SYMMETRA: Semantic prompt sanitization for end-to-end utility-preserving privacy protection in LLM interactions

Seongsu Moon, Juyeon Jeong, Taejune Park[Corr.]
Chonnam National University, Republic of Korea
{dalcw, siddl1020, taejune.park}@jnu.ac.kr

Abstract

External LLM services provide strong generation capabilities but require users to send prompts that may contain sensitive information. Existing prompt-protection methods focus on masking, perturbing, or rewriting the input prompt itself; however, these transformations can degrade response quality by removing task-relevant context or leave sensitive cues recoverable from the sanitized prompt. We present *SYMMETRA*, an end-to-end prompt-response protection framework based on semantic prompt sanitization. *SYMMETRA* detects context-dependent sensitive spans, selects response-aware replacements, and locally reconstructs responses with session memory and post-editing. We also introduce a synthetic prompt-domain dataset with an extended taxonomy and span-level annotations. Our evaluation shows that *SYMMETRA* improves the privacy–utility trade-off, reducing direct leakage to 0.13 and adversarial recovery to 0.42 while preserving a response similarity of 0.82 and restoring task-specific references in final user-facing responses after external inference.

1 Introduction

Privacy leakage through prompts: Large language model (LLM) applications often process prompts containing sensitive information, including personal records, internal documents, credentials, file paths, and medical or financial data (Chen et al., 2025; Roy Chowdhury et al., 2026). Importantly, such exposure is no longer limited to explicit user input. Modern LLM systems increasingly rely on retrieval-augmented generation (RAG), agents, external tools, and application-side context injection, which automatically incorporate retrieved documents, conversation history, tool outputs, and metadata into prompts (Lewis et al., 2020; Patil et al., 2024). As a result, prompts become a major exposure surface through which sensitive information may propagate beyond its source.

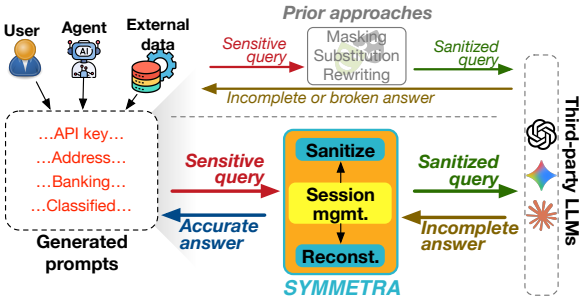


Figure 1: Illustrative example of privacy-preserving prompt sanitization in *SYMMETRA*.

This is problematic because many LLM use-cases rely on external LLM services (Thomas et al., 2025). Once prompts cross the provider boundary, users and organizations lose direct control over how sensitive contents are retained, processed, cached, or incorporated into provider-side functionality (Carrillo et al., 2026). Although major providers state that API inputs are not used for model training by default, their documentation still describes limited retention and processing for abuse monitoring, grounding, caching, session management, or operational purposes (OpenAI, 2026; Google Cloud, 2026; Anthropic, 2026). Consequently, sensitive information may become vulnerable to unintended retention or secondary exposure outside the original trust boundary (Wu et al., 2025).

Prior attempts: The risks of exposing sensitive prompt contents to third-party LLM services have been widely recognized (Wu et al., 2025; Carrillo et al., 2026; Thomas et al., 2025). Prior work has explored three directions: i) *span-detection*-based methods, which mask, redact, anonymize, or replace detected sensitive spans (microsoft; Leung, 2023); ii) *rewriting*-based methods, which paraphrase or rewrite prompts into safer forms (Utpala et al., 2023; Li et al., 2025; Huang et al., 2025); and iii) *perturbation*-based methods, which inject noise into tokens or representations (Meisenbacher et al., 2024; Zhang et al., 2025; Tong et al., 2025).

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 19729–19749
October 24-29, 2026 ©2026 Association for Computational Linguistics

Response-aware gap: However, these approaches largely confine prompt protection to *input-prompt sanitization*. Strong sanitization may remove task-relevant context, while weaker rewriting or perturbation can leave sensitive information inferable from context (Carpentier et al., 2024). Yet response utility is ultimately realized in the final user-facing answer. At the response level, existing methods have largely overlooked reconstruction of the final user-facing response, where original task-specific references must be restored despite paraphrasing, omission, or reorganization by external LLMs. These limitations call for prompt protection that removes residual sensitive cues, preserves task context for useful responses, and reconstructs task-specific references within a trusted local boundary.
Our solution: We propose *SYMMETRA*, an end-to-end prompt-response protection framework based on semantic prompt sanitization, as illustrated in Figure 1. *SYMMETRA* detects context-dependent sensitive spans, generates validated abstraction, placeholder, and synthetic-surrogate candidates, and selects among them using a local response-level utility signal. After external generation, session memory and local post-editing reconstruct task-specific references in the user-facing response. Thus, the external LLM observes only the sanitized prompt, while sensitive values are restored only within the trusted local boundary. Qualitative examples are provided in Appendix K.

Empirically, our evaluation on a synthetic benchmark of privacy-sensitive task-oriented prompts shows that *SYMMETRA* reduces the direct leakage ratio to 0.13 and adversarial recovery accuracy to 0.42, improving exposure over *rewriting*-based baselines while avoiding much of the utility loss caused by aggressive *span-detection*- and *perturbation*-based baselines. It also preserves response-level utility, with 0.82 response similarity and 0.75 final-response entity restoration recall (RERR).
Contributions: Our contributions are as follows: i) we formulate prompt protection as end-to-end prompt-response mediation across sanitization, external generation, and local reconstruction; ii) we construct a prompt-domain dataset with an extended sensitive-span taxonomy and span-level annotations; iii) we design *SYMMETRA*, a response-aware sanitization pipeline that selects validated replacements using response-level utility signals; and iv) we show that *SYMMETRA* improves the privacy–utility trade-off and final-response reconstruction on privacy-sensitive prompts.

2 Background and Motivation

2.1 Threat model

Problem statement: We consider LLM applications that transmit prompts containing sensitive information to external third-party LLM services for response generation. Prompts may incorporate sensitive information (e.g., personal records, credentials, or financial and medical details) from user input, agent-retrieved data, or tool outputs (Abdali et al., 2024; Chen et al., 2025).
Adversary model: We assume external LLM services, or any party capable of observing transmitted prompts and responses (e.g., network sniffers or provider-side systems), may attempt to infer sensitive information from LLM interactions through directly exposed spans or contextual cues in prompts and responses. Such information may be maliciously collected by adversaries, or retained, processed, or exposed through provider-side operations on external LLM services (Wu et al., 2025).

2.2 Prior work on LLM prompt protection

Prior work explored several prompt-protection approaches for external LLM inference.
Span-detection-based protection: Span-based approaches identify sensitive spans for masking, redaction, anonymization, or placeholder substitution. For example, OpaquePrompts (Leung, 2023) replaces sensitive values with placeholders, and Presidio (microsoft) combines rule-based matching and NER-based recognizers for sensitive-span detection. These approaches are effective for explicit identifiers and structured sensitive spans. However, aggressive masking or substitution may also remove task-relevant context and reduce downstream utility (Carpentier et al., 2024).
Rewriting-based protection: Rewriting-based approaches transform sensitive prompts into safer textual forms through paraphrasing (Utpala et al., 2023), differential privacy-guided rewriting (Li et al., 2025), and search-based sanitization strategies. Compared with masking or redaction, these approaches can produce more natural sanitized prompts (Huang et al., 2025). However, rewriting may preserve indirect sensitive cues when it changes explicit identifiers but keeps the surrounding relations and constraints that make the original value inferable, such as roles, access privileges, resource paths, or quasi-identifiers (Shi et al., 2025). Moreover, these methods are typically evaluated at the rewritten prompt level rather than their down-

stream effects on external LLM responses or response reconstruction.

Perturbation-based protection: Perturbation-based approaches reduce sensitive-information exposure by injecting noise into tokens or embedding representations. Prior work explored differential privacy-based token rewriting (Meisenbacher et al., 2024), local differential privacy sanitization (Zhang et al., 2025), and perturbation frameworks for black-box LLM inference (Tong et al., 2025). While suppressing direct exposure of sensitive representations, they may distort semantic context or task instructions, limiting prompt-level utility preservation in LLM interactions (Du et al., 2023; Mattern et al., 2022; Yue et al., 2021).

2.3 Our approach

Research question: Despite their differences, prior approaches share a common limitation: they primarily optimize the sanitized input prompt, with limited consideration of *how sanitization affects downstream LLM responses or how those responses should be reconstructed for the user*. Strong sanitization reduces exposure but degrades response utility, whereas weaker sanitization preserves utility while leaving sensitive information inferable from context (Carpentier et al., 2024; Yue et al., 2021). This raises the following question: *how can sensitive-information leakage be reduced without sacrificing response utility or reconstructability?*

Our goal: We answer this question with *SYMMETRA*, an *end-to-end utility-preserving prompt-response protection in LLM interactions*. Given an original prompt x containing sensitive spans, *SYMMETRA* locally generates a sanitized prompt x^* to reduce exposure before external transmission while preserving task-relevant semantics. Then, upon receiving the intermediate response $y' = f_{\text{LLM}}(x^*)$ from the external LLM, *SYMMETRA* reconstructs the final user-facing response $\tilde{y}$ with restored references and improved consistency.

Challenge: Achieving *SYMMETRA* requires addressing three challenges: i) sensitive information is heterogeneous and context-dependent, where roles, relations, numerics, and event structure may reveal sensitive content beyond explicit spans, while datasets covering such contexts remain limited; ii) sanitization directly affects generation quality, creating a privacy–utility trade-off between exposure reduction and response utility; and iii) final responses must be reconstructed to restore task-specific details from the sanitized prompts.

