Shielding PII to Prevent Re-identification and Preserve Utility

SHUHAO LIU, Shenzhen Institute of Computing Sciences, China
WENFEI FAN, Shenzhen Institute of Computing Sciences, China and University of Edinburgh, United Kingdom
YIJIA XU, South China University of Technology, China

This paper addresses the challenge of protecting Personally Identifiable Information (PII) in textual data, identifying and anonymizing PII to ensure privacy and regulatory compliance, while preserving data utility. We model this bi-criteria optimization problem as a two-player Stackelberg game, where an attacker seeks to link anonymized data back to individuals and a protector anonymizes the data to prevent re-identification. We show that the problem is intractable. Thus we develop SHIELD, an attack-aware PII protection system that iteratively engages the protector and attacker to prevent both PII breaches and over-scrubbing. SHIELD integrates logical reasoning with machine learning to identify PII, and supports pluggable attackers for robustness against re-identification. It achieves a constant-factor approximation for utility loss while mitigating risk. Using synthetic and real-world datasets, we empirically show that SHIELD offers better privacy–utility trade-off than prior PII protection systems, while remaining efficient and scalable.

CCS Concepts: • **Security and privacy** → **Pseudonymity, anonymity and untraceability**; **Usability in security and privacy**; • **Theory of computation** → *Constraint and logic programming*; • **Computing methodologies** → Natural language processing.

Additional Key Words and Phrases: personally identifiable information, re-identification, Stackelberg game, logical reasoning, Large Language Model

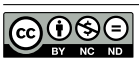
ACM Reference Format:
Shuhao Liu, Wenfei Fan, and Yijia Xu. 2026. Shielding PII to Prevent Re-identification and Preserve Utility . *Proc. ACM Manag. Data* 4, 3 (SIGMOD), Article 232 (June 2026), 28 pages. https://doi.org/10.1145/3802109

1 Introduction

Personally Identifiable Information (PII) refers to data that can identify a specific individual, either directly or indirectly. It includes information about identity, location, contact, finance, medical conditions, employment, and education. PII can be direct identifiers (*e.g.,* name, Social Security Number), or quasi-identifiers (*e.g.,* age, ZIP code, and job title when combined with other data) [58].

Protecting PII aims to safeguard sensitive personal data from unauthorized access, misuse, and breaches, ensuring privacy, security, and regulatory compliance [48]. Effective protection helps prevent (a) identity theft, fraud, and unauthorized transactions; (b) workplace and healthcare discrimination; (c) espionage, blackmail, and geopolitical attacks; (d) manipulation of user behavior on social media platforms, among others. Globally, PII protection is enforced through laws and regulations such as HIPAA (USA) [74], GDPR (EU) [20], PIPEDA (Canada) [28], and CCPA (California) [7]. As data collection and AI-driven analytics proliferate, privacy concerns over potential PII exposure become increasingly critical.

Authors' Contact Information: Shuhao Liu, shuhao@sics.ac.cn, Shenzhen Institute of Computing Sciences, Shenzhen, Guangdong, China; Wenfei Fan, wenfei@inf.ed.ac.uk, Shenzhen Institute of Computing Sciences, Shenzhen, China and University of Edinburgh, Edinburgh, United Kingdom; Yijia Xu, xuyijia606@outlook.com, South China University of Technology, Guangzhou, China.



ACM 2836-6573/2026/6-ART232
https://doi.org/10.1145/3802109

Table 1. A PII cleaning example.

Original	The issue related to the Dijkstra's algorithm bug was assigned to John Doe (Employee #A723, jd at x.com) from Samsung. He is my neighbor in Seoul, and we don't work with Skype.
Presidio [49]	The issue related to the <NAME>'s algorithm bug was assigned to <NAME> (Employee <ID>, jd at x.com) from <ORG>. He is my neighbor in <CITY>, and we don't work with Skype.
RoBERTa [14]	The issue related to the Dijkstra's algorithm bug was assigned to <NAME> (Employee #A723, <EMAIL_ADDR>) from <ORG>. He is my neighbor in <CITY>, and we don't work with <ORG>.
DP-MLM [46] ($e^{\epsilon} = 100$)	This issue related to the Skype's algorithm crash was reserved to John Doe (Case #A, jan at x) from Samsung. J is my neighbors in Pyongyang, and we do not work with Skype.
Qwen3-235B-A22B [68]	The issue related to the Dijkstra's algorithm bug was assigned to <NAME> (Employee <ID>, <EMAIL_ADDR>) from <ORG>. He is my neighbor in <CITY>, and we don't work with Skype.
Ours (SHIELD) ($e^{\epsilon} = 10$)	The issue related to the Dijkstra's algorithm bug was assigned to A3BD8E (Employee <ID>, <EMAIL_ADDR>) from Samsung. He is my neighbor in Korea, and we don't work with Skype.

Protecting PII is challenging as it requires balancing privacy with utility. On the one hand, not only explicit but also quasi-identifiers need to be anonymized, as seemingly harmless attributes can be combined to re-identify individuals; for example, gender, birth date, and postal code can uniquely identify 63% of the U.S. population when cross-referenced with Census data [27]. On the other hand, excessive anonymization can strip valuable information, impairing analysis; for instance, over-sanitizing clinical reports may remove patterns essential for understanding disease transmission [42].

Existing PII protection systems [2, 32, 49, 65, 68] often fail to satisfy both privacy and utility requirements. Their PII identification may be inadequate, resulting in (1) false negatives (FNs), where sensitive data is overlooked; and (2) false positives (FPs), where non-sensitive data is misclassified as PII. Moreover, their anonymization can unnecessarily degrade data utility by (3) over-sanitizing quasi-identifiers; and (4) completely transforming and obfuscating the input text, making it unclear what was altered or why. Fundamentally, these limitations stem from their *attack-agnostic* design: without anticipating potential adversarial exploits, they are unable to fully capture contextual information that could jeopardize privacy or to sanitize data effectively while maintaining its utility.

Example 1: Table 1 compares multiple state-of-the-art (SOTA) PII protection systems, including hybrid Presidio [49], Named Entity Recognition (NER)-based RoBERTa [14], differentially-private DP-MLM [46] and a large language model (LLM). Observe the following.

(1) Presidio redacts the algorithm name "Dijkstra" as name PII. Although such an FP does not leak sensitive information, it leads to unnecessary loss of critical technical context. Conversely, Presidio fails to detect "jd at x.com" as an email address (an FN), due to its unconventional format, which eludes standard detection patterns.

(2) RoBERTa misclassifies "Skype" as organizational PII (an FP) but overlooks the employee ID (an FN) due to limited training coverage.

(3) Even under a loose privacy constraint ($e^{\epsilon} = 100$), DP-MLM produces an insensible and unauditable output, altering the bug source, introducing spurious facts ("Seoul" with "Pyongyang"), and failing to anonymize "John Doe", incurring a severe privacy breach.

(4) Qwen3-235B-A22B [78], a reasoning LLM, can properly sanitize the short input text with [68]. However, it most likely relies on a cloud-hosted model, which incurs additional API costs and introduces potential risks of exposing PII to the service provider.

(5) Most SOTA systems redact both "Samsung" and "Seoul." This is over-sanitization: generalizing "Seoul" to "Korea" suffices to preserve privacy given Samsung's large workforce, while keeping the company name and partial location detail improves data utility. □

How can we achieve the optimal privacy–utility tradeoff, while maintaining explainability and efficiency? Can we enhance attacker contextual awareness to enable robust PII protection?

Contributions & Organization. We propose SHIELD, a System for Holistic Identification and ELimination of PII from Datasets, designed for textual data, which poses unique challenges [16, 41].

(1) Problem and complexity (Section 3). To enable explicit attack awareness, we model PII protection as a Stackelberg game [76] between a protector and an attacker. The protector first anonymizes a dataset $\mathcal{D}$ to maximize the utility of the output $\mathcal{D}'$ under privacy constraints. The attacker then seeks to maximize re-identification success. The optimal Stackelberg equilibrium is derived via backward induction. We prove that its computation is NP-complete.

(2) An iterative workflow (Section 4). We introduce SHIELD, a fully automated PII protection system that approximates the Stackelberg equilibrium via a multi-round workflow. In each round, the protector builds a re-identification risk graph and applies one anonymization action, followed by attackers probing for residual vulnerabilities. This adaptive process simplifies the original game and achieves balanced privacy and utility with a provable trade-off guarantee.

(3) Re-identification risk graph construction (Section 5). A key contribution of SHIELD is the *re-identification risk graph*, which consolidates contextual information from extracted PII to guide anonymization decisions. PII is detected using *PII Extraction Rules* (PERs) that integrate pluggable external tools, NER models, and entity relations to reduce FPs and FNs, and are automatically learned via the LLM-as-a-Judge paradigm [11, 30] for accuracy and adaptability.

(4) Adaptive PII anonymization (Section 6). SHIELD advances beyond standard scrubbing by introducing graph-driven anonymization actions, *i.e., node splitting* and *edge removal*, which operate directly on the re-identification risk graph. These targeted actions disrupt attacker inference while preserving contextual utility. Moreover, we propose a Risk-Utility Cover heuristic to guide action selection in each round, balancing privacy gain against utility loss and achieving a constant-factor approximation to the optimal trade-off.

(5) Experimental study (Section 7). Using real-life and synthetic datasets, we empirically verify the accuracy and efficiency of SHIELD. We find the following. Compared with the SOTA PII protection systems, (a) SHIELD detects PII more reliably, reducing FNRs (False Negative Rate) by up to 98.4%; (b) it preserves semantics with minimal edits, achieving 2.6–62.1 utility-point gains; (c) its structured logs enable direct explainability and traceability; and (d) it reduces runtime by 54.1%, and dollar cost by 31.0×. Moreover, (e) our ablation study justifies the effectiveness of the risk graph and the RUC heuristic, along with a recommended configuration strategy.

Section 2 reviews the background and related work. Section 8 summarizes our contributions. Detailed proofs are provided in [39].

2 Background and Related Work

We review PII identification, anonymization, and re-identification.

Personally Identifiable Information (PII). Global privacy regulations consistently stress the protection of PII, though their definitions vary slightly. For example, the EU's GDPR [20] defines personal data as "any information relating to an identified or identifiable natural person," including names, ID numbers, location data, and online identifiers. The California CCPA [7] likewise defines PII as data that identifies or can be linked, directly or indirectly, to a consumer or household. Both highlight the need for strong safeguards of information enabling individual identification.

PII identification. Identifying PII within texts is a critical initial step. The SOTA methods can be broadly categorized as follows:

(1) Rule-based methods rely on expert-crafted pattern matchers [47, 51], *e.g.,* regular expressions for ID, dictionaries [55], and lookup tables for common names. They achieve reasonable accuracy for specific PII types without requiring annotated data [70]. However, they suffer from (a) poor cross-domain generalization, causing FPs (*e.g.,* misidentifying disease names derived from scientists); (b) limited ability to detect complex or context-dependent PII, causing FNs (*e.g.,* missed abbreviations [52]); and (c) heavy maintenance demands to adapt to evolving data and patterns [38].

(2) ML-based methods frame PII identification as token-sequence classification via Named Entity Recognition (NER) [12, 45]. Progression from early statistical models [16, 41, 51, 79] to deep learning [8, 16, 50] and transformers [15, 17, 34, 43] has improved accuracy and generalizability. However, evaluations [64] show that even the SOTA models struggle with context-dependent PII. These methods also require extensive labeled data [52] and heavy computation. Recent approaches [5, 68] exploit LLMs' contextual reasoning [66], but still face (1) input length limits, (2) high cost and latency, and (3) reliance on proprietary models that raise privacy and security risks.

(3) Hybrid solutions integrate rule-based and ML-based methods into toolboxes with streamlined interfaces. Industry-grade examples include Flair [2], SpaCy [65] and Microsoft Presidio [49]. Yet, these systems typically depend on manually crafted selectors to resolve conflicts across detectors, requiring significant human expertise and limiting scalability and adaptability in large deployments.

Unlike prior work, SHIELD introduces a new hybrid approach to PII protection. (a) It unifies rule-based and ML-based methods through PERs and a context-rich re-identification risk graph. PERs embed multiple detectors as predicates, optimizing strategies per PII type and reducing both FPs and FNs to enhance generalizability and context-awareness. (b) It automatically learns PERs via LLM-as-a-Judge rather than relying on hand-crafted priority schemes over tool outputs, offering more flexible conflict resolution and better accuracy than hybrids. (c) PERs can support deep (recursive) detection, unlike hybrids that perform only single-pass checks.

Anonymization. Text anonymization aims to remove PII to ensure data privacy while maintaining utility. Key approaches include:

(1) PII scrubbing, also known as de-identification, directly replaces detected PII tokens [37, 58]. Common techniques include redaction (deletion), masking (placeholders such as "`[NAME]`"), and pseudonymization [60, 72], which substitutes PII with realistic synthetic surrogates. While effective, these methods struggle to balance privacy and utility: redaction and masking disrupt coherence and meaning [40], whereas pseudonymization may preserve hidden associations, heightening re-identification risks [67, 69].

(2) Privacy-preserving data publishing methods provide formal privacy guarantees for anonymized text through techniques such as PII generalization and embedding noising. Generalization replaces specific values with broader categories (*e.g.,* "in 1998" to "in the 1990s") to satisfy criteria like k-anonymity [61, 71], though selecting the right level of abstraction remains challenging [6, 9]. Embedding noising seeks guarantees such as differential privacy [18, 19] [32, 44, 46, 75, 81], t-plausibility [3], or C-sanitize [62, 63]. Despite strong theoretical appeal, these methods often require heavy, opaque text transformations that significantly degrade utility [58].

(3) LLM-based rewriting [5, 68] rewrites text using LLMs to obfuscate PII while preserving readability, but suffers from poor scalability to long documents, high cost, privacy concerns, and limited formal guarantees due to its black-box nature.

