PRvL: Quantifying the Capabilities and Risks of Large Language Models for PII Redaction

Leon Garza*, Anantaa Kotal*, Aritran Piplai*, Lavanya Elluri†,
Prajit Kumar Das‡, Aman Chadha§
* Dept. of C.S., The University of Texas at El Paso, {lgarza3, akotal, apiplai}@utep.edu
†Texas A&M University-Central Texas, elluri@tamuct.edu
‡Cisco Systems Inc., prajdas@cisco.com
§Amazon Web Services, hi@aman.ai

***Abstract*—Redacting Personally Identifiable Information (PII) from unstructured text is critical for ensuring data privacy in regulated domains. While earlier approaches have relied on rule-based systems and domain-specific NER models, these methods fail to generalize across formats and contexts. Recent advances in Large Language Models (LLMs) offer a promising alternative, yet the effect of architectural and training choices on redaction performance remains underexplored. LLMs have demonstrated strong performance in tasks that require contextual language understanding, including the redaction of PII in free-form text. Prior work suggests that with appropriate adaptation, LLMs can become effective contextual privacy learners. However, the consequences of architectural and training choices for PII Redaction remain underexplored. In this work, we present a comprehensive analysis of LLMs as privacy-preserving PII Redaction systems. We evaluate a range of Large Language Model (LLM) architectures and training strategies for their effectiveness in PII Redaction. Our analysis measures redaction performance, semantic preservation, and PII leakage, and compares these outcomes against latency and computational cost. The results provide practical guidance for configuring LLM-based redactors that are accurate, efficient, and privacy-aware. To support reproducibility and real-world deployment, we release PRvL, an open-source suite of fine-tuned models, and evaluation tools for general-purpose PII Redaction. PRvL is built entirely on open-source LLMs and supports multiple inference settings for flexibility and compliance. It is designed to be easily customized for different domains and fully operable within secure, self-managed environments. This enables data owners to perform redactions without relying on third-party services or exposing sensitive content beyond their own infrastructure.**

***Index Terms*—Personally Identifiable Information, PII Redaction, Large Language Models, Retrieval Augmented Generation**

I. Introduction

PII refers to any data that may be used to directly or indirectly identify an individual, such as names, addresses, social security numbers, phone numbers, and financial and health records. As digital systems manage increasing amounts of sensitive textual data, accurate and scalable PII redaction is more important than ever. This is especially important in regulated industries such as healthcare, law, finance, and education, where mishandling personal data can have serious legal, ethical, and financial consequences. For example, HIPAA prohibits sharing electronic health records (EHRs) for research unless they are properly de-identified [1]. Inadequate redaction can breach compliance and compromise patient trust [2]. Legal documents like court transcripts also require manual redaction which is subject to error, as evidenced by the 2010 data breach at Legal Aid that exposed client data [3]. However, the inclusion of unredacted personally identifiable information (PII) in training data can lead to memorization and unintended disclosure. For instance, Carlini et al. [4] demonstrated that generative models such as GPT-2 and GPT-3 can reproduce exact strings of sensitive training data, including phone numbers and email addresses, when prompted adversarially. This raises serious concerns about the privacy risks associated with unfiltered data corpora.

PII redaction has traditionally used rule-based or statistical methods. These methods rely on deterministic pattern matching or regular expression matching. While fast and interpretable, their use case is extremely limited as PII redaction requires language and context understanding. Patterns rarely generalize across languages or domains. For e.g., U.S. phone numbers are completely different from those in the UK. Consequently, their regex patterns would also be different, making these solutions unreliable at scale. Transformer-based Named Entity Recognition (NER) models [5] trained on labeled corpora like CoNLL-2003 [6] or OntoNotes [7] are frequently used to identify entities such as people, places, organizations, etc, and can be used for PII identification. However, these models can only work on their training domain and lack cross-domain generalization. For instance, a NER-based PII redactor trained on English corpora performs poorly when applied to Spanish texts, as demonstrated later in our experiments.

In recent years, proprietary services have attempted to fill this gap with deep learning-based commercial PII Redaction solutions. Offerings like AWS Comprehend [8], Microsoft Presidio [9], and Google Cloud Data Loss Prevention [10] provide APIs for detecting and redacting sensitive information in documents. These services benefit from large-scale models, achieving high accuracy in many settings. However, they come with significant drawbacks. First, they are closed-source, offering no transparency into the underlying models, data handling practices, or redaction logic. The lack of auditability makes them hard to adopt in compliance-heavy industries. Second, they require organizations to trust third parties with

Work done outside role at Amazon.

sensitive internal data. This in itself can be a regulatory violation. For example, hospitals or banks are legally restricted from sharing unencrypted data with external platforms, even for processing. These limitations motivate us to build an open-source, generalizable, and customizable PII Redaction tool.

To address these challenges, we propose the use of LLMs as customized PII Redaction tools. LLMs possess powerful language understanding capabilities, enabling them to identify PII types that evade pattern matching and traditional NER. LLMs are pre-trained on diverse corpora, making them adaptable and generalizable for redaction in multiple domains with minimal task-specific tuning. Many high-performing LLMs such as LLaMA, Falcon, Mixtral, etc., are open-source. These models can be deployed within secure, self-managed infrastructure, preserving data sovereignty and minimizing exposure of sensitive information. The use of LLMs offers several technical advantages: effective cross-domain transferability (e.g., applying knowledge from one language to another), robust handling of diverse formats and registers without handcrafted rules, and full transparency for auditing, retraining, and fine-tuning within controlled environments.

In this work, we investigate the capabilities and limitations of LLMs as privacy-preserving PII Redaction systems. While prior efforts have demonstrated promising results using large models as contextual privacy learners [11], there has been little systematic study on how architectural class, training paradigm, and inference strategy influence redaction performance across diverse domains. Our goal is to establish empirical foundations for choosing and adapting language models for high-accuracy, customizable PII Redaction.

We focus on two core research questions:

1) Can large language models be effectively adapted into generalizable and domain-agnostic systems for high accuracy PII Redaction?
2) What combinations of model architecture, training paradigm, and inference strategy yield the best trade-offs between redaction performance, latency, and cross-domain generalization?

To this end, we make the following contributions:

- We present a **comprehensive benchmark** evaluating a range of model architectures, including Dense LLM (e.g., LLaMA 3.1–8B [12], GPT-4 [13]), Small Language Model (SLM) (e.g., T5 [14], LLaMA 3.2–3B [12]), Mixture of Experts (MoE) (e.g., Mixtral [15]), Large Reasoning Model (LRM) (e.g., DeepSeek-R1 [16], DeepSeek-Q1 [17]), Structured State Model (SSM) (e.g., FalconMamba [18]), and a strong NER baseline (BERT-NER [5]). These models are assessed under multiple training paradigms, including vanilla (zero-shot), full fine-tuning, and instruction-tuning, as well as inference strategies such as standard generation and retrieval-augmented generation (RAG). Our evaluation spans span-correct and label-exact redaction, and includes metrics for redaction accuracy, and semantic preservation.
- We release **PRvL (PII Redaction via Language Models)**, a publicly available suite of fine-tuned models and evaluation tools for general-purpose PII Redaction. PRvL is built on open-source architectures and trained on a standardized taxonomy of PII types, as described in Appendix VIII. The models support instruction-tuned and RAG-based inference settings and are designed for extensibility to unseen domains. All code, model checkpoints, and evaluation scripts are made available via GitHub (https://anonymous.4open.science/r/PRvL-C1BF) to support reproducibility.

The remainder of this paper presents our Methodology in Section III, Experimental Setup in Section IV, Evaluation in Section V and Result and Analysis in Section VI.

II. Related Work

A. PII Redaction and Privacy-Preserving NLP

The task of detecting and redacting PII has been widely explored in domains such as healthcare [2], finance [19], and social media [20]. Early approaches have primarily relied on handcrafted rule-based systems [21]–[23] and regular-expression-based pattern matching [24], [25]. These methods are typically effective for structured or semi-structured data where entity formats are predictable. However, their performance tends to degrade on noisy, domain-specific, or context-sensitive text, where lexical cues alone are insufficient to distinguish PII from non-sensitive content [26].