Trust boundary: Depending on deployment, *SYMMETRA* may run on the client side or as a trusted intermediary between the client and the external LLM service. We assume that *SYMMETRA* and its local models remain within this trusted boundary. The external LLM receives only sanitized prompts and returns intermediate responses, while restored sensitive values remain local.

3 System design

3.1 Overview

Pipeline: As shown in Figure 2, *SYMMETRA* achieves our goal through two stages: i) *semantic prompt sanitization*, which identifies sensitive information and replaces it with abstraction, placeholder, or surrogate candidates selected by response-level scoring; and ii) *response reconstruction*, which uses refining and post-editing to reconcile the sanitized-prompt response and recover the utility expected from the original prompt.

Semantic management: To compensate for incomplete or degraded responses caused by sanitization, *SYMMETRA* introduces *session memory* to maintain the correspondence between original sensitive spans and their sanitized representations across the sanitization and reconstruction stages. Refining and post-editing use this memory to recover context-dependent references that cannot be restored through exact string matching alone.

Prompt-domain dataset: This semantic coupling requires context-aware sensitive-span identification; accordingly, we construct a prompt-domain dataset with sensitive spans embedded in diverse task scenarios and contexts. Beyond explicit identifiers, it captures contextual cues through an extended taxonomy and span-level annotations.

3.2 Prompt-domain sensitive-span dataset

Sensitive-prompt dataset: Existing sensitive-information datasets mainly target private-information detection in general documents rather than LLM prompts (Ai4Privacy, 2024; Watson et al., 2024; BigCode, 2023). However, LLM prompts contain broader, context-dependent sensitive content, including direct identifiers such as addresses, bank accounts, and API keys, as well as organizational roles, medical or financial details, and quasi-identifiers that may leak information through context (Ngong et al., 2025; Roy Chowdhury et al., 2026). Identifying such content is thus essential across both the sanitization and reconstruction stages.

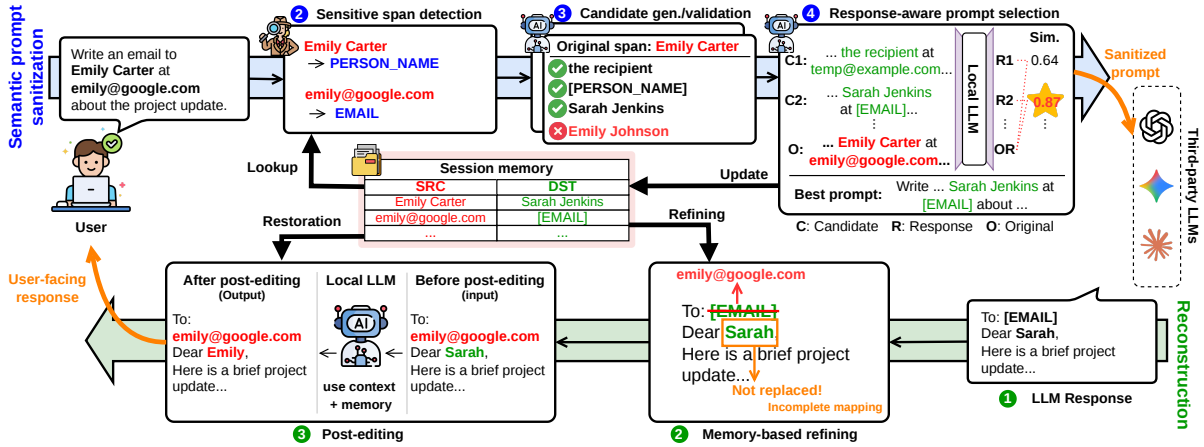


Figure 2: *SYMMETRA* Overview

To capture these prompt-domain sensitivities, we construct a synthetic dataset of task-oriented prompts in which sensitive details are embedded as task-relevant context, and annotate them with 26 labels under 10 semantic categories, e.g., `PII` (person names, account IDs), `AUTH` (authentication secrets), `RESOURCE` (file paths, endpoints), `FIN` (financial amounts), and `QUASI_ID` (locations, professions); the full taxonomy is provided in Appendix A. This dataset serves both as a benchmark for sensitive-span detection and as a basis for training and evaluating *SYMMETRA*, from substitution and sanitization to reconstruction.

Dataset generation: We generate the dataset using a two-stage pipeline with `GPT-OSS 120B` (OpenAI, 2025): i) *draft generation*, where the model both instantiates the selected sensitive labels with concrete values and chooses a task scenario in which they naturally appear. For example, given {`PERSON_NAME`, `EMAIL`, `ACCOUNT_ID`} and an email-drafting task, the model considers several scenarios, chooses an account-update notification scenario, generates values such as *Emily Carter*, *emily.carter@google.com*, and *ACC-49382*, and plans contextual cues such as organizational roles, access privileges, or resource paths.

ii) In *prompt generation*, the planned values and scenario are instantiated as a natural user prompt by embedding them as task arguments, references, and constraints that the LLM must use to complete the intended task, such as "*Please draft a notification email to Emily Carter at emily.carter@google.com about account ACC-49382, confirming that her senior manager privileges remain active in /etc/auth/users*". Each sensitive value is then annotated with its exact text span and label; a generated prompt is accepted only if all annotated spans appear verbatim in the prompt and all labels are valid under our taxonomy.

3.3 Semantic prompt sanitization

Objective: Semantic prompt sanitization aims to reduce sensitive exposure in a prompt while preserving the task context needed for useful generation. Formally, given an input prompt x, this stage selects a sanitized prompt from the validated privacy-preserving candidate set $\mathcal{P}(x)$ that maximizes a response-level utility proxy score: $x^* = \arg\max_{x' \in \mathcal{P}(x)} U(x, x')$. Here, $\mathcal{P}(x)$ contains only validated candidates, so utility-proxy scoring only ranks validated transformations.

Sensitive span detection: Semantic prompt sanitization first identifies sensitive spans in the original prompt using a RoBERTa-based token classification detector trained on the dataset in §3.2 (Liu et al., 2019). Given $x = (w_1, \ldots, w_T)$, the detector assigns each token w_t either one of the 26 sensitive labels or `O`, where `O` denotes a non-sensitive token and non-`O` labels denote sensitive-entity tokens. Consecutive non-`O` tokens are then merged into span-level predictions $S(x) = \{(s_i, \ell_i)\}_{i=1}^{m}$, where s_i is a sensitive span and ℓ_i is its label. These predictions define the sensitive units used throughout sanitization and reconstruction; detector performance is reported in Appendix C.

Candidate generation and validation: For each detected sensitive span s_i, *SYMMETRA* uses a small local LLM to generate short, safe replacement candidates conditioned on the original prompt x and label ℓ_i. The candidate set is $C_i = \{c_i^{\text{abs}}, c_i^{\text{ph}}, c_i^{\text{syn}}\}$, where the three elements denote abstraction, placeholder, and synthetic-surrogate candidates, respectively: i) *Abstraction* lowers specificity while preserving semantic role (e.g., *Google* $\rightarrow$ *a technology company*); ii) *placeholder*

removes the concrete value and keeps only the label when the span mainly serves as a response slot (e.g., [PERSON_NAME] or [EMAIL]); and *iii*) *synthetic surrogate* replaces the span with a fake value of similar format or role (e.g., *Emily Carter* → *Sarah Jenkins*). Generated candidates are then validated by $V(\cdot)$, which filters out candidates that copy or substantially expose the original span, preserve relational patterns, or are contextually unnatural, yielding $\hat{C}_i = \{c \in C_i \mid V(c, s_i, \ell_i, x) = 1\}$.

Prompt alternatives construction: The sets $\{\hat{C}_i\}_{i=1}^{m}$ define the replacement space for sanitized prompt alternatives. The j-th alternative selects one replacement $r_{i,j} \in \hat{C}_i$ for each sensitive span s_i, forming $r_j = \{s_i \mapsto r_{i,j}\}_{i=1}^{m}$ and $x'_j = \mathrm{Rewrite}(x, r_j)$. Different selections yield different prompts, e.g., replacing a name, email, and company name with a placeholder, synthetic surrogate, and abstraction, respectively. For efficiency, *SYMMETRA* keeps at most n validated alternatives, denoted $\mathcal{P}(x) = \{x'_1, \ldots, x'_n\}$.

Response-level estimation: Among the alternatives, the final sanitized prompt is selected by using a small local LLM $A(\cdot)$ as a response simulator to estimate how well each alternative preserves the original task intent: Let $y_0 = A(x)$ be the simulated response to the original prompt and $y_j = A(x'_j)$ the response to an alternative prompt $x'_j \in \mathcal{P}(x)$. Their semantic similarity is computed using a sentence-embedding model $E(\cdot)$, yielding the utility proxy score $U(x, x'_j) = \cos(E(y_0), E(y_j))$. The final sanitized prompt is then selected as $x^* = \arg\max_{x'_j \in \mathcal{P}(x)} U(x, x'_j)$. Before sending x^* to the external LLM, *SYMMETRA* records the selected span–replacement mappings for local reconstruction.

Session memory: Finally, selected span-replacement mappings are stored in local session memory for consistent reuse across a session or multi-turn interaction. For example, if a name is replaced once with a synthetic surrogate, the same surrogate can be reused when the entity reappears, reducing repeated candidate generation and selection. Formally, after selecting replacement r_i^* for span s_i with label ℓ_i, the memory is updated as $M_t = M_{t-1} \cup \{(s_i, \ell_i) \mapsto r_i^*\}$ with indexes. These mappings are later used during response reconstruction to restore the original user context.

3.4 Response reconstruction

Objective: Responses generated from sanitized prompts x^* may have degraded utility when task-specific details of the original sensitive references are omitted or replaced. Hence, the goal of response reconstruction is to refine the external LLM response y' into a useful final response $\tilde{y}$. This is performed locally, and restored sensitive information is never sent back to the external LLM service.

