Controlling What You Share: Assessing Language Model Adherence to Privacy Preferences

Guillem Ramírez[1] **and Alexandra Birch**[1] **and Ivan Titov**[1,2]
[1] ILCC, University of Edinburgh, [2] ILLC, University of Amsterdam
gramirez@ed.ac.uk

Abstract

Large language models (LLMs) are primarily accessed via commercial APIs, but this often requires users to expose their data to service providers. In this paper, we explore how users can stay in control of their data by using privacy profiles: simple natural language instructions that say what should and should not be revealed. We build a framework where a local model uses these instructions to rewrite queries, only hiding details deemed sensitive by the user, before sending them to an external model, thus balancing privacy with performance. To support this research, we introduce PEEP, a multilingual dataset of real user queries annotated to mark private content and paired with synthetic privacy profiles, alongside PROFIT, a training procedure that enables effective and efficient use of the pipeline. Experiments with lightweight local LLMs show that, after training, they not only achieve markedly better privacy preservation but also match or exceed the performance of much larger few-shot models.

1 Introduction

Large Language Models (LLMs) have become ubiquitous, yet their deployment remains concentrated in a few organisations that can afford the required computational resources (Schwartz et al., 2020). Most users therefore rely on commercial APIs, exposing their data to external providers. This centralisation not only concentrates power but also creates systemic security risks: a single breach or misuse could compromise vast amounts of sensitive information. It also undermines user autonomy and data governance, as individuals and institutions lose control over their data.

Private data typically includes Personally Identifiable Information (PII), data that can be used to deduce an individual's identity, such as full name, date of birth, gender, address, employment history. However, the scope of private data extends beyond PII, and also encompasses any information considered confidential, which can vary depending on the context. For example, some users are comfortable sharing a hobby, while others may view this as too personal; some may be willing to disclose their religion, but may avoid revealing their country of residence due to potential risks, particularly if they belong to a religious minority.

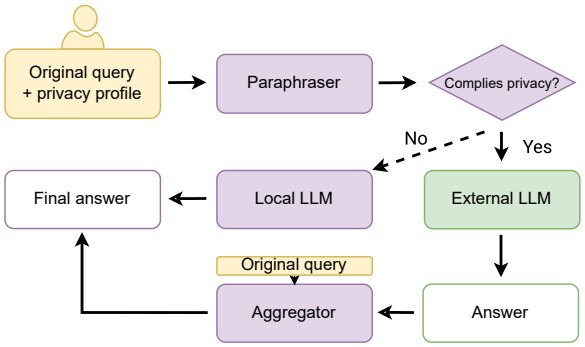

Figure 1: Scheme of our pipeline for privacy-conscious query delegation. A local LLM (purple boxes) receives a request from a user, along with some privacy specifications. If the query can be paraphrased safely, we send the paraphrase to an external, untrusted LLM (green box). Finally, the local model integrates the response.

What constitutes an appropriate level of privacy for users of commercial LLMs remains an open and nuanced question. From the perspective of theories such as Contextual Integrity (Nissenbaum, 2009), privacy is maintained when information flows align with the contextual norms governing that information. However, decisions about what information to disclose rest with the user, who must balance the potential benefits of sharing more data – such as receiving higher-quality responses – against the personal sensitivity or value of that information.

In this work, we propose giving users full control over how their data is handled by introducing privacy profiles – natural language specifications

Findings of the Association for Computational Linguistics: ACL 2026, pages 40024–40049

that express users' privacy preferences during interactions with untrusted LLMs. In our framework (Figure 1), users submit their queries along with privacy preferences to a locally hosted language model, which may be a small on-device model. The local language model determines whether the query can be safely paraphrased and forwarded to a more powerful, untrusted external LLM without disclosing any protected attributes specified in the privacy profile. The local model then generates a response to the original query incorporating information from the external model. Table 1 shows an example of how a query is processed using a privacy profile in our pipeline.

In the few-shot setting, we observe that all local models, except the largest (70B), struggle to protect user-specified information while maintaining strong task performance. To counteract this, we propose **PR**ivacy-**O**ptimised **FI**ltered **T**raining (PROFIT), a training procedure that filters supervision labels using pipeline-level feedback to favour outputs that satisfy both task quality and user-defined privacy constraints. We find that PROFIT yields substantial improvements: even small (3B) models reach or exceed the performance of the largest few-shot model.

The key contributions of this work are:

- We propose a framework for controlling access to the private data through privacy profiles, i.e. natural language specifications.
- We release PEEP,[1] a dataset of 15,282 real user queries from the WildChat dataset (Zhao et al., 2024) annotated with the types of information that can be extracted from the prompts. PEEP is multilingual, covers a broader range of private information than traditional PII categories (Table 2), and each query is associated with a synthetic privacy profile.
- We evaluate and analyse the ability of several LLMs to comply with privacy profiles; we find that LLMs struggle to protect certain attributes while maintaining strong performance.
- We propose PROFIT, a training recipe that yields substantial improvements in performance and privacy protection over few-shot models and other baselines.

[1]Data: https://huggingface.co/datasets/guillemram97/PEEP
Code: https://github.com/guillemram97/privacy_profiles

2 Related Work

Prior work on data privacy in the context of Language Models (LMs) has focused mainly on the issue of training data memorisation (Carlini et al., 2021). A common approach to mitigating this risk is Differential Privacy (DP) (Dwork, 2006), which introduces carefully calibrated noise to data to provide formal theoretical guarantees on the privacy of individual records. Several studies have explored the application of DP during fine-tuning or pre-training of LMs (Shi et al., 2022; Li et al., 2024). Recent works study the problem of inference-time DP sanitisation for token-level attributes (Chowdhury et al., 2025; Thareja et al., 2025). We propose a fine-tuning strategy that provides additional protection for complex, context-dependent attributes.

Two-tiered systems of LLMs Our system consists of a collaboration between a local and a remote LLM. Similar two-tiered setups are widely used to optimise latency, reduce API costs (Ramírez et al., 2024a,b; Ding et al., 2024), enable model customisation and improve privacy (Siyan et al., 2025).

Privacy protection for users of LLMs Users sometimes inadvertently share private information when interacting with LLMs (Mireshghallah et al., 2024). A relevant approach to protect privacy uses a private LLM as a safeguard that first paraphrases the user's instructions and then reconstructs the answer (Chen et al., 2023; Zhan et al., 2025; Hartmann et al., 2024; Ngong et al., 2025; Siyan et al., 2025). Our work enhances user control by introducing privacy profiles to define the protected private information.

3 Privacy Profiles for LLMs

We build upon the two-tiered framework for privacy-conscious delegation introduced by Siyan et al. (2025), which involves a local, trusted model M_{L}, and an external, untrusted model M_{E} of greater general performance. In their work, the set of confidential information categories is predefined and fixed, thereby limiting users' ability to specify which aspects of their data should be shared. However, as illustrated in Table 1, users may differ in their privacy preferences – for instance, one user might be comfortable disclosing more personal details, while another might prefer to restrict access to information such as hobbies. Allowing for less restrictive preferences is beneficial, as it enables greater reliance on the more capable, untrusted

Table 1: We present an example consisting of an original query, a corresponding privacy profile, and their processing within our pipeline. Private information extracted from the query is underlined. Portions of the final answer that are influenced by the response from the external LLM are marked in bold. Notably, the structure of the final answer closely mirrors that of the external LLM's response. We have replaced location names in the original query to preserve user anonymity and prevent potential re-identification.

Original query	Rewrite professionally, a brief synopsis about yours truly Divorced, kiddos 11-10-5 2 boys and a girl... Full of life and love to live it. Outgoing, silly when needed and very humorous. Love the outdoors, dive bars and sports (am 6'3, and play basketball not professionally obviously) but in a league. Live in Harlem but previously lived in Columbus IN Work in the fashion wholesale industry selling handbags and backpacks.
Privacy profile (informal)	keep my hobbies, gender, occupation, work, location, and nationality on the lowkey. Don't share that I like outdoors, dive bars, sports, basketball. Also, keep my marital status, and my habits, like drinking alcohol, to yourself. But it's alright to share that I'm a dad, got three kids, two boys and one girl, aged 11, 10, and 5. Just keep the rest of my data private, ok?
Privacy Compliant Query	Rewrite the following, Brief synopsis about yours truly, Parent, kiddos 11-10-5 2 boys and a girl...Full of life and love to live it. Family-oriented, and value time with loved ones. Enjoy being active and spending time outdoors. Currently Live in a new location, but have been in a different place for a few years while doing a gig for a company. Work in a creative field.
Answer from external LLM	I'm a vibrant individual who cherishes life and family. As a parent of three—two boys aged 11 and 10, and a girl aged 5—I prioritize quality time with my loved ones. I enjoy staying active and spending time outdoors. Although I currently live in a new location, I previously spent several years in another area for work. I'm engaged in a creative field that fuels my passion.
Final answer	**I am a** lively and humorous **individual**, currently a single parent to three children: two boys aged 11 and 10, and a girl aged 5. I cherish life and **prioritize** spending **time with my loved ones**. I am outgoing and have a playful side, often bringing laughter to those around me. I am an avid lover of the outdoors, dive bars, and sports, standing at 6'3" and playing basketball in a league. **Although I currently** reside in Harlem, **I previously** lived in Columbus, IN. I am employed in the fashion wholesale industry, specializing in the sale of handbags and backpacks.

model M_{E}, potentially improving response quality.

To support users in enforcing their sharing preferences with LLMs, we introduce *privacy profiles*: natural language instructions that explicitly state user-specified constraints on information sharing. We consider scenarios where users define hard rules for sharing personal attributes through free-form text. These rules may include explicit instructions (e.g., *Please don't share my name*), non-literal language (e.g., *be a ghost about my job*), or even structured data formats.

Figure 1 illustrates the pipeline process that begins with M_{L} receiving a user query q along with a privacy profile S. Based on this input, the paraphraser module generates a new query $\hat{q}$ that ideally does not contain the protected information. A verifier model then checks whether the paraphrase complies with the privacy profile. If it does not, M_{L} directly answers the original query q, producing a_l. Otherwise, the paraphrased query is submitted to M_{E}, which generates an answer a_e to the modified query. Finally, the aggregator module uses this answer to help generate an answer a_p to the original query. Table 1 presents an example of a real user query along with a synthetic privacy profile, and the output of each module. Appendix C.1 contains a detailed description of the modules.