More recently, [27] reviewed key advances in PII de-identification, highlighting the shift toward deep learning approaches, particularly in clinical text processing. State-of-the-art systems in this area now predominantly leverage neural architectures, including recurrent neural networks (RNNs) [28], long short-term memory (LSTM) networks, and gated recurrent units (GRUs) [29]. Transformer-based models have also gained traction for their ability to capture long-range dependencies and contextual cues more effectively [30], [31]. In parallel, hybrid systems that combine rule-based heuristics with neural models, as well as ensemble approaches that aggregate predictions across multiple architectures, continue to be active areas of research due to their potential to improve robustness and adaptability.

To overcome these limitations, researchers have explored the adaptation of LLMs for contextual PII Redaction. Unlike conventional models, LLMs exhibit strong generalization capabilities and nuanced language understanding, making them well-suited for identifying context-dependent PII in diverse domains. Recent work has proposed strategies to mitigate the privacy risks associated with LLMs through (1) pretraining corpus curation, (2) conditional or task-specific pretraining, and (3) post-training alignment with privacy constraints [32]–[35]. These methods aim to reduce the likelihood of memorizing and regurgitating sensitive information while preserving model utility. In parallel, efficient fine-tuning techniques have emerged to enhance contextual privacy, focusing on aligning model outputs with normative privacy expectations rather than relying solely on explicit identifiers. This shift is informed by theories of privacy as contextual integrity [11], [36] and operationalized in recent empirical work evaluating LLMs through this lens [35], [37].

B. Adaptation and Training Strategies for LLMs

Adapting LLMs to downstream tasks like PII Redaction requires strategies that balance performance, cost, and privacy constraints. Full fine-tuning remains effective but is often infeasible at scale. Parameter-efficient methods such as LoRA [38], prompt tuning [39], and prefix tuning [40] allow targeted adaptation with minimal overhead. Instruction tuning [41], [42] improves zero-shot generalization by aligning models to task-formatted prompts, while RAG setups [43] introduce external knowledge to aid contextual understanding. Recent work has also explored reinforcement learning from human feedback (RLHF) [44] to align LLM outputs with human values, though its application to structured redaction remains limited. Despite progress, there is limited comparative understanding of how training paradigms interact with model size and architecture in privacy-sensitive applications. Our work evaluates these strategies across architectural families to establish practical recommendations for scalable, accurate, and compliant PII redaction.

C. Architectural Variants of Language Models

Recent advancements in language model architectures have introduced diverse trade-offs between accuracy, latency, and context capacity. Standard dense models (e.g., GPT-3, LLaMA [12]) provide strong baselines but are computationally intensive. Small language models (SLMs) such as LLaMA-3 3B and T5-small [45] offer efficiency with minimal accuracy loss when task-aligned. Long-range models (LRMs), like DeepSeek-R1 and OpenAI-o3 [46], extend context beyond 32K tokens, essential for document-level redaction. Mixture-of-Experts (MoE) architectures (e.g., Mixtral [47], DeepSeek-MoE) scale capacity while limiting active compute. State space models (SSMs), including Mamba [48], show promise in low-latency, long-sequence processing. RAG models such as RETRO [49] incorporate external memory for grounded generation but remain underexplored in privacy contexts. These diverse designs inform our evaluation of architecture-specific strengths in generalizable PII Redaction.

D. Evaluation of Privacy in LLMs

As LLMs grow in scale and utility, their propensity to memorize and leak sensitive data has become a central concern. Carlini et al. [4] demonstrated that autoregressive models like GPT-2 and GPT-3 can regurgitate training data verbatim under adversarial prompting, prompting widespread investigation into privacy risks. Subsequent studies [37], [50] have proposed membership inference, extraction-based probing, and contextual integrity analysis to quantify leakage. Metrics such as exposure, precision of secret recall, and entropy reduction have become common tools for auditing memorization. However, these metrics often fail to capture the subtleties of contextual PII, which may not be explicitly memorized but inferred through latent associations. Differential privacy has been proposed as a training-time safeguard [51], but it remains challenging to apply at LLM scale without sacrificing performance. As a result, evaluating privacy remains an open problem—especially for downstream tasks like redaction, where private information may surface implicitly through model outputs or hallucinations.

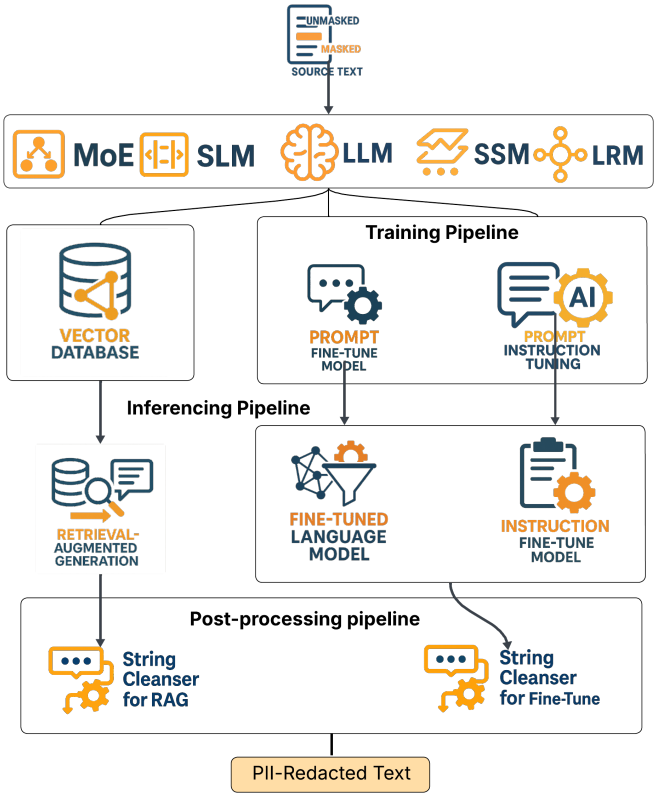

Fig. 1: End-to-end decision workflow for training or deploying a PII-aware redaction language model. The diagram outlines multiple adaptation paths: Fine-Tuning, Instruction Tuning, and Retrieval-Augmented Generation (RAG) and model selection spanning both proprietary (P) and open-source (OS) architectures.

Recent benchmarks, such as by Lukas et al. [35] and Shao et al. [52] have begun to evaluate privacy leakage across model scales and families systematically. However, few efforts have connected these evaluations to real-world tasks like PII redaction [53], leaving a gap in understanding practical privacy guarantees.

III. Methodology

A. Task Definition

We formalize *PII Redaction* as a token-level transformation task over a natural language sequence. Given an input sentence, denoted as $x = [x_1, x_2, \ldots, x_n]$, the goal is to generate a masked sequence $y = [y_1, y_2, \ldots, y_n]$ such that any span corresponding to PII in x is replaced with a corresponding type-specific placeholder, while non-PII tokens are preserved.

Each token x_i is associated with a label $l_i \in \{0, 1\}$ indicating whether the token belongs to a PII span. Let $T(x)$ be a set of annotated spans (s_j, e_j, t_j), where s_j and e_j are token-level start and end indices of the j^{th} PII entity, and t_j is

TABLE I: Overview of model capabilities across architecture types. A checkmark (✓) indicates support for the corresponding capability.

