3e-6	32
Pathology	Base	28.89%	0 (6)	0 (4)	0 (7)	0 (6)	-	-
	SFT	86.21%	0 (11)	0 (7)	0 (10)	0 (8)	5e-5	96
	DP-ϵ6	55.13%	0 (9)	0 (6)	0 (7)	0 (7)	2e-4	512
	UnDial-40%	**71.89%**	**0 (6)**	**0 (5)**	**0 (7)**	**0 (5)**	1e-5	16
Discharge Summary	Base	0.5227	1 (1940)	1 (719)	25 (9638)	15 (3974)	-	-
	SFT	0.7147	1 (1604)	1 (1334)	91 (11754)	15 (4453)	2e-4	128
	DP-ϵ2	0.6906	0 (1143)	0 (733)	43 (17405)	16 (5994)	1e-3	1024
	DP-ϵ6	0.6993	**0 (161)**	**0 (154)**	**30 (5624)**	**11 (1589)**	1e-3	1024
	UnDial-20%	0.6725	1 (1587)	1 (1103)	31 (9456)	13 (3593)	1e-5	16
	Reg-20%	0.6770	2006 (5388)	17 (2227)	6841 (17102)	142 (6601)	1e-5	16
	DPO-β0.01	**0.7084**	1 (1163)	1 (1009)	31 (6298)	13 (2860)	1e-7	32

Table 1: Comparison of PII memorization and task performance across methods and datasets. More details on task performance and memorization evaluation can be found in § 3.3. Cross-memorization results are in parentheses.

(e.g., GretelAI), results vary across languages (Figure 3). The specialized German medical language in Pathology likely contributes to lower extraction rates, suggesting that both **language** and **domain specificity** influence memorization.

Additionally, GretelAI's high baseline leakage indicates that models retain strong pre-training priors, amplifying memorization when new inputs contain familiar PII tokens. Other contributing factors may include task type, PII repetition, model capacity, and TPA prefix length (Appendix A).

Post-training methods offer robustness, but DP can outperform in specific cases. Across datasets (Table 1), post-training mitigation methods such as DPO and UnDial generally yield more consistent privacy–utility trade-offs and are more robust to hyperparameter variation. They are also less resource-intensive than preventive techniques like DP and regularization. Differential privacy (DP), however, shows strong privacy potential in specific scenarios. In the DS task, it reduces cross-memorization by over 60%, the highest among all methods, even without using seed PII data. Yet, DP remains unstable to train, often requiring larger batch sizes, higher learning rates, and longer training, with results varying substantially across runs. We also observe that DP models occasionally produce repetitive outputs under TPA, indicating possible degradation in generation quality. Regularization suffers from conflicting training objectives, preserving task performance but retaining more PII. Unlearning and alignment methods are sensitive to the quality and size of the seed set, requiring careful tuning to balance effectiveness and utility. Overall, while DP can outperform in isolated cases, post-training methods offer more stable and reproducible results. Crucially, even the most effective methods achieve only around a 30% reduction in direct PII memorization, indicating substantial room for improvement.

5 Discussion

This work provides a systematic analysis of unintended PII memorization in fine-tuned language models. We identify key influencing factors and evaluate four mitigation strategies with varying trade-offs in privacy, utility, and stability. Fine-tuning on small, domain-specific datasets may lessen memorization but does not remove the risk. Post-training methods such as DPO and UnDial generally offer more consistent privacy–utility trade-offs. However, DP achieves the strongest leakage reduction in specific cases, despite being unstable and sensitive to hyperparameters, with occasional output degradation. Unintended memorization remains a persistent challenge, and even the best methods yield only moderate improvements. This highlights the need for further research into scalable, robust, and practical privacy-preserving fine-tuning techniques.

Limitations

Our study focuses on PEFT (QLoRA) of 1B-parameter LLaMA models. Exploring larger models, deeper architectures, other PEFT techniques, non-quantized models, or full-model FT may reveal different memorization dynamics and mitigation behaviors.

Another limitation of our study is dataset availability and label quality. Public, high-quality PII-annotated corpora are scarce, so we rely on (1) synthetic multilingual financial data, (2) a small, manually annotated private dataset, and (3) a larger private corpus with PII spans identified via a semi-automated local LLM pipeline. Despite this diversity, synthetic data and semi-automated labeling introduce noise, and none of the datasets are structured at the individual (e.g., per-patient) level. This restricts our ability to explore privacy-preserving approaches such as federated learning or user-level differential privacy.