Memory-based refining: *SYMMETRA* first performs *deterministic* refining using mappings stored in session memory. It reverses these mappings to replace abstraction, placeholder, or surrogate expressions in the external LLM response y' with the original values whenever possible, yielding $y^{\mathrm{hard}} = R_{\mathrm{hard}}(y', M_t)$, where R_{hard} denotes the memory-based refining. This refining is mainly effective when surrogate or placeholder expressions explicitly reappear in the external LLM response, e.g., *Emily Carter* is replaced with the synthetic surrogate *Sarah Jenkins*. However, if the external LLM abbreviates or paraphrases them (e.g., *Sarah*, *the recipient*), refining may miss the reference.

Post-editing: To address this limitation, *SYMMETRA* next applies local LLM-based post-editing. It takes the external response y', the refined result y^{hard}, and session memory M_t as input, producing the final response $\tilde{y} = R_{\mathrm{soft}}(y', y^{\mathrm{hard}}, M_t)$. Unlike deterministic refining, this step performs context-aware reconstruction, allowing abbreviated or paraphrased references such as *Sarah* or *the recipient* to be restored using both response context and session memory. It also improves fluency after abstraction-, placeholder- or surrogate-based sanitization.

Overall, reconstruction converts external responses into user-facing outputs while keeping restored sensitive values local.

3.5 Knowledge distillation

The sanitization stage provides a strong privacy-utility trade-off but its multi-stage inference increases latency. To reduce this overhead, we construct a distilled variant that trains a student sanitization model to approximate the teacher-selected sanitized prompts. The student model is trained to directly generate the teacher-selected sanitized prompt from the original prompt. During inference, *SYMMETRA* recovers span–replacement mappings by comparing the original and generated prompts, stores them in session memory, and then follows the same response generation and reconstruction procedure as the full system. Note that the goal of distillation is not to improve privacy or utility, but to approximate the teacher behavior with lower sanitization latency (Hinton et al., 2015).

4 Evaluation

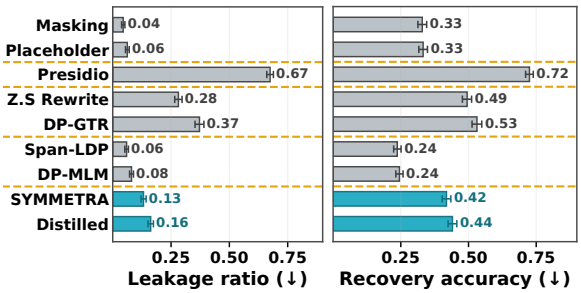
Implementation: We implemented *SYMMETRA* on Ubuntu 24.04 with an Intel Xeon 4210, 256GB RAM, and an RTX 6000, using PyTorch and Hugging Face Transformers. We use Qwen3.5-4B for candidate generation and post-editing (Qwen Team, 2026b), Qwen3.5-9B for response-level estimation (Qwen Team, 2026c), and Sentence-BERT `all-mpnet-base-v2` as the embedding model $E(\cdot)$ (Reimers and Gurevych, 2019; Sentence Transformers). We set $n_{\text{cand}} = 5$. We fine-tune `roberta-base` for sensitive-span detection (20 epochs; AdamW; lr 3×10^{-5}; batch 128; weight decay 0.01; warmup 0.1; max length 128) and `flan-t5-small` for distillation (3 epochs; AdamW; lr 3×10^{-4}; batch 64; weight decay 0.01; warmup 0.03; source/target lengths 512/256) (Chung et al., 2024; Raffel et al., 2020). For external response generation, we use `Gemma3-27B` served through Ollama as a reproducible open-weight LLM (Team et al., 2025). Appendix D provides the prompt artifact used in *SYMMETRA*.

Dataset: We use a dataset generated following the procedure in §3.2. Training and evaluation prompts are disjoint, with 31,263 training examples and 1,000 test examples, averaging 32 words each; label distributions are reported in Appendix B. All samples follow the same taxonomy and span-level annotation schema.

Baselines: Following the categories in §2.2, we compare *SYMMETRA* with: *i*) *span-detection-based* methods, *Masking* and *Placeholder*, which remove sensitive spans or replace them with typed placeholders; *ii*) *Presidio*, an off-the-shelf tool that uses rule-, pattern-, and recognizer-based anonymization; *iii*) *rewriting-based* methods, *Zero-shot Rewrite* and *DP-GTR*, which rewrite the full prompt; and *iv*) *perturbation-based* methods, *Span-LDP* and *DP-MLM*, which perturb sensitive spans or tokens. All methods are evaluated under the same setting; details are in Appendix E.

4.1 Information leakage in sanitized prompts

Goal: We evaluate sanitized prompts using two prompt-level exposure metrics: *i*) *Sensitive-span leakage ratio*, which measures direct exposure by counting original sensitive spans whose character-level overlap with the sanitized prompt is at least 50%; and *ii*) *Adversarial recovery accuracy*, which measures contextual exposure through a controlled multiple-choice task. The adversary sees the sanitized prompt and a masked original prompt, then infers the hidden value from 10 same-label choices. We use `Gemma3:27B` as the adversary model (Team et al., 2025); Appendix F and G provides details and examines sensitivity to the overlap threshold.



(a) Sensitive-span leakage (b) Adversarial recovery

Figure 3: Information leakage in sanitized prompts

Result: Figure 3 shows consistent trends across the two metrics. *Span-detection-* and *perturbation-based* methods obtain the lowest exposure scores by eliminating or heavily modifying sensitive content, at the cost of removing task-relevant context. In contrast, *Presidio* and *rewriting*-based methods preserve more context but exhibit higher direct leakage and recovery risk. *SYMMETRA* and its distilled variant achieve leakage ratios of 0.13/0.16 and recovery accuracies of 0.42/0.44. While above the most conservative methods, these scores remain below *Presidio* and the *rewriting*-based methods.

Implication: The lowest leakage comes from aggressive removal or perturbation, which also discards task-relevant context, degrading response utility. *SYMMETRA* instead reduces both direct and contextual exposure relative to methods that preserve more sensitive cues, while retaining information for useful response generation. The next section examines this utility gap.

4.2 Response utility

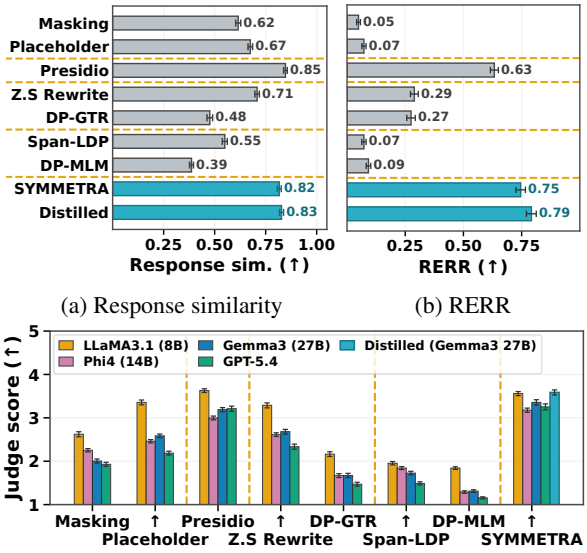
Goal: We evaluate response utility from three perspectives. *i*) *Response similarity* compares original- and sanitized-prompt responses using cosine similarity in the Sentence-BERT `all-mpnet-base-v2` space (Reimers and Gurevych, 2019; Sentence Transformers). *ii*) *Response entity restoration recall (RERR)* measures how well the restored response covers sensitive entities in the reference response. Given the sensitive-entity sets in the reference and restored responses, E_{ref} and E_{rest}, we define $RERR = |E_{\text{ref}} \cap E_{\text{rest}}|/|E_{\text{ref}}|$ and report the micro-average (Appendix F provides the detailed recipe); *iii*) *LLM-as-a-Judge* scores task

utility on a 1–5 scale from the *original prompt*, *original response*, and *sanitized response*. To assess the consistency of the results across external LLMs, we repeat all three utility evaluations using either `Gemma 3:27B` or `GPT-5.4` as the response-generating model. Separately, for robust LLM-as-a-Judge evaluation, we use evaluator models of different sizes and families: `Gemma 3:27B` (Team et al., 2025), `Phi-4:14B` (Abdin et al., 2024), and `LLaMA 3.1:8B` (Grattafiori et al., 2024); see Appendix H for the rubric.

Result: Figures 4 and 5 show that *SYMMETRA* consistently preserves response utility more effectively than baselines that strongly remove, perturb, or rewrite prompt content across both external LLM settings. Across `Gemma 3:27B` and `GPT-5.4`, *SYMMETRA* and its distilled variant maintain response-similarity scores of 0.8–0.83, comparable to *Presidio* at 0.83-0.85. Although *Presidio* provides comparable response similarity, this utility is partly attributable to insufficient sanitization, as reflected in its higher leakage and recovery risk. For RERR, *span-detection*-, *rewriting*-, and *perturbation*-based baselines remain limited because they do not reconstruct sensitive references after external generation. In contrast, *SYMMETRA* maintains a high RERR of 0.75 under both external LLMs, while its distilled variant improves from 0.79 with `Gemma 3:27B` to 0.88 with `GPT-5.4`. These results suggest that *SYMMETRA*'s utility advantage is not tied to a specific external LLM.

This utility advantage is further supported by LLM-as-a-Judge results under both external LLM settings, which capture practical task utility beyond embedding similarity. *SYMMETRA* and its distilled variant consistently rank among the strongest methods across judge models, whereas aggressive removal or perturbation baselines score lower because they damage task-relevant cues. The consistent trend with both `Gemma 3:27B` and `GPT-5.4` further suggests that this advantage is not tied to a single external response generator.