Model	Model Type	Open Source	Fine-Tuned	Instruction-Tuned	RAG
BERT-NER	NER	✓			
Llama3.1-8b	D-LLM	✓	✓	✓	✓
T5	SLM	✓	✓		
Llama3.2-3B	SLM	✓	✓	✓	✓
DeepSeek-Q1	LRM	✓	✓	✓	✓
Mixtral	MoE	✓	✓	✓	✓
GPT-4	D-LLM				✓
OpenAI-o3	LRM				✓
FalconMamba	SSM	✓			✓

its corresponding type (e.g., `NAME`, `EMAIL`, `LOCATION`). The output sequence is defined by the following transformation:

$$y_i = \begin{cases} \texttt{[MASK\_}t_j\texttt{]} & \text{if } x_i \in [x_{s_j}, \ldots, x_{e_j}], \text{ for some } j \\ x_i & \text{otherwise} \end{cases}$$

We emphasize that correct redaction requires not only accurate entity recognition but also context-sensitive disambiguation, as certain terms may be PII in one context and not in another (e.g., "Jordan" as a name vs. a country). Depending on the model employed, the task is modeled as a conditional sequence generation or classification task.

B. PII Redaction via Language Models (PRvL)

Building on this definition, we develop **PRvL**, a modular redaction framework that adapts diverse language models to the PII redaction task through targeted training and inference strategies. PRvL is an open-source suite of fine-tuned models, inference templates, and evaluation tools for general-purpose PII redaction. PRvL supports multiple model architectures and is compatible with both instruction-tuned and RAG-based inference workflows. All models are trained on a standardized taxonomy of PII types (see Appendix VIII) and designed for extensibility across domains such as healthcare, legal, and finance. PRvL enables deployment within secure, self-hosted environments, allowing users to redact sensitive content without relying on third-party APIs. The toolkit includes redaction benchmarks, evaluation metrics, and integration utilities for downstream pipelines. An end-to-end workflow is illustrated in Figure 1. All code, trained checkpoints, and documentation are available at: Anonymous-GitHub.

C. Model Architectures

We evaluate six families of model architectures, chosen to reflect a broad range of design principles, including parameter count, sparsity, retrieval integration, and computational efficiency.

1) **Dense Large Language Model (D-LLM):** Dense LLMs are typically transformer-based architectures with billions of parameters, trained on large-scale corpora using self-supervised learning. Their size allows them to generalize well across tasks, but they require significant compute for training and inference. Models of this class include Llama3.1-8b, GPT-4 etc.
2) **Small Language Model (SLM):** SLMs use simplified or pruned transformer architectures, sometimes with quantization or knowledge distillation to reduce size and complexity. While they sacrifice some performance, they are ideal for edge devices and low-latency applications due to reduced memory and compute demands. Models of this class include Llama3.2-3B etc.
3) **Mixture-of-Expert (MoE):** MoE architectures consist of many parallel subnetworks ("experts"), with a gating network dynamically selecting a few to activate per input. This sparse activation allows scaling to hundreds of billions of parameters with relatively constant compute per forward pass, offering high capacity without proportional cost. Models of this class include Mixtral, etc.
4) **Long-Range Model (LRM):** LRMs are designed to handle extended contexts by modifying attention mechanisms (e.g., sparse, linear, or memory-based attention) or by using recurrence/state structures. They can outperform standard transformers on tasks requiring deep context understanding while often using less memory. Models of this class include DeepSeek-Q1, OpenAI-o3, etc.
5) **Structured State Model (SSM):** SSMs use linear dynamical systems to model sequences, replacing self-attention with state transitions that evolve over time. Architectures like Mamba or S4 offer efficient long-range modeling with sub-quadratic complexity, making them faster and more scalable than transformers in some tasks.
6) **NER Baseline:** As a point of comparison, we include a strong NER baseline based on a BERT classifier fine-tuned for span-level entity recognition. While not generative, this model is fast and interpretable, and provides a traditional reference point for PII Redaction tasks.

We include an overview of model capabilities across architecture types, along with their compatibility with training (fine-tuned and instruction-tuned) and inferencing (RAG) capabilities (see Table I).

D. Training Strategies

We employ two primary adaptation strategies to configure language models for contextual redaction. In both cases, we use parameter-efficient fine-tuning via LoRA, enabling scalable model updates without modifying the full weight matrices.

1) **Fine-Tuning:** In this approach, models are trained on parallel corpora consisting of original sequences containing PII and corresponding fully redacted outputs. Each target output replaces annotated spans with consistent, type-aware placeholder tokens (e.g., `<NAME>`, `<EMAIL>`), while preserving all non-sensitive tokens. Training is conducted in a supervised manner, where the model is optimized to generate the redacted output sequence conditioned on the original text. The prompt template for PII Redaction with fine-tuning is provided below.

Fine-Tuning Example

Instruction:
Mask the PII in the following text:

Example Input:
Dear [Sejd], I am writing to inform you of an important ...

ExampleOutput:
Dear [[GIVENNAME1]], I am writing to inform you of an important ...

2) **Instruction Tuning:** Instruction tuning reframes redaction as a prompt-driven task using natural language instructions. Instead of training on the entire corpus, this approach uses a curated set of examples that demonstrate how unredacted inputs should be transformed into redacted outputs. Each instance consists of a prompt, a small number of illustrative input-output pairs, and a new input to redact.
The model is trained to follow the instruction and imitate the demonstrated behavior, thereby learning redaction patterns through alignment with task-level intent. Unlike full fine-tuning, this strategy emphasizes behavior induction over memorization and is particularly effective in low-resource or cross-domain settings where explicit instructions and exemplars guide the model to generalize from limited supervision. The instruction template for PII Redaction with instruction-tuning is provided below.

Instruction-Tuning Example

Instruction:
Below is a sentence. Sensitive information in the sentence should be replaced by placeholders like [NAME], [EMAIL], [DATE], etc.
Write:
(1) a privacy-protected version of the sentence.

Input

team addressed concerns from diverse participants, including students with Biesenkamp and Verdiani

Response

(1) a privacy-protected version of the sentence: team addressed concerns from diverse participants, including students with [LASTNAME] and [LASTNAME]

Example Input:
Dear [Sejd], I am writing to inform you of an important ...

ExampleOutput:
Dear [[GIVENNAME1]], I am writing to inform you of an important ...

E. Inference Strategies

At inference time, we employ two strategies for applying trained models to redaction tasks: (1) standard generation (vanilla), and (2) retrieval-augmented generation (RAG). While vanilla decoding directly maps raw input to redacted output, RAG augments the model input with retrieved examples or policies to guide more accurate and context-sensitive redaction.

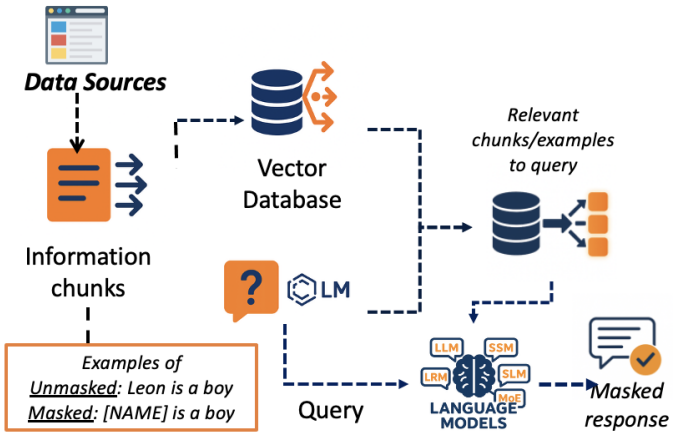

Fig. 2: Overview of the RAG-based redaction pipeline: retrieved examples inform context-aware masking.

1) Retrieval-Augmented Generation (RAG): RAG enhances redaction performance by explicitly conditioning the model on retrieved context relevant to the input. This enables the model to resolve ambiguous cases, handle rare PII types, and follow domain-specific redaction conventions.