SHIELD differs from prior work in the following: (1) it maximizes data utility by explicitly modeling potential re-identification attacks as adversaries; (2) beyond existing anonymization actions, it introduces two new ones (node splitting and edge removal) on the re-identification risk graph, to break associations between anonymized subjects and improve flexibility; and (3) it offers a constant-factor approximation to the optimal privacy–utility trade-off.

Re-identification. Re-identification seeks to link anonymized data back to original subjects, posing major privacy risks [80]. Common attacks include: (a) statistical inference [82], which exploits quasi-identifiers and background knowledge to uncover sensitive information; and (b) LLM inference [17, 67, 73], which leverages in-context and parametric knowledge to re-identify individuals.

In contrast to prior work, SHIELD explicitly incorporates known re-identification attacks as adversaries, allowing it to anticipate and mitigate existing vulnerabilities. Its modular design also supports the rapid integration of newly discovered attacks, ensuring continuous robustness against evolving privacy threats.

3 The PII Protection Problem

This section formulates the PII protection problem.

Challenges & Motivation. As remarked in Sections 1 and 2, existing PII protection systems face several critical challenges.

(1) FNs in identification. SOTA PII protection systems frequently miss context-dependent PII (*e.g.,* indirect references or atypically formatted phone numbers), enabling attackers to correlate residual information with external sources and compromise privacy.

(2) FPs in identification. Many detectors are overly cautious, misclassifying non-sensitive data as PII. This causes unnecessary anonymization, reduces utility, and can render text incoherent; *e.g.,* common words resembling names or locations are wrongly flagged.

(3) Over-sanitization. SOTA systems over-sanitize quasi-identifiers without assessing their context-specific re-identification risk, leading to avoidable utility loss (*e.g.,* over-sanitizing low-risk geographic details when a more nuanced approach would suffice).

(4) Lack of explainability. Some systems operate as black boxes, providing limited justification for why certain data is flagged and anonymized, hindering verification, error diagnosis, regulatory compliance, and adaptation to evolving privacy requirements.

These challenges motivate an *attack-aware* design that explicitly models realistic re-identification strategies, reducing false negatives and false positives, avoiding over-sanitization, and improving explainability through threat-model–grounded decisions.

Metrics. In light of the insights above, we advocate a game-theoretic setup for the problem, starting with the relevant metrics.

Privacy. To quantify the privacy level of an anonymized dataset $\mathcal{D}'$, we employ separate metrics for direct and quasi-identifier protection, reflecting the varying risks associated with their exposure.

(1) *Direct identifiers.* We evaluate the FNR (False Negative Rate) of direct identifier protection, since any FN directly exposes a specific individual in the dataset. We do not consider FPs in this context, since erroneously anonymizing a non-PII impacts the data utility only, which will be evaluated separately from privacy protection.

(2) *Quasi-identifiers.* We do not prioritize strict accuracy in quasi-identifier identification, as leaving some quasi-identifiers intact is acceptable if they do not substantially increase re-identification risk.

We introduce an attacker-aware privacy metric called the *de-identification indistinguishability gap* (DIG), a notion adapted from the multiplicative form of ϵ-differential privacy (DP) [18, 19].

More specifically, we model the PII protection procedure as a mechanism M over dataset $\mathcal{D}$, which outputs an anonymized dataset $\mathcal{D}' = M(\mathcal{D})$. Let $\mathcal{I}$ denote the set of targeted individuals subject to re-identification. For any $i \in \mathcal{I}$, let $\mathcal{D}_{-i}$ be an *adjacent dataset* to $\mathcal{D}$ obtained by removing all information associated with i. The DIG level $\epsilon_{\mathcal{A}}(M(\mathcal{D}))$ of $M(\mathcal{D})$ relative to $\mathcal{D}$ *w.r.t.* attacker $\mathcal{A}$ is thus quantified by the smallest number $\epsilon > 0$ such that

$$\Pr[\mathsf{DeID}_{\mathcal{A}}(i, M(\mathcal{D}))] \leq e^{\epsilon} \cdot \Pr[\mathsf{DeID}_{\mathcal{A}}(i, M(\mathcal{D}_{-j}))],\ \forall i, j \in \mathcal{I}. \tag{1}$$

Here $\mathsf{DeID}_{\mathcal{A}}(i, M(\mathcal{D}))$ is the acceptance event of a binary distinguisher run by $\mathcal{A}$ on $M(\mathcal{D})$. It can represent re-identification of i or, more generally, membership inference for deciding whether i is in $\mathcal{D}$. Intuitively, this ensures that, given $M(\mathcal{D})$, attacker $\mathcal{A}$ cannot reliably distinguish whether any $i \in \mathcal{I}$ is present in $\mathcal{D}$.

Let $\delta_i = \Pr[\mathsf{DeID}_{\mathcal{A}}(i, M(\mathcal{D}))]$ and $\delta_i^* = \min_{j \in \mathcal{I}} \Pr[\mathsf{DeID}_{\mathcal{A}}(i, M(\mathcal{D}_{-j}))]$. The DIG level of $\mathcal{D}' = M(\mathcal{D})$ can be calculated as

$$\epsilon_{\mathcal{A}}(\mathcal{D}') = \max_{i \in \mathcal{I}} \ln(\delta_i / \delta_i^*). \tag{2}$$

Intuitively, $\epsilon_{\mathcal{A}}(\mathcal{D}')$ measures the worst-case gap in attacker $\mathcal{A}$'s success rate of de-identification from the anonymized $\mathcal{D}'$, compared to the best-case baseline from an anonymized adjacent dataset. When it is clear from context, we omit the subscript $\mathcal{A}$ for brevity.

Note that $\epsilon_{\mathcal{A}}(\mathcal{D}')$ is an attacker-aware metric *w.r.t.* the specific DeID methods used by $\mathcal{A}$ (*e.g.,* statistical and LLM-based). It is directly inspired by DP's indistinguishability principle: under a threat model where $\mathcal{A}$ models the strongest known distinguishing attacks considered, a small $\epsilon_{\mathcal{A}}(\mathcal{D}')$ provides a DP-like indistinguishability guarantee against that attack class. As attackers evolve in strategies and background knowledge, the privacy may diminish, requiring adaptive anonymization to maintain robust protection.

Data utility. Recent studies [42] have shown that no single metric can fully capture the utility of anonymized datasets. We thus employ *relative utility* $U(\mathcal{D}', \mathcal{D})$, which captures both the retention of semantic meaning and the preservation of linguistic quality through a composition of established metrics. It is defined as

$$U(\mathcal{D}, \mathcal{D}') = w_1 \cdot \mathsf{Sim}(\mathcal{D}, \mathcal{D}') - w_2 \Delta_{\mathsf{CoLA}}(\mathcal{D}, \mathcal{D}') - w_3 \Delta_{\mathsf{PPL}}(\mathcal{D}, \mathcal{D}'),$$

where w_1, w_2 and w_3 are non-negative weights;

- *Semantic similarity* $\mathsf{Sim}(\mathcal{D}, \mathcal{D}')$ measures the preservation of meaning between $\mathcal{D}$ and $\mathcal{D}'$ (higher values are preferred). We use language model–based *BERTScore* [83], which is more nuanced and context-aware than traditional BLEU/ROUGE [36, 57];
- *CoLA* [77] *delta* $\Delta_{\mathsf{CoLA}}(\mathcal{D}, \mathcal{D}') = \mathsf{CoLA}(\mathcal{D}') - \mathsf{CoLA}(\mathcal{D})$ quantifies the grammatical acceptability degradation from $\mathcal{D}$ to $\mathcal{D}'$, where higher values indicate more degradation; and
- *Perplexity* [59] *delta* $\Delta_{\mathsf{PPL}}(\mathcal{D}, \mathcal{D}') = 1/\mathsf{PPL}(\mathcal{D}) - 1/\mathsf{PPL}(\mathcal{D}')$ assesses readability and fluency degradation from $\mathcal{D}$ to $\mathcal{D}'$, using inverse perplexity to normalize scores to $[0, 1]$, where higher values indicate worse quality of dataset $\mathcal{D}'$.

The weights w_1, w_2 and w_3 are selected via a small grid search, after normalizing each utility metric on a validation split and balancing their contributions. Such a lightweight procedure can already ensure stable PII protection even with minor perturbation (see [39]).

Here we omit FPR because a protector may intentionally retain some quasi-identifiers. An FP affects the output only if the token is anonymized, and the downstream cost of unnecessary edits is captured by $U(\mathcal{D}, \mathcal{D}')$ through lower Sim and larger $\Delta_{\mathsf{CoLA}}/\Delta_{\mathsf{PPL}}$. Thus, the composite utility more faithfully reflects the end-to-end cost of PII protection, including degradation induced by FPs.

Table 2. Data utility metrics for examples in Table 1.

Metric	$\uparrow \mathsf{Sim}(\mathcal{D}, \mathcal{D}')$	$\downarrow \Delta_{\mathsf{CoLA}}(\mathcal{D}, \mathcal{D}')$	$\downarrow \Delta_{\mathsf{PPL}}(\mathcal{D}, \mathcal{D}')$
Presidio	0.9145	0.0425	0.0184
RoBERTa	0.9095	0.0469	0.0249
DP-MLM	0.9130	0.0638	0.0656
Qwen3-235B-A22B	0.9060	0.0448	**0.0038**
SHIELD	**0.9363**	**0.0406**	**0.0146**

Example 2: Continuing with Example 1, Table 2 reports utility scores for Table 1. The metrics are complementary: aggressive anonymization may increase Δ_{CoLA} and Δ_{PPL} even when Sim remains high. SHIELD achieves the highest similarity (0.9363) by avoiding over-sanitization, and yields the best grammatical acceptability ($\Delta_{\mathsf{CoLA}} = 0.0406$), with competitive fluency ($\Delta_{\mathsf{PPL}} = 0.0146$) that ranks among the top two, just behind Qwen3-235B-A22B. □

The PII protection problem. PII protection involves the inherent trade-off between safeguarding privacy and maintaining the utility of the anonymized dataset. We model this interaction as a zero-sum Stackelberg game [76], where a protector (leader) and an attacker (follower) take turns playing to maximize their respective payoffs.

Game setup. We set up the PII protection game as follows:

(1) *The protector* moves first by selecting a strategy p from its strategy space $\mathcal{P}$. More specifically, strategy $p = \langle a_1, a_2, \ldots, a_{|p|} \rangle$ is a sequence of *atomic* PII anonymization actions, where a_i is an action from the action space A for $1 \leq i \leq |p|$. The strategy is applied to the original dataset $\mathcal{D}$, producing an anonymized dataset $\mathcal{D}' = p(\mathcal{D})$.

(2) *The attacker* then responds by selecting a strategy q from its strategy space $\mathcal{Q}$ to attempt re-identification. It evaluates the privacy risk of individuals in terms of the $\mathsf{FNR}(\mathcal{D}, \mathcal{D}')$ of direct identifiers and the DIG level $\epsilon(\mathcal{D}')$ of quasi-identifiers.

(3) *The payoffs*. Given the protector's strategy p and the attacker's strategy q, the protector's payoff $\pi(p, q)$ is:

$$\pi(p, q) = U(\mathcal{D}, \mathcal{D}') \cdot \mathbb{I}[\mathsf{FNR}(\mathcal{D}, \mathcal{D}') \leq \tau_0 \wedge \epsilon(\mathcal{D}') \leq \epsilon_0],$$

where τ_0 and ϵ_0 are the maximum tolerable values for FNR and ϵ, respectively, and $\mathbb{I}[\cdot]$ is the indicator function that returns 1 if the condition is true and 0 otherwise. The attacker's payoff is $-\pi(p, q)$. Intuitively, the protector's payoff maximizes utility under strategy p subject to privacy constraints τ_0 and ϵ_0, while the attacker seeks to exploit vulnerabilities in $\mathcal{D}'$ using strategy q.

The problem. An optimal solution to the PII protection game is a Stackelberg equilibrium [76], where the protector and the attacker choose their strategies p^* and q^*, respectively, such that neither can improve their payoff by unilaterally changing their strategy.

We thus formulate the PII protection problem as an optimization problem, which solves the game via backward induction:

- *Input:* A collection $\mathcal{D}$ of texts, privacy thresholds τ_0 and ϵ_0.
- *Output:* A protector strategy $p^* \in \mathcal{P}$.
- *Objective:* Find the solution $p^* = \arg\max_{p \in \mathcal{P}} U(\mathcal{D}, p(\mathcal{D}))$.
- *Constraints:* $\mathsf{FNR}(\mathcal{D}, p^*(\mathcal{D})) \leq \tau_0$ and $\epsilon(p^*(\mathcal{D})) \leq \epsilon_0$.

The goal is to find an optimal protector strategy p^* that maximizes the relative utility $U(\mathcal{D}, \mathcal{D}')$ between the original dataset $\mathcal{D}$ and the anonymized $\mathcal{D}' = p^*(\mathcal{D})$, *i.e.,* seeking a $\mathcal{D}'$ that remains as

close as possible to $\mathcal{D}$ semantically while preserving grammatical coherence and fluency. A valid strategy p^* must ensure that the privacy risks of $\mathcal{D}'$ are below the specified thresholds.

Example 3: Continuing with Example 1, casting PII protection as a protector–attacker game addresses key challenges. (1) It mitigates FPs by retaining benign terms *e.g.,* "Dijkstra," which may match name patterns but are not exploitable by attackers. The context indicates a generic reference, so its anonymization would unnecessarily hurt utility. (2) It reduces FNs by flagging high-risk identifiers, *e.g.,* "jd at x.com," whose exposure enables re-identification. (3) It curbs over-sanitization by calibrating generalization to attacker utility. "Seoul" can be safely generalized to "Korea" to reduce location-induced risk with minimal utility loss. (4) It improves explainability by grounding anonymization in concrete adversarial inference and providing actionable justifications. □

Complexity. The PII protection problem is nontrivial. Its decision problem DPP is to decide, given a dataset $\mathcal{D}$ and a utility threshold U_0, whether there exists a protector strategy $p \in \mathcal{P}$ such that $U(\mathcal{D}, p(\mathcal{D})) \geq U_0$ and the privacy thresholds τ_0 and ϵ_0 are met.