Finally, we do not evaluate robustness against adversarial extraction, such as jailbreak prompts, instruction-based attacks, or white-box gradient leaks, which could undermine DP, UnDial, or DPO defenses. Systematic red-teaming and adversarial threat modeling remain important avenues for future validation.

Ethics Statement

This study involves the analysis of fine-tuned language models on datasets containing annotated PII spans to evaluate memorization risks. All data handling and annotation procedures were conducted in compliance with applicable data protection regulations and were approved by the institutional ethics review board. Sensitive datasets were processed exclusively on secure, in-house infrastructure using local models, ensuring that no data left the organization or was exposed to third-party services. Appropriate safeguards were implemented throughout to protect individual privacy and maintain data confidentiality.

References

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying Memorization Across Neural Language Models. In *The Eleventh International Conference on Learning Representations*.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting Training Data from Large Language Models. In *30th USENIX Security Symposium*, pages 2633–2650.

Yijiang River Dong, Hongzhou Lin, Mikhail Belkin, Ramon Huerta, and Ivan Vulić. 2024. UNDIAL: Self-Distillation with Adjusted Logits for Robust Unlearning in Large Language Models. *arXiv preprint*. ArXiv:2402.10052 [cs].

Cynthia Dwork. 2006. Differential Privacy. In *Automata, Languages and Programming*, pages 1–12, Berlin, Heidelberg. Springer.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. 2025. Simplicity Prevails: Rethinking Negative Preference Optimization for LLM Unlearning. *arXiv preprint*. ArXiv:2410.07163 [cs].

Vitaly Feldman and Chiyuan Zhang. 2020. What Neural Networks Memorize and Why: Discovering the Long Tail via Influence Estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 Herd of Models. *arXiv preprint*. ArXiv:2407.21783 [cs].

Shlomo Hoory, Amir Feder, Avichai Tendler, Sofia Erell, Alon Peled-Cohen, Itay Laish, Hootan Nakhost, Uri Stemmer, Ayelet Benjamini, Avinatan Hassidim, and Yossi Matias. 2021. Learning and Evaluating a Differentially Private Pre-trained Language Model. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1178–1189, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S. Yu, and Xuyun Zhang. 2022. Membership Inference Attacks on Machine Learning: A Survey. *ACM Comput. Surv.*, 54(11s):235:1–235:37.

Bogdan Kulynych, Juan Felipe Gomez, Georgios Kaissis, Jamie Hayes, Borja Balle, Flavio du Pin Calmon, and Jean Louis Raisaro. 2025. Unifying Re-Identification, Attribute Inference, and Data Reconstruction Risks in Differential Privacy. *arXiv preprint*. ArXiv:2507.06969 [cs].

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. 2021. Large Language Models Can Be Strong Differentially Private Learners. 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. ISSN: 2375-1207.

Fatemehsadat Mireshghallah, Archit Uniyal, Tianhao Wang, David Evans, and Taylor Berg-Kirkpatrick. 2022. Memorization in NLP Fine-tuning Methods. *arXiv preprint*. ArXiv:2205.12506 [cs].

John X. Morris, Chawin Sitawarin, Chuan Guo, Narine Kokhlikyan, G. Edward Suh, Alexander M. Rush, Kamalika Chaudhuri, and Saeed Mahloujifar. 2025. How much do language models memorize? *arXiv preprint*. ArXiv:2505.24832 [cs].

Krishna Kanth Nakka, Ahmed Frikha, Ricardo Mendes, Xue Jiang, and Xuebing Zhou. 2024a. 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.

Krishna Kanth Nakka, Ahmed Frikha, Ricardo Mendes, Xue Jiang, and Xuebing Zhou. 2024b. PII-Scope: A Comprehensive Study on Training Data PII Extraction Attacks in LLMs. *arXiv preprint*. ArXiv:2410.06704 [cs].

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. 2024. MUSE: Machine Unlearning Six-Way Evaluation for Language Models. In *International Conference on Learning Representations*.

Marton Szep, Daniel Rueckert, Rüdiger von Eisenhart-Rothe, and Florian Hinterwimmer. 2024. A Practical Guide to Fine-tuning Language Models with Limited Data. *arXiv preprint*. ArXiv:2411.09539 [cs].