The RAG pipeline involves three stages (illustrated in Figure 2):

1) **Query Construction:** Given an input sequence x, a query q is generated to retrieve relevant redaction examples. In the default case, $q = x$, but optionally we encode x with a dense retriever to emphasize redaction-critical spans (e.g., suspected PII markers or tags). An example of the RAG prompt used for retrieval is provided in Appendix.
2) **Document Retrieval:** Using q, we retrieve the top-k redaction exemplars $\{d_1, d_2, \ldots, d_k\}$ from a pre-encoded index. The retrieval corpus includes previously annotated redaction pairs or curated templates representing valid redaction behavior across domains. Retrieved documents may be filtered by entity type overlap or similarity thresholds.
3) **Contextualized Generation:** The model input is constructed as a concatenation of retrieved examples and the original query:

$$x' = \texttt{[CONTEXT]} \parallel d_1 \parallel \ldots \parallel d_k \parallel \texttt{[INPUT]} \parallel x$$

The model generates a redacted output conditioned on x'. Instruction-tuned models additionally receive prompts specifying the redaction task (e.g., "Redact all PII based on the examples above") to align behavior with retrieved demonstrations.

We use static retrieval during evaluation for consistency, but the setup supports real-time dynamic retrieval for deployment. The RAG mechanism is architecture-agnostic and applies to both encoder-decoder and decoder-only models within their context limits.

IV. Experimental setup

A. Training and Inference Setup

1) **Supervision Format:** For token-classification-compatible models (e.g., BERT), we use BIO or span-label encoding for each token. During postprocessing, identified PII spans are replaced with type-specific placeholders such as <NAME> or <EMAIL>. For generative models (e.g., GPT-style), we format inputs as described in Section III-D
2) **Optimization:** All models are optimized using AdamW with linear warmup and cosine learning rate decay. We apply parameter-efficient fine-tuning (PEFT) via LoRA, updating only a small subset of adapter parameters while keeping the base model frozen. Hyperparameters are selected via grid search per model class, and early stopping is based on validation loss and task-specific redaction accuracy.
3) **Infrastructure:** All training was conducted on NVIDIA RTX 6000 GPUs with 48GB of memory. Larger MoE models were distributed across 2 nodes using model parallelism. Experiments were run in Dockerized environments with identical software stacks to eliminate confounding from infrastructure heterogeneity.

B. Dataset

We evaluate models on three variants of the AI4Privacy-300K dataset [54]—English, Spanish and Italian and AI4Privacy-500K [55]—each comprising synthetic English text augmented with rich, contextually embedded PII annotations. All datasets share a common schema with fine-grained entity labels (full list provided in Appendix VIII), enabling both token-level and generative-style redaction evaluations. Larger dataset variants increase diversity, entity density, and narrative complexity, providing a scalable benchmark for studying model generalization across redaction strategies. For all reported evaluations, we use a held-out test split of 1K examples.

V. Evaluation

To assess the effectiveness of LLM-based PII Redaction systems, we conduct a systematic evaluation across domains, languages, and model configurations. Our analysis focuses on measuring redaction accuracy, semantic preservation, and privacy leakage.

A. Correctness: Span-Correct and Label-Exact

Standard span-level metrics for NER or PII Redaction assume exact token-label alignment, which breaks down in generative redaction where models may obfuscate PII correctly but diverge syntactically from references. To better capture practical privacy behavior, we design a custom evaluation using **structural edit distance** with semantic interpretation of errors. Our objective is to quantify both the *correctness of redaction* (i.e., whether all PII was masked) and the *fidelity of redaction* (i.e., whether the right label was applied). We compute the minimum sequence of edit operations needed to transform the model-generated output into the ground truth redacted output. The valid operations include the following:

- Insertions: A missing masked token is added. This corresponds to a *false negative* (FN), where a PII span was not masked.
- Deletions: A spurious masked token is removed. This is a *false positive* (FP), indicating a non-PII span was incorrectly masked.
- Substitutions: A token is changed—this is interpreted as a misclassification, either:
  - A correct PII span masked with the wrong label (e.g., <EMAIL> → <NAME>)
  - A spurious or hallucinated redaction label.

  We define:
  - TP (True Positives) = Correctly masked spans with correct labels
  - FP (False Positives) = Non-PII spans incorrectly masked (deletion)
  - FN (False Negatives) = PII spans not masked (insertion)
  - TN (True Negatives) = Correctly identified Non-PII spans

  From these, we compute standard metrics:
  - Precision = TP / (TP + FP)
  - Recall = TP / (TP + FN)
  - Accuracy = TP + TN / (TP + FP + FN +TN)

  We introduce two complementary evaluation settings:
- **Span-Correct Evaluation:** In this setting, a redaction operation is counted as correct if the model identifies the correct *span* of PII, regardless of whether the assigned label matches the gold-standard tag. For example, if a model masks "Google" as <NAME> instead of <ORG>, it is still treated as a true positive under relaxed evaluation. This setting reflects practical goals of privacy preservation: the PII was successfully obscured, even if its type was misclassified.
- **Label-Exact Evaluation:** Here, both the span and the label must match the ground truth. Using the previous example, redaction "Google" as <NAME> would be penalized, as the correct tag is <ORG>. Mislabels are treated as *type-level errors* and penalized accordingly. This setting reflects applications where tag semantics matter (e.g., typed redaction for auditability or compliance).

Mislabel Count: To further distinguish error modes, we report the raw number of mislabeling errors—cases where the span is masked, but the tag is incorrect. These are counted separately from insertions or deletions and help isolate semantic confusion from redaction omission or overreach.

B. Sequence Level Overlap

Since our models generate entire masked outputs, not just labels or token tags, it is essential to assess whether the redacted text maintains structural and semantic fidelity to the intended form. To this end, we use two common sequence-level overlap metrics: ROUGE and BLEU. These metrics are not used to measure redaction performance per se, but rather

to assess the linguistic quality and structural preservation of the output compared to the reference redacted sentence.

ROUGE: ROUGE (Recall-Oriented Understudy for Gisting Evaluation) measures n-gram overlap between the model output and the reference output. We report three variants:

- ROUGE-1: Overlap of unigrams (single words).
- ROUGE-2: Overlap of bigrams (two-word sequences).
- ROUGE-L: Longest common subsequence (LCS) between the two sequences, capturing structural similarity.

Each variant is reported as an F1 score:

$$\text{ROUGE-F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

where precision and recall are defined over overlapping n-grams between the hypothesis and reference.

BLEU: BLEU (Bilingual Evaluation Understudy) measures n-gram precision of the model output against the reference and incorporates a brevity penalty to penalize overly short outputs. BLEU is defined as:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

where p_n is the precision of n-grams, w_n are weights (typically uniform), and BP is the brevity penalty:

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

with c and r representing candidate and reference lengths, respectively.

C. Privacy Leakage: SPriV

To directly quantify residual privacy risk in model outputs, we use the **SPriV** (Sensitive Privacy Violation) score [11]. SPriV measures the proportion of ground-truth PII tokens that remain unmasked in the generated output, normalized by the total number of output tokens.

Let $G = [g_1, g_2, \ldots, g_{|G|}]$ be the generated output sequence of length $|G|$, and let $T \subseteq \{1, 2, \ldots, |G|\}$ denote the set of token indices corresponding to PII tokens in the ground-truth annotations.

We define an indicator function $m_i \in \{0, 1\}$ for each token g_i as:

$$m_i = \begin{cases} 1 & \text{if } i \in T \text{ and } g_i \text{ is not masked} \\ 0 & \text{otherwise} \end{cases}$$

Then, the SPriV score is computed as:

$$\text{SPriV} = \frac{\sum_{i=1}^{|G|} m_i}{|G|}$$

A SPriV score of 0 indicates perfect masking of all sensitive content, while higher values reflect greater privacy leakage. This makes SPriV a critical metric for evaluating redaction systems deployed in compliance-sensitive or high-risk environments.

D. Cross-Domain Generalization:

To evaluate the generalization capability of PII Redaction models, we perform a cross-domain assessment in which models trained solely on English-language data from the Ai4Privacy-300K dataset are tested on novel domains and languages without any additional fine-tuning. The training data consists of PII-annotated English text drawn from structured domains such as email, chat logs, and customer service records. For evaluation, we use the more diverse Ai4Privacy-500K benchmark, which includes documents from heterogeneous domains such as legal, medical, web, and social media. In addition to domain variation, we assess cross-lingual transfer by evaluating model performance on manually annotated Spanish and Italian subsets. These examples are either professionally translated or synthetically generated from English templates, with PII spans verified and re-aligned for language-specific morphology. All evaluations are conducted using the same span-level and sequence-level metrics defined in Section V. Results are discussed in Section VI-H.