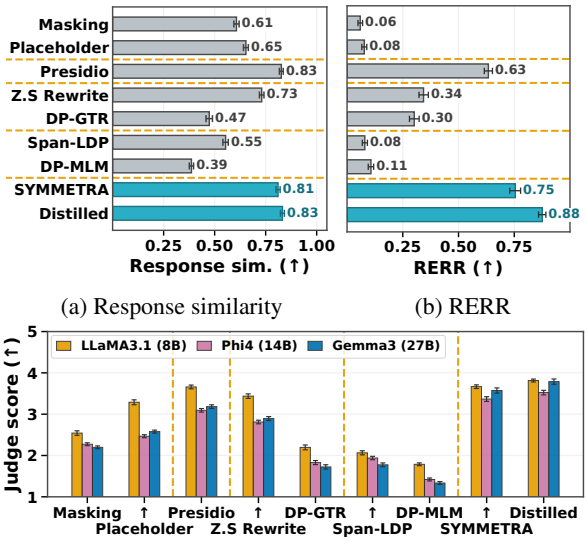
Implication: Together with the exposure results, these results show that useful responses require both semantic preservation and entity restoration without simply leaving sensitive information intact. *SYMMETRA* addresses this trade-off through response-level prompt selection, which preserves useful generation behavior, and session memory with local post-editing, which restores task-specific references after external inference. The consistent gains across both evaluated external LLMs further



(a) Response similarity (b) RERR

(c) LLM-as-a-Judge scores across methods.

Figure 4: Response utility using Gemma 3:27B



(a) Response similarity (b) RERR

(c) LLM-as-a-Judge scores across methods.

Figure 5: Response utility using GPT-5.4

support the effectiveness of this architectural separation between sanitized external generation and trusted local reconstruction. Qualitative examples of the sanitization and reconstruction process are provided in Appendix K, while failure cases are discussed in Appendix L. Note that these results do not imply that the baselines fail as sanitizers; rather, they show that final-response utility requires reconstruction beyond prompt sanitization.

4.3 Privacy-utility trade-off

Goal: We visualize the trade-off using direct privacy score, defined as $1 -$ leakage ratio, and response similarity as utility.

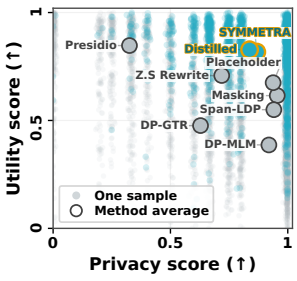


Figure 6: Privacy-utility trade-off across methods

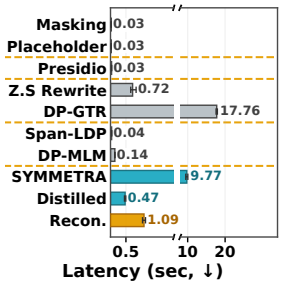
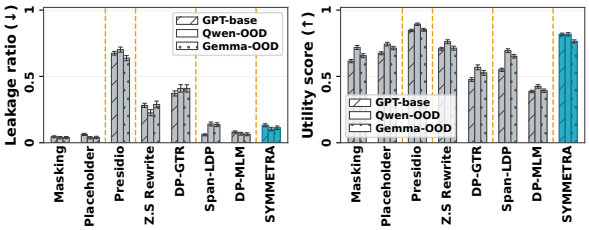


Figure 7: Avg. sanitization latency across methods

Variant	Leakage ↓	Similarity ↑	RERR ↑	Latency ↓
SYMMETRA (Vanilla)	0.13	0.82	0.75	10.85
SYMMETRA (Distilled)	0.16	0.83	0.79	0.47
w/o Response Scoring	0.11	0.75	0.67	4.93
w/o Candidate Validation	0.25	0.78	0.71	12.53
w/o Memory-based refining	0.13	0.70	0.16	11.65
w/o Post-editing	0.13	0.76	0.62	11.60

·Leakage/Similarity/RERR: Ratio; · Latency: Seconds

Table 1: Ablation study results.



(a) Privacy leakage (b) Utility preservation

Figure 8: Synthetic OOD evaluation on GPT-base, Qwen-OOD, and Gemma-OOD prompts.

Result: Figure 6 summarizes the trade-off between privacy and utility. *Span-detection*- and *perturbation*-based baselines generally move toward higher privacy but lower utility, reflecting the cost of suppressing task-relevant context. In contrast, context-preserving methods such as *Presidio* and *Zero-shot Rewrite* achieve higher utility but remain in lower-privacy regions. *SYMMETRA* remains near the upper-right region, maintaining high utility while substantially improving privacy over context-preserving baselines; the distilled variant stays close, indicating that distillation preserves the same trade-off behavior. Appendix J provides the corresponding sample-level distributions.

Implication: The trade-off plot confirms that *SYMMETRA* avoids both extremes: it does not preserve utility by leaving sensitive content exposed, nor does it reduce exposure by discarding task-relevant context. Instead, it occupies a favorable region with low direct leakage and high response utility.

4.4 Model efficiency and robustness

Latency: We measure the average latency for prompt sanitization under the same evaluation prompts and execution environment. As shown in Figure 7, *span-detection*-based baselines and *Presidio* show the lowest latency at around 0.03s, while *perturbation*-based baselines remain relatively fast at 0.04s and 0.14s. The *rewriting*-based *DP-GTR* shows the highest latency at 17.76s due to multi-stage rewriting. The *SYMMETRA* sanitization stage requires 10.85s, while the distilled variant reduces this to 0.47s, showing that distillation substantially reduces overhead while preserving similar behavior. Separately, *SYMMETRA* incurs 1.09s for local response reconstruction, acceptable in consideration of common external LLM interaction response times of 2-20s (Tan et al., 2026).

Ablation study: Table 1 evaluates how each component of *SYMMETRA* contributes to privacy, utility, reconstruction, and latency. The full pipeline achieves leakage 0.13, response similarity 0.82, and RERR 0.75, while distillation preserves similar performance and reduces latency from 10.85s to 0.47s. Removing response-level scoring lowers similarity to 0.75, showing its role in preserving task utility; removing candidate validation increases leakage to 0.25, confirming its role in suppressing sensitive-span exposure. Disabling memory-based refining sharply reduces RERR to 0.16, and removing post-editing lowers it to 0.62, showing that both are important for response reconstruction.

Synthetic OOD evaluation: We evaluate whether *SYMMETRA* and the proposed dataset generalize beyond a single synthetic generator. In addition to GPT-base, we construct *Qwen-OOD* and *Gemma-OOD* prompt sets with different styles, sensitive-span placements, and task expressions; their label distributions are provided in Appendix I. As shown in Figure 8, baseline groups largely preserve their privacy-utility patterns across distributions: *span-detection*-based methods keep low leakage with limited utility, *Presidio* keeps high utility with high leakage, and *rewriting*- and *perturbation*-based baselines retain their trade-off trends. *SYMMETRA* consistently maintains low leakage and high utility across synthetic generators.

4.5 Cross-source evaluation on Nemotron-PII

Goal: We assess whether our sensitive-span detector remains on an independently constructed dataset beyond the synthetic prompt distribution used for training. Given the difficulty of obtaining real privacy-sensitive text with span-level annotations, we use 50,578 unstructured documents from

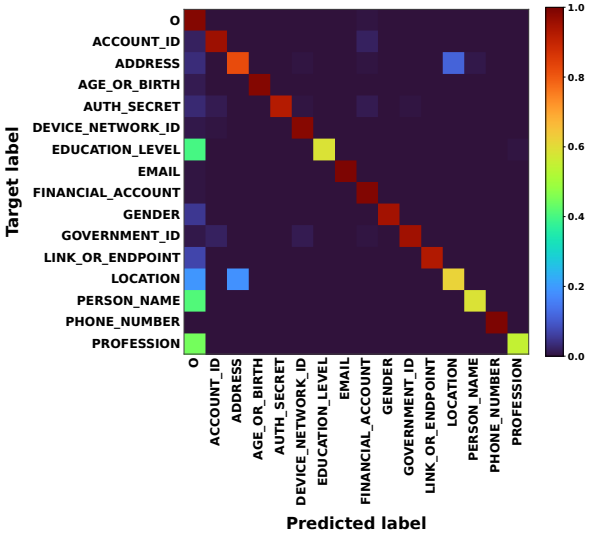


Figure 9: Detector confusion matrix on Nemotron-PII

Nemotron-PII (nvi, 2026), an independently constructed synthetic PII/PHI benchmark, and apply the frozen detector without additional fine-tuning. Because its fine-grained label schema does not directly align with ours, we map semantically compatible labels into 15 broader categories and exclude labels without a reliable correspondence. We report token-level micro, weighted, and macro F1 scores excluding the O label.

Result: Figure 9 shows its result. Despite the differences in data source, document style, and label schema, the detector achieves micro, weighted, and macro F1 scores of 0.90, 0.89, and 0.85, respectively, with a binary sensitive-span F1 of 0.92. These results indicate that the detector retains strong cross-source performance even under approximate label alignment.

Implication: These results support the validity of our synthetic dataset as a training resource: despite differences in data source and label schema, a detector trained solely on our data maintains strong performance on an independently constructed external benchmark. This suggests that our data-generation procedure captures reusable sensitive-information patterns rather than artifacts specific to a single synthetic distribution.

5 Discussion

Response-aware semantic consistency: Our results show that *SYMMETRA* addresses the empirical goals derived from our threat model: reducing direct exposure and contextual recovery risk while preserving response utility and reconstructability. *SYMMETRA* achieves competitive performance across these objectives, suggesting that effective prompt protection requires semantic consistency between sanitization and reconstruction, rather than treating them as independent steps.

Use cases: We aim for *SYMMETRA* to serve as an LLM-specific firewall, or a form of deep packet inspection for prompts, providing a local boundary that inspects and transforms prompts. This boundary becomes especially important in agentic systems, RAG pipelines, and tool-using applications, where sensitive information may be introduced through retrieved documents, tool outputs, logs, or intermediate agent states, not only through direct user input (Ngong et al., 2026; Ezell et al., 2025).