Theorem 1: *Problem* DPP *is* NP*-complete.* □

Proof sketch: DPP is in NP since for any protector strategy $p \in \mathcal{P}$, one can verify that it is in PTIME to check whether its utility $U(\mathcal{D}, p(\mathcal{D})) \geq U_0$ and whether the privacy constraints (τ_0, ϵ_0) are satisfied. We prove NP-hardness by reduction from the set cover problem (see [39] for details), which is NP-complete [25]. □

4 SHIELD: System Overview

As an effective solution to the PII protection problem, this section presents SHIELD, a system to balance data utility and privacy.

A multi-round simplification. As shown in Theorem 1, finding the optimal protector strategy p^* for the single-round Stackelberg game is NP-complete. The intractability stems from the combinatorial explosion of possible action sequences in $\mathcal{P} = A^*$. Consequently, evaluating the payoff for every strategy $p \in \mathcal{P}$ is computationally infeasible for any non-trivial problem instance.

To address this, we reformulate the problem as a multi-round game to enable a tractable approximation. Each round consists of a *single* action by the protector followed by the attacker's response, decomposing the complex decision process into manageable steps.

The multi-round structure reduces a global optimization over $|A|^k$ action sequences to a series of single-action decisions. In the single-round Stackelberg game, the protector must select a full action sequence $p = \langle a_1, \ldots, a_k \rangle$ from $\mathcal{P} = A^*$, evaluating the utility-privacy tradeoff over $|A|^k$ possibilities, which is computationally prohibitive. In contrast, in the multi-round approach, each round selects one action based on the current state $\mathcal{D}_{t-1}$, enabling PTIME greedy heuristics that locally balance utility and privacy, adapt to adversarial feedback, and permit practical approximation.

Workflow. At the core of SHIELD's operation is a *re-identification risk graph* that captures all PII instances, individuals, and their associations. It serves as the central structure for risk assessment and anonymization, providing a unified view of sensitive information. Its graph structure supports efficient traversal and incremental updates in response to attacker feedback, making it a versatile foundation for iterative risk analysis and protection in SHIELD.

Figure 1 shows the iterative three-stage workflow built around this graph. Its entire pipeline, from extraction to protection to attack, is fully automated, with no human-in-the-loop during deployment. Instead of selecting a complete strategy p in advance, the protector operates in rounds, updating the graph and refining its actions based on attacker-identified vulnerabilities. Let $\mathcal{D}_0 = \mathcal{D}$ denote the initial dataset and $\mathcal{D}_{t-1}$ the dataset at the start of round t.

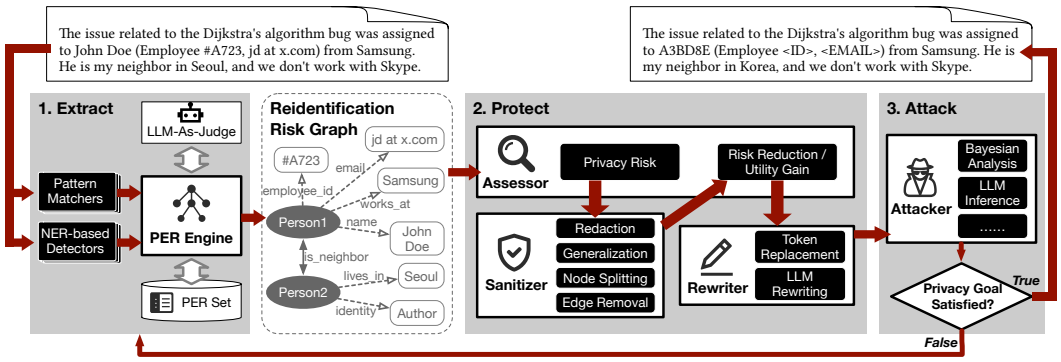


Fig. 1. The workflow of SHIELD.

(Stage 1) Extract. This stage builds and updates a *re-identification risk graph* G_{t-1} that encodes identified PII, target individuals, and their associations within the current dataset $\mathcal{D}_{t-1}$. To extract PII effectively, SHIELD uses a unified framework based on *PII Extraction Rules* (PERs), which integrate logic, regular expressions, and ML detectors. PERs are automatically learned using the LLM-as-a-Judge paradigm [11, 30]. The resulting PER Engine populates and maintains the risk graph G_{t-1}, with structured PII relationships, forming the input to the next stage. Details are discussed in Section 5.

(Stage 2) Protect. An assessor analyzes the graph G_{t-1} to identify high-risk individuals. A sanitizer generates candidate anonymization actions, including redaction, generalization, node splitting, and edge removal. The utility and privacy impact of each action is estimated. Then a rewriter applies the selected action to produce ΔG_t, which is propagated to the dataset as $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \Delta G_t$, using either token replacement or LLM-based rephrasing (Section 6).

(Stage 3) Attack. The attacker attempts re-identification in $\mathcal{D}_t$ adopting a combination of Bayesian reasoning and LLM inference, as they reflect SOTA re-identification strategies: the former captures statistical linkage attacks, whereas the latter captures semantic and contextual inference. Importantly, it is modular and flexible, supporting plug-in attack strategies and auxiliary knowledge sources to model emerging adversaries and uncover latent privacy risks.

Note that the attackers are components internal to SHIELD. All intermediary datasets $\mathcal{D}_t$ and graph G_t are not observable by real-world attackers, consuming no privacy budget until termination.

Termination. The workflow terminates when the privacy goal is achieved: the FNR is below threshold ($\text{FNR} \leq \tau_0$), and the inferred DIG level satisfies $\epsilon(\mathcal{D}_t) \leq \epsilon_0$. Otherwise, it continues to round $t+1$. The final anonymized dataset is released $\mathcal{D}' = \mathcal{D}_t$, and the resulting protector strategy guarantees to be valid and correct.

Remarks. We remark the following on the workflow of SHIELD:

(1) *Termination guarantee.* The workflow is guaranteed to terminate because (a) each anonymization action $a \in A$ strictly reduces or preserves the current level of privacy risk (see Theorem 3), ensuring that $\text{FNR}(\mathcal{D}, \mathcal{D}_t)$ and $\epsilon(\mathcal{D}_t)$ are non-increasing; and (b) for any positive τ_0 and ϵ_0, a finite sequence of actions can reduce privacy risk below these thresholds. These ensure monotonic progress toward correct termination and bound SHIELD's iterations.

(2) *Solving core challenges.* Through its multi-round, feedback-rich workflow, SHIELD adapts to evolving attackers and auxiliary knowledge without redeployment. This enables: (a) fewer FNs

by refining detectors after successful attacks; (b) fewer FPs by preserving utility-critical content when risk is low; (c) less over-sanitization via threat-specific minimal anonymization; and (d) better explainability by tying each action to an explicit adversarial signal.

(3) *Efficiency*. SHIELD is designed to balance efficiency and utility through a restrained use of LLMs. The most expensive LLM operations are confined to offline phases for PERs learning, Online, LLMs are used sparingly and small local models suffice. Thus, although SHIELD is more expensive than rule- or NER-based scrubbing, it incurs only modest overhead while achieving substantially stronger anonymization, and remains significantly more efficient than DP-based rewriters and fully LLM-driven anonymizers.

The following theorem formalizes an idealized bridge result showing when SHIELD recovers an attacker-agnostic DP guarantee.

Theorem 2: *Assume that* SHIELD *integrates a strong attacker* $\mathcal{A}^\star$ *whose attack family covers all information-theoretic* membership-inference *(distinguishing) attacks between adjacent datasets* $\mathcal{D}$ *and* $\mathcal{D}_{-j}$ *based on the released output. Then, the anonymization mechanism* M *induced by* SHIELD *satisfies* ϵ_0*-differential privacy.* □

Proof sketch: By the termination of SHIELD above, for every input dataset $\mathcal{D}$, SHIELD outputs $M(\mathcal{D})$ with $\epsilon_{\mathcal{A}^\star}(M(\mathcal{D})) \leq \epsilon_0$. Thus, the DIG criterion yields an e^{ϵ_0} multiplicative bound on membership-inference attack events in the family of $\mathcal{A}^\star$ for every adjacent $\mathcal{D}$ and $\mathcal{D}_{-j}$. Since $\mathcal{A}^\star$ covers all information-theoretic membership-inference attacks, for any measurable event $S \subseteq \mathcal{Y}$ over the output space $\mathcal{Y}$ of M, consider the binary distinguisher $T_S(y) = \mathbf{1}[y \in S]$. Applying Equation 1 to the acceptance event $\{T_S(M(\cdot)) = 1\}$ yields $\Pr[M(\mathcal{D}) \in S] \leq e^{\epsilon_0} \Pr[M(\mathcal{D}_{-j}) \in S]$. Since this holds for all adjacent datasets and all measurable events S, M satisfies ϵ_0-DP. □

5 PII Identification Methods

This section details our method for PII identification and re-identification. We define PERs, PII Extraction Rules, to accurately identify PII in a dataset $\mathcal{D}$ (Section 5.1). Using the identified PII, we construct re-identification risk graph G of $\mathcal{D}$ (Section 5.2).

5.1 PII Extraction Rules (PERs)

PERs are rules for detecting PII in textual datasets, synthesizing diverse evidence into a unified representation. They combine signals from syntactic patterns, PII identification tools, NER models, and binary relations to enable rich, context-aware detection. With extensible, semantically grounded conflict resolution, PERs deliver accurate, robust performance across varied PII types and domains.

Preliminaries. We start with basic notations.

Datasets. We model a textual dataset $\mathcal{D}$ as a document represented by a finite sequence of tokens $\langle w_1, \ldots, w_{|\mathcal{D}|} \rangle$ over a vocabulary $\mathcal{V}$, assumed to be syntactically valid and semantically coherent. Each token $w_i \in \mathcal{V}$ denotes a word, punctuation mark, or symbol.

Substrings. A *substring* of a dataset $\mathcal{D} = \langle w_1, w_2, \ldots, w_{|\mathcal{D}|} \rangle$ is a contiguous sequence of tokens $\langle w_i, \ldots, w_j \rangle$ for $1 \leq i \leq j \leq |\mathcal{D}|$. The objective of PII identification is to identify and extract substrings of dataset $\mathcal{D}$ that correspond to instances of PII.

Predicates. A *predicate* of a textual dataset $\mathcal{D}$ is defined as

$$p ::= S(s) \mid \mathcal{T}(T, s) \mid \neg\mathcal{T}(T, s) \mid \mathcal{M}(T, s) \oplus c \mid \mathcal{R}(s, s'),$$

where s is a variable bounded by $S(s)$ for substrings of $\mathcal{D}$, $\oplus$ is a comparison operator in $\{=, \neq, <, \leq, >, \geq\}$, and c is a constant. Here (1) $\mathcal{T}$ (resp. $\neg\mathcal{T}$) denotes an existing PII identification tool that

determines whether a substring s corresponds to (resp. is not) PII of type T; (2) $\mathcal{M}$ denotes an NER model that returns a prediction strength in $[0, 1]$ for the likelihood that s is PII of type T; and (3) $\mathcal{R}$ denotes a binary relation (*e.g.,* co-reference, temporal precedence) between substrings s and s', enabling reasoning over contextual semantics.

(1) A *tool predicate* $\mathcal{T}(T, s)$ evaluates to true if an existing identification tool classifies substring s as PII of type T. Such tools include regular expressions that match specific PII, dictionaries, lookup tables, *etc.*. They are pre-defined and can be extended by domain experts, effective for detecting PII with well-defined syntactic patterns. SHIELD integrates various components of Presidio [49] as tools. Each tool is an optional, pluggable component that serves as an auxiliary signal source; no single tool is a mandatory dependency.

(2) An *NER predicate* $\mathcal{M}(T, s) \oplus c$ holds if a pre-trained NER model $\mathcal{M}$ assigns to substring s a confidence d of type T satisfying $d \oplus c$. This captures contextual PII beyond rigid syntactic patterns, such as personal names or organizations. Adjusting the threshold c allows a PER to tune the precision–recall trade-off in PII detection. We support RoBERTa [14], SpaCy [65], FLAIR [2], and ai4privacy [1] as $\mathcal{M}$.

(3) A *binary relation predicate* $\mathcal{R}(s, s')$ captures a semantic or structural relation between two substrings s and s'. These relations can include co-reference (*e.g.,* both s and s' are linked to the same entity), temporal ordering, or discourse-based dependencies. Such predicates are essential for context-aware detection, allowing rules to account for dependencies between different parts of a document. We support pre-trained NLP models as $\mathcal{R}(s, s')$, including AllenNLP [24] for co-reference, TempRel [53] for temporal tagging, and LLM-based `langextract` [26] for relationship parsing.

PII Extraction Rules. A PER φ has the following form:

$$\varphi : X \rightarrow q.$$

Here X is a conjunction $p_1 \wedge p_2 \wedge \cdots \wedge p_k$ of predicates over $\mathcal{D}$, where each p_j is one of the predicates above. Predicate q is an assertion $\mathsf{IsPII}(T, s)$ (or its negation), for substring variable s bounded in X, saying that s is (or is not) PII of type T. We refer to X and q as the *precondition* and *consequence* of PER φ, respectively. This structure supports the composition of fine-grained, reusable rules that capture complex, context-sensitive PII identification logic.

Intuitively, a PER aggregates signals from multiple PII detectors into a single decision. Each predicate contributes structured evidence, *e.g.,* syntactic matches from tools $\mathcal{T}$, contextual confidence from models $\mathcal{M}$, or semantic dependencies captured by relations $\mathcal{R}$. Instead of trusting any detector in isolation, a PER specifies how these signals must agree, or how one signal may override another.

This explicit conflict resolution mitigates cascading errors. Noisy matches from brittle tools are filtered unless corroborated, while high-confidence contradictions can veto spurious detections. By encoding precedence and redundancy directly, PERs prevent early errors from propagating into downstream anchoring, graph construction, or protection, yielding more stable performance.