TABLE II: Span-Correct Evaluation: Metrics reflect detection of correct PII spans regardless of label accuracy. Precision and recall are computed from edit-distance alignment with span-only matching.

Model	Accuracy	Precision	Recall
BERT-NER	0.986	0.907	0.982
Fine-Tuned			
Llama3.1-8B	0.986	0.915	0.969
T5	0.883	0.727	0.830
Llama3.2-3B	0.843	0.429	0.689
DeepSeek-Q1	0.993	0.963	0.978
Mixtral	0.988	0.940	0.957
Instruction-Tuned			
Llama3.1-8B	0.992	**0.975**	0.962
Llama3.2-3B	0.983	0.942	0.909
DeepSeek-Q1	**0.994**	0.973	**0.981**
Mixtral	0.973	0.937	0.834
RAG			
Llama3-8B	0.930	0.827	0.717
Llama3.2-2B	0.919	0.751	0.657
DeepSeek-Q1	0.878	0.628	0.521
Mixtral	0.939	0.803	0.776
GPT-4	0.975	0.886	0.900
OpenAI-o3	0.970	0.880	0.860
FalconMamba	0.884	0.688	0.443

VI. Results and Analysis

We analyze our experimental findings along four practical axes: training efficiency, inference latency, architectural trade-offs between scale and efficiency, and the impact of different adaptation paradigms. These results illuminate the operational and strategic implications of deploying PII Redaction systems

across real-world environments with varying resource constraints.

TABLE III: Label-Exact Evaluation: Metrics reflect strict matching of both span and entity label. Mislabel # indicates type errors on correctly identified spans.

Model	Mislabel #	Accuracy	Precision	Recall
BERT-NER	195	0.986	0.904	0.982
Fine-Tuned				
Llama3.1-8B	2974	0.985	0.835	0.936
T5	1211	0.884	0.700	0.825
Llama3.2-3B	2005	0.836	0.269	0.513
DeepSeek-Q1	3033	0.992	0.925	0.953
Mixtral	4324	0.987	0.773	0.831
Instruction-Tuned				
Llama3.1-8B	2968	0.992	**0.949**	0.922
Llama3.2-3B	2673	0.982	0.889	0.832
DeepSeek-Q1	3047	**0.994**	0.945	**0.960**
Mixtral	3640	0.972	0.785	0.553
RAG				
Llama3-8B	869	0.926	0.780	0.653
Llama3.2-3B	768	0.916	0.683	0.578
DeepSeek-Q1	818	0.874	0.509	0.400
Mixtral	865	0.936	0.750	0.718
GPT-4	1209	0.974	0.873	0.857
OpenAI-o3	1035	0.970	0.851	0.830
FalconMamba	468	0.883	0.620	0.366

A. Summary of Evaluation Results

We analyze model performance across span-level correctness, label fidelity, output fluency, and privacy leakage, drawing from Table II, Table III, and Table IV. Instruction-tuned models, particularly DeepSeek-Q1 and Llama3.1-8B, consistently demonstrate strong span-level accuracy. In Span-Correct evaluation, instruction-tuned DeepSeek-Q1 achieves the highest overall accuracy (0.994) and recall (0.981), while Llama3.1-8B attains the highest precision (0.975), highlighting its conservative masking behavior. These results indicate that with instruction tuning, models can reliably identify PII spans even under relaxed evaluation criteria.

Under the stricter Label-Exact evaluation, which penalizes incorrect type assignments, performance drops across the board. Nonetheless, instruction-tuned DeepSeek-Q1 retains top performance with the highest accuracy (0.994) and recall (0.960), and Llama3.1-8B again leads in precision (0.949). Mislabeling errors are substantial for fine-tuned models—DeepSeek-Q1 (fine-tuned) shows over 3,000 mislabels—while instruction-tuned variants reduce this number, suggesting better semantic understanding of entity types.

Sequence-level metrics highlight the generative fluency and structure of the redacted outputs. T5 achieves the highest ROUGE-1/2/L scores (0.940 / 0.857 / 0.934), indicating close structural alignment with reference outputs. However, instruction-tuned DeepSeek-Q1 achieves the best BLEU score (0.908) and the lowest SPriV score (0.002), balancing fluency with privacy robustness. SPriV results show that some models, such as Llama3.2-3B (RAG) and FalconMamba, exhibit significant leakage despite producing grammatically fluent outputs.

Overall, instruction tuning proves critical to redaction effectiveness. Instruction-tuned models outperform fine-tuned and retrieval-augmented counterparts across all dimensions, demonstrating superior span detection, label precision, structural fidelity, and minimized privacy leakage.

TABLE IV: Sequence-Level Metrics: ROUGE and BLEU measure structural fidelity of the masked output; SPriV quantifies proportion of redacted PII tokens.

Model	ROUGE-1/2/L	BLEU	SPriV
Fine-Tuned			
Llama3.1-8B	0.915 / 0.847 / 0.915	0.872	0.003
T5	**0.940 / 0.857 / 0.934**	0.830	0.024
Llama3.2-3B	0..602 / 0.544 / 0.598	0.497	0.036
DeepSeek-Q1	0.915 / 0.845 / 0.915	0.906	**0.002**
Mixtral	0.876 / 0.781 / 0.876	0.864	0.004
Instruction-Tuned			
Llama3.1-8B	0.910 / 0.842 / 0.910	0.882	0.004
Llama3.2-3B	0.911 / 0.843 / 0.911	0.882	0.010
DeepSeek-Q1	0.915 / 0.846 / 0.915	**0.908**	**0.002**
Mixtral	0.855 / 0.750 / 0.854	0.837	0.019
RAG			
Llama3.1-8B	0.841 / 0.777 / 0.837	0.743	0.028
Llama3.2-3B	0.792 / 0.713 / 0.784	0.740	0.205
DeepSeek-Q1	0.645 / 0.556 / 0.631	0.607	0.027
Mixtral	0.840 / 0.769 / 0.835	0.799	0.028
GPT-4	0.928 / 0.881 / 0.929	0.900	0.011
OpenAI-o3	0.810 / 0.688 / 0.800	0.720	0.016
FalconMamba	0.734 / 0.649 / 0.721	0.659	0.024

B. Taxonomy of redaction Errors

To understand model behavior beyond aggregate scores, we analyze common failure modes observed across model outputs. These error patterns correspond to specific degradations in performance in the metrics reported in Tables II, III, and IV.

1) **Overredaction (False Positives)**: Redacting non-PII content due to superficial lexical signals (e.g., capitalization, rarity) leads to reduced precision.

   `Input:` I met them at Quantum Bistro near the coast.
   `Prediction:` I met them at `<ORG>` near the coast.

   This is frequent in low-capacity models such as Mixtral-RAG and FalconMamba, whose relaxed precision scores fall below 0.71 in Table II.

2) **Underredaction (False Negatives)**: Failure to redact valid PII—often due to ambiguous phrasing or domain-specific patterns—leads to direct privacy leakage.

   `Input:` Here's what Jordan emailed on the 22nd.
   `Prediction:` Here's what Jordan emailed on `<DATE>`.

Models like Llama3.2-3B exhibit high SPriV scores (0.75 in Table IV), indicating incomplete coverage despite fluent output.

3) **Mislabeling (Type Confusion)**: Models correctly identify PII spans but assign incorrect labels, which affects strict evaluation metrics.

   `Input:` You can reach me at stanford.edu
   `Prediction:` You can reach me at `<ORG>`
   `Ground truth:` `<EMAIL>`

   Finetuned models like Llama3.1-8B and DeepSeek-Q1 show high mislabel counts (3120 and 2764 in Table III) and strict precision below 0.92 despite high span accuracy.