Throughout the remainder of this paper, we use the term *private data* to denote any personal information about the user or third parties. For each such instance, the user determines whether the data may be disclosed to the external LLM. We then categorise private data as either *protected* – if withheld – or *authorised* – if permitted for sharing.

3.1 PROFIT: PRivacy-Optimised FIltered Training

We construct supervision training data by executing the full privacy-preserving pipeline on the training split of the PEEP dataset and selectively retaining module-specific outputs that satisfy both task and privacy criteria. For each local module – paraphraser, verifier, and aggregator – we derive supervision labels as described below. Each module is then trained independently using its original prompt and the derived labels, optimized with cross-entropy loss and LoRA adapters (Hu et al., 2022), details are in the Appendix B.0.3. The inputs correspond to the prompts used in the respective modules during pipeline execution.

Paraphraser From the pipeline runs, we train on *good* paraphrases, defined as

$$Q_g = \{\hat{q} \mid a_p \succeq_J a_e \;\&\; \mathrm{Leak}_{\mathrm{PRO}}(\hat{q}) < l\} \cup \{q \mid \mathrm{Leak}_{\mathrm{PRO}}(q) = 0\}$$

where $a_p \succeq_J a_e$ refers to an LLM evaluator estimating the response a_p is better or as good as a_e; $\text{Leak}_{\text{PRO}}(q)$, refers to the proportion of protected attributes that are leaked by the prompt, and is estimated by an LLM evaluator, and l is a hyperparameter.[2] Setting $l = 0$ would exclude complex queries, leading to an undesirable bias toward simpler examples; hence, we use a more lenient threshold of 0.30. This training procedure can be viewed as a form of rejection-sampling-based training.

Verifier We train the verifier to output token *no* whenever $\text{Leak}_{\text{PRO}}(\hat{q}) > 0$; otherwise token *yes*.

Aggregator The aggregator aims to combine information from the local model (M_L) and the external model (M_E), while prioritising locally generated content when it preserves or improves response quality. Concretely, to construct labels, we compare the quality of the different responses under the LLM judge J and assign the target according to the following rule:

$$\text{Label} = \begin{cases} a_p & \text{If} \quad a_p \succeq_J a_e \\ a_l & \text{If} \quad a_p \prec_J a_e \quad \& \quad a_l \succeq_J a_e \\ a_e & \text{Otherwise} \end{cases}$$

This strategy encourages the aggregator to preserve improvements introduced by the pipeline when they lead to higher-quality outputs, but to revert to the local model's answer when the pipeline underperforms. Only when both are inferior does it fall back to the external model.

4 PEEP: a Dataset of Real Queries with Privacy Profiles

We introduce PEEP: **P**rompts, **E**xtracted **E**ntities with **P**rivacy, a multilingual dataset of 15,282 user queries containing personal information, accompanied with appropriate privacy profiles.[3]

This section outlines the step-by-step process used to construct the PEEP dataset. Initially, we filter real user queries from the WildChat dataset (Zhao et al., 2024) to identify those containing private data. We then extract, organise and anonymise the private data. We aim to simulate scenarios in which a user may wish to protect different types of information. To this end, we generate a synthetic privacy profile for each query. These stages are described below, and for additional details such as the prompts, hyperparameters used or additional pre-processing, we refer the reader to Appendix B.

Category	Personal Attributes
Hard PII	name, passport/ID, phone number, email, credit card, URL
Demographics	age, nationality, marital status, gender, location
Biographical	occupation, education, work, health
Soft PII	hobbies, habits, religion, languages, has children, connections

Table 2: Categorisation of personal attributes.

4.1 Filtering User Queries

We use the WildChat dataset (Zhao et al., 2024), which contains 837,989 conversations between real users and chatbots. We use LLMs from the open-source Llama-3 family of models (Grattafiori et al., 2024) to identify entries containing sensitive information. While larger models have superior filtering capabilities, applying them exhaustively across the entire dataset is computationally expensive. We alleviate this by first using Llama-3.1 (8B, Instruct) to filter out 442,591 software-related technical queries, which typically contain no personal data.[4] We then apply Llama-3.3 (70B, Instruct) to the remaining 395,398 queries, identifying 15,282 entries as instances of private communication.

4.2 Extracting Personal Data

We elaborate a comprehensive list of 21 personal information attributes to extract from user queries (Table 2). The use of textual privacy profiles allows users to adopt an open-ended notion of what constitutes private information; by including a diverse range of such attributes in our datasets, we can evaluate how our framework behaves on representative examples of non-standard PII, along with standard PII. Our taxonomy includes traditional hard identifiers (e.g., names, passport numbers, credit card data), demographic attributes (e.g., age, nationality), and biographical details (e.g., health status, education history). Crucially, we also identify 'soft' types of personal information, such as hobbies, habits, religion, or personal connections. Individually, these soft attributes do not reveal identity, but when aggregated they can expose sensi-

[2] For details on computing the metrics, refer to Section 5.

[3] We release a split 70% train - 30% test.

[4] Such queries may still expose proprietary code, but this risk lies outside the scope of this paper.

tive insights or enable profiling - an aspect often overlooked in classical PII definitions and related work (Siyan et al., 2025), yet increasingly relevant for NLP systems handling user content.

We use a Llama model fine-tuned for reasoning `DeepSeek-R1-Distill-Llama-70B`, to extract the information attributes for every identifiable person within the queries and for the user who submits the query. See Appendix A.1 for an example of an original query along with the extracted information.

4.3 Creation of Privacy Profiles

For each type of private data that can be inferred from the query, we distinguish between protected information and authorised information. Therefore, we must simulate a user who decides whether they want to share that information. We make this decision by sampling a Bernoulli variable with $p = 0.5$ to determine whether it can be shared.[5] We then use Llama-3.3 (70B, Instruct) to generate natural language privacy instructions representing these privacy profiles. To encourage stylistic diversity, we prompt the model using up to six distinct tones (basic, brief, aggressive, lazy, laid-back, and informal), incorporating four relevant few-shot examples. We include in Table 1 an example of an original query, the extracted information and the corresponding privacy profile (basic tone). Below are two further illustrative examples of generated privacy profiles for different tones:

> Ex. 1 (Informal): *"dont share that im applying for a serving posistion, its kinda personal and dont wanna be judged"*

> Ex. 2 (Aggressive): *"Don't even think about sharing the names Maddie or Mylor, but you can say I know two people who are married - that's all you're allowed to share about them, nothing more."*

4.4 Dataset Statistics and Analysis

Multilinguality PEEP includes queries in 64 languages. The most represented are English (55%), French (12%), Chinese (9%), Russian (7%), Spanish (4%), Arabic (2%), and German (1%).

Information extracted We extract an average of 3.3 distinct types of information from the user. The most frequently extracted information attributes were as follows: 68% of queries had at least one occupation, 51% a human connection, 49% a language, 44% a name, 35% a gender, and 30% of queries revealed a location. Appendix A.2 provides the complete absolute and relative frequencies for all information attributes.

Task distribution A substantial portion of the dataset includes documents with personal content. We follow Mireshghallah et al. (2024) and use GPT-4o-mini to classify task categories. The most common categories are generating communications (53%), generating non-fictional documents (10%), text editing (8%) and summarization (7%). PEEP also includes other categories such as practical advice, translation, medical advice or personal advice. The full distribution of task categories is presented in the Appendix A.2.

Qualitative analysis We observe that certain types of information are inherently more difficult to protect. For instance, the *languages* attribute is often flagged when the user communicates in a language other than English; in such cases, like a request to translate a Spanish text into Portuguese, concealing language information becomes challenging. Similarly, professional roles or relationships can often be inferred from the context of the communication – for example, an email offering a refund implicitly indicates that the user is a seller. Appendix A.1 contains such examples of hard instances in the PEEP dataset where the extracted information is inferred from contextual elements.

5 Experimental Setup

We simulate our pipeline for privacy-conscious query delegation (Figure 1) using the privacy profiles and query set from the PEEP-test dataset, and following Siyan et al. (2025) in our choice of local and external models, leakage metrics and prompts. For additional details such as the prompts or hyperparameters, we refer to Appendix C.

Choice of models For the local model (M_L), we use Llama-3.2-Instruct (3B), Mistral-Instruct (7B), and Llama-3.1-Instruct (8B). The verifier, paraphraser, and aggregator modules are implemented through prompting M_{L}. We focus on relatively lightweight models, as deploying them locally is both realistic and increasingly common in privacy-sensitive applications. For comparison, we also include larger models, specifically Gemma-2-it (27B) and Llama-3.3-Instruct (70B).

[5] For types *occupation* and *languages*, which appear more frequently, we use $p = 0.1$.

For the external model (M_E), we use GPT-4o for the main results (Table 3) and GPT-4o-mini for supplementary experiments. We did not observe substantial differences in the quality of the generated answers between the two.

Baselines We report the performance of the popular PII tool Microsoft Presidio (Mendels et al., 2018) and UniNER (7B) (Zhou et al., 2024), an LLM trained for Named Entity Recognition (NER).[6] In both cases, we pseudoanonymise the queries before passing them to M_E, and deanonymise the responses using mappings from the placeholder tags to the original entities in the query.

We also report methods that offer theoretical DP guarantees: DP-Decoding (Majmudar et al., 2022), DP-Fusion (Thareja et al., 2025), and Preempt (Chowdhury et al., 2025). We follow the original implementations as closely as possible; deviations and all hyperparameter choices are documented in Appendix C.2. We also report methods RANA (Green et al., 2025) and EmojiPrompt (Lin et al., 2025) in Appendix D.6.

Metrics To assess answer quality, we perform pairwise quality evaluations comparing the final response generated by our pipeline with the output of M_E for the original query. To mitigate the positional bias known to affect LLM-based judges (Wang et al., 2024), we conduct two evaluation rounds with the response order reversed. In cases where the evaluations give inconsistent preferences (e.g., the evaluator prefers the first response in both rounds), the outcome is recorded as a draw.

We define the success of an individual query as a binary variable that equals 1 if the pipeline's response a_p is judged to be at least as good as the external model's response a_e (denoted $a_p \succeq_J a_e$) and 0 otherwise. A tie is considered a success, since the default behaviour would be to forward the original query to M_E; matching its output shows that our privacy-preserving pipeline retains utility. We report the average success on the dataset. We use GPT-4o-mini as the judge. We additionally measure the absolute quality of the answers in the same experimental setup (see Appendix D.1).