Example 4: Consider the original text snippet from Table 1 as dataset $\mathcal{D}$. We may apply the following PERs to extract PII:

(1) φ_1: $S(s) \wedge \mathcal{T}(\mathsf{Name}, s) \wedge \mathcal{M}(\mathsf{Name}, s) < 0.1 \rightarrow \neg\mathsf{IsPII}(\mathsf{Name}, s)$. This PER states that a substring s is not name-PII if an NER model $\mathcal{M}$ deems it unlikely (< 0.1), overriding its presence in the common name lookup table $\mathcal{T}$. Applied to the snippet, substring s = Dijkstra will no longer be flagged despite being a common name. This improves recall by trusting high-confidence predictions from contextual NER models when pattern-based tools produce FPs.

(2) φ_2: $S(s) \wedge \neg\mathcal{T}_{\mathsf{regex}}(\mathsf{Email}, s) \wedge \mathcal{M}(\mathsf{Email}, s) > 0.8 \rightarrow \mathsf{IsPII}(\mathsf{Email}, s)$. This PER captures email

addresses with nonstandard formats that bypass traditional regex $\mathcal{T}_{\text{regex}}$, *e.g.,* "`jd at x.com`", based on high-confidence model prediction by $\mathcal{M}$. This rule reduces FNs by enabling detection of subtle or malformed PII instances.

(3) φ_3: $S(s) \wedge S(s') \wedge \mathcal{T}(\texttt{STAFF}, s) \wedge \mathcal{M}(\texttt{ORG}, s') > 0.7 \wedge \mathcal{R}(s, s') \rightarrow \texttt{IsPII}(\texttt{DIRECT}, s)$. It flags a staff ID s (detected by tool $\mathcal{T}$) as direct identifier only if it is linked to (via binary relation $\mathcal{R}$) an organizational entity s' (detected by $\mathcal{M}$). By common sense, Employee ID "#A723" only becomes a direct identifier when "Samsung" is in context. This rule facilitates direct identifier detection by filtering out innocuous ones that lack entity-specific associations.

All three PERs contain upstream errors and lead to better privacy–utility trade-offs. An FP from a name detector might otherwise propagate downstream, but φ_1 contains this by vetoing low-confidence hits. Conversely, φ_2 recovers missed PII when tools fail, reducing FNs that would increase privacy risk. Rule φ_3 further gates high-impact decisions on contextual agreement, preventing benign identifiers from being escalated unless semantically linked. □

Semantics. A *valuation* h of variables of a PER φ over $\mathcal{D}$, or simply a valuation of φ, is a mapping that instantiates a substring variable s of φ with a sequence $h(s)$ of tokens from $\mathcal{D}$.

We say that valuation h *satisfies* a predicate p, denoted $h \models p$, if: (1) when p is $S(s)$, $h(s)$ is a substring of $\mathcal{D}$. (2) When p is $\mathcal{T}(T, s)$ (or $\neg\mathcal{T}(T, s)$), the PII identification tool $\mathcal{T}$ classifies (or does not classify) the substring $h(s)$ as type-T PII. (3) When p is $\mathcal{M}(T, s) \oplus c$, the model $\mathcal{M}$ assigns a confidence score $\alpha \oplus c$ to $h(s)$ being of type T. (4) When p is $\mathcal{R}(s, s')$, the relation $\mathcal{R}$ holds between $h(s)$ and $h(s')$.

For precondition X, $h \models X$ iff $h \models p$ for every predicate p in X. For a PER $\varphi = X \rightarrow q$, $h \models \varphi$ iff $h \models X$ implies $h \models q$. Dataset $\mathcal{D}$ *satisfies* φ, written $\mathcal{D} \models \varphi$, if every valuation h of φ over $\mathcal{D}$ satisfies it. For a set Σ of PERs, we write $\mathcal{D} \models \Sigma$ if $\mathcal{D} \models \varphi$ for all $\varphi \in \Sigma$.

PER Discovery. We now outline how to learn high-quality PERs.

Procedure. We first aggregate detection signals from regex rules, PII tools, NER models, and context-based relations to construct a pool of predicate-based candidates. When full ground truth is unavailable, we use *LLM-as-a-Judge* to assess whether a candidate substring is valid PII, enabling supervision with partial annotations.

We then model PER discovery as a meta-learning problem and adopt a Bayesian-based optimization algorithm to identify valuable PERs, along the same lines as [10]. This formulation avoids the exponential enumeration of PER predicates and selects hyperparameters through Bayesian optimization. We adopt the accuracy (F-score) of PII identification as the criterion to identify effective rules, instead of traditional metrics such as support and confidence [31, 56].

Remarks. (1) Departing from rules such as denial constraints [4], conditional functional dependencies (CFDs) [21] and entity enhancing rules [22, 23], PERs support tool predicates $\mathcal{T}$ (and their negations), NER models and binary relations $\mathcal{R}$, tailored for PII detection.

(2) *Generalizability and extensibility*. Learned PERs generalize beyond the training data by combining complementary PII detectors whose properties are largely dataset-agnostic, preserving accuracy on unseen datasets. This is validated in Section 7, where the same set of PERs is reused across datasets and consistently achieves high accuracy in PII identification. New third-party tools can be added as predicate types without modifying existing rules, enabling extensibility while maintaining backward compatibility.

(3) *Concerns on labeling*. PER learning requires minimal labeled data, using an LLM-based oracle to verify labels, enabling low-resource learning and domain adaptation without heavy annotation.

Privacy risks from LLM-as-a-Judge are limited, since it is used only offline for PER discovery, and is never invoked during online anonymization. Moreover, PERs can be learned on public data and then transferred to private datasets. At deployment time, SHIELD relies only on local, open-weight LLMs for extraction, rewriting, and attack simulation, so sensitive data never leaves the system.

5.2 Re-identification Risk Graph Construction

PII found by PERs is used to build a re-identification risk graph G.

A re-identification risk graph G captures links between individuals and their PII, structuring evidence that could be used for re-identification and enabling systematic analysis. Formally, $G = (V, E, L_V, L_E, A)$ is a directed labeled graph, where V is the vertex set, E is the edge set, L_V and L_E are labeling functions, and A assigns a set of key–value attributes to each vertex.

Vertices. Each vertex $v \in V$ is assigned a label $L_V(v)$, which is either (a) `person`, representing a target individual in $\mathcal{D}$ to whom PII may be attributed, or (b) a specific PII type (*e.g.,* `Name`, `Email`). Moreover, each vertex carries a set of key–value attributes $A(v)$, including *e.g.,* the extracted PII value, the position of evidence tokens supporting the extraction. These attributes facilitate traceability.

Edges. Each edge $e = \langle v_i, v_j \rangle$ in E is assigned an edge label $L_E(e)$. There are two types of edges: (1) An individual–PII edge $\langle v_i, v_p \rangle$ indicates that PII instance v_p is associated with individual i, with $L_E(\langle v_i, v_p \rangle) = \texttt{has\_pii}$; and (2) an individual–individual edge $\langle v_i, v_j \rangle$ represents an inferred interpersonal relationship between individuals i and j, labeled by the specific relation type (*e.g.,* `is_neighbor`).

Edges are unweighted. Nevertheless, different edges may contribute to re-identification risk to different degrees, *e.g.,* by enabling distinct pathways for PII propagation. Extending the graph with explicit weights is possible, but requires corresponding extensions to the attacker model, protector, and privacy guarantee, *e.g.,* via a Bayesian formulation that treats edge existence probabilistically. Such weighted semantics is not assumed here and is mentioned only as a possible extension.

Risk evaluation. A risk graph G preserves all contextual information in dataset $\mathcal{D}$ that may be exploited by a re-identification attacker $\mathcal{A}$. To simplify notation, we use G and $\mathcal{D}$ interchangeably when describing anonymization transformations. For example, $\mathsf{DeID}_{\mathcal{A}}(i, \mathcal{D})$ is equivalent to $\mathsf{DeID}_{\mathcal{A}}(i, G)$, and for a protector strategy p, $p(G)$ denotes the risk graph constructed from $p(\mathcal{D})$.

A key benefit of using G is that it provides a convenient proxy for evaluating the privacy level of an anonymized dataset $\mathcal{D}' = p(\mathcal{D})$, echoing the notion of node-level differential privacy [33] over the risk graph. Let G' be the risk graph constructed from $\mathcal{D}'$. For a target individual $i \in \mathcal{I}$, let G_{-i} be the graph obtained from G by removing the vertex v_i and all incident edges. By construction, G_{-i} is the risk graph of the adjacent dataset $\mathcal{D}_{-i}$ to $\mathcal{D}$. By Equation 2, we have $\epsilon_{\mathcal{A}}(\mathcal{D}') = \epsilon_{\mathcal{A}}(p(G)) = \max_{i \in \mathcal{I}} \ln \frac{\Pr[\mathsf{DeID}_{\mathcal{A}}(i, p(G))]}{\min_{j \in \mathcal{I}} \Pr[\mathsf{DeID}_{\mathcal{A}}(i, p(G_{-j}))]}$.

Graph Construction. We build the re-identification risk graph G in two stages, driven by PERs and a lightweight, local LLM working under the closed schema induced by L_V and L_E. We instantiate vertices and edges from concrete textual evidence. Each node corresponds to a latent individual, and an edge is created only when the text *explicitly supports* the corresponding relation. Throughout, we retain *provenance* for all extracted vertices and edges, capturing the originating substrings in dataset $\mathcal{D}$, the triggering PERs and necessary contextual cues that justify construction.

(Stage 1) Individual anchoring. When applying PERs to dataset $\mathcal{D}$, we also *latent individuals* that anchor extracted PII. PERs identify person-bearing mentions (*e.g.,* names, pronouns), cluster co-references, and attach each PII span to its most likely individual using schema-consistent cues,

with an LLM resolving residual ambiguity. This yields one `person` vertex per individual cluster, PII-type vertices, and `has_pii` edges linking individuals to PII.

(Stage 2) Inter-individual relation extraction. Given anchored individuals, we infer edges between `person` vertices using contextual cues (*e.g.,* co-location, kinship, discourse relations), followed by consistency checks (*e.g.,* symmetry for `is_neighbor`, antisymmetry for `is_parent_of`) and cross-window deduplication. LLMs are employed under the closed schema for reliable edge extraction.

Efficiency and Privacy. Our graph construction approach enables structured analysis while maintaining computational efficiency. By enforcing a closed schema and retaining strict provenance for all graph elements, we leverage lightweight, local LLMs rather than large proprietary models. As will be seen in Section 7, this reduces resource usage without introducing additional privacy risks.

Scalability. The entire graph construction pipeline is *fully automatic*, and it scales well to handle long documents that exceed the local LLM's context limit. Since both individual and relation extraction do not require global document knowledge, we adopt windowed chunk processing for cost-efficient LLM queries. In particular, we find chunking by paragraphs sufficient for `langextract` to operate reliably. This is purely an engineering choice to reduce LLM cost and latency, and to enable parallelization, not a modeling assumption.

Example 5: Consider the original input in Table 1 as dataset $\mathcal{D}$. The constructed re-identification risk graph G (Figure 1) represents: (a) an individual named "John Doe" with associated employment and location information; (b) another individual, the snippet's author; and (c) a neighbor relationship linking the two. □

Reflect on adversarial cues. To support adaptive privacy protection, SHIELD incrementally incorporates attacker-discovered vulnerabilities into the risk graph. At the beginning of round $t+1$ in the iterative workflow (Figure 1), successful re-identifications from round t are analyzed and compared against graph G_{t-1}. If a novel vulnerability is not already present, it triggers a localized update, enabling SHIELD to refine re-identification pathways.

Attacker rationales (*e.g.,* inference chains from contextual associations) are parsed into cues such as relations or attribute combinations. These are checked against G_{t-1}; if absent, the extraction module conducts local graph construction to incrementally add missing PII nodes or relations, producing the updated G_t. This targeted patching uses existing provenance, limits changes to affected subgraphs, and keeps cost low. Incorporating adversarial insights enables SHIELD to adapt efficiently to evolving attack strategies.

6 PII Anonymization

This section presents the anonymization algorithms underlying SHIELD, which operate on a dataset $\mathcal{D}$ during each round.

6.1 Anonymization Actions over Risk Graph

Given a constructed re-identification risk graph G, SHIELD applies four anonymization actions. Each action operates on G and induces a corresponding transformation on the dataset $\mathcal{D}$. These actions differ in complexity and in privacy–utility trade-offs.

Action Types. Besides standard anonymization actions such as redaction/pseudonymization and generalization, SHIELD supports two novel graph-based operations, namely, node splitting and edge removal, offering finer control over anonymization.

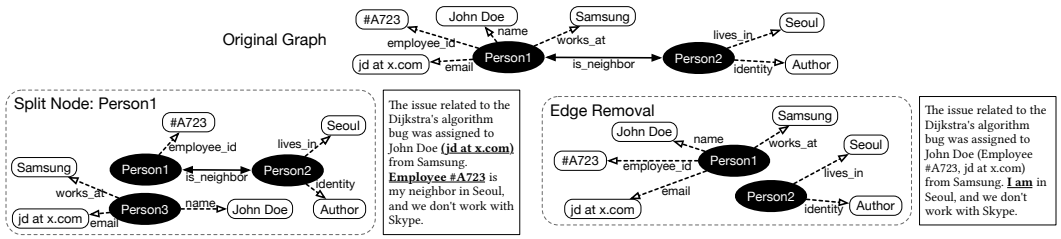


Fig. 2. Examples of node splitting and edge removal.

Redaction and pseudonymization. This action targets PII vertices in G, replacing them with tags (*e.g.,* `[NAME]`) or consistent pseudonyms (*e.g.,* hash strings), with pseudonymization preferred for preserving contextual and referential coherence. In G, this substitutes sensitive vertex attributes; in $\mathcal{D}$, it corresponds to direct token replacement. The approach is effective for direct identifiers, minimizing re-identification risk with negligible computational cost. However, its utility for quasi-identifiers is limited: outright removal or substitution can incur substantial information loss and offers little flexibility for fine-grained privacy–utility trade-offs.