4) **Mask Drift and Hallucination**: Some generative models hallucinate mask tokens in contexts that contain no actual PII, often due to weak grounding.

   `Input:` Thank you for your interest.
   `Prediction:` Thank you for your interest, `<NAME>`!

   These errors inflate SPriV and reduce relaxed precision, particularly in RAG variants (Table IV).

C. Training Resource Requirements

We benchmark GPU time against F1 score to evaluate fine-tuning efficiency across models and tuning strategies. All experiments were conducted on two 48GB NVIDIA RTX 6000 GPUs. As shown in Fig. 3, and detailed in Table VIII, models like DeepSeek-Q1(IT), LLaMA 3.1–8B(IT), and LLaMA 3.2–3B(IT) lie in the top-left quadrant, demonstrating strong performance with low GPU time. Mixtral(IT) achieves high F1 but incurs the largest compute cost. T5(FT), despite moderate GPU usage, underperforms significantly in F1. Fine-tuned variants such as DeepSeek-Q1(FT) and LLaMA 3.2–3B(FT) strike a good balance, while instruction-tuned models tend to yield better F1 efficiency.

D. Inference Latency and Cost

We evaluate a range of model architectures across fine-tuning and instruction-tuning setups by measuring their F1 score against average inference latency for generating 150 tokens. The trade-off is visualized in Fig. 4, with results detailed in Table IX. Models in the top-left quadrant, such as LLaMA 3.1–8B(FT), LLaMA 3.2–3B(FT), and DeepSeek-Q1(FT), achieve strong performance with low latency, representing the best efficiency-accuracy trade-off. GPT-4 and Mixtral(FT) exhibit high F1 but at higher computational cost. Instruction-tuned models generally show reduced F1, with T5(FT) notably underperforming in both metrics.

E. Model Scale vs. Efficiency

We analyze the relationship between model size (in billions of parameters) and F1 score across both fine-tuned and instruction-tuned setups. As shown in Fig. 5, the x-axis is log-scaled to capture size differences across multiple orders of magnitude. Several smaller models—such as DeepSeek-Q1(IT), DeepSeek-Q1(FT), and LLaMA 3.2–3B(FT)—achieve strong F1 scores, demonstrating that compact architectures can yield highly competitive performance. GPT-4 and Mixtral(FT), while significantly larger, also deliver high F1, illustrating that size still correlates with top-end performance. T5(FT) and instruction-tuned LLaMA models underperform relative to their size, appearing in the lower quadrants. Overall, fine-tuned variants show better efficiency, and some small models rival larger counterparts, suggesting that optimal tuning and architecture selection may outweigh raw scale for certain tasks.

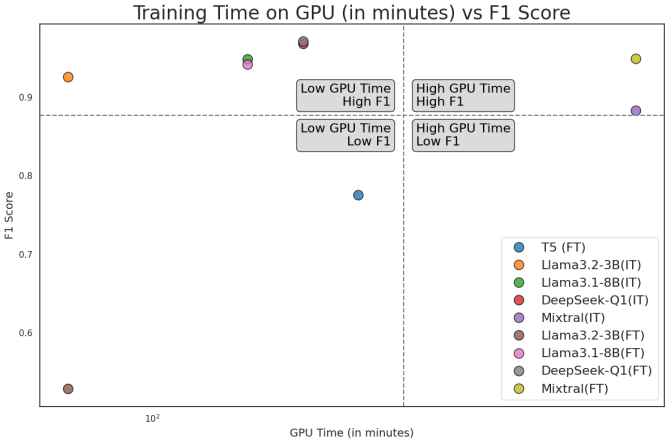

Fig. 3: This graph visualizes the trade-off between GPU usage and model performance. All experiments were run on two 48GB NVIDIA RTX 6000 GPUs. The plot is divided into quadrants: the top-left represents the optimal trade-off—high performance with low GPU usage. The top-right indicates high performance at high computational cost, while the bottom-left reflects both low resource usage and low performance.

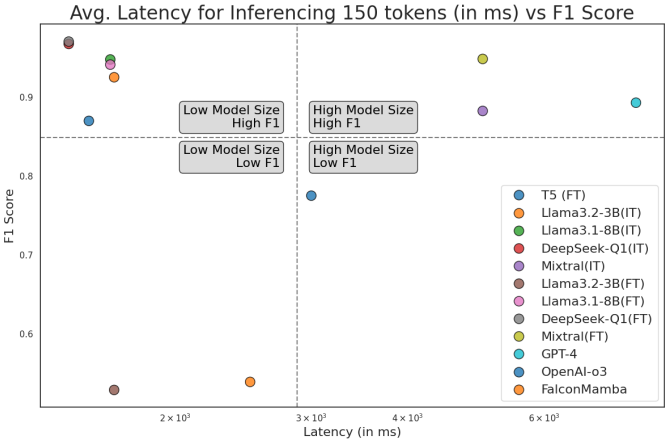

Fig. 4: This plot illustrates the trade-off between inference latency (ms) and F1 score for generating 150 tokens. Models are categorized by training strategy (FT: fine-tuned, IT: instruction-tuned). The top-left quadrant indicates the optimal balance—high F1 with low latency. The top-right captures high-performing but slower models, while the bottom-right reflects both high latency and low performance, notably T5(FT).

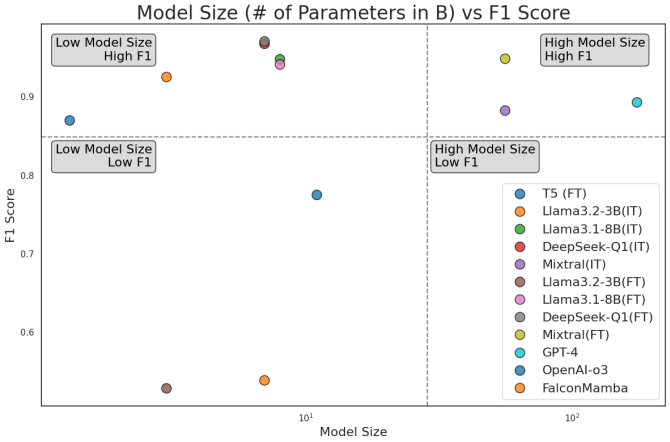

Fig. 5: This plot illustrates the trade-off between model size (in billions of parameters) and F1 score. The x-axis uses a log scale. Models in the top-left quadrant achieve high F1 with compact architectures, including DeepSeek-Q1(IT), LLaMA 3.2–3B(FT), and DeepSeek-Q1(FT). The top-right quadrant includes larger, high-performing models such as GPT-4 and Mixtral(FT). T5(FT) and instruction-tuned LLaMA variants lie in the lower quadrants, reflecting reduced performance despite varied model size.

F. Impact of Training and Inference Paradigm

We analyze how different training strategies affect model performance under equal parameter budgets. Table V shows a side-by-side comparison of Llama3.2-3B and DeepSeek-Q1 under all adaptation methods. Instruction tuning consistently improves both span and label accuracy, while reducing mislabels and SPriV. RAG methods improve structural fluency but provide less consistent gains in label fidelity or privacy protection, highlighting the importance of task-specific instruction design over raw retrieval.

TABLE V: Effect of training paradigms on redaction performance. **FT** = Fine-Tuned, **IT** = Instruction-Tuned, **RAG** = Retrieval-Augmented Generation. Metrics include span-level accuracy, label-level accuracy, number of mislabels, and SPriV score.

Model (Adaptation)	Span Acc.	Label Acc.	Mislabels	SPriV
Llama3.2-3B (FT)	0.843	0.836	2005	0.036
Llama3.2-3B (IT)	0.992	0.992	2968	0.010
Llama3.2-3B (RAG)	0.919	0.916	768	0.028
DeepSeek-Q1 (FT)	0.993	0.992	3033	**0.002**
DeepSeek-Q1 (IT)	**0.994**	**0.994**	3047	**0.002**
DeepSeek-Q1 (RAG)	0.878	0.874	818	0.027

G. Comparison with Baseline (Vanilla)

We analyze the performance of vanilla open-source models as baselines, observing their difficulty in consistently adhering to instructions and producing format-compliant outputs. Frequently, these models failed to generate the requested masked sentences, indicating limitations in their raw pretrained capabilities without targeted fine-tuning. Consequently, evaluation was restricted to sequence-level metrics such as ROUGE and BLEU due to the inconsistency of generated responses. Table VI summarizes these baseline results, providing context to the improvements observed through fine-tuning, instruction tuning, and RAG.