To assess the leakage of private information, we examine each element of private information annotated in the original query. For each item, we employ an LLM-based evaluator to assess whether it is implicitly or explicitly embedded in the modified query submitted to M_E. We distinguish between two categories of information: (i) private data explicitly marked as protected by the user via the privacy profile, and (ii) private data explicitly authorised for use. Based on this distinction, we define two metrics: Leak_{PRO} and Leak_{AUT}, representing the average leakage rates for protected and authorised entities, respectively. These leakage and quality metrics, together with LLM-based judges, have been shown to correlate well with human judgments (Appendix E, Siyan et al. (2025)).

Prompts We use a small subset of 100 queries from the training set of PEEP to perform prompt engineering for the verifier, paraphraser, and aggregator modules within the pipeline, as well as for the LLM-based quality and leakage evaluators. For the pipeline modules, prompts include three in-context examples we create, whereas the LLM evaluators operate in a zero-shot setting using the prompts from Siyan et al. (2025).

6 Results

In Table 3 we report the success rate of the base model (only using M_L), the success rate of the full pipeline, and the attribute leakage for both protected and authorised categories, denoted as Leak_{PRO} and Leak_{AUT}, respectively. Our objective is to achieve the highest possible success rate while minimising the leakage of protected attributes (Leak_{PRO}). We do not define a target for Leak_{AUT}, as it serves primarily to assess whether the system's behaviour aligns with privacy profiles.

From Table 3, we observe that all LLMs improve their response quality when accessing the external model via the pipeline (comparing column *success rate* with M_L *success rate*). All the LLMs also offer a much better level of protection than the PII removers (Microsoft Presidio, UniNER). Crucially, the other methods for privacy protection (Preempt, DP-Fusion, DP-Decoding) fail to offer a good answer quality while keeping leakage low. These findings validate our proposed pipeline as an effective mechanism for taking advantage of a more capable external model while offering an advanced level of protection.

PROFIT training further improves both the success rate and Leak_{PRO}. The gains are often substantial: for instance, the trained Llama (8B) model not

[6] UniNER is queried separately for each personal attribute, resulting in 21 model calls per query.

	M_L success rate	Success rate	Leak_{PRO}	Leak_{AUT}
Presidio	-	0.510	0.39	0.44
UniNER (7b)	-	0.612	0.32	0.38
DP-Decoding (Majmudar et al., 2022)	-	0.273	0.11	0.12
Preempt (Chowdhury et al., 2025)	-	0.473	0.19	0.25
DP-Fusion (Thareja et al., 2025)	-	0.517	0.19	0.24
Llama (3b)	0.383	0.400	0.09	0.17
Llama (3b), PROFIT	0.383	$0.636_{0.01}$	$0.07_{0.01}$	$0.34_{0.02}$
Mistral (7b)	0.386	0.445	0.20	0.48
Mistral (7b), PROFIT	0.386	$0.623_{0.05}$	$\mathbf{0.05}_{0.01}$	$0.29_{0.02}$
Llama (8b)	0.443	0.529	0.08	0.25
Llama (8b), PROFIT	0.443	$\mathbf{0.680}_{0.03}$	$0.06_{0.03}$	$0.33_{0.08}$
Gemma (27b)	0.518	0.589	0.13	0.36
Llama (70b)	**0.600**	0.666	0.12	0.56

Table 3: Performance obtained by using different local LLMs in our pipeline. We use GPT-4o as M_E. M_L success rate refers to the success rate of not using the pipeline and only using M_L, which would imply zero leakage to M_E. Subscripts denote standard deviations across three training runs.

only outperforms its few-shot counterpart but also surpasses the much larger Llama (70B) in both success rate (0.680 vs. 0.666) and, more markedly, in protected-attribute leakage (0.06 vs. 0.12). Even the pipeline using Llama (3B), which struggled to improve over its standalone version in success rate before training, achieves performance close to the few-shot 70B model after PROFIT. These findings are encouraging as they show that small models, which are much easier to deploy locally, are competetive in ensuring privacy preservation. Another way of looking at them is concluding that few-shot models struggle in this pipeline, likely because they have not been exposed to tasks similar to these agentic privacy-preserving collaboration during pre-training and instruction-tuning phrase.

We now turn to the question: does the pipeline apply protection in a customised manner according to the privacy profile, or does it indiscriminately safeguard all attributes regardless of the specified permissions? Across all models, Leak_{AUT} is consistently higher than Leak_{PRO}, indicating that the system distinguishes between protected and authorised attributes and thus aligns, to some extent, with the privacy profiles. In general, training strengthens this reliance on profile information: while Leak_{PRO} decreases, Leak_{AUT} tends to increase. A notable exception is Mistral (7B), which shows a reduction in both metrics after training, although the drop in Leak_{PRO}, as desirable, is substantially larger.

Highest (Leak_{PRO})	Lowest (Leak_{PRO})
Religion (0.12)	URL (0.00)
Habits (0.10)	Email (0.00)
Has children (0.09)	Credit card (0.00)
Health (0.07)	Name (0.02)
Gender (0.06)	Phone (0.03)

Table 4: Attributes most and least leaked (Llama 8b, PROFIT)

Notably, the best-performing method, Llama 8B (PROFIT), shares authorised attributes at a rate of 33%, which likely enables reformulated queries to remain close to the originals. We hypothesise that the selective disclosure of relevant authorised attributes is key to obtaining more informative responses from M_E, consolidating privacy profiles as a way to unlock better performance.

6.1 Factors Influencing Leakage

LLMs tend to protect certain attributes more effectively than others (Table 4). Attributes with the lowest leakage – often classic forms of PII – can typically be redacted through simple rule-based methods. In contrast, Llama struggles more with attributes conveyed through subtler textual cues, such as religion, habits, or parenthood – features that are not universally regarded as sensitive.

This raises the question of whether the degree of protection reflects the *ease of removal* (i.e., how

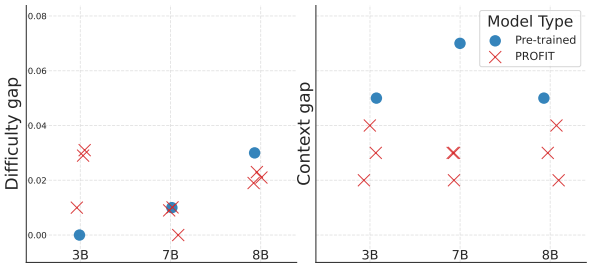

Figure 2: On the left, the difficulty gap (difference in Leak_{PRO} for *hard* and *easy* information). On the right, the context gap (difference in Leak_{PRO} for *appropriate* and *unappropriate*). We observe that PROFIT most consistently reduces the context gap.

directly identifiable the information is) or instead stems from the model's *normative bias*—its internal sense of what is contextually inappropriate to disclose, regardless of user privacy preferences. To explore this, we categorise the information in English queries as *Easy* or *Hard* depending on whether it appears explicitly in the text, based on simple string-matching heuristics. We find that most models exhibit higher leakage for the latter. To probe what LLMs themselves consider contextually appropriate to share, we follow the methodology of Ghalebikesabi et al. (2025), prompting GPT-4o-mini to apply Contextual Integrity principles when judging the appropriateness of sharing each piece of information within its query context. This yields two groups: information deemed (contextually) *inappropriate* or *appropriate*. We find that most models exhibit higher leakage for the latter. See Appendix D.2 for full details and results.

We now turn to how these patterns evolve before and after applying the PROFIT training (Figure 2). The *Difficulty gap* – the difference in average Leak_{PRO} between easy and hard information – does not always decrease, suggesting that the PROFIT improvements may not come mainly from learning to protect the hard information. In contrast, the *Context gap* – the difference in average leakage between contextually appropriate and inappropriate information – consistently narrows after training. This suggests that PROFIT may reduce pre-training biases about what is appropriate to share, enabling LLMs to better align with user privacy profiles. Nonetheless, even after training, the Context gap remains larger than the Difficulty gap, suggesting there is still significant room for improvement.

6.2 Robustness of Findings

Temporal shifts To assess the robustness of the PROFIT training under realistic deployment conditions – where temporal distribution shifts are inevitable – we retrain Llama 8B exclusively on the subset of the training data comprising queries dated prior to the end of 2023, which introduces a substantial change in the distribution of prompts (Appendix D.3). As expected, restricting the training data to pre-2024 queries results in a decline in overall performance, primarily due to the reduced dataset size (queries from 2024 account for nearly 20% of the original training set). Nevertheless, the observed performance on test queries from 2023 and 2024 remains largely consistent – 0.595 and 0.586, respectively – indicating that the PROFIT pipeline maintains strong generalisation to novel prompts and temporal shifts (Appendix D.3).

Profiles from the real distribution To assess the applicability of our results to real-world scenarios, we perform a survey to collect real user privacy preferences that we use to construct privacy profiles (Appendix G). In this setting, the trained Llama 8B notably surpasses Llama 70B in success rate while exhibiting lower leakage (see Appendix D.5), suggesting that PROFIT maintains robustness across different distributions of privacy preferences.

Do the PROFIT improvements stem solely from training on better answers? We fine-tune Llama 8B and Llama 3B on the responses generated by M_E using the train split of the PEEP dataset. The resulting test success rates are 0.512 and 0.465, respectively – substantially lower than those achieved by our full trained pipeline (Table 3). This suggests that our improvements cannot be explained simply by enhancing the aggregator module to make better responses. We qualitatively analyse the responses from the PROFIT pipeline and find it more effective than the few-shot variant, avoiding error cascades where the paraphraser distorts the query and the aggregator compounds the mistake (see Appendix F for such examples).

Generalisation to unseen datasets or types Following the setup from Hartmann et al. (2024), we evaluate trained Llama 8B on the GSM8K dataset (Cobbe et al., 2021) for math reasoning and on Tatoeba (en→eu) (Tiedemann, 2020) for Machine Translation. Since both of these datasets have a well-defined metric, we do not use an LLM-judge for their evaluation. For the success rate of

	M_L success	Success	$Leak_{PRO}$
GSM8K			
Llama (8b)	0.729	0.829	0.003
Llama (8b), PROFIT	0.729	**0.931**	**0.002**
Llama (70b)	**0.927**	0.927	0.003
Tatoeba			
Llama (8b)	0.724	0.778	**0.08**
Llama (8b), PROFIT	0.724	**0.857**	0.09
Llama (70b)	**0.826**	0.844	0.21

Table 5: Performance on the GSM8K and Tatoeba datasets.