Generalization. This coarsens a PII attribute (*e.g.,* "Seoul" to "Korea"), implemented in G via a predefined hierarchy or abstraction rule, and in $\mathcal{D}$ via token replacement. Generalization balances risk reduction and information retention, suited for quasi-identifiers where full redaction is too destructive. Hierarchies are derived from structured knowledge bases or dynamically via LLM prompting, and are slightly more expensive than redaction/pseudonymization.

Node splitting. This action partitions a `person` vertex in the graph G into two distinct vertices, redistributing the connected PII among them. The goal is to dissociate combinations of sensitive attributes that may jointly contribute to re-identification, thus reducing the overall risk by fragmentation. In the dataset $\mathcal{D}$, this is reflected by introducing two distinct pseudonyms for the individual. In simple cases, this requires token substitution only. In more complex cases, especially where multiple PII are syntactically entangled, LLM-based rewriting may be required to ensure coherence.

Node splitting mitigates linkage- and co-occurrence-based attacks, reducing joint re-identification risk with minimal utility loss. By separating co-occurring quasi-identifiers into distinct vertices, it reduces joint re-identification risk while preserving topical content, local syntax, and document-level semantics.

Edge removal. This action targets inter-individual relationships by removing edges between `person` vertices. These edges represent contextual links that may implicitly propagate sensitive information, such as shared affiliations or co-occurrences. In dataset $\mathcal{D}$, this action typically corresponds to rephrasing the specific evidence token (*e.g.,* "my *neighbor*") that establishes the relationship.

Edge removal, unlike PII-only anonymization, suppresses non-PII relational cues that enable re-identification, blocking auxiliary evidence propagation, preventing advanced inference attacks while preserving per-entity attributes, topical content, and local syntax.

Example 6: Continuing with Example 5, Figure 2 illustrates how node splitting and edge removal operate on the constructed re-identification risk graph G and subsequently on the dataset $\mathcal{D}$. These actions enable mild yet effective anonymization, offering greater flexibility for fine-grained privacy protection. □

Theorem 3: *Given a dataset $\mathcal{D}$ with risk graph G, a re-identification attacker $\mathcal{A}$, and an anonymization action* a *of* SHIELD*, applying action* a *neither decreases privacy nor increases utility.* □

Proof sketch: All four actions remove or coarsen identifying evidence in G, and thus yield a post-processed view with weakly less information for $\mathcal{A}$. Therefore, $\mathcal{A}$ cannot increase its re-identification success probability, so privacy risk is monotone non-increasing.

Each action takes edits, so similarity to the original $\mathcal{D}$ cannot improve, and utility cannot increase. See [39] for the full proof. □

Action impact evaluation. We define the *privacy gain* and *utility loss* to guide anonymization action selection, striking a principled balance between privacy enhancement and utility preservation.

Privacy gain $\Delta R(G, a)$. We quantify privacy impact using the notion of *excess risk*, which captures the extent to which an individual's re-identification risk surpasses the thresholds. For a target individual i in the risk graph G, the excess risk $R_i(G)$ is defined as:

$$R_i(G) = \begin{cases} \mathbb{I}[i \text{ is directly identifiable in } G], & \text{if } \mathsf{FNR}(G) > \tau_0; \\ \max\left\{0, \ln \frac{\Pr[\mathsf{DeID}_{\mathcal{A}}(i,G)]}{\min_{j \in \mathcal{I}} \Pr[\mathsf{DeID}_{\mathcal{A}}(i,G_{-j})]} - \epsilon_0\right\}, & \text{otherwise.} \end{cases} \tag{3}$$

Here $\mathcal{I}$ denotes the set of all target individuals. Intuitively, R_i quantifies the extent to which i's privacy risk exceeds either the FNR threshold τ_0 or the relative leakage threshold ϵ_0. When excess risk $R_i(G) = 0$ for all $i \in \mathcal{I}$, the privacy objective is already met with the current risk graph G. It implies both $\mathsf{FNR}(G) > \tau_0$ and $\epsilon(G) \leq \epsilon_0$.

Given an anonymization action a applied to graph G, we thus define the *privacy gain* as $\Delta R(G, a) = \sum_{i \in \mathcal{I}} (R_i(G) - R_i(G'))$, where $G' = G + a(G)$ is the updated graph. It is calculated by (a) hypothetically applying a to G to obtain G'; (b) re-running attacker $\mathcal{A}$ only for those individuals with changed incident nodes or edges, in order to obtain updated re-identification probabilities $\Pr[\mathsf{DeID}_{\mathcal{A}}(i, G')]$; and (c) computing the privacy gain $\Delta R(G, a)$ using Equation 3 as the total excess risk reduction of affected individuals.

For an edge removal action a, the privacy gain comes from blocking attribute propagation along that relation in the risk graph. We thus treat a as deleting a dependency that couples the two endpoints' attribute evidence, so the recomputed risks $R_i(G')$ reflect only paths that remain in the updated graph G'. For a node split action a, we treat it as introducing two distinct latent individuals that inherit disjoint subsets of the original incident PII, and we evaluate the risks of i_1 and i_2 separately under the attacker $\mathcal{A}$.

Utility loss $\Delta U(\mathcal{D}, a)$. Utility loss is measured as relative utility $\Delta U(\mathcal{D}, a) = U(\mathcal{D}, \mathcal{D} \oplus a(G))$ (Section 3), where $\mathcal{D} \oplus a(G)$ denotes the dataset produced by action a. It is estimated by recomputing the utility metrics on the minimally edited output of action a.

6.2 Greedy Text Anonymization

SHIELD's effectiveness critically depends on the dynamic selection of anonymization actions by the protector. Here we develop an efficient yet effective strategy for action selection at each round.

Challenges. SHIELD supports multiple anonymization actions (redaction, generalization, node splitting, edge removal), each affecting privacy and utility differently. Selecting the optimal action a_t in round t is difficult because (1) their effects are interdependent and highly non-linear, and (2) transformations are irreversible and non-commutative, so action order crucially shapes both current and future outcomes. Thus, even local choices make approximating a globally optimal privacy–utility trade-off nontrivial.

A simple approach is the greatest-risk-reduction-first heuristic, which targets PII contributing most to re-identification risk (*e.g.,* by FNR or privacy budget) and applies the most effective

Algorithm 1: Iterative PII Protection in SHIELD.

Input: Dataset $\mathcal{D}$, protector's action space A, target individual set T, thresholds τ_0 and ϵ_0.
Output: Strategy p_{RUC}.

1 Initialize $t := 1$, $\mathcal{D}_0 := \mathcal{D}$, $G_0 := \text{build}(\mathcal{D}_0)$;
2 **while** $\exists i \in \mathcal{I}$, such that $R_i(G_{t-1}) > 0$ **do**
3 **foreach** action $a \in A$ **do**
4 privacy gain $\Delta R(a) := \Delta R(G_{t-1}, a)$;
5 utility loss $\Delta U(a) := \Delta U(\mathcal{D}_{t-1}, a)$;
6 select action $a_t = \arg\max_{a \in A} \frac{\Delta R(a)}{\Delta U(a)}$; // RUC heuristic.
7 $G_t := G_{t-1} + a_t(G_{t-1})$; $\mathcal{D}_t := \mathcal{D}_{t-1} \oplus a_t(G_{t-1})$; $t := t + 1$;
8 **return** $p_{\text{RUC}} := \langle a_0, a_1, \ldots, a_{t-1} \rangle$;

mitigating action. However, it is often overly aggressive, selecting high-impact actions that over-sanitize data and severely degrade utility. Conversely, the minimum-utility-loss-first heuristic chooses actions with minimal utility loss each round. While it can eventually achieve acceptable privacy, it typically requires many steps, leading to long runtimes and slow convergence that limit practical use.

Risk-Utility Cover Heuristic. Effective anonymization requires balancing privacy and utility. We address this with a heuristic that formalizes the trade-off in an optimization framework.

The key idea to formulate the PII protection problem as a *weighted set cover* (WSC) instance. Given a universe of elements and a family of sets with nonnegative weights, WSC is to select a subfamily of *minimum total weight* whose union covers the entire universe. Here the "elements" are individuals with nonzero excess risk (Equation 3), and each anonymization action corresponds to a "set" covering those individuals whose risk can reduce to acceptable levels. Each action also incurs a "weight," measured as its utility loss.

Framing the problem this way enables a natural greedy approximation strategy: at each iteration, we select the action that maximizes the ratio of privacy gain to utility loss, akin to the greedy algorithm for the weighted set cover problem. This provides the foundation for the heuristic algorithm we present next.

Algorithm. We describe our *Risk-Utility Cover* (RUC) heuristic in Algorithm 1, which operationalizes SHIELD's workflow by incrementally selecting actions that balance privacy gain and utility loss.

The algorithm begins by constructing the initial re-identification risk graph G_0 from the dataset $\mathcal{D}$ (Line 1). It then enters an iterative process that continues until every target individual $i \in \mathcal{I}$ has no excess risk (Line 2). In each round, it evaluates all possible anonymization actions $a \in A$ (Line 3) that can be applied to the current graph G_{t-1}. For each action, it computes the privacy gain $\Delta R(a)$ (Line 4), and the utility loss $\Delta U(a)$ (Line 5).

The core of the heuristic lies in selecting the action a_t that maximizes the ratio $\frac{\Delta R(a)}{\Delta U(a)}$ (Line 6), analogous to Chvátal's greedy rule [13] for weighted set cover. Actions with zero utility loss and nonzero privacy gain are prioritized as having infinite relative gain.

Finally, the selected action is applied, updating the risk graph and dataset for the next round (Line 7). This process repeats until all excess risks are eliminated, *i.e.,* the privacy goals for both direct and quasi-identifiers are met, yielding a result strategy p_{RUC}.

Intuitively, the RUC heuristic always prefers actions that yield large privacy gains for small utility costs. For example, it will prioritize node splitting (*e.g.,* separating the "doctor," "MIT," and "02139" combination) or edge removal (*e.g.,* obfuscating "my neighbor"), since both sharply reduce re-identification risk while preserving topical content. In contrast, it tends to avoid heavy-handed

Table 3. Summary of benchmarking datasets.

Dataset	Type	#Docs	Avg. Doc Length	Domain
ai4privacy [1]	Synthetic	150,693	17.0 words	General
GretelAI [29]	Synthetic	5,000	28.17 words	General
TAB [58]	Real-world	254	859.0 words	Legal

edits with high semantic cost (*e.g.,* direct redaction) and minor tweaks with negligible benefit (*e.g.,* generalizing "$18.5" to "$18–19"). It drives steady risk reduction per unit utility spent, ensuring progress toward the privacy goal without unnecessary utility degradation.

The RUC heuristic greedily selects, in each round, the action maximizing excess-risk reduction per unit utility loss, mirroring Chvátal's greedy algorithm [13] for weighted set cover. Here excess-risk units across all $i \in \mathcal{I}$ serve as elements, and each action a covers $\Delta R(a)$ units at cost $\Delta U(a)$, ensuring steady progress toward the privacy goal with minimal utility loss.

Example 7: Consider the original text in Table 1. We evaluate candidate actions: a_1 redacts "Samsung" to <ORG>, a_2 masks "Seoul" as <CITY>, and a_3 generalizes "Seoul" to "Korea." Suppose the estimated privacy gains are $\Delta R(a_1) = 6$, $\Delta R(a_2) = 3$, and $\Delta R(a_3) = 2$. The utility losses for a_1 and a_2 are high: $\Delta U(a_1) = 6$ and $\Delta U(a_2) = 6$, due to contextual information loss, whereas a_3 incurs a low utility loss, $\Delta U(a_3) = 1$. RUC selects the action that maximizes $\Delta R/\Delta U$ per round; the ratios are 1.0, 0.5, and 2.0 for a_1, a_2, and a_3, respectively. Hence, RUC chooses a_3, which reduces re-identification risk while preserving regional context with minimal utility loss. □

Approximation ratio. Algorithm 1 achieves a constant-factor approximation of the optimal solution, as established by Chvátal's analysis of the weighted set cover problem [13]. Enforcing the privacy goal, RUC approximates the optimal solution within a factor of $H(R_{\max})$, where $H(k)$ is the k-th harmonic number and $R_{\max}$ denotes the maximum risk reduction by any single action.

This performance guarantee improves upon the naive "largest-risk-first" heuristic, which considers only the absolute risk reduction and ignores the associated utility cost. In contrast, RUC explicitly optimizes the trade-off between privacy gain and utility loss, steering the solution toward a more balanced outcome.

We now formally prove the approximation ratio of Algorithm 1. Let p^* be the optimal sequence of actions minimizing total utility loss while satisfying all privacy constraints. Let U^* be the total utility loss of p^*, p_{RUC} the sequence produced by Algorithm 1, and U_{RUC} its total utility loss. Let $R_{\max} = \lceil \max_{a \in A} \Delta R(a) \rceil$ be the maximum risk reduction achievable by a single action. Applying Chvátal's result to our setting yields the following theorem.

Theorem 4: *The RUC heuristic (Algorithm 1) produces a solution p_{RUC} whose total utility loss U_{RUC} satisfies $U_{RUC} \leq H(R_{max}) \cdot U^*$, where $H(k) = \sum_{i=1}^{k} \frac{1}{i}$ is the k-th harmonic number.* □

Proof sketch: Following WSC [13], we treat excess risk as the elements to be covered. Each action a covers $\Delta R(a)$ units at cost $\Delta U(a)$, and RUC greedily selects the minimum cost-per-unit-risk action. This bounds the marginal cost by $U^*/R^{(t)}$, where $R^{(t)}$ is the remaining risk and U^* is the optimal cost. Summing over rounds yields a total utility loss of at most $H(R_{\max}) \cdot U^*$. □

7 Experimental Study

Using real-life and synthetic data, we evaluated SHIELD on (a) PII detection accuracy, (b) anonymization utility, (c) explainability, (d) cost effectiveness, and (e) robustness across diverse configurations.

Experimental settings. We start with the settings.