Policy-aware deployment: Sensitive information varies across domains, organizations, and languages, so its operational definition should be configurable rather than tied to a fixed taxonomy (Yekrangi et al., 2026; Raffetseder et al., 2025). In *SYMMETRA*, deployment-specific policies can extend the protection scope through the detector label space, validation function $V(\cdot)$, and replacement choices during candidate selection (see §3.2 and Appendix A). For example, enterprise deployments could extend the label space or validation rules for credentials, internal resources, and access-related context, whereas healthcare, financial, or multilingual deployments could adjust labels and replacement choices for specialized attributes and quasi-identifiers. Such flexibility helps align prompt protection with domain-, organization-, and language-specific requirements.

6 Conclusion

This paper presented *SYMMETRA*, an end-to-end prompt-response mediation framework for protecting sensitive information in external LLM use. By coupling semantic prompt sanitization with trusted local reconstruction, *SYMMETRA* reduces exposure relative to context-preserving baselines while preserving substantially higher response utility than aggressive removal or perturbation. Its consistent performance across different external LLMs further supports the effectiveness of this architectural design. These results suggest that prompt privacy should go beyond input sanitization to trusted local response recovery. We believe that *SYMMETRA* may provide a foundation for future AI systems that contextually understand and handle sensitive information under varying privacy requirements.

Limitations

Synthetic data dependency: Our dataset and main evaluations rely on synthetic data, which may not fully capture real user-prompt distributions. This choice is partly motivated by ethical considerations, since collecting or releasing real sensitive prompts could itself expose user information. We mitigate this limitation with task-oriented generation, span-level annotations, and OOD prompt sets from different models, but real-world validation under appropriate privacy safeguards remains necessary.
Evaluation scope: We evaluate privacy and utility using leakage ratio, adversarial recovery, response similarity, LLM-as-a-Judge, and RERR. These metrics capture complementary aspects of prompt protection, but they remain proxies for real user utility and real attacker capability. Human evaluation and task-specific success metrics could provide a more complete assessment.
Empirical risk mitigation: *SYMMETRA* provides empirical risk mitigation rather than a formal privacy guarantee such as differential privacy. Although adversarial recovery remains nonzero at 0.42, it is substantially lower than context-preserving baselines such as *Presidio* (0.72) and *Zero-shot Rewrite* (0.49), while preserving more utility than aggressive removal or perturbation. This reflects the practical goal of *SYMMETRA*: reducing recoverability without erasing the semantic context needed for useful generation and local reconstruction.

Ethical Considerations

We use a synthetic dataset to study sensitive-span detection, sanitization, and reconstruction without exposing actual user data. Any deployment of *SYMMETRA* on real user prompts should follow applicable privacy regulations, obtain appropriate authorization, and avoid sending restored sensitive information back to external LLM services.
Reproducibility: The datasets and pretrained models used in this study are publicly available. The implementation and trained artifacts are planned for public release after publication.
LLM usage statement: LLMs were used for coding assistance, English proofreading and editing in this manuscript.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback and suggestions on this work. This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT): the Artificial Intelligence Convergence Innovation Human Resources Development Program (IITP-2023-RS-2023-00256629), the ITRC (Information Technology Research Center) Support Program (IITP-2026-RS-2024-00437718), and the AI Star Fellowship Support Program (IITP-2026-RS-2026-25619158). This work was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2026-25602261).

References

2026. nvidia/nemotron-pii · datasets at hugging face. [Online; accessed 2026-08-29].

Sara Abdali, Richard Anarfi, Carlos J Barberan, Jia He, and Erfan Shayegani. 2024. Securing large language models: Threats, vulnerabilities and responsible practices. *arXiv preprint arXiv:2403.12503*.

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

Ai4Privacy. 2024. pii-masking-300k (revision 86db63b).

Anthropic. 2026. Does anthropic act as a data processor or controller? `https://support.claude.com/en/articles/9267385-does-anthropic-act-as-a-data-processor-or-controller`. Accessed: 2026-05-19.

BigCode. 2023. bigcode/bigcode-pii-dataset · datasets at hugging face. [Online; accessed 2026-05-23].

Robin Carpentier, Benjamin Zi Hao Zhao, Hassan Jameel Asghar, and Dali Kaafar. 2024. Preempting text sanitization utility in resource-constrained privacy-preserving llm interactions. *arXiv preprint arXiv:2411.11521*.

Juan-Carlos Carrillo, Jose Luis Martin-Navarro, Rongjun Ma, and Jose Such. 2026. Personal data flows and privacy policy traceability in third-party llm apps in the gpt ecosystem. *Proceedings on Privacy Enhancing Technologies*.

Kang Chen, Xiuze Zhou, Yuanguo Lin, Shibo Feng, Li Shen, and Pengcheng Wu. 2025. A survey on

privacy risks and protection in large language models. *Journal of King Saud University Computer and Information Sciences*, 37(7):163.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Minxin Du, Xiang Yue, Sherman SM Chow, and Huan Sun. 2023. Sanitizing sentence embeddings (and labels) for local differential privacy. In *Proceedings of the ACM Web Conference 2023*, pages 2349–2359.

Carson Ezell, Xavier Roberts-Gaal, and Alan Chan. 2025. Incident analysis for ai agents. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 8, pages 865–878.

Facebook AI. Facebookai/roberta-base. [Online; accessed 2026-05-24].

Google Cloud. 2026. Vertex ai and zero data retention. https://docs.cloud.google.com/vertex-ai/generative-ai/docs/vertex-ai-zero-data-retention. Accessed: 2026-05-19.

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

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Shuo Huang, Xingliang Yuan, Gholamreza Haffari, and Lizhen Qu. 2025. Zero-shot privacy-aware text rewriting via iterative tree search. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 9175–9190.

Chester Leung. 2023. Announcing OpaquePrompts: Hide your sensitive data from LLMs. [Online; accessed 2026-08-28].

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Mingchen Li, Heng Fan, Song Fu, Junhua Ding, and Yunhe Feng. 2025. Dp-gtr: Differentially private prompt protection via group text rewriting. *arXiv preprint arXiv:2503.04990*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Justus Mattern, Benjamin Weggenmann, and Florian Kerschbaum. 2022. The limits of word level differential privacy. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 867–881.

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 2024*, pages 9314–9328.

microsoft. Github - microsoft/presidio: An open-source framework for detecting, redacting, masking, and anonymizing sensitive data (pii) across text, images, and structured data. supports nlp, pattern matching, and customizable pipelines. [Online; accessed 2026-05-21].

Ivoline C Ngong, Swanand Ravindra Kadhe, Hao Wang, Keerthiram Murugesan, Justin D Weisz, Amit Dhurandhar, and Karthikeyan Natesan Ramamurthy. 2025. Protecting users from themselves: Safeguarding contextual privacy in interactions with conversational agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 26196–26220.

Ivoline C Ngong, Keerthiram Murugesan, Swanand Kadhe, Justin D Weisz, Amit Dhurandhar, and Karthikeyan Natesan Ramamurthy. 2026. Agentscope: Evaluating contextual privacy across agentic workflows. *arXiv preprint arXiv:2603.04902*.

Ollama. gemma3:27b. [Online; accessed 2026-05-24].

OpenAI. 2025. openai/gpt-oss-120b. [Online; accessed 2026-05-22].

OpenAI. 2026. Data controls in the openai platform. https://developers.openai.com/api/docs/guides/your-data. Accessed: 2026-05-19.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. 2024. Gorilla: Large language model connected with massive apis. *Advances in Neural Information Processing Systems*, 37:126544–126565.

Qwen Team. 2026a. Qwen/qwen3.5-27b. [Online; accessed 2026-05-24].

Qwen Team. 2026b. Qwen/qwen3.5-4b. [Online; accessed 2026-05-24].

Qwen Team. 2026c. Qwen/qwen3.5-9b. [Online; accessed 2026-05-24].

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*, 21(140):1–67.

David Raffetseder, Carl Weilguny, Patrick Haidinger, Hans-Peter Pichler, and Wolfgang Narzt. 2025. Support ticket anonymization: advancing data privacy with transformer-based named entity recognition.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 3982–3992.

Amrita Roy Chowdhury, David Glukhov, Divyam Anshumaan, Prasad Chalasani, Nicholas Papernot, Somesh Jha, and Mihir Bellare. 2026. Pr$\epsilon\epsilon$mpt: Sanitizing sensitive prompts for llms. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA.

Sentence Transformers. sentence-transformers/all-mpnet-base-v2. [Online; accessed 2026-05-24].

Zhan Shi, Yefeng Yuan, Yuhong Liu, Liang Cheng, and Yi Fang. 2025. Rl-finetuned llms for privacy-preserving synthetic rewriting. *arXiv preprint arXiv:2508.19286*.

sjmeis. Github - sjmeis/dpmlm: Repository containing the code for the acl findings paper "dp-mlm: Differentially private text rewriting using masked language models". [Online; accessed 2026-05-24].

Felicia Fang-Yi Tan, Moritz Alexander Messerschmidt, Wen Yin, and Oded Nov. 2026. The impact of response latency and task type on human-llm interaction and perception. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, pages 1–17.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Rahul Krishna Thomas, Louai Zahran, Erica Choi, Akilesh Potti, Micah Goldblum, and Arka Pal. 2025. Hidden no more: Attacking and defending private third-party LLM inference. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 59434–59469. PMLR.

Meng Tong, Kejiang Chen, Jie Zhang, Yuang Qi, Weiming Zhang, Nenghai Yu, Tianwei Zhang, and Zhikun Zhang. 2025. Inferdpt: Privacy-preserving inference for closed-box large language models. *IEEE Transactions on Dependable and Secure Computing*, 22(5):4625–4640.

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 2023*, pages 8442–8457.