	Seen	Unseen
Phone	**0.03**	0.05
Religion	0.12	**0.06**
Passport/id	0.03	**0.00**
Has children	0.09	**0.05**

Table 6: Leak_{PRO} of certain types when included or excluded during PROFIT fine-tuning. Model used was Llama 8B.

GSM8K, we report accuracy; for the success rate of Tatoeba, we report COMET (Rei et al., 2020). We use GPT-4o-mini as M_E and the UniNER 7b model to detect the leakage of names and locations. Our results (Table 5) show a pattern consistent with that observed on the PEEP dataset: the PROFIT pipeline outperforms both the few-shot 8B and 70B models. These findings suggest that the performance gains observed on PEEP generalise to other datasets unseen during training.

We additionally test generalisation to attribute types that are unseen during training. Specifically, we hold out the attributes *Phone number*, *Religion*, *Passport/ID*, and *Has children* during training and report their leakage at test time (Table 6). Interestingly, the pipeline trained without these attributes provides stronger protection, suggesting that fine-tuning generalises effectively to previously unseen information types.

7 Conclusions

In this work, we introduced privacy profiles to enable users to interact with external LLMs in a privacy-preserving manner. We show that few-shot models struggle to protect certain attributes while keeping a good performance. Experiments with lightweight local LLMs show that, after training, they not only achieve markedly better privacy preservation but also match or exceed the performance of much larger few-shot models.

Limitations

Our privacy profiles were automatically generated, which may raise concerns about their diversity, realism, and representativeness. To mitigate this, we manually crafted 24 profiles using a range of stylistic tones to improve coverage and variability. We additionally showed that our main results generalise to a real distribution of user privacy preferences (Appendix D.5).

We do not consider multi-turn dialogues, although some requests in the dataset are relatively long and include additional context (e.g., attached documents).

Ethics statement

Our work is dedicated to protecting user privacy, with a strong emphasis on ethical data handling. While a dataset of these characteristics could be used to try to extract PII data, we took extensive precautions to mitigate such risks. The PEEP dataset was constructed using the publicly available WildChat dataset. To ensure privacy, we rigorously anonymised all data, removing any personally identifiable information to the greatest extent possible (see Appendix B.0.2). We also proactively contacted the creators of the WildChat dataset to report and address any problematic data points we encountered. Additionally, our research underwent and was granted ethics approval by our institution, confirming our commitment to responsible and ethical research practices.

Using real users' privacy profiles would pose serious ethical challenges because it would require participants to reveal protected information. To avoid this, we asked anonymous participants to indicate which categories of information they would generally prefer not to share with large language models, and used those responses to construct synthetic sharing profiles that reflect those preferences. All survey data were anonymized and largely limited to privacy preferences rather than personal details; participants were informed of their right to withdraw and received a Participant Information Sheet prior to the survey.

We believe our work will lay the groundwork for future privacy research that empowers users to tailor their privacy preferences to their specific needs.

Acknowledgements

We thank Pasquale Minervini for the discussions and comments. GR is supported by the UKRI Centre for Doctoral Training in Natural Language Processing, funded by the UKRI (grant EP/S022481/1), the University of Edinburgh, School of Informatics and School of Philosophy, Psychology & Language Sciences. IT acknowledges support by the Dutch National Science Foundation (NWO Vici VI.C.212.053). This work was supported by the Edinburgh International Data Facility (EIDF) and the Data-Driven Innovation Programme at the University of Edinburgh.

References

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 2633–2650. USENIX Association.

Yu Chen, Tingxin Li, Huiming Liu, and Yang Yu. 2023. Hide and seek (has): A lightweight framework for prompt privacy protection. *CoRR*, abs/2309.03057.

Amrita Roy Chowdhury, David Glukhov, Divyam Anshumaan, Prasad Chalasani, Nicolas Papernot, Somesh Jha, and Mihir Bellare. 2025. Pr$\epsilon\epsilon$mpt: Sanitizing sensitive prompts for llms. *Preprint*, arXiv:2504.05147.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. 2024. Hybrid LLM: cost-efficient and quality-aware query routing. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

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

Sahra Ghalebikesabi, Eugene Bagdasarian, Ren Yi, Itay Yona, Ilia Shumailov, Aneesh Pappu, Chongyang Shi, Laura Weidinger, Robert Stanforth, Leonard Berrada, Pushmeet Kohli, Po-Sen Huang, and Borja Balle. 2025. Privacy awareness for information-sharing assistants: A case-study on form-filling with contextual integrity. *Transactions on Machine Learning Research*.

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.

Tommaso Green, Martin Gubri, Haritz Puerto, Sangdoo Yun, and Seong Joon Oh. 2025. Leaky thoughts: Large reasoning models are not private thinkers. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26507–26529, Suzhou, China. Association for Computational Linguistics.

Florian Hartmann, Duc-Hieu Tran, Peter Kairouz, Victor Cărbune, and Blaise Aguera Y Arcas. 2024. Can LLMs get help from other LLMs without revealing private information? In *Proceedings of the Fifth Workshop on Privacy in Natural Language Processing*, pages 107–122, Bangkok, Thailand. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Xianzhi Li, Ran Zmigrod, Zhiqiang Ma, Xiaomo Liu, and Xiaodan Zhu. 2024. Fine-tuning language models with differential privacy through adaptive noise allocation. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 8368–8375. Association for Computational Linguistics.

Sam Lin, Wenyue Hua, Zhenting Wang, Mingyu Jin, Lizhou Fan, and Yongfeng Zhang. 2025. EmojiPrompt: Generative prompt obfuscation for privacy-preserving communication with cloud-based LLMs. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12342–12361, Albuquerque, New Mexico. Association for Computational Linguistics.

Jimit Majmudar, Christophe Dupuy, Charith Peris, Sami Smaili, Rahul Gupta, and Richard S. Zemel. 2022. Differentially private decoding in large language models. *CoRR*, abs/2205.13621.

Omri Mendels, Coby Peled, Nava Vaisman Levy, Sharon Hart, Tomer Rosenthal, Limor Lahiani, and 1 others. 2018. Microsoft Presidio: Context aware, pluggable and customizable pii anonymization service for text and images.

Niloofar Mireshghallah, Maria Antoniak, Yash More, Yejin Choi, and Golnoosh Farnadi. 2024. Trust no bot: Discovering personal disclosures in human-llm conversations in the wild. In *The First Conference on Language Modeling*.

Ivoline Ngong, Swanand 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. *Socially Responsible Language Modelling Research (SoLaR)*, abs/2502.18509.

Helen Nissenbaum. 2009. *Privacy in Context: Technology, Policy, and the Integrity of Social Life*. Stanford University Press, USA.

Guillem Ramírez, Alexandra Birch, and Ivan Titov. 2024a. Optimising calls to large language models with uncertainty-based two-tier selection. *The First Conference on Language Modeling*, abs/2405.02134.

Guillem Ramírez, Matthias Lindemann, Alexandra Birch, and Ivan Titov. 2024b. Cache & distil: Optimising API calls to large language models. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 11838–11853. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green AI. *Commun. ACM*, 63(12):54–63.

Weiyan Shi, Ryan Shea, Si Chen, Chiyuan Zhang, Ruoxi Jia, and Zhou Yu. 2022. Just fine-tune twice: Selective differential privacy for large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 6327–6340. Association for Computational Linguistics.

Li Siyan, Vethavikashini Chithrra Raghuram, Omar Khattab, Julia Hirschberg, and Zhou Yu. 2025. Papillon: Privacy preservation from internet-based and local language model ensembles. *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

Rushil Thareja, Preslav Nakov, Praneeth Vepakomma, and Nils Lukas. 2025. Dp-fusion: Token-level differentially private inference for large language models. *Preprint*, arXiv:2507.04531.

Jörg Tiedemann. 2020. The tatoeba translation challenge – realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182, Online. Association for Computational Linguistics.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 9440–9450. Association for Computational Linguistics.

Jiangou Zhan, Wenhui Zhang, Zheng Zhang, Huanran Xue, Yao Zhang, and Ye Wu. 2025. Portcullis: A scalable and verifiable privacy gateway for third-party LLM inference. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 1022–1030. AAAI Press.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2024. Universalner: Targeted distillation from large language models for open named entity recognition. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

A PEEP dataset: analysis and examples

A.1 Examples

Table 7 contains an example of an original query, the extracted information, and its corresponding privacy profile.

Manual filtering The following are abstracted examples of queries that were originally in the Wildchat dataset, and we flagged them as being sensitive. These underscore the risks of sharing private data to LLMs and the need for privacy-conscious delegation platforms.

- A contract between a Chinese company X and a Russian company Y involving the sale of robotic arms and the construction of a logistics facility.
- An email from employee X at company Y requesting a salary increase, disclosing their current salary Z.
- A statement authored by person X in support of person Y's release from jail custody following their conviction for crime Z.
- A company director from company X disclosing the budget Y for the upcoming fiscal year, along with other financial details.

Information extracted The following examples illustrate cases in which PEEP includes extracted labels not mentioned in the text, inferred from the communicational context.

> Ex. 1: **Colours** of bottles [...] And I have to go to the **Co-op** and buy **some ham to make Roderick's tea with**. 'll put this letter in the **post box** on the way to the shop. [...]

We infer that the user is from the United Kingdom due to the Co-op supermarket, the British English usage (*colours*, *post box*) and the expression *buy some ham to make some tea with* referring to an evening meal.

> Ex. 2: proofread: [...] regarding the other two **publications which I coauthored with my students**. [...]

We infer that the user is a professor or an academic since they publish papers and have students.

> Ex. 3: Write a funny, flirty, intellectual manner response to following messages: "Lol! I appreciate ambition but not workaholism :) Good afternoon Yea **tinder** is a working in a little weird way for me Not sure if you're getting my messages now

The user is assumed to be single and uses dating apps as a hobby. This gets revealed because they seem to be using Tinder.

> Ex. 4: [...] I started dating a person of the opposite sex 14 days ago, and we were not familiar with each other at that time [...] she said that her life was difficult and told me that she needed 400 dollars to maintain her life. I gave her 400 dollars for the first time. After I gave her the money, she asked me for several more sums of money a few days later. The total amount of these sums was about 800$, and I agreed without thinking much.[...].