Datasets. We used three widely adopted benchmark datasets for evaluating PII protection systems, summarized in Table 3: (a) `ai4privacy` [1] is a large synthetic English corpus of ~150k short documents designed to stress-test anonymization at scale. (b) `GretelAI` [29], the Gretel PII masking en-v1 dataset [29] (test split), contains 5k synthetic documents with diverse structures, serving as an industrial benchmark for PII detection. (c) `TAB` [58] (test and dev splits) comprises 254 court cases with manually verified annotations, capturing domain-specific PII in legal texts.

Baselines. We compared SHIELD with the SOTA PII protection systems, spanning all 3 categories of PII anonymizers (see Section 2).

- *PII scrubbers*, using various systems for PII identification:
  (1) Presidio [49], a SOTA hybrid system with built-in PII detection rules and integrated SpaCy [65] as the NLP module.
  (2) RoBERTa [14], a general NER PII identification model.
  (3) Gretel GLiNER [29], an NER model for PII detection, fine-tuned on the 50k train split of the `GretelAI` dataset.
- *PPDP* (privacy-preserving data publishing) methods (4) DP-Prompt [75], and (5) DP-MLM [46]. Both provide DP guarantees, under the same privacy parameter ϵ as SHIELD. We omit systems with custom privacy metrics as DP is the *de-facto* standard.
- The SOTA *LLM-based rewriter* Staab *et al.* [68]. It is backed by a leading reasoning LLM, either (6) the open-weight Qwen3-235B-A22B [78], or (7) the proprietary GPT-5 [54]. The LLMs were accessed via their official API endpoints, under default settings.

SHIELD configurations. By default, the PER engine integrates Presidio, a RoBERTa NER model, and Gretel GLiNER as tool/NER predicates. For attack simulation it supports both statistical inference [82] and LLM-based attackers [67]. We assume attackers may hold background knowledge about target individuals, modeled as a prior probability distribution $p(\cdot)$ over known quasi-identifiers. By default, the privacy parameter is set to $e^{\epsilon} = 10$. Its LLM backend is `qwen3:8b`.

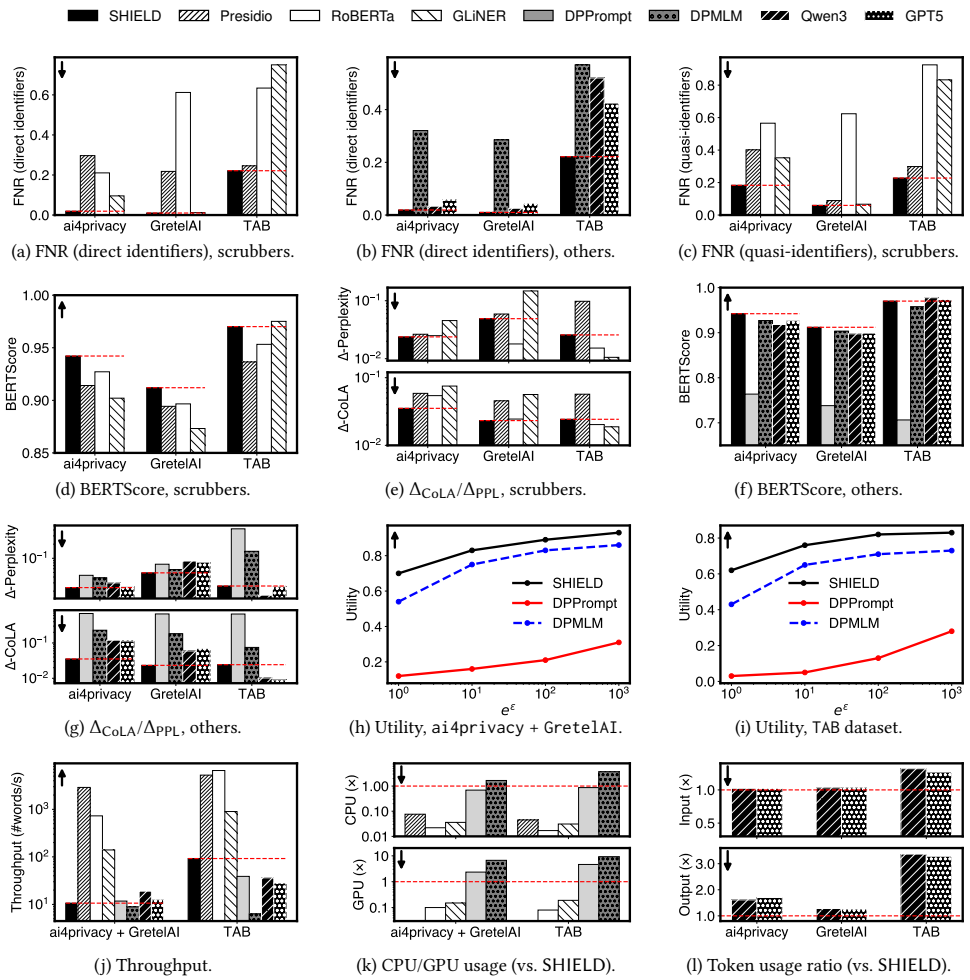
PERs. We used the same set of PERs in experiments to show that these rules are dataset-agnostic. The PER set is learned from the 50k training samples of the `GretelAI` dataset [29]. It consists of 63 rules (see [39]), all of which are reasonably simple; the average, maximum, and minimum number of predicates are 3.8, 8, and 2, respectively. We observe no conflicts among the learned rules.

Testbed. Our primary testbed is a consumer-grade workstation powered by an Intel Core i7-11700 @2.50GHz CPU, with 8 physical cores and 64 GB DDR4 RAM. It is also equipped with an NVIDIA GeForce RTX 3090 24GB GPU. Unless specified otherwise, SHIELD is backed by a local LLM through the vLLM inference server [35].

Experimental results. We next report our findings. To verify that SHIELD addresses the challenges outlined earlier, we measured FNs in PII detection (Exp-1), quantified the effect of FPs and over-sanitization via utility loss (Exp-2), and justify explainability with auditable logs (Exp-3). We also empirically evaluated its operations cost (Exp-4) and explore its optimal configurations (Exp-5).

Exp-1: Privacy. We compared SHIELD with all baselines. For de-identifier systems, we measured the FNR of PII detection. For PPDP methods and LLM-based rewriters, we examined whether a specific direct identifier remained in the final anonymized output by exact string matching. If detected, it was counted as a false negative (FN).

Accuracy of direct identifier detection. Figures 3a–3b report the results across the three datasets. We find the following:



(a) FNR (direct identifiers), scrubbers. (b) FNR (direct identifiers), others. (c) FNR (quasi-identifiers), scrubbers.

(d) BERTScore, scrubbers. (e) $\Delta_{\text{CoLA}}/\Delta_{\text{PPL}}$, scrubbers. (f) BERTScore, others.

(g) $\Delta_{\text{CoLA}}/\Delta_{\text{PPL}}$, others. (h) Utility, `ai4privacy` + `GretelAI`. (i) Utility, `TAB` dataset.

(j) Throughput. (k) CPU/GPU usage (vs. SHIELD). (l) Token usage ratio (vs. SHIELD).

Fig. 3. Evaluation of PII protection systems. Arrows (↑/↓) indicate whether higher or lower values are preferred.

(1) SHIELD achieves FNRs of 1.0%, 1.9% and 22.1% on `ai4privacy`, `GretelAI` and `TAB`, respectively, lower than all baselines. Compared with de-identifiers, it reduces false negatives by 9.9–98.4% through PERs that fuse multi-detector signals for comprehensive decisions. It further lowers FNs of DP-MLM and LLM rewriters by 42.9–96.5%, *i.e.*, SHIELD identifies sensitive PII more reliably than SOTA LLMs, particularly in lengthy documents containing frequent PII instances.

(2) DP-Prompt (omitted from the figures) removes >99.0% of direct identifiers across all datasets. However, this comes at the cost of unacceptably low-utility output (see Exp-2). For example, it transforms "21/01/1999 is the birthday of Si Anayra" (from `ai4privacy`) into "Tr depending arise snugglkinderpreis Honda1921997 Programragekindouche", which is entirely incomprehensible.

Accuracy of quasi-identifier detection. Figure 3c shows the performance on quasi-identifiers. As expected, SHIELD attains much higher accuracy than all PII scrubbers. It achieves 23.7–54.4%, 67.6–90.5% and 10.9–72.6% lower FNRs on `ai4privacy`, `GretelAI` and `TAB`, respectively. Unlike scrubbers that redact all detected quasi-identifiers, SHIELD may retain instances whose estimated

re-identification risk is negligible. A lower FNR improves risk awareness, thereby enabling a more balanced privacy–utility trade-off.

Because PPDP and LLM-based methods anonymize text end-to-end, evaluating their accuracy on quasi-identifiers is not meaningful. We therefore omit the accuracy metrics for these baselines.

Cross-dataset generalizability. Learned from the 50k GretelAI training samples solely, the PER set transfers effectively to unseen datasets. On ai4privacy and the domain-shifted TAB legal documents, SHIELD attains consistently lower FNRs than all integrated tools, supporting the claim that the learned PERs generalize beyond the training distribution and are dataset-agnostic.

In contrast, the NER models show poor cross-dataset generalizability. GLiNER performs strongly on GretelAI (FNR 1.2% for direct and 3.0% for quasi-identifiers) but its FNR soars above 75.0% on TAB. RoBERTa exhibits FNRs exceeding 21.1% across all datasets.

Exp-2: Utility. We next compare the utility of the anonymized text, in terms of average BERTScore, CoLA delta, and perplexity delta.

Compared with PII scrubbers. As shown in Figures 3d–3e, SHIELD attains 0.91–0.97 BERTScore, < 0.035 CoLA delta, and <0.049 Perplexity delta across all datasets, demonstrating strong semantic preservation and linguistic quality. It surpasses PII scrubbers by up to 4.4 points, 78.8% and 73.7% on the three metrics, respectively. This stems from SHIELD 's selective anonymization: unlike de-identifiers that indiscriminately mask all detected PII, causing severe information loss, SHIELD anonymizes only when necessary, leveraging attack awareness to safely retain non-exploitable identifiers, despite detecting them more accurately.

Two exceptions arise: RoBERTa (GretelAI/TAB) and GLiNER (TAB) show slightly higher utility (lower CoLA/perplexity deltas) by retaining over 60% of direct and 80% of quasi-identifiers. However, such "utility" is illusory, since substantial sensitive content remains exposed. This underscores SHIELD 's balanced design, which aims to improve utility only after ensuring privacy.

Compared with LLM-based rewriters. Figures 3f–3g show that the LLM-based anonymizer [68] yields strong performance in terms of preserving utility. This is expected, as they rewrite the text into fluent and grammatically correct sentences. Yet SHIELD still beats them by up to 2.6 points in BERTScore, 60.8% in CoLA delta, and 44.0% in Perplexity delta. This is because SHIELD minimizes rewriting and applies local replacements whenever possible, thus preserving semantics of the original text more faithfully.

Qwen3 and GPT-5 appear to achieve higher utility than SHIELD on TAB (Figure 3g); however, this is largely illusory, as both models fail to anonymize over 40% of direct identifiers (Figure 3b). In contrast, SHIELD detects and pseudonymizes more than 80% of these highly sensitive tokens. Although this may slightly reduce linguistic fluency, it is essential for meeting privacy guarantees.

Compared with PPDP methods. SHIELD consistently offers higher utility than DP-based DP-Prompt and DP-MLM across all three metrics, as shown in Figures 3e–3g. It substantially outperforms DP-Prompt, which produces highly incoherent text. Compared to DP-MLM, it yields similar BERTScores (<1.6 points higher); however, it is up to 87.3% and 81.7% lower on CoLA and Perplexity delta, respectively. Although both SHIELD and DP anonymizers operate under the same differential privacy guarantee ($e^{\epsilon} = 10$), the latter often generate semantically problematic text which undermines readability and fluency measured by the latter two metrics.

Figures 3h–3i show the effect of the privacy parameter ϵ on utility. SHIELD consistently outperforms both DP anonymizers. As e^{ϵ} increases from 1 to 1000, utility rises monotonically across all systems. SHIELD mirrors the trend of DP-MLM but exceeds it by 8.1–44.2% across datasets and settings. DP-Prompt remains unusable even under the most permissive (high-risk) setting of

$e^{\epsilon} = 1000$. Overall, SHIELD delivers strong privacy guarantees without compromising semantic fidelity or linguistic quality, confirming its effectiveness in mitigating over-sanitization.

Exp-3: Explainability. SHIELD's decision-making process is fully transparent and explainable via its trace logs, as shown below. Each protection decision is attributed to a concrete attacker with quantitative evidence, *e.g.,* the probability of matching a specific individual and the implied k-anonymity level. The log first enumerates candidate anonymization actions with scores and predicted utility impact, then records the selected action. Finally, a post-action verification reports residual risk and realized utility change, enabling external auditors to reproduce the decision path.

```
1. [TRACE] pii_id=DOC173:Span("Mhedi Phong", type=NAME, loc=9-20)
2. [ATTACKER] linkage_attack:v1 risk=0.87 evidence={match_prob=0.93, k_anonymity=1, background_attrs=["17th
     Birthday: 27/11/1976"]}
3. [DECISION] candidates:
  - REDACT score=0.62 expected_utility_delta=-0.31
  - MASK(NAME->[PERSON]) score=0.78 expected_utility_delta=-0.12
  - PSEUDONYMIZE(NAME->"Alex A3C8") score=0.84 expected_utility_delta=-0.06
4. [ACTION] apply SYNTHESIZE(NAME->"Alex A3C8") to span:9-20
5. [VERIFICATION] post_risk=0.12 realized_utility_delta=-0.06
```

The sample log above illustrates a representative snippet of SHIELD's decision-making process. It (1) identifies the precise PII span under review in the document, (2) invokes an explicit attacker model to quantify re-identification risk from the available evidence, (3) enumerates a small set of candidate protections together with their predicted utility impact, (4) selects and applies the highest-scoring action to the span, and (5) verifies that the action reduces residual risk while incurring the expected utility change.