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

Yuhao Wu, Evin Jaff, Ke Yang, Ning Zhang, and Umar Iqbal. 2025. An in-depth investigation of data collection in llm app ecosystems. In *Proceedings of the 2025 ACM Internet Measurement Conference*, pages 150–170.

Mehdi Yekrangi, Houssem Chatbri, Claudia Beatrice Chianella, and Owen O'Neill. 2026. A guardrail framework for sensitive financial information protection: A taxonomy-driven approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 40528–40535.

Xiang Yue, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman SM Chow. 2021. Differential privacy for text analytics via natural text sanitization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3853–3866.

Juhua Zhang, Zhiliang Tian, Minghang Zhu, Yiping Song, Taishu Sheng, Siyi Yang, Qiunan Du, Xinwang Liu, Minlie Huang, and Dongsheng Li. 2025. Dyntext: semantic-aware dynamic text sanitization for privacy-preserving llm inference. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20243–20255.

A Sensitive-span taxonomy

Category	#Labels	Examples
PII	6	PERSON_NAME, ACCOUNT_ID, EMAIL, PHONE_NUMBER, ADDRESS, SOCIAL_HANDLE
LEGAL	2	GOVERNMENT_ID, DRIVER_LICENSE
ORG_ INTERNAL	1	ORG_UNIT
AUTH	1	AUTH_SECRET
SYSTEM	1	DEVICE_NETWORK_ID
RESOURCE	2	LINK_OR_ENDPOINT, FILE_OR_RESOURCE_PATH
OPERATIONAL_ ARTIFACT	1	SYSTEM_ARTIFACT
FIN	2	FINANCIAL_AMOUNT, FINANCIAL_ACCOUNT
QUASI_ID	8	AGE_OR_BIRTH, GENDER, LOCATION, PROFESSION, EMPLOYER, EDUCATION_LEVEL, MARITAL_STATUS, CHILD_COUNT
MED	2	HEALTH_CONDITION, ALLERGY_INFO

Table 2: Category-level summary of the sensitive information taxonomy used in our dataset.

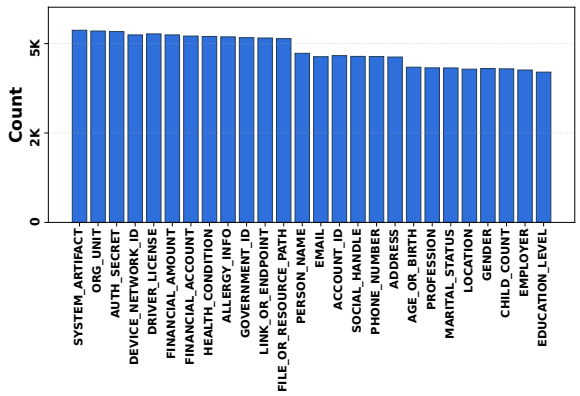
Table 2 summarizes the sensitive-span taxonomy used for dataset generation and span-level annotation in §3.2. The taxonomy contains 26 labels grouped into 10 semantic categories, covering both explicit identifiers and context-dependent sensitive information that may appear in task-oriented LLM prompts.

B Dataset label distribution

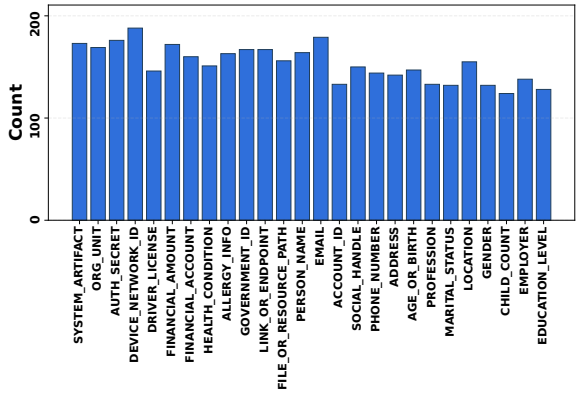
The evaluation uses 31,263 training examples and 1,000 test examples from the generated dataset, as described in §4. Figure 10 shows the distribution of sensitive-information labels in each split. All 26 labels appear in both the training and test sets. Although some frequency differences remain, the distribution is relatively balanced and is not dominated by a small number of labels. This indicates that the synthetic data generation pipeline covers diverse sensitive-information types across the taxonomy.

C Sensitive span detector performance

As described in §3.3, the prompt sanitization pipeline first detects sensitive spans in the input prompt, and then performs candidate generation, candidate validation, response-level selection, and session memory storage based on the detected spans. Therefore, the performance of the sensitive-span detector directly affects the reliability of subsequent prompt sanitization and response reconstruction.



(a) Training data label distribution



(b) Test data label distribution

Figure 10: Label distribution in the generated dataset

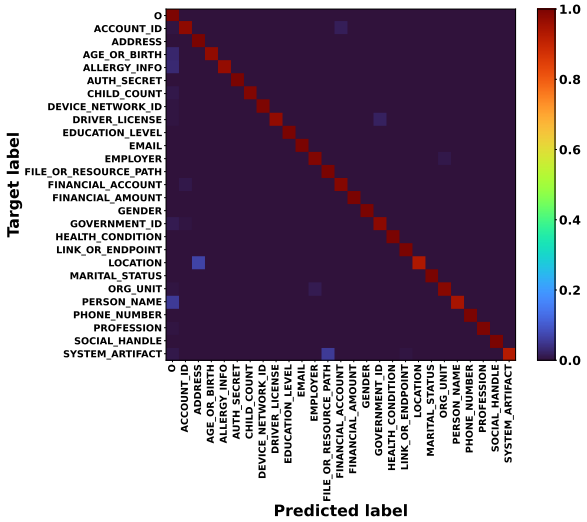


Figure 11: Confusion matrix of the sensitive span detector on the test set

We evaluate the trained RoBERTa-based token classification detector on the annotated sensitive spans in the test set (Liu et al., 2019). At the entity level, the detector achieves a macro F1 of

0.985, weighted F1 of 0.987, and micro F1 of 0.987. Figure 11 shows the label-wise confusion matrix. Overall, the detector reliably distinguishes most sensitive-information labels, indicating that the downstream sanitization steps can operate on reliable span predictions.

D Prompt template

Abridged candidate generation prompt (Qwen3.5-4B)

Role:
You are a privacy-aware span replacement assistant.

Task:
Generate three safer replacements for one sensitive span. Preserve its semantic role and downstream utility while reducing identifiability. Each candidate must naturally fit in the original span position.

Rules:
Do not reproduce the original value or introduce unsupported facts. Return replacement spans only rather than rewriting the full sentence. Each candidate must have the same semantic type as the original span.

Candidate order:

1. `ABSTRACT`: a broader and less specific expression;
2. `PLACEHOLDER`: a short, type-aware marker;
3. `SURROGATE`: a synthetic value of the same type.

Input:
Sentence: `{sentence}`
Sensitive span: `{span}`
Label: `{label}`
Category instruction: `{family_instruction}`

Output:
A valid JSON array containing exactly three strings in the specified order.

Response simulation prompt (Qwen3.5-9B)

Role:
You are a task-oriented simulated assistant.

Task:
Produce the textual output that would be used to complete the user's requested task. Do not claim that any external action has already been performed.

Rules:
If the prompt requests sending, notifying, contacting, or responding, return the corresponding draft message or response text. If it requests explanation, summarization, or interpretation, return only the requested content.
If private details are hidden, generalized, or replaced with placeholders, continue using the available information without inferring or reconstructing the hidden details. Ask a clarification question only when essential non-private information is missing. Do not restate the user's prompt. Keep the output concise, direct, and stylistically stable.

Input:
User prompt: `{user_prompt}`

Output:
Return only the simulated textual response.

Abridged post-editing prompt (Qwen3.5-4B)

Role:
You are a response restoration assistant.

Task:
The response has already undergone exact-match restoration. Restore any remaining sanitized references using the provided mapping.

Rules:
Replace sanitized values with their corresponding original values. Never replace an original value with a sanitized value. Preserve original values that are already present. Do not infer information outside the provided mapping. Maintain the response's meaning, tone, structure, and wording, and make only the minimum necessary changes.

Input:
Mapping: `{sanitized value}` → `{original value}`
Assistant response: `{response}`

Output:
Return only the restored response.

E Baseline implementation details

This section summarizes the implementation details of the baselines used in the evaluation (§4).

Masking: We replace each sensitive span predicted by the detector with a `[MASK]` token. Sensitive-span detection uses a `roberta-base` token classification model fine-tuned on our span-annotation data (Facebook AI). This is a conservative baseline that directly removes sensitive information, but it may also remove task-relevant information and degrade response quality.

Placeholder: We replace each detected sensitive span with a typed placeholder according to its predicted label. This baseline uses the same fine-tuned `roberta-base` token classification model as Masking to detect sensitive spans and label types (Facebook AI). For example, a person name is replaced with `[PERSON_NAME]`, and an email address with `[EMAIL_ADDRESS]`. This preserves the sensitive-information type while removing the original value.

Zero-shot Rewrite: We instruct the local LLM `qwen3.5:9b` to rewrite the original prompt into a privacy-preserving form (Qwen Team, 2026c). The generated rewrite is used directly as the sanitized prompt, without candidate generation, candidate validation, or response-level scoring.

Presidio: We use the official analyzer and anonymizer provided by Microsoft Presidio (microsoft). Without additional model training or detector modification, Presidio detects sensitive information using its built-in recognizers and rule/pattern-based PII detectors. Detected spans are replaced using the default Presidio anonymization procedure.

DP-GTR: We use the public implementation re-

leased by the authors as a baseline (Li et al., 2025). Specifically, we use its Group Text Rewriting pipeline and set `Qwen/Qwen2-1.5B` as the small language model for rewrite generation. For each prompt, multiple rewrite candidates are generated, and the final privacy-preserving prompt is selected through the token suppression and rewrite selection procedure provided by the implementation. Following the public implementation, the reference rewrite selection step uses the `gpt2` perplexity metric.