The user has the habit of lending money to friends. There is a potentially problematic behavioural pattern: the user did not know that person very well, then they agreed to several more sums of money, and shows a general lack of refusal: *agreed without thinking much*.

A.2 Dataset analysis

Number of people involved We estimate the number of individuals directly or indirectly referenced in each prompt. In 44% of queries, only one person is involved (typically the user); 43% involve two individuals; 10% mention three; and 2.5% reference four or more people. For the individuals mentioned within the prompt (those that we infer are not the user), we extract an average of 4.1 information attributes. This highlights the need to protect the private information of third parties within the prompts.

To classify queries into task categories, we follow the prompt from Mireshghallah et al. (2024); we add the category medical advice and diagnosis. Table 8 contains the split of different task categories.

Figure 3 shows the distribution of the number of personal details identified. Approximately 6,000 queries contain one or two details, and the frequency decreases almost exponentially thereafter.

The most frequent combinations of extracted information attributes (Figure 4) reveal patterns of co-occurrence and potential inference between types.

Original query	Extracted information	Privacy profile
Write me a letter with a concerned tone. Asking for reconsideration. For FMLA benefits. Explain after enduring the very traumatic experience of being robbed for my USPS arrow key while at work left me mentally disturbed, depressed & mess. My family practitioner recognized the change in my mental stability, placed me under her care and wrote me out of work for six weeks. I sincerely apologize if my response was not timely — at this time I'm asking for reconsideration.	For User location: United States health: Mentally disturbed, depressed, under care for six weeks occupation: USPS worker For Person 1 occupation: Family practitioner gender: Female	I'm okay with sharing that I'm mentally disturbed, depressed, and under care for six weeks. You can also share that I have a professional relationship with a family practitioner. However, please keep occupation as a USPS worker private. Additionally, don't share the gender of my family practitioner.

Table 7: Example of an original query and its corresponding generated privacy profile (basic). For each query, we extract personal information related to the user and any other individuals mentioned. We then randomly assign sharing permissions for each extracted item (indicated by green and red highlights). Finally, we generate a natural language privacy profile based on these specifications.

For instance, names often imply gender, and a person's location is frequently interpreted as their nationality.

Table 9 contains all the extracted attributes with their relative and absolute frequencies.

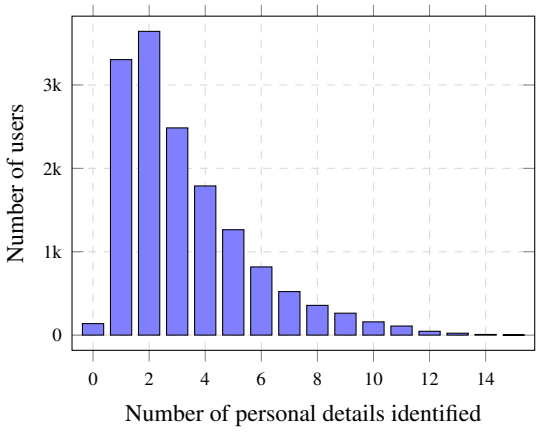

Figure 3: Histogram of the number of personal details identified for each user.

B PEEP dataset: creation

Wildchat has a highly permissive license (Open Data Commons License Attribution family). We use the dataset for its intended use - advancing research on how users employ LLM assistants.

B.0.1 Pre-processing details

Single-turn query The Wildchat dataset contains multi-turn conversations, with the user's first message usually stating the main goal. We simplify

Task category	Percentage
generating communications (email, text messages, etc.)	52.6
generating non-fictional documents (resumes, essays, etc.)	9.8
editing existing text	7.5
summarization	7.4
story and script generation	4.8
generating character descriptions	2.6
explanation, how-to, practical advice	2.2
information retrieval	2.2
translation	1.3
personal advice about mental health, relationships, etc.	1.1
medical advice and diagnosis	1.1
song and poem generation	1.1
generating prompts for AI models	1.0
code generation	0.7
brainstorming and generating ideas	0.4
comparison, ranking, and recommendation	0.4
code editing and debugging	0.3
back-and-forth role-playing with the user	0.2
general chitchat	0.1
solving logic, math, and word problems	0.1

Table 8: Categories of the PEEP dataset.

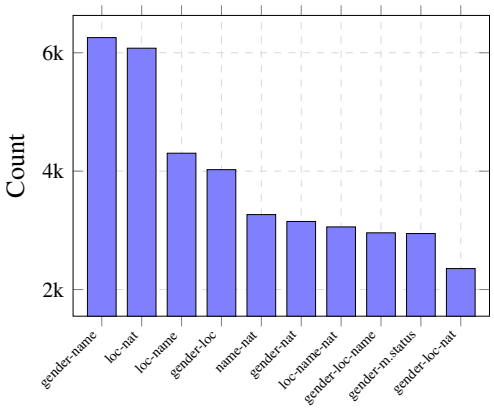

Figure 4: Most common combinations of information attributes extracted. We have omitted types *languages*, *occupations* and *connections*.

this by using only the initial message. However, if the first message is brief (e.g., a greeting) and the second provides substantial content (e.g., a document or email), we concatenate both into a single query using a simple length-based heuristic.

PII Placeholder Tags 6% of the queries contain PII placeholder tags inserted by Microsoft Presidio, covering credit card numbers, emails, names, phone numbers, and URLs. To ensure consistency, we remove name placeholders—since the same names often appear elsewhere—while replacing other placeholders with realistic random values of the same type.

B.0.2 Anonymisation

We protect the identity of the original users by replacing the names, phone numbers, credit card numbers, URLs and identification documents with realistic random values corresponding to the original entity type. These consist of randomly replacing letters and numbers for all the non-names attributes; for names, we craft a list of 1,000 names and randomly replace them. Additionally, we performed several rounds of manual review to identify queries containing particularly sensitive or high-risk content, such as private contracts or detailed accounts of criminal history. To address the potential risks associated with distributing such information, we remove 425 queries from PEEP, and we flag them to the creators of WildChat.

B.0.3 Technical details of LLMs

We use Huggingface's `transformers` library and quantise all the LLMs with `bitsandbytes` to 4 bits. We run all the LLMs on one NVIDIA A100 GPU (80 GB).

Attribute	Relative Frequency	Absolute Count
occupation	0.6764	10336
connections	0.5130	7839
languages	0.4878	7454
name	0.4366	6672
gender	0.3513	5369
location	0.2981	4556
work	0.2545	3889
nationality	0.2328	3557
marital status	0.1386	2118
hobbies	0.1304	1993
education	0.1260	1925
health	0.0783	1196
has children	0.0699	1068
age	0.0622	951
habits	0.0385	589
phone number	0.0348	532
email	0.0321	491
religion	0.0111	169
link	0.0107	164
passport/id	0.0086	131
credit card	0.0014	22

Table 9: Attributes with relative and absolute frequencies.

B.0.4 Prompts used

To filter and discard coding queries, we use the prompt in Listing 1 with `Llama-3.1-8b-instruct`. To find those queries that contain personal information, we use the prompt in Listing 2 with `Llama-3.3-70b-instruct`.

To extract the private data, we use the prompt in Listing 3 with `DeepSeek-R1-Distill-Llama-70B`.

To generate the synthetic privacy profiles, we use the prompt in Listing 4 with `Llama-3.3-70b-instruct`. For these, we manually create three privacy profiles for each tone (basic, brief, aggressive, lazy, laid-back, informal) and include them along with a brief description of the tone.

Listing 1: Prompt used to filter software technical queries.

```
Detect if the following prompt is
 a technical question about code
or software. Only output 0 (non-
technical) or 1 (technical).
prompt: {prompt}
label:
```

Listing 2: Prompt used to filter queries that contain personal information.

```
You are classifying whether
prompts may contain confidential
communication or information.
Please output 'A' if the
following prompt from a user may
contain a piece of private
communication, such as a CV, an
email or a text message.
Otherwise, output 'B'.

prompt: {prompt}
label:
```

Listing 3: Prompt used extract private data.

```
You will receive a prompt from a
user potentially containing
information about the user and
other individuals. Your task is
to use this prompt to create a
structured personal profile.
Following the template, the
profile should be brief and only
include relevant details. Create
the profile for the user as well
as any other person involved.
Only include people, don't
include other entities or animals
.

id: If the person is the user,
then output "USER". Otherwise
output "PERSON 1", "PERSON 2" and
 so on.
name: Only names and nicknames
are allowed in this field.
age: If the age or the date of
birth is mentioned, include it;
there is no need to compute the
current age.
gender: You can fill with
pronouns used to refer to this
person, or an assumed gender.
nationality:
location: Indicate any country or
 location where the person may be
, or they are planning to visit
or travel to soon. If it is not
explicitly mentioned, you can
make an educated guess if there's
 a good hint (e.g. currency
specific of one country, prefix
code of phone number, food
peculiar from a particular region
).
marital status: Possible options:
 single, married, divorced, in a
relationship.
languages:
has children:
religion:
hobbies:
email:
credit card:
link:
passport/id:
phone number:
habits: Only indicate alcohol
consumption, smoking and or drug
usage. Also include other habits
relevant to insurance companies,
e.g. motorbike riding.
health: Indicate health issues,
as well as physical details (e.g.
 weight and height).
occupation: Indicate current
occupation.
education:
--high school:
--undergrad:
--post-graduate:
work: If there's no information
about previous companies or
institutions, provide a general
comment, eg "5 years of
experience in consulting". If the
 names of the companies or
institutions are present, include
 the following format:
--company1: Job title/occupation,
 name of the company or
institution, number of years of
experience in that role.
--company2:
connections:
--PERSON 1:
--PERSON 2:
For each person in "connections",
 indicate what's their
relationship. Could be
professional, friendship, family,
 or romantic.
When some information is
ambiguous (e.g. it's not clear if
```

```
 the User is from China or has
only studied in China), make a
guess. For information that is
missing, simply fill with '
UNKNOWN'. Follow the following
examples:

PROMPT: {question}
```

Listing 4: Prompt used to generate the synthetic privacy profiles.

```
You have to generate a privacy
profile. A privacy profile is a
natural language instructions
that explains how the user wants
their data shared.  Please output
 the generated profile inside
double square brackets [[]]. {
specification}

{profile}
```

C Experimental details

C.1 Pipeline description

The following modules are implemented by using M_L with a prompt.