Alex Watson, Yev Meyer, Maarten Van Segbroeck, Matthew Grossman, Sami Torbey, Piotr Mlocek, and Johnny Greco. 2024. Synthetic-PII-Financial-Documents-North-America: A synthetic dataset for training language models to label and detect pii in domain specific formats.

Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. 2022. Opacus: User-Friendly Differential Privacy Library in PyTorch. *arXiv preprint*. ArXiv:2109.12298 [cs].

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. 2021. Differentially Private Fine-tuning of Language Models. In *International Conference on Learning Representations*.

Shenglai Zeng, Yaxin Li, Jie Ren, Yiding Liu, Han Xu, Pengfei He, Yue Xing, Shuaiqiang Wang, Jiliang Tang, and Dawei Yin. 2024. Exploring Memorization in Fine-tuned Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3917–3948, Bangkok, Thailand. Association for Computational Linguistics.

A Additional Findings

Why does DP not (always) prevent PII leakage? Differential privacy protects against singling out individual records or users. It implicitly assigns a privacy cost to using information in the training dataset at the level of records, not tokens, hence it is oblivious to different occurrences of the same information across records or users. This is an effective method to mitigate risks of disclosing *by whom* data was contributed, but it does not take into account *about whom* the content is (Lukas et al., 2023).

Effect of token length in TPA. Following Carlini et al. (2022), we evaluate attack success using varying prefix lengths $\in \{10, 25, 50, 100\}$ tokens preceding the target PII (Figure 4). Consistent with their findings, we observe a sharp increase in effectiveness between 25 and 50 tokens, with only marginal gains beyond 50. Based on this, we standardize a 50-token prefix for most evaluations.

Figure 4 shows a logarithmic increase in attack success with prefix length for fine-tuned models. The pre-trained model shows a similar trend on GretelAI, but on the DS dataset, success decreases with longer prefixes. While attacks benefit from prefixes up to 200 tokens on GretelAI, they plateau between 50–100 tokens on DS. These patterns align partially with Carlini et al. (2022) but suggest possible dataset-specific trends in unintended PII memorization.

Comparison of different data-centric attack methods Nakka et al. (2024b) benchmark Template, In-Context Learning (ICL), and PII-Compass attacks alongside TPA. Template attacks use adversarial prompts to query target PII, ICL augments these prompts with examples of known PII in the same format, and PII-Compass combines TPA with Template by adding another true PII prefix. Parallel to TPA, we attempted to construct templates for the

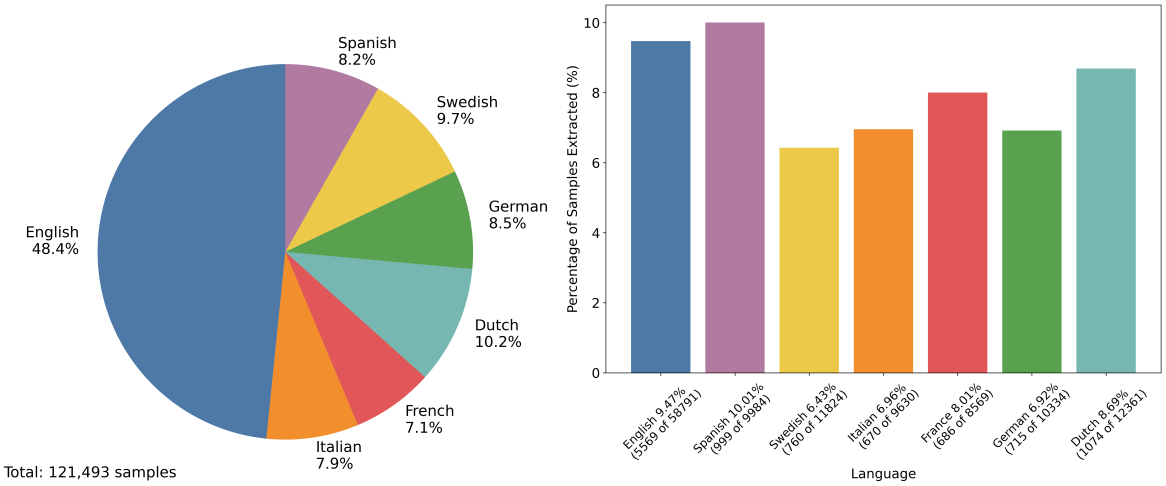

Figure 3: Distribution of PII across languages in the GretelAI dataset training split (left). PII extraction success ratio across languages (right).