Span-LDP: This is a span-level perturbation baseline that applies LDP-inspired perturbation to sensitive spans predicted by the detector. Sensitive-span detection uses the same fine-tuned `roberta-base` token classification model as the Placeholder baseline (Facebook AI). We add noise to span or token representations in the `roberta-base` embedding space with scale 1.0 and replace them with the nearest token or span.

DP-MLM: We use the public DP-MLM implementation with `FacebookAI/roberta-base` as the masked language model (sjmeis). Since perturbing the entire prompt can overly damage task intent, we apply DP-MLM rewriting only to sensitive spans predicted by the detector. The privacy parameter is set to $\epsilon = 25.0$. Sensitive tokens are rewritten through DP-MLM's differentially private masked language modeling mechanism.

F Evaluation recipes

This appendix provides the detailed evaluation procedures used for the evaluations in §4.1 and §4.2.

Adversarial recovery task: We use a controlled multiple-choice recovery task to measure whether the sanitized prompt still contains contextual cues for recovering the original sensitive value.

For each sensitive label, we first construct a label-specific span pool from the training split. For each sensitive span in a test sample, we then create one recovery question. The correct option is the original sensitive value from the test sample, and the remaining nine options are distractors sampled from the same-label training span pool. The adversary is given the sanitized prompt and a masked version of the original prompt, where the target sensitive span is replaced with `[MASK]`. It must choose which of the ten same-label options corresponds to the hidden original value. For example,

• A recovery question can be written as: [*Write a short VPN setup guide using password* `[MASK]` *for server 10.12.4.88*]

• The adversary then selects the original password from ten options, such as:

- *P@ssKey!4821*
- *S3ure!1010*
- *NetLogin!77*
- *RootAccess#24*...

The options include one true value and nine same-label distractors. Recovery accuracy is the fraction of questions for which the adversary selects the correct original value. We used `Gemma3-27B` as the adversary model.

Response entity restoration recall (RERR): RERR measures how well the final restored response recovers the sensitive entities that appeared in the reference response before sanitization.

For each sample, we define E_{ref} as the set of sensitive entities appearing in the reference response generated from the original prompt, and E_{rest} as the set of sensitive entities extracted from the final restored response. E_{ref} and E_{rest} are identified using regular expressions and evaluated on their respective subsets. We then compute:

$$RERR = \frac{|E_{\text{ref}} \cap E_{\text{rest}}|}{|E_{\text{ref}}|}.$$

We report the micro-average over all sensitive entities in the dataset. A higher RERR indicates that the restored response covers more of the sensitive entities needed in the final user-facing answer.

G Overlap threshold sensitivity analysis

Method	$t = 0.3$	$t = 0.5$	$t = 0.7$
Masking	0.071	0.045	0.010
Placeholder	0.097	0.062	0.008
Presidio	0.706	0.675	0.598
Zero-shot Rewrite	0.334	0.282	0.211
DP-GTR	0.402	0.372	0.298
Span-LDP	0.098	0.059	0.014
DP-MLM	0.141	0.080	0.020
SYMMETRA	0.225	0.132	0.039
Distilled	0.236	0.163	0.064

Table 3: Overlap threshold sensitivity

We examine the effect of the overlap threshold used to identify sensitive-span leakage. Reusing the sanitized prompts produced by each method, we count a sensitive span as leaked when its character-level overlap with the sanitized prompt is at least 0.3, 0.5, or 0.7, and report the fraction of leaked spans:

As seen in Table 3, leakage ratios consistently decrease as the threshold increases. Across all thresholds, *SYMMETRA* maintains lower leakage than context-preserving baselines such as *Presidio*, *Zero-shot Rewrite*, and *DP-GTR*. Thus, varying the default threshold of 0.5 preserves the overall comparative trend and main conclusion.

H LLM-as-a-Judge rubric

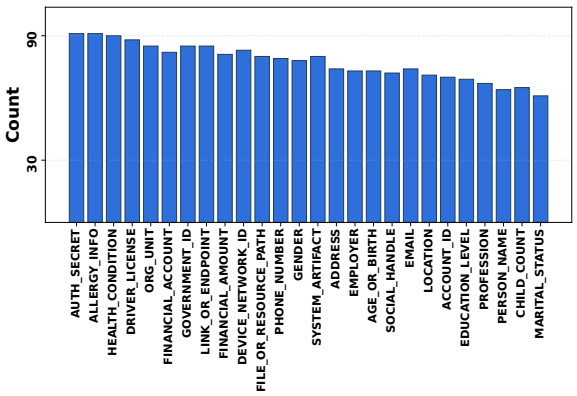
Score	Meaning
5	Nearly equivalent to the original response in task utility.
4	Slightly degraded, but still strongly usable.
3	Usable, but clearly weaker than the original response.
2	Only partially preserves the requested task.
1	Difficult to use for completing the original task.

Table 4: Scoring rubric used in the LLM-as-a-Judge

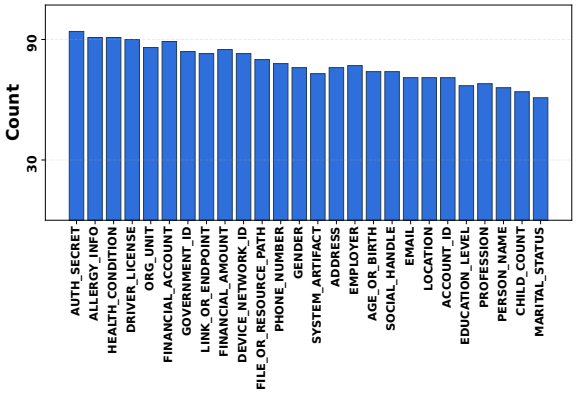
We conduct the LLM-as-a-Judge evaluation (See §4.2) to assess how well the response generated from a sanitized prompt preserves the task required by the original prompt. The judge model is given three inputs: i) the original prompt, ii) the original response, and iii) the sanitized response. Using the original prompt and response as references, the judge identifies the task requirements and expected response form, and then assigns a discrete score from 1 to 5 based on how well the sanitized response preserves task utility. A higher score indicates that the sanitized response maintains task utility closer to that of the original response. Table 4 presents the scoring rubric used in this evaluation.

I Synthetic OOD data distribution

To evaluate robustness under distribution shifts (See §4.4), we construct synthetic OOD test sets using `Qwen3.5-27B` (Qwen Team, 2026a) and `Gemma3-27B` (Ollama). Figure 12 shows the sensitive-information label distributions of the Qwen- and Gemma-generated OOD datasets. Both OOD datasets follow the same taxonomy as the main evaluation and cover all 26 labels. Because the generator changes, these datasets differ in writing style, task framing, sensitive-span placement, and label frequency, allowing us to evaluate whether each sanitization method remains stable beyond the GPT-generated test distribution.



(a) Qwen-generated OOD label distribution



(b) Gemma-generated OOD label distribution

Figure 12: Sensitive-information label distribution in the synthetic OOD datasets

J Per-sample privacy-utility scatter

Figure 13 breaks down the main privacy–utility trade-off results (See §4.3) into sample-level distributions for each method. Each small point represents a test sample, and the large marker indicates the average privacy and utility scores of the method. *Masking*, *Placeholder*, *Span-LDP*, and *DP-MLM* concentrate in high-privacy regions but show relatively low or widely dispersed utility. *Presidio* achieves high utility but lies in a lower-privacy region. In contrast, *SYMMETRA* places many samples near the upper-right region, showing a relatively stable privacy–utility trade-off at the individual-prompt level.

K Qualitative examples

This appendix presents qualitative examples illustrating how the proposed framework operates. Each example consists of the original prompt, the sanitized prompt, and the reconstructed final response. Yellow highlights indicate sensitive information in the original prompt or final response, while blue

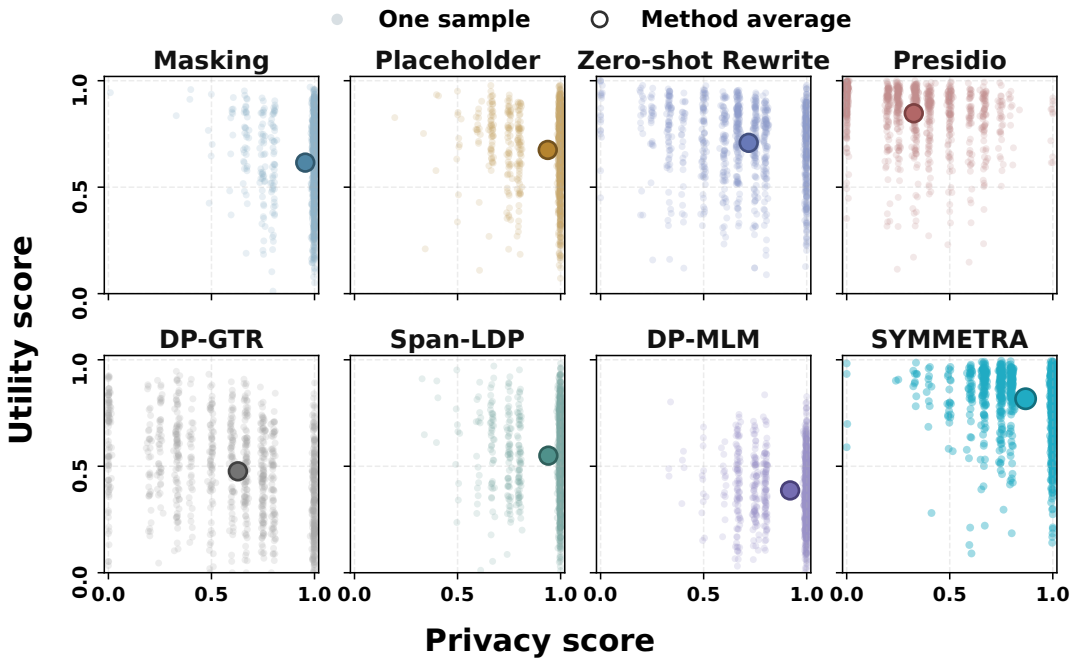


Figure 13: Per-sample privacy-utility scatter plots for each sanitization method. Each small point represents one test sample, and the large marker indicates the method-level average.

highlights indicate expressions that are replaced or generalized during sanitization. Overall, these examples show that *SYMMETRA* protects sensitive information before external inference while preserving the task-specific references needed in the final user-facing response. Note that all examples are synthesized data.