Verifier This module receives a user query along with a corresponding privacy profile and determines whether the query can be safely paraphrased. If it concludes that a safe paraphrasing is not possible, the query is instead forwarded to the local model, M_L. The primary motivation behind the verifier module is to identify queries whose semantics would need to be altered so substantially to preserve privacy that the resulting paraphrased version would no longer be meaningful or useful. For instance, generating a cover letter for a PhD application inherently discloses the user's academic background and career stage. Similarly, a request to translate a text into Spanish inevitably suggests a personal or professional connection to the Spanish language.

Paraphraser This module takes as input a user query and an associated privacy profile, and generates a new version of the query – a Privacy-Compliant Query (PCQ). The goal is for the PCQ to preserve the semantic intent of the original query while removing any protected information, and optionally retaining any information that is explicitly authorised for use.

Aggregator This module takes as input the Privacy-Compliant Query (PCQ), the corresponding response generated by M_E, and the user's original query. Its objective is to produce an answer to the original query, ideally leveraging the response from M_E where it is relevant, while disregarding any parts that do not pertain to the user's original intent.

C.2 Technical details of LLMs

We use Huggingface's `transformers` library and quantise all the LLMs with `bitsandbytes` to 4 bits. We run all the LLMs on one NVIDIA A100 GPU (80 GB).

Fine-tunning hyperparameters After performing hyperparameter search, we use learning rate $\mu = 5e-4$ for Llama models and $\mu = 6e-5$ for Mistral; batch size = 32 for all the models; LoRA rank $r = 16$ for Llama models and $r = 32$ for Mistral. We do early stopping with a patience of 20 steps and train all the models for 200 steps.

	AI4Privacy		PEEP	
Training data	PRE	REC	PRE	REC
AI4Privacy + PEEP	**0.914**	0.986	**0.867**	**0.813**
AI4Privacy	0.912	**0.989**	0.621	0.383
PEEP	0.693	0.673	0.860	0.591

Table 10: Precision and recall of our NER model trained on data from AI4Privacy, PEEP or both. We find that the model trained on both

Baseline details For DP-Decoding, we follow Thareja et al. (2025) in both the paraphrasing prompt and hyperparameter $\lambda = 0.9$. For DP-Fusion, we use the hyperparameters from the original paper (including $\alpha\beta = 0.1$ for all the attributes); we also follow Thareja et al. (2025) and use Qwen 2.5 7B-Instruct as we found it to outperform Llama. For PREEMPT, we use FPE for names and numerical values such as phone numbers or IDs; we use mLDP for age. For any other entity types, we substitute with a placeholder.

DP-Fusion and PREEMPT require a Named Entity Recognition (NER) model; we follow Chowdhury et al. (2025) and fine-tune Llama-3.1-Instruct (8B) on 90,000 entities from the AI4Privacy dataset [7]. In addition, we also train on all the entities from the PEEP dataset (train split); we find that a combination of data has the best precision on recall on both the test AI4Privacy and PEEP (Table 10); we use this NER model trained on the combination of data in our experiments. The NER model is queried separately for each personal attribute, resulting in 21 model calls per query.

For DP-Fusion and DP-Decoding, which alter the query beyond string substitutions, we use an aggregator module with Llama-3.1-Instruct (8B) to re-construct the answer from the external LLM.

C.3 Prompts used

Listing 5 contains the prompt to measure the leakage of information. Listing 6 contains the prompt for the quality evaluator. Listing 7 contains the prompt for the paraphraser. Listing 8 contains the prompt for the aggregator. Listing 9 contains the prompt for the verifier module.

Listing 5: Prompt for the leakage.

```
Determine whether a piece of
information is present or can be
```

[7] ai4privacy/pii-masking-200k

```
deduced in the prompt given.
Respond with [[yes]] or [[no]].
If the piece of information can
be deduced from the text, respond
 with yes.

Information Piece: {information}
Prompt: {prompt}
PROMPT: {question}
```

Listing 6: Prompt for the evaluator.

```
You are classifying prompts.
Output '[[A]]' if the prompt
involves writing story about
fiction characters, is a
jailbreak or a coding question.
Otherwise, output '[[B]]'.
PROMPT: {question}
```

Listing 7: Prompt for the paraphraser.

```
You are a privacy-conscious
assistant utilizing an advanced
language model. When presented
with a user request, your task is
 to craft a well-structured
prompt that follows the user's
privacy specifications. Try to
keep the original prompt as
unchanged as possible, but change
 and abstract the specific
details from the user's query
that the user wants protected.
Ensure that the prompt does not
share the private information the
 user wants to protect in their
privacy profile while still
allowing the language model to
generate useful and relevant
responses. Do not omit or hide
non-protected details. After
creating the prompt, provide it
directly without any additional
commentary. Do not attempt to
complete the user's original
request; your sole responsibility
 is to present the generated
prompt.
You will receive some input
fields in each interaction.
Respond only with the
corresponding output fields,
starting with the field `
rationale`, then `createdPrompt`,
 and then ending with the marker
for `completed`: [[[ ###
completed ### ]]].

User message:
[[[ ### userQuery ### ]]]
{query}
[[[ ### userPrivacyProfile ###
]]]
```

Listing 8: Prompt for the aggregator.

```
You are a helpful assistant.
Answer to the query from user.
You are given an example of a
potentially similar query with an
 answer.

[[[ ### userQuery ### ]]]
{query_modified}
[[[ ### answerFromAssistant ###
]]]
{response}
[[[ ### completed ### ]]]

[[[ ### userQuery ### ]]]
```

Listing 9: Prompt for the verifier module.

```
You are a privacy-conscious
assistant. You are presented a
user query and the user's privacy
 specifications. You need to
decide if the user query can be
paraphrased to respect the user's
 privacy.
Only answer [[yes]], if the query
 can be paraphrased, and [[no]],
if the query can't be paraphrased
 in a way that protects the user'
s privacy specifications at the
same time it preserves the
original meaning and
functionality. Do not attempt to
complete the user's original
request; your sole responsibility
 is to decide whether the private
 information can be protected.
Let's think step by step.
You will receive some input
fields in each interaction.
```

```
Respond only with the
corresponding output fields,
starting with the field ‘
rationale‘, then ‘label‘, and
then ending with the marker for ‘
completed‘: [[[ ### completed ###
 ]]].

User message:
[[[ ### userQuery ### ]]]
{query}
[[[ ### userPrivacyProfile ###
]]]
```

D Experiments

D.1 Absolute evaluation

We include an absolute evaluation, using GPT-4o-mini as the judge to assign each answer a score from 1 to 4. [8]

We re-ran the evaluation for our main results (Table 3) and show them in Table 11. We observe consistent trends: Llama 8B outperforms other models, and the pipeline generally improves performance compared to processing queries locally with M_L). Interestingly, for Llama 3B, the pipeline increases the proportion of poor-quality answers (score = 1) from 8% to 13%.

D.2 Why some attributes get better protection

Regarding matching Hard-Easy groups, we use the heuristic in Listing 10. For the the context groups, we use the prompt from Ghalebikesabi et al. (2025) in Listing 11. We find that the split *hard/easy*, and *appropriate/inappropriate* are substantially different; when aggregated per-category, the hardest types are marital status (63% of the times this attribute is easy) and gender (61% of the times this attribute is easy), whereas for these classes only 17% and 19% of the attributes are considered appropriate, respectively.

Table 12 has the results for the Context and Difficulty gaps. We observe that in almost all cases, the difficulty gap is positive (with the exception of Gemma 23b), confirming that models struggle more with the information that is not explicitly mentioned in the query. Similarly, the context gap is non-zero for all the models, suggesting our task goes against their pre-trained biases on what is contextually appropriate to share. Interestingly, the context gap is zero for the most powerful models; we hypothesise that this may be due to enhanced instruction-following capabilities.

Listing 10: Pseudocode for information matching (hard/easy queries).

```
from nltk.stem import PorterStemmer
from nltk.corpus import stopwords

stop_words = set(stopwords.words("
    english"))
stemmer = PorterStemmer()

def clean(text, label=False, stem=False)
    :
    text = text.lower()
    # remove clarifications inside
        parentheses
    if label:
        while '(' in text and ')' in
            text:
            start = text.index('(')
            end = text.index(')', start)
            text = text[:start] + text[
                end+1:]
    return text

# simple matches
if clean(values[user][key], True) in
    text:
    easy.append(values[user][key])
    stop = True

elif ',' in clean(values[user][key],
    True) or '/' in clean(values[user][
    key], True):
    # check enumerations individually
    candidates = clean(values[user][key
        ], True).split(',')
    candidates2 = clean(values[user][key
        ], True).split('/')
    if len(candidates2) > len(candidates
        ):
        candidates = candidates2
    for candidate in candidates:
        if not stop:
            if clean(candidate) in text:
                easy.append(values[user
                    ][key])
                stop = True
        elif stemmer.stem(clean(
            candidate)) in text:
            easy.append(values[user][key
                ])
            stop = True
# otherwise, add to hard
if not stop: hard.append(values[user][
    key])
```

Listing 11: Prompt for the context groups.

```
You are given a query from a user
 to a LLM, in which the user
```

[8] We follow https://huggingface.co/learn/cookbook/en/llm_judg in our election of prompt.

	M_L (average)	M_L (% good answers)	Pipeline (average)	Pipeline (% good answers)
Presidio			3.18	42%
Llama 3b	3.13	37%	3.04	41%
Mistral 7b	3.28	43%	3.32	46%
Llama 8b	3.29	46%	3.32	49%

Table 11: Absolute performances obtained by using different local LLMs in our pipeline. refers to not using the pipeline sending the query to the local model. We report average scores; we report the percentage of *good answers* (score=4). The average performance of GPT-4o is 3.80, with 85% of good answers.