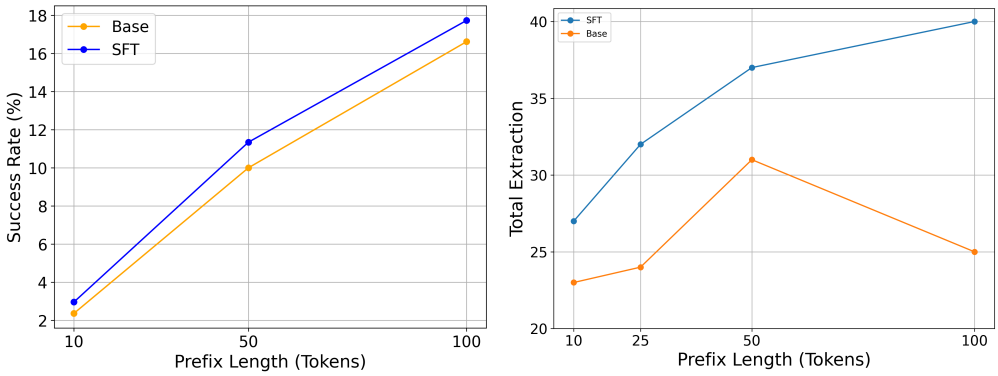

Figure 4: PII extraction success vs. prefix length (in tokens) for LLaMA-3.2-1B base and fine-tuned models. Left: GretelAI dataset (extraction rate of all training PII). Right: Discharge summaries dataset (approximate count of PII extracted).

Template, ICL, and PII-Compass attacks in an automated manner by extracting co-occurring PII pairs (e.g., a "name" span followed within N characters by another PII type). Table 2 summarizes the best data extraction attempts using the FT model on the GretelAI dataset.

Method	4 examples	8 examples	16 examples	24 examples
ICL Attack	0.686%	1.521%	1.184%	1.101%
ICL Attack - 2	0.692%	1.120%	1.322%	1.104%
	Prefix 1	**Prefix 2**	**Prefix 3**	**Prefix 4**
PII-Compass	0.843%	0.885%	0.843%	0.311%

Table 2: Comparison of the amount of trainable parameters in LLaMa-3.2-1B fine-tuned models and their effect on total memorization and unique PII memorization on the GretelAI dataset.

Similar to (Nakka et al., 2024a), we found that the effectiveness of these attacks is highly sensitive to template design, example selection, and the PII targets. Our initial runs showed TPA memorization rising from 10.0% (pre-trained) to 11.3% (fine-tuned), while the template-based attacks yielded near-zero recall on both models, likely due to high variance in PII associations and low-quality synthetic templates. We conducted additional evaluations of these data extraction attacks using more realistic PII and manually selected templates from the Enron dataset, but their effectiveness was still far from TPA.

Unintended PII memorization scales with model capacity. As shown in Table 3, an eight-fold increase in LoRA rank does **not** increase the **total** number of PII extractions, which remains equal under both query numbers per prefix setting. However, the number of **distinct PII** increases under both settings. This reveals that while additional parameters do not cause higher total memorization, they broaden the memorization of distinct PII.

LoRA Rank	Trainable Parameters	Total PII		Distinct PII	
		32 Query	1 Query	32 Query	1 Query
8	5.6M	849	40	165	17
64	45.1M	849	40	199	23

Table 3: Comparison of the amount of trainable parameters in LLaMa-3.2-1B fine-tuned models and their effect on total memorization and unique PII memorization for the Discharge summaries.

These findings mirror the findings of Mireshghallah et al. (2022), where they attribute memorization to the location of the trainable parameters in the model, and not the total quantity (i.e., header-tuning only causing more memorization than full FT). In our case, by scaling the LoRA rank, we still fine-tune all linear layers in the model, but increase the number of trainable parameters.

Unintended PII memorization does not directly correlate with PII repetitions. Contrary to Carlini et al. (2022), we did not find a direct correlation between the frequency of PII occurrences in the data and unintended memorization rates. We hypothesize that memorization is influenced more by the PII's textual context and its utility to the downstream task than by raw repetition. For example, in our DS dataset, only the model fine-tuned with a learning rate an order of magnitude larger reproduced PII contained in the document headers, which were some of the most frequent PII in the dataset corpus (and most irrelevant to the downstream task).