Exp-4: Cost. We next examine the cost of PII protection systems.

Efficiency. Figure 3j reports system throughput, measured as the average number of processed words per second. As micro-benchmarks, Figure 3k compares total CPU time and local GPU usage, computed as average SM Efficiency multiplied by execution time, with SHIELD as the baseline. (1) PII scrubbers achieve orders-of-magnitude higher throughput than all other systems, since their lightweight detectors and limited anonymization actions require 92.4–96.9% less CPU and 81.0–99.9% less GPU usage. (2) DP rewriters consume 2.4–9.2× more GPU resources than SHIELD, leading to up to 14.6× lower throughput. (3) SHIELD processes short `ai4privacy` and `GretelAI` documents at 10.6 words/s, 15.0–32.0% slower than LLM-based rewriters. The gap is primarily due to the longer Time-To-First-Token of the local LLM in SHIELD, relative to cloud APIs. Even so, the absolute difference in end-to-end execution time is under one second.

While all systems (except DP-MLM) process longer documents faster, SHIELD scales much better: on `TAB`, for instance, it is 54.1% and 27.3% faster than Qwen3 and GPT-5, respectively. Its throughput improves by 8.69×, whereas others increase by less than 2.82×. These demonstrate that SHIELD is both efficient and scalable in practice, while maintaining the best privacy–utility balance.

Monetary cost. Figure 3l shows the relative token usage of LLM-based anonymizers, normalized to that of SHIELD: on `TAB`, Qwen3 (resp. GPT-5) consumes 32.0% (resp. 26.9%) more input tokens and 238.8% (resp. 228.4%) more output tokens than SHIELD. Based on official cloud API pricing, this translates to 25.7× (resp. 31.0×) higher monetary cost. In practice, SHIELD can run locally on a desktop with a consumer-grade GPU, incurring zero API expenses.

SHIELD thus offers two clear advantages. (1) It is highly cost-efficient, requiring significantly less monetary resources to achieve equal or superior privacy and utility. (2) It supports local deployment with open-weight models, eliminating the privacy risks of sending sensitive documents to external LLM service providers.

Table 4. Ablation studies. FNR_{DI} and FNR_{all} refer to the FNR of detecting direct identifier and all PII, respectively. "–" indicates no change.

Config	ai4privacy					GretelAI					TAB				
	FNR_{DI}	FNR_{all}	Utility	Time	LLM($)	FNR_{DI}	FNR_{all}	Utility	Time	LLM($)	FNR_{DI}	FNR_{all}	Utility	Time	LLM($)
NoPresidio	+0.034	+0.082	+1.2	-4.1%	0.98×	+0.020	+0.013	+0.1	-3.2%	0.99×	**+0.229**	**+0.250**	**+3.3**	**-22.0%**	0.95×
NoRoBERTa	+0.047	+0.054	+0.5	-4.5%	0.99×	+0.010	+0.005	+0.0	-4.2%	1.00×	+0.019	+0.028	+0.3	-2.0%	0.99×
NoGLiNER	+0.054	+0.087	+0.6	**-8.7%**	0.98×	**+0.161**	**+0.102**	**+1.8**	**-16.6%**	0.97×	+0.006	+0.035	+0.3	-6.2%	0.98×
NTool	+0.021	+0.034	+0.8	-1.5%	1.02×	+0.016	+0.009	+0.2	-1.3%	0.97×	+0.031	+0.038	+0.7	-0.8%	1.08×
NModel	+0.014	+0.030	+0.3	-3.1%	0.98×	+0.026	+0.039	+0.3	-5.1%	0.96×	+0.010	+0.012	+0.1	-2.0%	0.99×
NoNodeSplit	–	–	**-1.3**	+2.1%	–	–	–	**-1.6**	+1.8%	–	–	–	**-3.2**	+3.6%	–
NoEdgeRemoval	–	–	-0.6	+0.8%	0.98×	–	–	-0.8	+0.9%	0.98×	–	–	**-5.6**	+3.7%	0.93×
UtilGreedy	–	–	+0.1	**+46.2%**	1.08×	–	–	-0.0	**+51.6%**	1.07×	–	–	+0.2	**+103.3%**	**1.18×**
PrivacyGreedy	–	–	**-3.4**	**-6.9%**	1.00×	–	–	**-3.2**	**-8.1%**	0.99×	–	–	**-5.6**	**-18.5%**	0.95×
StatInfOnly	+0.013	+0.031	**-2.9**	**-18.1%**	0.92×	+0.007	+0.016	**-2.2**	**-17.0%**	0.94×	+0.017	+0.042	**-3.5**	**-26.9%**	0.86×
LLMInfOnly	–	–	-1.0	-1.5%	1.00×	–	–	-0.9	-1.3%	1.01×	–	–	**-2.1**	-1.9%	1.03×
LLM:4b	+0.011	+0.027	**-1.9**	-3.28%	**0.61×**	-0.001	+0.012	**-2.0**	-3.76%	**0.61×**	+0.009	+0.038	**-2.8**	-6.90%	**0.62×**
LLM:235b	+0.006	+0.003	+0.1	+6.8%	**3.79×**	-0.005	+0.000	-0.2	+6.5%	**3.75×**	**-0.010**	-0.008	-0.2	**+8.4%**	**3.72×**
LLM:gpt5	**-0.020**	**-0.026**	+0.3	**+9.4%**	**5.31×**	**-0.011**	**-0.010**	+0.5	**+9.0%**	**5.43×**	**-0.036**	**-0.029**	+0.2	**+12.1%**	**5.21×**

Exp-5: Ablation study. Table 4 summarizes key metrics of SHIELD under alternative configurations. This ablation study isolates the contribution of each component and identifies the optimal configuration that balances privacy, utility, and cost.

Impact of integrated detectors. We tested three variants: NoPresidio, NoRoBERTa and NoGLiNER, by disabling the corresponding PII detectors in PERs while keeping other components unchanged. (1) Removing detectors raises the FNR for direct identifiers by up to 5.4, 16.1 and 22.9 percentage points on `ai4privacy`, `GretelAI` and `TAB`, respectively, with similar increases for overall PII. (2) Each detector contributes differently across datasets, highlighting the value of unified integration. (3) Utility remains largely unchanged, except for NoPresidio on `TAB` and NoGLiNER on `GretelAI`, where the apparent gains stem from higher FNRs. (4) Runtime drops by 3.2–22.0% due to fewer tool and candidate action evaluations, with negligible impact on LLM cost. Overall, combining all three detectors yields the best privacy–utility balance at minimal cost.

Sensitivity to detector errors. To assess robustness to detector noise, we evaluated NTool and NModel with injected random label flips, changing 10% of positive and 2% of negative detections. NTool flips Presidio's binary outputs, whereas NModel applies them to the confidence scores from RoBERTa and GLiNER, by setting the selected scores to 0.5. Overall FNR increases by at most 3.9%, and utility varies by at most 0.8 points, well below the injected noise rate. These confirm that PERs mitigate cascading detector errors.

Impact of novel actions. NoNodeSplit and NoEdgeRemove disable node-splitting and edge-removal actions, respectively. Both exhibit noticeably lower utility (up to 5.6 points) and slightly longer runtime (~3.7% slower) on `TAB`. Their impact on short documents is limited, as these actions are seldom required when only a single individual appears in the text. These confirm that SHIELD's graph-driven anonymization actions enhance privacy gains while minimizing redundant redactions, particularly in long documents.

Impact of action selection strategy. We assessed UtilGreedy, which picks the highest-utility action at each step, and PrivacyGreedy, which selects the highest-privacy action, against the default RUC-guided heuristic. UtilGreedy maintains utility comparable to the default (within 0.2 points) but increases runtime by 46.2–103.3% and LLM cost by up to 18%, as it conservatively explores

benign rewrites. In contrast, PrivacyGreedy shortens runtime by 6.9–18.5% but lowers utility by 3.2–5.6 points, particularly on TAB where over-sanitization degrades fluency and content retention. This justifies our RUC heuristic, which balances utility and performance.

Impact of attackers. We tested StatInfOnly (statistical inference only) and LLMInfOnly (LLM inference only) against the default hybrid setup, treating them as pluggable modules. Adding stronger attackers strengthens protection without requiring architectural changes. StatInfOnly slightly raises FNRs by 0.7–4.2 points and reduces utility by up to 3.5 points, primarily due to mischaracterized vulnerabilities. It cuts runtime by up to 26.9% and LLM cost by up to 14% through fewer inference calls. Conversely, LLMInfOnly shows a modest utility drop (0.9–2.1 points) but incurs up to 3% higher cost, especially on long TAB documents. Overall, the hybrid setup best balances privacy, utility, and efficiency, while StatInfOnly provides a low-cost alternative with minor accuracy and utility degradation.

Choice of LLMs. We compared open-weight `qwen3:4b` (`LLM:4b`), Qwen3-235B-A22B (`LLM:235b`), and proprietary GPT-5 (`LLM:gpt5`) against the default `qwen3:8b` backend, all accessed via cloud APIs for consistency. The smaller `LLM:4b` lags behind the default by 1.9–2.8 utility points despite being faster and cheaper. The two larger models achieve only marginal FNR reductions (<3.6 points) and minor utility gains (<0.5 points), while increasing runtime by up to 12.1% and monetary cost by 3.72–5.43×. These results confirm that SHIELD does not depend on frontier LLMs; a local 8B model offers the most balanced trade-off among privacy, utility, and cost.

Summary. SHIELD addresses key challenges of PII protection:

- *Accurate PII Identification.* Detect PII accurately on synthetic and real datasets, cutting FNRs by up to 98.4% over the SOTA.
- *High-Utility Anonymization.* Produce fluent outputs with minimal edits, yielding utility gains of up to 4.4, 62.1 and 2.6 points over de-identifiers, PPDP and LLM-based rewriters, respectively.
- *Transparency and Auditability.* Provide structured logs of attacker evidence, candidate actions, selected actions, and verification, ensuring traceability and explainability.
- *Balanced Trade-offs.* Improve utility levels at a fixed privacy target, while reducing runtime by up to 54.1% and cost by 31.0×.
- *Actionable Insights.* Ablations recommend: combining diverse detectors, RUC-guided action selection, hybrid attackers, and a local `qwen3:8b` backend for balanced privacy–utility–cost.

8 Conclusion

This work proposes (1) an attack-aware approach for explicitly assessing re-identification risks; (2) a PII protection model that balances privacy and utility via a zero-sum Stackelberg game, together with a proof of the intractability of the bi-criteria optimization problem; (3) SHIELD, a multi-round PII protection framework that guarantees a constant-factor approximation to the optimal trade-off; (4) the notion of a re-identification risk graph and a class of PII extraction rules that consolidate logical reasoning, NER predictions, and existing tools to identify context-dependent PII; and (5) new anonymization operations for balancing the privacy–utility trade-off. Our experiments verify that SHIELD is effective in practice.

Future work includes (a) extending SHIELD to protect PII in non-textual data, such as images and audio for multi-modal privacy protection, and (b) optimizing SHIELD to further improve scalability, enabling efficient PII detection and scrubbing in large-scale datasets.

References

[1] AI4Privacy. 2025. English Anonymiser OpenPII (Ai4Privacy) on HuggingFace. https://huggingface.co/ai4privacy/llama-ai4privacy-english-anonymiser-openpii

[2] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An Easy-to-Use Framework for State-of-the-Art NLP. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. 54–59.

[3] Balamurugan Anandan, Chris Clifton, Wei Jiang, Mummoorthy Murugesan, Pedro Pastrana-Camacho, and Luo Si. 2012. T-Plausibility: Generalizing Words to Desensitize Text. *Transaction on Data Privacy (TDP)* 5, 3 (Dec. 2012), 505–534.

[4] Marcelo Arenas, Leopoldo Bertossi, and Jan Chomicki. 1999. Consistent Query Answers in Inconsistent Databases. In *Symposium on Principles of Database Systems (PODS)*. 68–79.

[5] Shubhi Asthana, Ruchi Mahindru, Bing Zhang, and Jorge Sanz. 2025. Adaptive PII Mitigation Framework for Large Language Models. In *AAAI Workshop on Privacy-Preserving Artificial Intelligence (PPAI)*. arXiv:2501.12465 [cs]

[6] Vanessa Ayala-Rivera, Patrick McDonagh, Thomas Cerqueus, Liam Murphy, and Christina Thorpe. 2017. Enhancing the Utility of Anonymized Data by Improving the Quality of Generalization Hierarchies. *Transations on Data Privacy (TDP)* 10, 1 (April 2017), 27–59.

[7] California State Legislature. 2018. California Consumer Privacy Act. https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375

[8] Rosario Catelli, Valentina Casola, Giuseppe De Pietro, Hamido Fujita, and Massimo Esposito. 2021. Combining Contextualized Word Representation and Sub-Document Level Analysis through Bi-LSTM+CRF Architecture for Clinical de-Identification. *Knowledge-Based Systems* 213 (Feb. 2021), 106649.

[9] Venkatesan T. Chakaravarthy, Himanshu Gupta, Prasan Roy, and Mukesh K. Mohania. 2008. Efficient Techniques for Document Sanitization. In *ACM Conference on Information and Knowledge Management (CIKM)*. 843–852.

[10] Shenglin Chen, Wenfei Fan, and Ruochun Jin. 2025. Outliers: The Good, the Bad and the Ugly. *Proc. ACM Manag. Data (SIGMOD)* 3, 4 (2025), 259:1–259:30.

[11] Cheng-Han Chiang and Hung-yi Lee. 2023. Can Large Language Models Be an Alternative to Human Evaluations?. In *Annual Meeting of the Association for Computational Linguistics (ACL)*. 15607–15631.

[12] Jason P.C. Chiu and Eric Nichols. 2016. Named Entity Recognition with Bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics (TACL)* 4 (2016), 357–370.

[13] Vasek Chvatal. 1979. A Greedy Heuristic for the Set-Covering Problem. *Mathematics of Operations Research (MOOR)* 4, 3 (1979), 233–235.