TABLE VI: Baseline (Vanilla) performance comparison. Metrics include ROUGE-1/2/L and BLEU scores.

Model	ROUGE-1/2/L	BLEU
LLama3.1-8B (Vanilla)	0.431/0.338/0.405	0.286
LLama3.1-8B (IT)	0.910/0.842/0.910	0.882
DeepSeek-Q1 (Vanilla)	0.272/0.179/0.236	0.163
DeepSeek-Q1 (FT)	0.915/0.845/0.915	**0.906**
GPT-4 (Vanilla)	0.540/0.342/0.529	0.325
GPT-4 (RAG)	**0.928/0.881/0.929**	0.900

H. Cross-Domain Generalization

We evaluate the robustness of models to cross-domain generalization by testing on three out-of-distribution datasets: a smaller held-out Spanish set, a smaller held-out Italian set, and an external English dataset that was not used during training or RAG construction. Across domains, LLMs demonstrated strong generalization on Spanish and Italian sets, benefiting from structural and semantic alignment with the original training data. Furthermore, BERT-NER limitations on unseen languages and sentence structures during training are reflected in its low performance compared to the more robust LLMs. However, SPriV is not seen to be affected because of the tendency of BERT-NER to over-mask. Table VII summarizes these results.

TABLE VII: Cross-Domain generalization performance using span-exact evaluation.

Model	Accuracy	Precision	Recall	SPriV
Spanish				
BERT-NER	0.845	0.408	0.885	0.013
Llama3.1-8B (IT)	0.984	0.981	0.878	0.014
DeepSeek-Q1 (FT)	0.984	0.927	0.942	0.006
GPT-4 (RAG)	0.974	0.856	0.942	0.006
Italian				
BERT-NER	0.721	0.301	0.915	0.012
Llama3.1-8B (IT)	**0.993**	0.964	0.966	0.002
DeepSeek-Q1 (FT)	**0.993**	**0.996**	**0.989**	**0.001**
GPT-4 (RAG)	0.967	0.865	0.921	0.011
External Dataset				
BERT-NER	0.878	0.663	0.737	0.053
Llama3.1-8B (IT)	0.922	0.826	0.700	0.046
DeepSeek-Q1 (FT)	0.915	0.866	0.687	0.064
GPT-4 (RAG)	0.934	0.914	0.776	0.045

VII. Conclusion

We present a comprehensive study of large language models (LLMs) for contextual redaction of personally identifiable information (PII) in unstructured text. Our evaluation across model architectures, training paradigms, and inference strategies reveals that instruction-tuned and fine-tuned open-source

models achieve high accuracy, low latency, and minimal privacy leakage. Instruction tuning emerges as the most effective adaptation strategy, while smaller models like DeepSeek-Q1 offer strong performance at lower computational cost. RAG improves fluency but is less reliable for strict redaction needs. Cross-lingual and cross-domain evaluations confirm that LLM-based redactors generalize well with minimal task-specific tuning. As a core contribution, we release PRvL, a fully open-source toolkit that includes fine-tuned models, evaluation metrics, and deployment-ready utilities for secure, compliant redaction. PRvL supports instruction tuning, RAG, and domain customization, enabling end-to-end privacy-preserving workflows without relying on third-party services. Our findings establish a strong empirical foundation for building accurate, efficient, and trustworthy redaction systems using open LLMs.

REFERENCES

[1] "Summary of the hipaa privacy rule — hhs.gov," 3 2025, [Online; accessed 2025-06-20]. [Online]. Available: https://www.hhs.gov/hipaa/for-professionals/privacy/laws-regulations/index.html

[2] I. Neamatullah, M. M. Douglass, L.-W. H. Lehman, A. Reisner, M. Villarroel, W. J. Long, P. Szolovits, G. B. Moody, R. G. Mark, and G. D. Clifford, "Automated de-identification of free-text medical records," *BMC medical informatics and decision making*, vol. 8, pp. 1–17, 2008.

[3] A. Hope, "Legal aid data breach leaks millions of sensitive records, moj's poor cybersecurity practices slammed - cpo magazine," 5 2025, [Online; accessed 2025-06-20]. [Online]. Available: https://tinyurl.com/53d697hz

[4] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson *et al.*, "Extracting training data from large language models," in *30th USENIX security symposium (USENIX Security 21)*, 2021, pp. 2633–2650.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019, pp. 4171–4186.

[6] E. F. Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," *arXiv preprint cs/0306050*, 2003.

[7] S. S. Pradhan, E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel, "Ontonotes: A unified relational semantic representation," in *International Conference on Semantic Computing (ICSC 2007)*, 2007, pp. 517–526.

[8] Amazon Web Services, "Natural Language Processing – Amazon Comprehend," AWS, 2025, [Online]. Available: URL [Accessed: Date]. [Online]. Available: https://aws.amazon.com/comprehend/

[9] Microsoft, "Presidio - data protection and de-identification sdk." [Online]. Available: https://github.com/microsoft/presidio

[10] "Cloud data loss prevention — google cloud," [Online; accessed 2025-06-21]. [Online]. Available: https://cloud.google.com/security/products/dlp

[11] Y. Xiao, Y. Jin, Y. Bai, Y. Wu, X. Yang, X. Luo, W. Yu, X. Zhao, Y. Liu, Q. Gu *et al.*, "Privacymind: large language models can be contextual privacy protection learners," *arXiv preprint arXiv:2310.02469*, 2023.

[12] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[13] OpenAI, "Gpt-4 technical report," https://openai.com/research/gpt-4, 2023.

[14] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *JMLR*, 2020.

[15] M. AI, "Mixtral of experts," https://mistral.ai/news/mixtral-of-experts/, 2023.

[16] DeepSeek, "Deepseek-r: Retrieval-augmented language models," https://github.com/deepseek-ai, 2024.

[17] ——, "Deepseek-q: Mixture of experts for multitask language understanding," https://github.com/deepseek-ai, 2024.

[18] T. I. Institute, "Falconmamba: Combining falcon and mamba for efficient long-context modeling," https://huggingface.co/tiiuae/falcon-mamba, 2024.

[19] L. Chiticariu, Y. Li, and F. Reiss, "Rule-based information extraction is dead! long live rule-based information extraction systems!" *EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, vol. October, pp. 827–832, 01 2013.

[20] Y. Liu, F. Y. Lin, M. Ebrahimi, W. Li, and H. Chen, "Automated pii extraction from social media for raising privacy awareness: A deep transfer learning approach," in *2021 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE, 2021, pp. 1–6.

[21] S. M. Meystre, F. J. Friedlin, B. R. South, S. Shen, and M. H. Samore, "Automatic de-identification of textual documents in the electronic health record: a review of recent research," *BMC medical research methodology*, vol. 10, pp. 1–16, 2010.

[22] J. Aberdeen, S. Bayer, R. Yeniterzi, B. Wellner, C. Clark, D. Hanauer, B. Malin, and L. Hirschman, "The mitre identification scrubber toolkit: design, training, and assessment," *International journal of medical informatics*, vol. 79, no. 12, pp. 849–859, 2010.

[23] B. Norgeot, K. Muenzen, T. A. Peterson, X. Fan, B. S. Glicksberg, G. Schenk, E. Rutenberg, B. Oskotsky, M. Sirota, J. Yazdany *et al.*, "Protected health information filter (philter): accurately and securely de-identifying free-text clinical notes," *NPJ digital medicine*, vol. 3, no. 1, p. 57, 2020.