Effect of downstream task on PII memorization. Previous research has shown that the nature of the target downstream task can affect general sequences memorization Zeng et al. (2024). Fully fine-tuned LLMs tend to memorize more training sequences on generative tasks, such as summarization or chat/conversational tasks, than when fine-tuned for discriminative tasks, e.g., classification or question-answering.

However, our experimental findings reveal that this pattern does not necessarily extend to unintended PII memorization (other factors could be just as important). Summarizing our memorization results presented previously, after FT, models memorized significantly more PII in the GretelAI (and GretelAI+Enron) dataset, followed by the Discharge summaries dataset, and did not memorize any PII from the training data in the Pathology dataset. The tasks of these datasets correspond to document classification, text generation/summarization, and information extraction/classification, respectively.

Although we find that the nature of the FT task does not have a direct impact on unintended memorization, a closer qualitative analysis at the fine-tuned model's outputs suggests that the output format of the task might influence memorization, or at least, mitigate the effectiveness of the different data extraction attacks.

- **Patho Dataset:** The FT model often emits JSON-formatted responses even under TPA, Q&A, or translation instructions prompts, indicating that the rigid output schema learned during FT constrains free-form PII generation.
- **Discharge Summary Dataset:** Because PII tokens are masked in the training targets, the FT model increasingly produces masked placeholders post-tuning (1788 masked tokens $\rightarrow$ 4507 masked tokens), partially reducing direct PII exposures.

These observations imply that output alignment (i.e. training the model to emit structured or masked formats) can mitigate unintended PII leakage similar to our DPO setup. Designing FT objectives that enforce strict output schemas may thus serve as an additional privacy safeguard when possible.

Other PII Types We report additional results on all PII types in both the GretelAI dataset (Table 4 and Table 5) and the Pathology dataset (Table 6 and Table 7). The reduced PII leakage in the Pathology dataset highlights the assumptions about the memorization dynamics on this dataset made in § 4.

Model	Total PII	Distinct PII	Performance
Base	9589	5162	12.08%
SFT	10473	5302	87.17%
DP-ϵ2	9448	4904	66.16%
DP-ϵ6	10001	4846	74.84%
UnDial-40	9889	4969	76.53%
DPO-β0.01	**7666**	**3947**	**79.24%**

Table 4: TPA results in the GretelAI dataset including all PII types.

Sampling reveals more memorized PII For this experiments, we repeat the TPA, sampling the models 32-times per prefix, setting the model temperature to 1. Our results in Table 8 show an increase in succesfully extracted PII.

Model	Name	Comp.	Email	Add.	Other
Base	3402	3092	2176	577	342
SFT	3601	2892	2695	990	295
DP-ϵ2	3304	2853	2281	706	304
DP-ϵ6	3563	3030	2332	739	337
UnDial-40	4031	3007	2088	457	306
DPO-β0.01	**2616**	**2024**	**2586**	**295**	**145**

Table 5: TPA results in the GretelAI dataset itemized by PII types: Name, Company, Email, Address, Others.

Model	Total PII	Distinct PII	Performance
Base	265	17	28.89%
SFT	130	21	86.21%
DP-ϵ6	269	19	55.13%

Table 6: Cross-Memorization results of models in the Pathology dataset including all PII types except dates.

Model	Name	Serial Nr.	Location	Contact Info
Base	196	37	25	7
SFT	81	31	8	10
DP-ϵ6	172	70	18	9

Table 7: Cross-Memorization results of models in the Pathology dataset itemized by PII types: Name, Serial Number, Location and Contact Information.

Model	LR	1 Query		32 Queries	
		Total	Distinct	Total	Distinct
Base	-	25	15	815	205
SFT	5e-5	40	17	849	165
SFT	2e-4	91	15	1980	126

Table 8: Comparison of results of the enhanced TPA, using 1 and 32 queries per True Prefix, on the DS dataset.

B Data Preprocessing

B.1 GretelAI - Text classification

During preprocessing, we excluded from GretelAI-Financial[1] eight classes with trivial classification due to rigid text structure: *CSV, EDI, SWIFT Message, FIX Protocol, BAI Format, XBRL, FpML,* and *MT940*. We also removed documents with quality scores below 90/100. The resulting class distribution is shown in Figure 5.

Upon inspection, we identified AI-generated documents and filtered them using heuristic rules.The final dataset comprised 27,636 training and 3,136 test samples (from 50,346 and 5,594 originally).