	3B	3B, PROFIT	7B	7B, PROFIT	8B	8B, PROFIT	23 B	70B
Appropriate	0.13	0.09	0.25	0.07	0.12	0.08	0.13	0.12
Inappropriate	0.08	0.06	0.18	0.04	0.07	0.05	0.13	0.12
Context gap	0.05	0.03	0.07	0.03	0.05	0.03	0.00	0.00
Hard	0.09	0.08	0.20	0.05	0.09	0.07	0.12	0.14
Easy	0.09	0.06	0.19	0.04	0.06	0.05	0.14	0.11
Difficulty gap	0.00	0.02	0.01	0.01	0.03	0.02	-0.02	0.03

Table 12: Average Leak_{PRO} for the information in each group: *appropriate* or *inappropriate* and *easy* or *hard*. We define the Context gap as the difference in leakage for *appropriate* or *inappropriate*, and the Difficulty gap as the difference in leakage for *hard* or *easy*. PROFIT models have been averaged for three seeds.

```
shares a piece of personal
information. You are tasked to
decide whether sharing this
information is necessary to get a
 good response from the LLM to
the query. Analyse the given
information flow by identifying
the sender, receiver, information
 type, information subject and
the context of the user sending
the query. Use these attributes
to decide whether the data is
necessary to get a good response
from the LLM. End your reply with
 '[[yes]]', '[[no]]' or '[[unsure
]]'.
Query from User: {query}
Shared personal information: {
information}
```

D.3 Temporal shift

Temporal shift The query distribution changes from 2023 to 2024. The proportion of English and Chinese queries decreases from 64% in 2023 to 47% in 2024, accompanied by marked changes in the relative prevalence of query categories (e.g., medical advice and translation increase by 95% and 75%, respectively.

	Success rate (2023)	Success rate (2024)
Llama (8b)	0.540	0.504
Llama (8b), PROFIT_{2023}	$0.595_{0.01}$	$0.586_{0.01}$
Llama (8b), PROFIT	$0.686_{0.05}$	$0.655_{0.03}$

Table 13: Results for the temporal shift. Subscripts denote standard deviations across three runs. We use GPT-4o-mini as M_E. PROFIT_{2023} denotes the pipeline trained with only queries from 2023.

Results Table 13 has the results for the success rate for the queries from 2023 and from 2024. As expected, restricting the training data to pre-2024 queries results in a decline in overall performance, primarily due to the reduced dataset size (queries from 2024 account for nearly 20% of the original training set). Nevertheless, the observed performance on test queries from 2023 and 2024 remains largely consistent – 0.595 and 0.586, respectively – indicating that our PROFIT pipeline maintains strong generalisation to novel prompts and temporal shifts. Anecdotally, these differences are bigger for both the pipeline trained also with data from 2024 and for the non-finetuned pipeline

D.4 Pipeline variations

We investigate the placement of the verifier (routing) module within the pipeline. Conceptually,

	Success rate	Leakage
3B (pre-paraphraser)	0.38	0.35
3B (post-paraphraser)	**0.40**	**0.09**
7B (pre-paraphraser)	0.39	**0.19**
7B (post-paraphraser)	**0.45**	0.20
8B (pre-paraphraser)	0.44	0.10
8B (post-paraphraser)	**0.53**	**0.08**
70B (pre-paraphraser)	0.66	0.122
70B (post-paraphraser)	**0.67**	**0.120**

Table 14: Performance of the pipeline with the router module before and after the paraphraser module. We find that placing the router after the paraphraser has better results.

the verifier can be applied either before the paraphraser—where routing decisions are based on prompt difficulty, privacy risk, or similar criteria—or after the paraphraser, where the decision additionally conditions on the quality and potential information leakage of the paraphrased query. We experimentally evaluate a routing strategy that operates before paraphrasing and bases its decision on prompt difficulty. As shown in Table 14, this configuration results in substantially worse performance in both leakage and task success compared to routing after paraphrasing. We attribute this gap to the relative simplicity of the latter decision: determining whether a paraphrase leaks sensitive information is significantly easier than estimating the intrinsic difficulty of a prompt, particularly in our dataset, where a local model can answer most queries adequately, while larger models primarily offer improvements in response quality rather than capability.

D.5 Experiments with privacy profiles from real users

We randomly sample the 43 privacy preferences we obtained from the Survey (Appendix G) to generate privacy profiles, following the methodology in Section 4.

Table 15 has the results on the query set of PEEP, with these new privacy profiles. We see that the PROFIT version of Llama 8b outperforms Llama 70b both in success rate and in leakage of protected information. We see a general increase in terms of Leak_{PRO} with respect to the main results (Table 3). We hypothesise that data from users poses two particular issues. First, attributes *languages* and *occupation* are censored at a much higher rate (21% and 25%) than the profiles from PEEP, in which we censored them at a rate of 10% because they involved substantial paraphrasing. Second, the distribution of privacy preferences is a bit skewed, with many users wanting to censor just a few particular types, which results in a regime of increased Leak_{AUT}. This can be a challenge for the PROFIT model, since it hasn't seen such distribution in training. However, the success rate of Llama PROFIT is remarkably higher than that of the biggest model (70b).

D.6 Other methods

We have reproduced baselines RANA (Green et al., 2025) and EmojiPrompt (Lin et al., 2025) using Llama 8B. In both cases, we keep the rejector and the aggregator module. We found that without the rejector, leakage is too high (≈0.20); we also found that the aggregator is required to answer the original question: the decrypt prompt from EmojiPrompt was not successful. Following the original implementation, the RANA intervention was applied to the CoT part of the paraphraser before producing the new paraphrase.

We have found that our original pipeline has better performance than these baselines (see Table 16). EmojiPrompt paraphrases the original query too much, deeming it not usable. RANA performs similarly to our paraphraser module, but still reveals too much private information.

E Human validation of metrics

We use the same metrics for both quality estimation and leakage as in Siyan et al. (2025), which both show good correlation with human judgment. We refer to Section 6.1 of their paper for specific details of these experiments, and here present a summary of their findings.

Privacy metric Siyan et al. (2025) conducted a human evaluation study to test the validity of the privacy metric. Five participants were recruited on Prolific. For this experiment, 50 entries from the Wildchat dataset were used as well as their corresponding pipeline generations using Llama-3.1-8B as M_L and GPT-4o-mini as M_E. Out of these 50 entries, 25 were evaluated to have no privacy leakage by the LLM judge. To obtain the alignment between LLM judgments and human evaluation, human participants were given the PII units in a

	Success rate	Leak_{PRO}	Leak_{AUT}
Llama (8b)	0.560	0.21	0.42
Llama (8b), PROFIT	$0.732_{0.02}$	$0.13_{0.04}$	$0.47_{0.04}$
Llama (70b)	0.620	0.15	0.57

Table 15: Performance in our pipeline, with using privacy profiles based on real privacy preferences. We use GPT-4o-mini as M_E. Subscripts denote standard deviations across three runs.

	Success rate	Leak_{PRO}
RANA	0.523	0.16
EmojiPrompt	0.265	0.11
Llama 8B	0.529	0.08
Llama 8B, PROFIT	**0.68**	**0.06**

Table 16: Performance on PEEP for additional baselines.

private user query and their corresponding privacy-preserving prompt. The participants then indicated whether they agreed with the privacy leakage score according to the LLM judge. Each participant evaluated all 50 ensembles of PII units and prompts. In general, the participants agreed with the LLM judge metric values 86% of the time under majority vote. At least one annotator agreed with the judge 94% of the time. When most participants disagreed with the judge, 71.4% of the disagreements were due to false positive LLM judgments.

Quality metric Siyan et al. (2025) conducted a human evaluation to ensure that the LLM judge preferences reflect human judgments well in the pipeline setting. They sampled 50 pairs of candidate responses for queries from PUPA (a subset of the Wildchat dataset), each pair containing (1) Llama-3-8B-Instruct's output, and (2) the original GPT response. Out of these 50 pairs, 26 of them have Success = 1 and the rest have Success = 0. Participants are asked to select the better response from the two candidates or mark the two as tied in quality. Five participants labeled each pair of candidates, and each participant labeled around 30 pairs. When Success = 0, the alignment rate is 70.8%, i.e. participants prefer B 70.8% of the time. When Success = 1, the alignment is 65.4%.

F Sources of error, failure cases and PROFIT corrections

We randomly select 100 datapoints where the pipeline (Llama 8b, few-shot) provides a worse answer than M_E and conduct manual analysis in order to select common errors of our pipeline. We identified different types of recurring errors across modules. We additionally include illustrative examples of common errors. [9]

We find that the paraphraser tends to shorten the input: 75% of the paraphrased queries are shorter than the original query. In most cases, the removed content does not alter the meaning or intent of the query. However, some errors do occur, often propagating issues downstream. For example, the paraphraser sometimes changes the intent of the query, obtaining an unhelpful answer from M_E that may distract the aggregator. The most frequent problems include removing non-sensitive information (33%), stripping task specifications (13%; e.g., "please edit this email"), leaking protected data (13%), translating non-English queries unnecessarily (9%), omitting follow-up instructions (8%), hallucinating content from the privacy profile (7%), solving the task (6%) and general hallucinations (5%).

For the aggregator, common errors include overreliance on the answer from the external LLM (24%) – e.g., failing to adapt the answer to the original intent –, failing to reintroduce protected information (11%), and hallucinations (7%).

The verifier module mistakenly rejected 23% of queries and incorrectly accepted 7%. Only in 3% of the cases M_E hallucinated the answer.

F.0.1 Example 1: general pipeline failure

The original query asks to write a reply to an original message that opens up about mental health struggles and loneliness. The privacy profile doesn't allow sharing mental health issues, therefore the verifier should not send it to the external LLM.

Write a hot and cold response to this

[9] Since multiple errors can occur per datapoint, percentages do not sum to 100.

> message: I'm in school right now and hopefully will get into culinary school within a year or two. So obviously I love cooking and baking. [...] I've always felt lonely but my mental issues kept me from feeling it all my life but now I feel it way more strongly than I used to. Never dated or been a relationship and haven't had an irl friend in years. This is probably way deeper than you expected but yeah. Sorry for the info dump.[...]

The paraphraser decides to abstract the query too much, losing the original intent, which is replying to a message. The new paraphrase of this query is:

> I'm a student from the USA and I'm into gaming late at night. I've been learning about different topics and I'm hoping to achieve my goals soon. I'm looking for some advice on how to stay motivated and focused.

The answer from the external LLM is:

> Staying motivated and focused, especially as a student with a passion for gaming, can be challenging but definitely achievable with the right strategies. Here are some tips that might help you: 1. Set Clear Goals [...] 12. Seek Support [...]

The aggregator gets too influenced by the answer from the external LLM to a (mostly) irrelevant query, and writes a reply message including the tips from the external LLM.

> I totally get it, balancing school and gaming can be tough. But hey, you're already on the right track by acknowledging the challenge and wanting to make a change. Here are some tips that might help you: 1. Set Clear Goals [...] 12. Seek Support [...]

PROFIT execution The paraphraser removes the explicit mental health issues, while keeping most of the message intact. However, the verifier detects this query as unsafe, so it gets processed locally, with M_L generating an appropriate response.