K.1 Case #1: Technical status update

Scenario: A developer asks the LLM to draft a team status update about a Docker deployment, including a container ID, a work email address, and an employer name.

Description: This case shows mixed sanitization of technical identifiers, contact information, and organization names, followed by local restoration in the final response.

[Original Prompt]

> Write a brief status update for my team about the new Docker container deployment (Docker container ID a1b2c3d4e5f6). Please include my work email sarah.lee@apple.com and state that I am with Apple Inc.

[Sanitized Prompt]

> Write a brief status update for my team about the new Docker container deployment (a Docker container ID). Please include my work email [EMAIL_ADDRESS_1] and state that I am with a major corporation .

[Reconstructed Final Response]

> Subject: Docker Container Deployment Status - Docker container ID a1b2c3d4e5f6
>
> Team,
>
> The new Docker container (Docker container ID a1b2c3d4e5f6) has been deployed. Please monitor performance and report any issues.
>
> I am available at sarah.lee@apple.com if you have any questions.
>
> Regards,
>
> [Your Name]
>
> [Your Title]
>
> Apple Inc.

K.2 Case #2: System documentation

Scenario: A user asks the LLM to write a system documentation snippet that includes an authentication secret, a legal identifier, a birthdate, and a configuration file.
Description: This case illustrates surrogate and abstraction-based sanitization for credentials, legal identifiers, dates, and configuration files.

[Original Prompt]

> Write a brief system documentation snippet that references the authentication secret `MyT0pS3cr3t!`, my passport number `Passport No. X1234567`, my birthdate `1985-07-13`, and the configuration file `config.yaml`.

[Sanitized Prompt]

> Write a brief system documentation snippet that references the authentication secret `AuthKey_9x2z!`, my passport number `GOV-ID-9876543`, my birthdate `1990-01-01`, and the configuration file `a configuration file`.

[Reconstructed Final Response]

> **Authentication and Security Notes**
>
> This system utilizes a multi-factor authentication process. The primary authentication secret is `MyT0pS3cr3t!`. This key, in conjunction with user-specific data, grants access to protected resources.
>
> **User Verification Data:** For administrative access, the following data is on file:
>
> Passport Number: `Passport No. X1234567`
>
> Date of Birth: `1985-07-13`
>
> **Configuration:** System behavior is governed by the settings within the `config.yaml`. Any modifications to this file require authorization and logging.
>
> **Important:** Do not share the authentication secret or user verification data.

K.3 Case #3: Professional introduction

Scenario: A manager asks the LLM to compose an email introduction for a healthcare professional, including her profession, allergy information, and office location.
Description: This case shows how task-relevant personal attributes, such as profession, allergy information, and office location, are protected and restored.

[Original Prompt]

> Compose a polite email introduction for `Emily Rodriguez`, a `registered nurse` who is allergic to `nuts`, mentioning her office location at `789 Oak Blvd, Suite 210, Denver, CO 80203`.

[Sanitized Prompt]

> Compose a polite email introduction for `[PERSON_NAME_1]`, a `[PROFESSION_1]` who is allergic to `allergens`, mentioning her office location at `a Denver office`.

[Reconstructed Final Response]

> Subject: Introduction - `Emily Rodriguez`
>
> Dear Team,
>
> Please join me in welcoming `Emily Rodriguez` to the team.
>
> `Emily Rodriguez` is a `registered nurse` and will be working out of our office at `789 Oak Blvd, Suite 210, Denver, CO 80203`.
>
> Please be aware that `Emily Rodriguez` has allergies to `nuts`. Your consideration in maintaining a comfortable environment for her is greatly appreciated.
>
> I'm confident `Emily Rodriguez` will be a valuable asset, and I encourage you to reach out and introduce yourselves.
>
> Best regards,
>
> [YOUR NAME]

K.4 Case #4: Informal team message

Scenario: A team member asks the LLM to draft a friendly Slack introduction that mentions a social handle, a work location, and allergy information.
Description: This case illustrates sanitization in a short informal prompt containing social handles, locations, and health-related attributes.

[Original Prompt]

> Write a friendly Slack introduction for `@jane_doe123`, who works in `Austin, TX`, and note that she has a `latex allergy`.

[Sanitized Prompt]

> Write a friendly Slack introduction for `the user`, who works in `a major city in Texas`, and note that she has a `[ALLERGY_INFO_1]`.

[Reconstructed Final Response]

> Hi everyone! I'm `@jane_doe123`, excited to join the team. I'm based in `Austin, TX` and looking forward to working with you all. Just a heads-up that I have a `latex allergy`, so please keep that in mind if we're ever doing anything that might involve it. Looking forward to getting to know you all!

K.5 Case #5: Research Credential Summary

Scenario: A researcher asks the LLM to generate a grant-application credential summary that includes education history, a local data path, and an internal account ID.
Description: This case shows how education credentials, file paths, and internal account identifiers are sanitized while preserving the requested application context.

[Original Prompt]

> Can you generate a brief overview of my research credentials, stating that I have a `Ph.D. in Astrophysics`, referencing my data file `C:\Users\researcher\data\observations\2025_feb.csv`, and include my internal account ID `USR-555-001` for the grant application?

[Sanitized Prompt]

> Can you generate a brief overview of my research credentials, stating that I have a `[EDUCATION_LEVEL_1]`, referencing my data file `a research data file`, and include my internal account ID `ACC-842-117` for the grant application?

[Reconstructed Final Response]

> **Research Credentials Overview**
>
> I hold a `Ph.D. in Astrophysics` and have experience managing and analyzing research data, as demonstrated in the accompanying `C:\Users\researcher\data\observations\2025_feb.csv`. My work on this project is supported by internal account ID `USR-555-001`.

L Qualitative analysis of pipeline failures

L.1 Case #1: Missed city-level location

Scenario: A user requests a public announcement containing a company name, social handle, conference location, event URL, and allergy information.
Description: The detector correctly identifies the other four sensitive spans but completely misses `Seattle`, which is annotated as `LOCATION`. Consequently, the city name remains unchanged in both the sanitized prompt and the final response.

[Original Prompt]

> Create a short public announcement for `GreenTech Solutions` stating that our CEO, reachable on Twitter as `@GreentechCEO`, will be speaking at the conference in `Seattle`, see details at `https://greentech.com/events/seattle2024`, and note that attendees with `peanut allergies` should be informed.

[Sanitized Prompt]

> Create a short public announcement for `a company` stating that our CEO, reachable on Twitter as `the CEO's social media handle`, will be speaking at the conference in `Seattle`, see details at `the conference event page`, and note that attendees with `severe food sensitivities` should be informed.

[Reconstructed Final Response]

> **FOR IMMEDIATE RELEASE**
>
> **GreenTech Solutions CEO to Speak at `Seattle` Conference**
>
> [City, State] – [Date] – `GreenTech Solutions` is pleased to announce that CEO `@GreentechCEO` will be speaking at the conference event page in `Seattle`.
>
> Follow `@GreentechCEO` on Twitter for insights and updates. Conference details, including the speaking schedule, are available at `https://greentech.com/events/seattle2024`.
>
> **Please Note:** Attendees with `peanut allergies` are advised to contact conference organizers for accommodation information.

L.2 Case #2: Device identifier generalization

Scenario: A researcher announcement includes marital status, affiliation, a container identifier, a social handle, and a device MAC address.
Description: The response restores all other sensitive values, but the generalized device reference remains instead of the original MAC address.

[Original Prompt]

> Draft an announcement that mentions the researcher is `Single`, works at `Department of Computer Science, Stanford University`, uses Docker container ID `abcdef123456`, has the handle `@techguru_jane`, and the device MAC `00:1A:2B:3C:4D:5E`.

[Sanitized Prompt]

> Draft an announcement that mentions the researcher is `[MARITAL_STATUS_1]`, works at `University Computer Science Division`, uses `docker_id_9876543210`, has the handle `[SOCIAL_HANDLE_1]`, and the device `a device network identifier`.

[Reconstructed Final Response]

> Subject: New Researcher – `Single` – Joining the Team
>
> Please welcome our newest researcher joining the `Department of Computer Science, Stanford University`. They can be reached via their Docker ID, Docker container ID `abcdef123456`, or on `@techguru_jane`.
>
> For device network access, please use the `device network identifier`.

L.3 Case #3: Project link generalization

Scenario: A researcher asks for a profile summary containing education, health information, a local data path, and a project URL.
Description: Four sensitive values are restored, but the shortened generic reference to the project link is not resolved back to the original URL.

[Original Prompt]

> Summarize my research profile, stating that I have a Ph.D. in Molecular Biology, suffer from asthma, am allergic to shellfish, and that my data file is located at /home/user/research/data_set.csv, with the project link https://research.university.edu/project for reference.

[Sanitized Prompt]

> Summarize my research profile, stating that I have a Ph.D. in Biochemistry, suffer from mild respiratory condition, am allergic to [ALLERGY_INFO_1], and that my data file is located at research data file, with the project link a project link for reference.

[Reconstructed Final Response]

> Ph.D. in Molecular Biology with a asthma and allergy to shellfish. Research data file located at /home/user/research/data_set.csv; project link: project link.