PII counts revealed 121,493 total spans with 51,206 unique entities after filtering spans shorter than three characters, and PII of classes not valuable. However, the dataset sometimes exhibits limited PII diversity, containing poor quality repetitive synthetic values like "John Doe" or "janedoe@mail.com".

B.2 Enron - Text classification

To enhance PII realism in our classification task, we created a hybrid dataset by combining samples from GretelAI and the Enron Email Corpus. The Enron dataset provides authentic email communications with naturally occurring PII patterns, including real name-email pairs and diverse email addresses, formatting conventions, and domains.

We randomly sampled 400 emails from Enron (ensuring at least 400 unique name-email pairs) and replaced all "EMAIL" class documents in our cleaned GretelAI dataset. This maintained the original class distribution across document types.

B.3 Pathology reports - Information extraction

For this dataset, we manually annotated PII (serial number, person name, contact info, date, and location) and bone tumor-related information with the help of medical professionals, including dignity (benign/malignant), intervention type (resection/biopsy/curettage), entity, subentity, and location. We filter out poor quality reports (very short or very limited tumor-related information; not bone tumor related) and perform extensive preprocessing (removing duplicates, text normalization, creating a labeling UI, and performing annotation consistency checks). This resulted in a structured format with pre-annotated PII spans and task labels. We split the 2,552 samples using an 80%/10%/10% train/validation/test distribution, yielding 2,041 training, 255 validation, and 256 test samples.

B.4 Discharge summaries - Medical Follow-up Planning

We applied a consistent cleaning pipeline to raw documents to eliminate formatting artifacts and ensure experimental stability. This included normalizing control characters (replacing line breaks, tabs, and non-printable symbols with spaces), collapsing consecutive spaces, trimming whitespace, and removing common report headers using regular expressions to isolate the free-text body of each summary.

[1] https://huggingface.co/datasets/gretelai/synthetic_pii_finance_multilingual

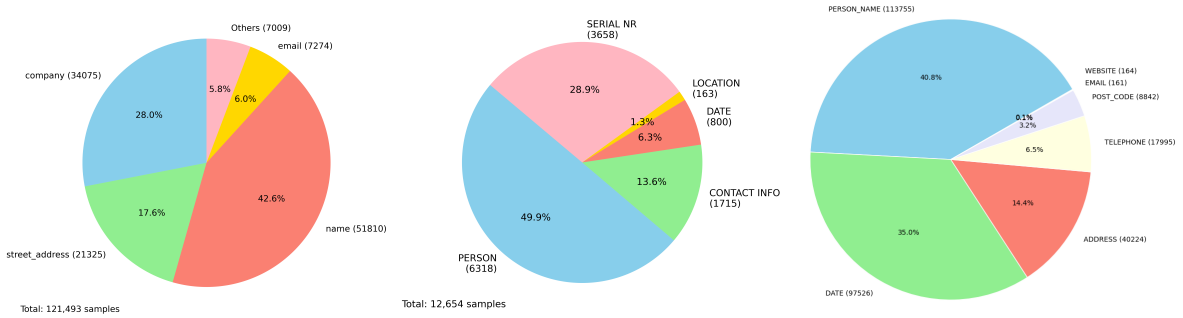

Figure 5: Distribution of PII types for the GretelAI synthetic dataset (left), the Pathology dataset (middle), and the Discharge summary dataset (right).

PII Span Labeling. Using Ollama[2], we ran LLaMa 4 Scout Q4 with structured generation to identify PII classes: PERSON_NAME, POST_CODE, ADDRESS, DATE, EMAIL, TELEPHONE, and WEBSITE. To remove false positives from model hallucinations, we used LLaMa 4 Scout to review and tag potential false positives, followed by manual review.

After extracting PII spans, we performed localization to map each instance to its exact character offsets in the original text, enabling reliable masking and targeted extraction attacks.

Target Extraction We extracted the *Procedere* section from each document, typically appearing near the end and introduced by phrases like "Procedere:" or "als weiteres Procedere...". Samples with sections under 50 characters or over 2,000 characters were discarded.

Since *Procedere* targets may contain PII, we masked all detected PII to preserve the "unintended" nature of our memorization study and prevent direct training on sensitive information.

The resulting dataset consists of 26,306 samples, split into 80%-10%-10% train-validation-test splits, each with a generation target and annotated PII spans.

C Training details

All our experiments have been run on an NVIDIA A100 80GB GPU. Fine-tuning took at most 24 hours, while attacks took at most 12 hours.