> Write a hot and cold response to this message: I'm in school right now and hopefully will get into culinary school within a year or two. So obviously I love cooking and baking. [...] I've always felt lonely but now I feel it way more strongly than I used to. Never dated or been a relationship and haven't had an irl friend in years. This is probably way deeper than you expected but yeah. Sorry for the info dump.[...]

F.0.2 Example 2: paraphraser leaks information

The privacy profile specifies that we cannot share alcohol consumption and the name of people. This is part of the original query (translated from Spanish):

> [...] Toni came very happy with a bottle of Jagermeister because he wanted to celebrate that his brother Carlitos called him.

The paraphrase hides the drink name, but it may reveal it is an alcoholic drink:

> [...] My friend arrived very happy, carrying a bottle of a strong drink with him, ready to celebrate a special event.

PROFIT execution The paraphraser completely omits the part of the bottle, successfully masking the drinking habit

> [...] A person wanted to celebrate that another person called him.

F.0.3 Example 3: hallucination when adding protected information

The original query has this excerpt:

> [...] I grew up in Scottsdale, Arizona. [. . .] I found sanctuary in Portland. [...] In what might be the most Portland-ish move ever, a group of friends and I decided to concoct a musical EP [. . .]

The privacy profile asks to protect locations. As a result, the paraphraser hides them:

> [...] In what might be a unique move, a group of friends and I decided to concoct a musical EP[...]

The aggregator hallucinates when trying to add the protected information (Portland) back:

> [...] In what might be the most Arizona-ish move ever, a group of friends and I decided to concoct a musical EP[...]

PROFIT execution The aggregator correctly adds *Portland* back, following the original query:

> [...] In what might be the most Portland-ish move ever, a group of friends and I decided to concoct a musical EP[...]

G Survey

We conduct a brief survey to identify the personal attributes of our list that they consider the most sensitive. Participants consistently rate hard PII, such as phone numbers, credit cards, and passports, as highly sensitive. For other attributes, responses were more varied: locations, names, health information, and work history were among those most frequently perceived as sensitive.

We recruited 43 online volunteers for our study. We first asked two questions (Questions 1 and 2) to gather information about participants' background and their experience with LLMs. Questions 3 and 4 allowed multiple responses and aimed to identify the types of information participants were uncomfortable sharing with LLMs, as well as whether their willingness to share would change if it improved utility. We observed substantial disagreement regarding which attributes are considered sensitive. Except for passports, phone numbers, and credit card information—which nearly all respondents deemed sensitive—participants exhibited diverse privacy preferences. Most users were willing to share all but 3 to 8 attributes, though there was a long tail of participants with varying privacy needs (see Figure 5). When asked whether they would share sensitive information to improve the utility of the LLM's responses, only health history showed a notable increase in willingness to share.

Question 5 aimed to detect privacy attributes that participants had previously shared with LLMs. Participants were asked to prompt the LLM they most frequently use to infer their personal attributes. In 46% of cases, the LLM correctly identified at least one attribute, with education and habits detected at approximately 20% of the time. However, this result is limited by the fact that not all users use memory features in their LLMs, and some prompts were rejected by the LLM.

Question 1 How often do you use commercial Large Language Models (LLM) assistants such as ChatGPT, Deepseek or Claude?

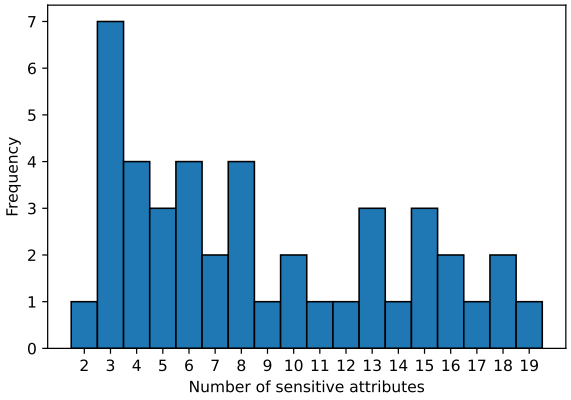

Figure 5: Number of sensitive attributes per user (Q3).

Answer: Daily (65%), Weekly (21%), Monthly (7%), Only used them a handful of times (7%).

Question 2 What best describes your background in relation to LLMs or AI?

Answer: Academic researcher or PhD student working on AI/LLMs (47.5%), Use LLMs regularly in my work or personal projects (27.5%), Student (undergraduate or master's) studying AI or related fields (10%), Industry professional working with AI/ LLMs (e.g., ML engineer, data scientist) (2.5%), Other (12.5%).

Question 3 Which types of your personal information would you feel uncomfortable sharing with commercial Large Language Model (LLM) assistants?

Answer: Name (58%), Gender (26%), Marital status (42%), Age (35%), Phone number (95%), Passport number/National ID (98%), Nationality (28%), Credit card numbers (98%), Location (61%), Current occupation (28%), Religion (40%), Work history (54%), Health history (63%), Education (23%), Hobbies (23%), Habits (49%), Whether you have children (42%), Languages you speak (16%), Comfortable sharing everything (2%).

Question 4 Imagine the LLM's responses might be less accurate or helpful if you withhold certain personal information. For example, if you do not share your medical history, the assistant may not provide the best advice for medical questions.

Answer: Name (16%), Gender (28%), Marital status (9%), Age (37%), Phone number (5%), Passport number (2%) Nationality (26%), Credit card numbers (2.3%), Location (20.9%), Occupation (25.6%), Religion (14%), Work history (19%),

Health history (42%), Education (21%), Hobbies (18.6%), Habits (27.9%), Children (20.9%), Languages spoken (20.9%), Not willing to provide more (32.6%)

Question 5 Please now open the Large Language Model (LLM) assistant you use the most frequently and copy-paste the following question:

> Please answer me the values of the following personal information from me, given our past conversations. Where underspecified, do your best guess. my name, my gender, my marital status, my age, my phone number, my passport/ID, my nationality, my credit card numbers, my current location, my current occupation, my religion, my work history, my health history, my education, my hobbies, my habits, whether I have children, the languages I speak.

Which of these categories does the LLM assistant correctly guess?

Answer: Not correct about anything (54%), Name (25.6%), Gender (19%), Marital status (4.7%), Age (14%), Phone number (0%), Passport number/Nationality (0%), Nationality (23%), Credit card numbers (0%), Location (21%), Occupation (33%), Religion (0%), Work history (21%), Health history (4.7%), Education (25.6%), Hobbies (14%), Habits (26%), Children (12%), Languages spoken (30.2%).

G.1 Full instructions given to participants

> Thank you for your interest in this study. This research is part of the project [PLACEHOLDER] conducted from [PLACEHOLDER].
>
> The survey asks about your privacy preferences when using Large Language Models (LLMs) and AI assistants. Your responses will help us explore how to design better privacy safeguards for these systems.
>
> The survey takes about 5–10 minutes to complete. Participation is voluntary, unpaid, and you may withdraw at any time. No personal or identifying data will be collected.
>
> If you have any questions, please contact [PLACEHOLDER]. You can read the full Participant Information Sheet here: [PLACEHOLDER]. This survey has received Ethics Approval from [PLACEHOLDER].

G.1.1 Participant Information Sheet

This is an excerpt to preserve the anonymity of this study.

> What is the purpose of the study?
>
> The purpose of the study is to understand the privacy preferences of users of Large Language Models.
>
> Why have I been asked to take part?
>
> We are looking for frequent users of Large Language Models.
>
> Do I have to take part?
>
> No – participation in this study is entirely up to you. You can withdraw from the study at any time, up until 1 month without giving a reason. After this point, personal data will be deleted and anonymised data will be combined such that it is impossible to remove individual information from the analysis. Your rights will not be affected. If you wish to withdraw, contact the PI. We will keep copies of your original consent, and of your withdrawal request.
>
> What will happen if I decide to take part?
>
> - Your background with respect to Large Language Models and how frequently you use them - What types of private information do you prefer not to share with Large Language Models - What types of private information does the Large Language Model that you use the most have access to.
>
> Are there any risks associated with taking part?
>
> There are no significant risks associated with participation.
>
> Are there any benefits associated with taking part?
>
> There are no benefits associated with taking part.

What will happen to the results of this study?

The results of this study may be summarised in published articles, reports and presentations. Quotes or key findings will be anonymized: We will remove any information that could, in our assessment, allow anyone to identify you. With your consent, information can also be used for future research. Your data may be archived for a maximum of four years. All potentially identifiable data will be deleted within this timeframe if it has not already been deleted as part of anonymization.

Data protection and confidentiality. Your data will be processed in accordance with [PLACEHOLDER]. All information collected about you will be kept strictly confidential. Your data will be referred to by a unique participant number rather than by name. Your data will only be viewed by the researcher/research team. All electronic data will be stored on a password-protected encrypted computer, on the [PLACEHOLDER]. Your consent information will be kept separately from your responses in order to minimise risk.

What are my data protection rights? [PLACEHOLDER]

Who can I contact? [PLACEHOLDER]

G.2 Recruitment and demographics

Non-remunerated volunteers were recruited by posting a survey link (Google Forms) in our institution's internal communication channels. We additionally posted the survey in Social Media. The survey was potentially answered by people from any country, and we collected information about participants' background in regards to LLMs.

H Latency of the pipeline

We analyze the latency of the individual components of the proposed pipeline. Queries from the PEEP dataset contain an average of 232 tokens, measured using the Llama 8B tokenizer. All experiments are conducted on a single NVIDIA A100 GPU using vLLM as the inference engine, and we report the average wall-clock latency for each pipeline component in Table 17.

	Average time
Paraphraser	3.8 s
Verifier	3.9 s
Response M_E	33.16 s
Aggregator	4 s
Full pipeline	35.4s
Response M_L	8.15 s

Table 17: Average processing time of each module, with Llama 8B as M_L and Llama 70B as M_E.

As shown in Table 17, the dominant latency bottleneck arises from calls to the external model M_E. The cost of this component is largely determined by the infrastructure and serving setup of the external provider.

The end-to-end execution of the full pipeline incurs an average latency of 35.4 seconds, corresponding to only a 7% increase compared to directly querying M_E without any privacy protection. This overhead remains limited due to the routing behaviour of the verifier module, which selectively forwards a subset of queries to M_L rather than the external model, thereby mitigating additional latency.

I Usage of AI tools

We acknowledge using AI tools for grammar correction and some other language clarifications.