[14] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-Lingual Representation Learning at Scale. In *Annual Meeting of the Association for Computational Linguistics (ACL)*. 8440–8451.

[15] M. Mikail Demir, Hakan T. Otal, and M. Abdullah Canbaz. 2025. LegalGuardian: A Privacy-Preserving Framework for Secure Integration of Large Language Models in Legal Practice. arXiv:2501.10915 [cs] http://arxiv.org/abs/2501.10915

[16] Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. 2017. De-Identification of Patient Notes with Recurrent Neural Networks. *Journal of the American Medical Informatics Association (JAMIA)* 24, 3 (2017), 596–606.

[17] Yao Dou, Isadora Krsek, Tarek Naous, Anubha Kabra, Sauvik Das, Alan Ritter, and Wei Xu. 2024. Reducing Privacy Risks in Online Self-Disclosures with Language Models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*. 13732–13754.

[18] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography (TTC)*. 265–284.

[19] Cynthia Dwork and Aaron Roth. 2014. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science* 9, 3–4 (Aug. 2014), 211–407.

[20] European Parliament and Council of the European Union. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC (General Data Protection Regulation). 88 pages. https://eur-lex.europa.eu/eli/reg/2016/679/oj

[21] Wenfei Fan, Floris Geerts, Xibei Jia, and Anastasios Kementsietsidis. 2008. Conditional functional dependencies for capturing data inconsistencies. *ACM Trans. Database Syst. (TODS)* 33, 2 (2008), 6:1–6:48.

[22] Wenfei Fan, Ping Lu, and Chao Tian. 2020. Unifying Logic Rules and Machine Learning for Entity Enhancing. *Sci. China Inf. Sci.* 63, 7 (2020).

[23] Wenfei Fan, Chao Tian, Yanghao Wang, and Qiang Yin. 2021. Parallel Discrepancy Detection and Incremental Detection. *Proceedings of the VLDB Endowment (PVLDB)* 14, 8 (2021), 1351–1364.

[24] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A Deep Semantic Natural Language Processing Platform. In *Workshop for NLP Open Source Software (NLPOSS)*. 1–6.

[25] Michael Garey and David Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company.

[26] Akshay Goel. 2025. LangExtract. https://github.com/google/langextract

[27] Philippe Golle. 2006. Revisiting the Uniqueness of Simple Demographics in the US Population. In *ACM Workshop on Privacy in Electronic Society (WPES)*. 77–80.

[28] Government of Canada. 2000. Personal Information Protection and Electronic Documents Act. https://laws-lois.justice.gc.ca/eng/acts/P-21/FullText.html

[29] Gretel AI. 2024. GLiNER Models for PII Detection through Fine-Tuning on Gretel-Generated Synthetic Documents. https://huggingface.co/datasets/gretelai/gretel-pii-masking-en-v1

[30] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2025. A Survey on LLM-as-a-Judge. arXiv:2411.15594 [cs] http://arxiv.org/abs/2411.15594

[31] Ykä Huhtala, Juha Kärkkäinen, Pasi Porkka, and Hannu Toivonen. 1999. TANE: An efficient algorithm for discovering functional and approximate dependencies. *Comput. J.* (1999).

[32] Timour Igamberdiev and Ivan Habernal. 2023. DP-BART for Privatized Text Rewriting under Local Differential Privacy. In *Findings of the Association for Computational Linguistics (ACL)*. 13914–13934.

[33] Jacob Imola, Takao Murakami, and Kamalika Chaudhuri. 2022. Differentially Private Triangle and 4-Cycle Counting in the Shuffle Model. In *ACM Conference on Computer and Communications Security (CCS)*. 1505–1519.

[34] Alistair E. W. Johnson, Lucas Bulgarelli, and Tom J. Pollard. 2020. Deidentification of Free-Text Medical Records Using Pre-Trained Bidirectional Transformers. In *ACM Conference on Health, Inference, and Learning (CHIL)*. 214–221.

[35] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Symposium on Operating Systems Principles (SOSP)*. 611–626.

[36] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. 74–81.

[37] Pierre Lison, Ildikó Pilán, David Sanchez, Montserrat Batet, and Lilja Øvrelid. 2021. Anonymisation Models for Text Data: State of the Art, Challenges and Future Directions. In *Annual Meeting of the Association for Computational Linguistics (ACL)*. 4188–4203.

[38] Leibo Liu, Oscar Perez-Concha, Anthony Nguyen, Vicki Bennett, Victoria Blake, Blanca Gallego, and Louisa Jorm. 2023. Web-Based Application Based on Human-in-the-Loop Deep Learning for Deidentifying Free-Text Data in Electronic Medical Records: Development and Usability Study. *Interactive Journal of Medical Research (IJMR)* 12 (Aug. 2023), e46322.

[39] Shuhao Liu, Wenfei Fan, and Yijia Xu. 2026. Full version paper with appendices. https://shuhaoliu.github.io/assets/papers/shuhao-sigmod26-shield-full.pdf.

[40] Xin Liu, Farima Fatahi Bayat, and Lu Wang. 2024. Enhancing Language Model Factuality via Activation-Based Confidence Calibration and Guided Decoding. arXiv:2406.13230 [cs] http://arxiv.org/abs/2406.13230

[41] Zengjian Liu, Buzhou Tang, Xiaolong Wang, and Qingcai Chen. 2017. De-Identification of Clinical Notes via Recurrent Neural Network and Conditional Random Field. *Journal of Biomedical Informatics (JBI)* 75 (Nov. 2017), S34–S42.

[42] Gabriel Loiseau, Damien Sileo, Damien Riquet, Maxime Meyer, and Marc Tommasi. 2025. Tau-Eval: A Unified Evaluation Framework for Useful and Private Text Anonymization. arXiv:2506.05979 [cs] http://arxiv.org/abs/2506.05979

[43] Courtney Mansfield, Amandalynne Paullada, and Kristen Howell. 2022. Behind the Mask: Demographic Bias in Name Detection for PII Masking. In *Workshop on Language Technology for Equality, Diversity and Inclusion (LTEDI)*. 76–89.

[44] Justus Mattern, Benjamin Weggenmann, and Florian Kerschbaum. 2022. The Limits of Word Level Differential Privacy. In *Findings of the Association for Computational Linguistics (NAACL)*. 867–881.

[45] Ehrmann Maud, Hamdi Ahmed, Elvys Linhares Pontes, Romanello Matteo, and Doucet Antoine. 2023. Named Entity Recognition and Classification in Historical Documents: A Survey. *Comput. Surveys* (Sept. 2023).

[46] Stephen Meisenbacher, Maulik Chevli, Juraj Vladika, and Florian Matthes. 2024. DP-MLM: Differentially Private Text Rewriting Using Masked Language Models. In *Findings of the Association for Computational Linguistics (ACL)*. 9314–9328.

[47] Vincent Menger, Floor Scheepers, Lisette Maria Van Wijk, and Marco Spruit. 2018. DEDUCE: A Pattern Matching Method for Automatic de-Identification of Dutch Medical Text. *Telematics and Informatics* 35, 4 (July 2018), 727–736.

[48] Michele Miranda, Elena Sofia Ruzzetti, Andrea Santilli, Fabio Massimo Zanzotto, Sebastien Bratieres, and Emanuele Rodola. 2025. Preserving Privacy in Large Language Models: A Survey on Current Threats and Solutions. *Transactions on Machine Learning Research (TMLR)* (2025).

[49] Microsoft. 2025. Presidio. https://microsoft.github.io/presidio/

[50] John Morris, Justin Chiu, Ramin Zabih, and Alexander Rush. 2022. Unsupervised Text Deidentification. In *Findings of the Association for Computational Linguistics (EMNLP)*. 4777–4788.

[51] Ishna Neamatullah, Margaret M Douglass, Li-wei H Lehman, Andrew Reisner, Mauricio Villarroel, William J Long, Peter Szolovits, George B Moody, Roger G Mark, and Gari D Clifford. 2008. Automated De-Identification of Free-Text Medical Records. *BMC Medical Informatics and Decision Making* 8, 1 (Dec. 2008), 32.

[52] Bekelu Negash, Alan Katz, Christine J. Neilson, Moniruzzaman Moni, Marcello Nesca, Alexander Singer, and Jennifer E. Enns. 2023. De-Identification of Free Text Data Containing Personal Health Information: A Scoping Review of Reviews. *International Journal of Population Data Science (IJPDS)* 8, 1 (Dec. 2023).

[53] Qiang Ning, Sanjay Subramanian, and Dan Roth. 2019. An Improved Neural Baseline for Temporal Relation Extraction. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 6203–6209.

[54] OpenAI. 2025. Introducing GPT-5. https://openai.com/index/introducing-gpt-5/

[55] Anthi Papadopoulou, Yunhao Yu, Pierre Lison, and Lilja Øvrelid. 2022. Neural Text Sanitization with Explicit Measures of Privacy Risk. In *Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the Joint Conference on Natural Language Processing (AACL-IJCNLP)*. 217–229.

[56] Thorsten Papenbrock and Felix Naumann. 2016. A Hybrid Approach to Functional Dependency Discovery. In *ACM International Conference on Management of Data (SIGMOD)*.

[57] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*. 311–318.

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

[59] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research (JMLR)* 21, 140 (2020), 1–67.

[60] Arij Riabi, Menel Mahamdi, Virginie Mouilleron, and Djamé Seddah. 2024. Cloaked Classifiers: Pseudonymization Strategies on Sensitive Classification Tasks. In *Workshop on Privacy in Natural Language Processing (PrivateNLP)*. 123–136.

[61] P. Samarati. Nov.-Dec./2001. Protecting Respondents Identities in Microdata Release. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 13, 6 (Nov.-Dec./2001), 1010–1027.

[62] David Sánchez and Montserrat Batet. 2016. C-sanitized: A Privacy Model for Document Redaction and Sanitization. *Journal of the Association for Information Science and Technology* 67, 1 (Jan. 2016), 148–163.

[63] David Sánchez and Montserrat Batet. 2017. Toward Sensitive Document Release with Privacy Guarantees. *Engineering Applications of Artificial Intelligence* 59 (March 2017), 23–34.

[64] Hao Shen, Zhouhong Gu, Haokai Hong, and Weili Han. 2025. PII-Bench: Evaluating Query-Aware Privacy Protection Systems. arXiv:2502.18545 [cs] http://arxiv.org/abs/2502.18545

[65] spaCy Developers. 2025. spaCy · Industrial-strength Natural Language Processing in Python. https://spacy.io/

[66] Robin Staab, Nikola Jovanović, Mislav Balunović, and Martin Vechev. 2024. From Principle to Practice: Vertical Data Minimization for Machine Learning. In *IEEE Symposium on Security and Privacy (SP)*. 4733–4752.

[67] Robin Staab, Mark Vero, Mislav Balunovic, and Martin Vechev. 2024. Beyond Memorization: Violating Privacy via Inference with Large Language Models. In *International Conference on Learning Representations (ICLR)*.

[68] Robin Staab, Mark Vero, Mislav Balunović, and Martin Vechev. 2025. Large Language Models Are Advanced Anonymizers. In *International Conference on Learning Representations (ICLR)*. arXiv:2402.13846 [cs]

[69] Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. 2022. Synthetic Data – Anonymisation Groundhog Day. In *USENIX Security Symposium (Security)*. 1451–1468.

[70] Nishant Subramani, Sasha Luccioni, Jesse Dodge, and Margaret Mitchell. 2023. Detecting Personal Information in Training Corpora: An Analysis. In *Workshop on Trustworthy Natural Language Processing (TrustNLP)*. 208–220.

[71] Latanya Sweeney. 2002. K-Anonymity: A Model for Protecting Privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems (IJUFKS)* 10, 05 (2002), 557–570.

[72] Maria Irena Szawerna, Simon Dobnik, Therese Lindström Tiedemann, Ricardo Muñoz Sánchez, Xuan-Son Vu, and Elena Volodina. 2024. Pseudonymization Categories across Domain Boundaries. In *Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*. 13303–13314.

[73] Thomas Brewster. 2023. ChatGPT Has Been Turned into a Social Media Surveillance Assistant. *Forbes* (Nov. 2023).

[74] United States Congress. 1996. Health Insurance Portability and Accountability Act of 1996. https://www.congress.gov/bill/104th-congress/house-bill/3103

[75] Saiteja Utpala, Sara Hooker, and Pin-Yu Chen. 2023. Locally Differentially Private Document Generation Using Zero Shot Prompting. In *Findings of the Association for Computational Linguistics (EMNLP)*. 8442–8457.

[76] Heinrich von Stackelberg. 2010. *Market Structure and Equilibrium*. Springer.

[77] Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural Network Acceptability Judgments. *Transactions of the Association for Computational Linguistics (TACL)* 7 (2019), 625–641.

[78] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 Technical Report. https://arxiv.org/abs/2505.09388

[79] Hui Yang and Jonathan M. Garibaldi. 2015. Automatic Detection of Protected Health Information from Clinic Narratives. *Journal of Biomedical Informatics (JBI)* 58 (Dec. 2015), S30–S38.

[80] Heung Youl Youm. 2020. An Overview of De-Identification Techniques and Their Standardization Directions. *IEICE Transactions on Information and Systems (TransInf)* E103.D, 7 (2020), 1448–1461.

[81] Xiang Yue, Huseyin Inan, Xuechen Li, Girish Kumar, Julia McAnallen, Hoda Shajari, Huan Sun, David Levitan, and Robert Sim. 2023. Synthetic Text Generation with Differential Privacy: A Simple and Practical Recipe. In *Annual Meeting of the Association for Computational Linguistics (ACL)*. 1321–1342.

[82] Hanna Yukhymenko, Robin Staab, Mark Vero, and Martin Vechev. 2024. A Synthetic Dataset for Personal Attribute Inference. In *Conference on Neural Information Processing Systems (NeurIPS)*.

[83] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations (ICLR)*.

Received October 2025; revised January 2026; accepted February 2026