C.1 Fine-tuning

We use HuggingFace's (HF) `SFTTrainer`[3], a high-level wrapper around the HF Trainer API, which simplifies the FT process by managing the training loop, loss computation, and optimizer updates. We monitor overfitting and guide early stopping on the validation set, using a patience of 3 validation checks. The frequency of validation is adjusted based on the total number of epochs and specific experimental configurations, as well as the dataset specification. For optimization, we use the `paged_adamw_32bit` optimizer, a memory-efficient variant of AdamW that supports paged memory loading and uses 32-bit precision for optimizer states. Our default FT hyperparameters are LoRA rank $r = 8$ (1.5M trainable parameters for the 1B model), scaling factor: $\alpha = 16$, dropout rate of 0.05, and a learning rate of 1×10^{-5}, with a linear warmup over 3% of training steps followed by cosine decay.

C.2 Differential Privacy

Our differential privacy experiments aim to match the downstream task performance of SFT models for fair PII memorization comparison. While training for more epochs with a fixed privacy budget spreads the privacy budget across additional steps (reducing the signal-to-noise ratio per update), we found better results by increasing learning rate and batch size instead, following recommendations from Li et al. (2021).

C.3 UnDial

We apply UnDial using Dong et al. (2024)'s implementation[4] (with minimal updates for Hugging Face Trainer compatibility). Following the original authors' guidelines, we began with conservative hyperparameters: learning rate of 10^{-6} and unlearning strength of 3, the default selection in their repository. However, our experiments revealed

[2] https://github.com/ollama/ollama

[3] https://huggingface.co/docs/trl/en/sft_trainer

[4] https://github.com/dong-river/LLM_unlearning

that moderately higher values achieved superior privacy-utility tradeoffs. Specifically:

- **Optimal configuration**: Learning rate $\in [1, 5] \times 10^{-5}$ (one order of magnitude higher than recommended) with unlearning strengths of 5-7
- **Performance preservation**: UnDial maintained >95% of original accuracy (compared to >12% degradation with DP-FT)
- **Memorization reduction**: Using 20% of total PII for unlearning reduced extractable distinct PII from 13.44% to 12.65%
- **Sequence length optimization**: 50-token contexts proved most effective, balancing sufficient context with computational efficiency

Importantly, we found that aggressive hyperparameters (learning rates larger than 10^{-4} and unlearning strength larger than 7) led to substantial performance drops without additional privacy benefits, highlighting the need for careful tuning.

C.4 DPO

We use HF's DPOTrainer [5]. A careful balance of learning rate and β is required to prevent catastrophic forgetting and maintain the model's utility while achieving the desired alignment goal. While a common value for β is 0.1 and learning rate one order of magnitude lower than the SFT learning rate, our empirical results revealed that a more aggressive $\beta = 0.01$ was required for achieving appropriate PII masking. Simultaneously, we found that learning rates $\geq 5e-6$ resulted in excessive token masking, causing catastrophic forgetting.

D Evaluation details

D.1 Memorization Assessment

Prior work has focused on exact matching, but PII memorization requires considering approximate matches due to the sensitive nature of content and its variability. For instance, abbreviations, formatting inconsistencies, or incomplete PII exposure can also be a privacy risk.

To address these challenges, we define two evaluation strategies depending on the dataset (and multiple criteria within the dataset) and type of PII under evaluation:

1. **Exact-Match (EM) Evaluation** For datasets where PII quality is lower and highly uniform (for instance, Gretel-AI's dataset), we consider a PII span memorized only if the model's normalized output contains an exact substring match of the target PII.

2. **Approximate-Match Evaluation** For real-PII datasets (Pathologie, Discharge Summary), we adopt fuzzy string matching via the Levenshtein distance using the `thefuzz` library[6] based on the PII type. We set a similarity threshold (e.g., 90%) so that minor variations, such as abbreviations, missing components, or misspellings, still count as memorization. With names, addresses, and similar types, we can apply this approach. However, for phone numbers, postcodes, or other numeric-only PII, we only apply normalization by removing all non-numeric characters. Finally, for other PII, such as email addresses or websites, where EM is important, we use EM.

By combining an upper-bound TPA with both exact and approximate matching criteria, we obtain a robust, worst-case estimate of PII memorization across our experimental settings.

[5] https://huggingface.co/docs/trl/main/en/dpo_trainer

[6] https://github.com/seatgeek/thefuzz