[24] M. Douglass, G. D. Clifford, A. Reisner, G. B. Moody, and R. G. Mark, "Computer-assisted de-identification of free text in the mimic ii database," in *Computers in Cardiology, 2004*. IEEE, 2004, pp. 341–344.

[25] T. Aura, T. A. Kuhn, and M. Roe, "Scanning electronic documents for personally identifiable information," in *Proceedings of the 5th ACM workshop on Privacy in electronic society*, 2006, pp. 41–50.

[26] D. Singh and S. Narayanan, "Unmasking the reality of pii masking models: Performance gaps and the call for accountability," *arXiv preprint arXiv:2504.12308*, 2025.

[27] V. Yogarajan, B. Pfahringer, and M. Mayo, "A review of automatic end-to-end de-identification: Is high accuracy the only metric?" *Applied Artificial Intelligence*, vol. 34, no. 3, pp. 251–269, 2020.

[28] F. Dernoncourt, J. Y. Lee, O. Uzuner, and P. Szolovits, "De-identification of patient notes with recurrent neural networks," *Journal of the American Medical Informatics Association*, vol. 24, no. 3, pp. 596–606, 2017.

[29] T. Ahmed, M. M. Al Aziz, and N. Mohammed, "De-identification of electronic health record using neural network," *Scientific Reports*, vol. 10, no. 1, pp. 1–11, 2020.

[30] A. E. Johnson, L. Bulgarelli, and T. J. Pollard, "Deidentification of free-text medical records using pre-trained bidirectional transformers," in *Proceedings of the ACM Conference on Health, Inference, and Learning*. ACM, 2020, pp. 214–221.

[31] K. Murugadoss, A. Rajasekharan, B. Malin, V. Agarwal, S. Bade, J. R. Anderson, J. L. Ross, W. A. J. Faubion, J. D. Halamka, V. Soundararajan *et al.*, "Building a best-in-class automated de-identification tool for electronic health records through ensemble learning," *Patterns*, vol. 2, no. 6, p. 100255, 2021.

[32] Y. Li, Z. Tan, and Y. Liu, "Privacy-preserving prompt tuning for large language model services," *arXiv preprint arXiv:2304.11635*, 2023.

[33] X. Zhang, S. Li, X. Yang, C. Tian, Y. Qin, and L. R. Petzold, "Enhancing small medical learners with privacy-preserving contextual prompting," *arXiv preprint arXiv:2308.04621*, 2023.

[34] S. Kim, S. Yun, H. Lee, M. Gubri, S. Yoon, and S. J. Oh, "Propile: Probing privacy leakage in large language models," in *Advances in Neural Information Processing Systems*, 2023.

[35] N. Lukas, A. Salem, R. Sim, S. Tople, L. Wutschitz, and S. Zanella-Béguelin, "Analyzing leakage of personally identifiable information in language models," *arXiv preprint arXiv:2305.17303*, 2023.

[36] H. Nissenbaum, "Privacy as contextual integrity," *Washington Law Review*, vol. 79, pp. 119–158, 2004.

[37] N. Mireshghallah, H. Kim, X. Zhou, Y. Tsvetkov, M. Sap, R. Shokri, and Y. Choi, "Can llms keep a secret? testing privacy implications of language models via contextual integrity theory," in *International Conference on Learning Representations (ICLR)*, 2024.

[38] E. J. Hu, P. Wallis, Z. Allen-Zhu *et al.*, "Lora: Low-rank adaptation of large language models," in *International Conference on Learning Representations (ICLR)*, 2022.

[39] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proceedings of the 2021 Con-*

ference on Empirical Methods in Natural Language Processing, 2021, pp. 3045–3059.

[40] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," *arXiv preprint arXiv:2101.00190*, 2021.

[41] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned language models are zero-shot learners," *arXiv preprint arXiv:2109.01652*, 2021.

[42] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, "Stanford alpaca: An instruction-following llama model," 2023.

[43] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.

[44] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[45] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[46] B. Jiang *et al.*, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[47] E. Beeching, G. Izacard, H. Touvron *et al.*, "Mixtral of experts: Sparse mixture of experts models are extremely effective," *arXiv preprint arXiv:2312.15842*, 2023.

[48] A. Gu, T. Dao, Y. He, A. R. Fu, C. Re *et al.*, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[49] S. Borgeaud, A. Mensch, J. Hoffmann *et al.*, "Improving language models by retrieving from trillions of tokens," in *Proceedings of the 39th International Conference on Machine Learning*, 2022.

[50] N. Carlini, D. Ippolito, M. Jagielski *et al.*, "Quantifying memorization across neural language models," *arXiv preprint arXiv:2202.07646*, 2022.

[51] X. Li, F. Tramer, P. Liang, and T. Hashimoto, "Large language models can be strong differentially private learners," in *International Conference on Learning Representations*, 2022.

[52] Y. Shao, T. Li, W. Shi, Y. Liu, and D. Yang, "Privacylens: Evaluating privacy norm awareness of language models in action," *arXiv preprint arXiv:2409.00138*, 2024.

[53] D. Pham, P. Kairouz, N. Mireshghallah, E. Bagdasarian, C. M. Pham, and A. Houmansadr, "Can large language models really recognize your name?" *arXiv preprint arXiv:2505.14549*, 2025.

[54] "ai4privacy/pii-masking-300k · datasets at hugging face," 6 2024, [Online; accessed 2025-06-21]. [Online]. Available: https://huggingface.co/datasets/ai4privacy/pii-masking-300k

[55] "ai4privacy/open-pii-masking-500k-ai4privacy · datasets at hugging face," 3 2025, [Online; accessed 2025-06-21]. [Online]. Available: https://huggingface.co/datasets/ai4privacy/open-pii-masking-500k-ai4privacy

VIII. Appendix

TABLE VIII: Training resource requirements: GPU hours and peak memory usage per model

Model (Adaptation)	GPU Hours	Peak Memory (GB)
T5	4 h	19 GB
Llama3.2-3B	1 h 10 mins	5.5 GB
Llama3.1-8B	2 h 30 mins	10 GB
DeepSeek-Q1	3h 10 mins	14 GB
Mixtral	13 h	43 GB

TABLE IX: Inference latency per model. Latency is average milliseconds per 150 tokens.

Model (Adaptation)	Latency (ms)	Tokens/sec
T5	3000	50
Llama3.2-3B	1667	90
Llama3.1-8B	1648	91
DeepSeek-Q1	1456	102
Mixtral	5000	30
OpenAI-o3	1546	97
Falcon-Mamba- 7B	2500	60
GPT-4	7895	19

List of Supported PII Labels

[STREET]	[USERNAME]
[GEOCOORD]	[GIVENNAME1]
[SOCIALNUMBER]	[GIVENNAME2]
[TEL]	[CARDISSUER]
[TITLE]	[EMAIL]
[PASSPORT]	[BUILDING]
[PASS]	[IP]
[COUNTRY]	[CITY]
[SEX]	[POSTCODE]
[BOD]	[SECADDRESS]
[LASTNAME3]	[STATE]
[TIME]	[LASTNAME1]
[LASTNAME2]	[DATE]
[IDCARD]	[DRIVERLICENSE]

RAG Prompt Example

Instruction:
Below is a `sentence-to-mask` and examples of `unmasked - masked` sentences. Based on the examples, write a privacy protection version of `sentence-to-mask` in the form of a `masked-sentence`.
Sensitive information should be replaced by placeholders like [NAME], [EMAIL], [ORG], etc.
Always put your response after `masked-sentence:`

Examples:
Example 1:
`unmasked:` Alice went to Stanford University.
`masked:` [NAME] went to [ORG].

Example 2:
`unmasked:` Bob emailed me at bob@gmail.com.
`masked:` [NAME] emailed me at [EMAIL].

Example 3:
`unmasked:` Carla was born on May 4, 1990.
`masked:` [NAME] was born on [DATE].

`End of examples`

Sentence-to-mask:
John registered for the app with email 1909@gmail.com

masked-sentence:
[NAME] registered for the app with email [EMAIL]