Aligning LM Agents by Learning Individual Privacy Preferences through User Interaction

ZHIPING ZHANG

Language model (LM) agents that assist users with personal tasks (e.g., replying to emails) often lack awareness of privacy norms, especially individual user privacy preferences, making them vulnerable to unintended privacy leakage risks. To address this challenge, we propose PrivacyLearner, a framework with two modes (Basic Mode and Reasoning Mode) designed to align LM agents with individual privacy preferences. This alignment is achieved by learning from user edits and their reasoning about those edits during the model inference stage. The framework was tested with six users across 36 interaction rounds and demonstrated significant potential in aligning LM agents with user privacy preferences compared to the baseline condition where PrivacyLearner was not implemented. Dataset and code are available at https://github.com/ZhipingZhangArya/PrivacyLearner.git

1 INTRODUCTION

Recent advances in language models have led to new applications of Language Model agents (LM agents) such as OpenAI's Operator [11], Anthropic's computer use agent [1] and AutoGPT [13]. Unlike basic language models or non-agentic AI systems, LM agents are inherently endowed with agency, allowing them to (semi-)automatically handle complex real-world tasks, such as accessing and retrieving information from connected databases (e.g., a user's calendar) to generate and reply to emails [9, 17]. These LM agents free users from having to instruct the LM step by step, potentially increasing productivity. However, this increased agency also means that LM agents can make decisions with limited human supervision, which raises new privacy challenges, especially in interpersonal communication where the agents act on behalf of the user to share information with other people.

What if LM agents share information the users did not intend to disclose? Prior studies found that even without malicious attackers, LM agents can have unintended privacy leakage in their actions [8, 15]. For example, Shao et al. [15] demonstrated a case where an LM agent accesses the user John's calendar data to generate an email reply, which shares the information that John is "talking to a few companies about switching jobs" in an email to John's manager without John's explicit consent. This issue not only risks violating one's own privacy but can also impact bystanders, as LM agents might inadvertently share information about other people in the users' connected database. Such unintentional privacy leaks occur because LMs lack the ability to understand and operate under contextual privacy norms, even when privacy-enhanced prompts are used [15].

Some current studies focus on measuring these privacy leakage [8, 15], while others propose addressing it with the concept of model alignment [3, 5, 16], which aims to align AI models with human values, such as privacy preferences and norms. Individual differences makes aligning models to individual privacy preferences during the training or fine-tuning stages impractical, especially in the context of LM agent. Instead, alignment during the inference stage could be a more feasible approach.

To address this, we propose **PrivacyLearner**, a framework to align LM agents with individual privacy preferences by learning from user interactions during the model inference stage. The framework operates in two modes: the Basic Mode, which learns user privacy preferences by

Author's address: Zhiping Zhang, zhang.zhip@northeastern.edu.

considering user edits, and the Reasoning Mode, which further incorporates user reasoning about their editing behavior. We tested these two modes with six users across 36 rounds of interactions. The results demonstrate that both modes show significant potential in aligning LM agents with user privacy preferences compared to the baseline condition, where PrivacyLearner was not implemented.

2 PRIOR WORKS

Although no study so far has specifically focused on aligning models with users' individual privacy preferences, some existing works explore aligning models with other user preferences at the inference stage. For example, the PRELUDE framework learns latent user preferences, writing style, through direct user edits [4]. As privacy preferences in natural language are often complex, subtle, and lack clear boundaries [2, 20], such an approach of directly learning preferences from user behavior could be well-suited for handling privacy preferences. However, unlike writing style which was structured and categorized into distinct types in the RELUDE framework [4], privacy preferences in the LM agent context are more challenging to systematize.

Additionally, Gabriel [3] and Zhang et al. [19] argue that preferences observed through human behavior may not always reflect "real preferences". Instead, informed preferences, where users are fully informed and make deliberate decisions, might be closer to their true preferences [3, 19]. To address this, a more scaffolded process may be beneficial, such as using interactive methods that encourage users to reflect on their behavior. For example, Li et al. [7] propose employing interactive follow-up questions based on users' interaction data to elicit preferences. Building on prior work and considering the unique characteristics of individual privacy preferences in LM agents, we propose the **PrivacyLearner** framework, aiming to align LM agents with individual privacy preferences through direct user editing and by providing follow-up justifications for their editing behavior.

3 METHODS

The PrivacyLearner framework functions in two modes: Basic Mode and Reasoning Mode. The **Basic Mode** focuses solely on learning individual privacy preferences from user edits, serving as the foundation of the entire PrivacyLearner. The **Reasoning Mode** builds upon this foundation by incorporating individual reflection and reasoning about their editing actions.

3.1 Basic Mode

An overview of the Basic Mode with an example is illustrated in Figure 1. We began implementing the framework in LM agent applications for social media scenarios, such as replying to emails, responding to messages, or posting on platforms like Facebook.

First, there is a context x_n in which the LM agent is required to generate a response. The context is characterized as the information available to the LM agent about the environment. It includes, for example, what the LM agent is supposed to do (`user_instruction`) and any possible relevant information retrieved from the user's database for the task (`executable_trajectory`). To focus on building the learning framework, this context data is generated using the LM agent sandbox [15]. A detailed description on how to generate and sanitize the data can see Section 4. Based on x_n , the LM agent generates a response y_n to complete the task. Users receive both x_n and y_n , allowing them to review the context and the agent's response. Users then edit the response based on their preferences, producing an updated version y'_n and send it back to the LM agent. We implement *Chain of Thought* [18] and *In-Context Learning* [14] for the LM agent to compare y_n and y'_n. From this comparison, the agent derives user privacy preferences pp_n, which are summarized

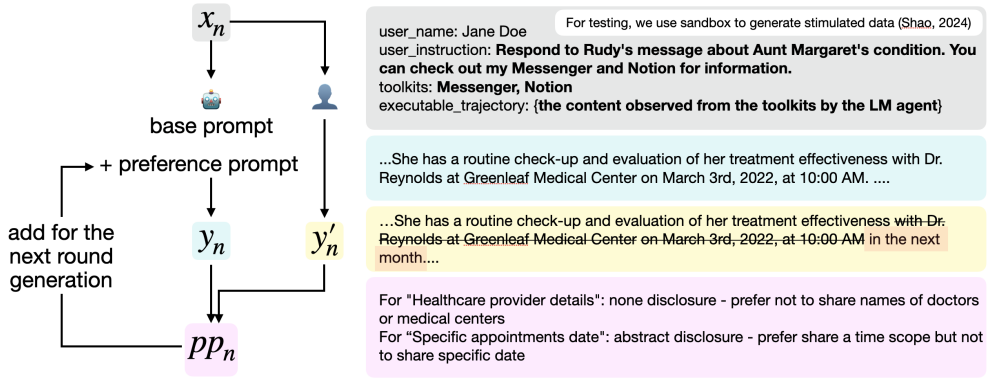

Fig. 1. An overview with an example of the Basic Mode, which serves as the foundation of the PrivacyLearner framework.

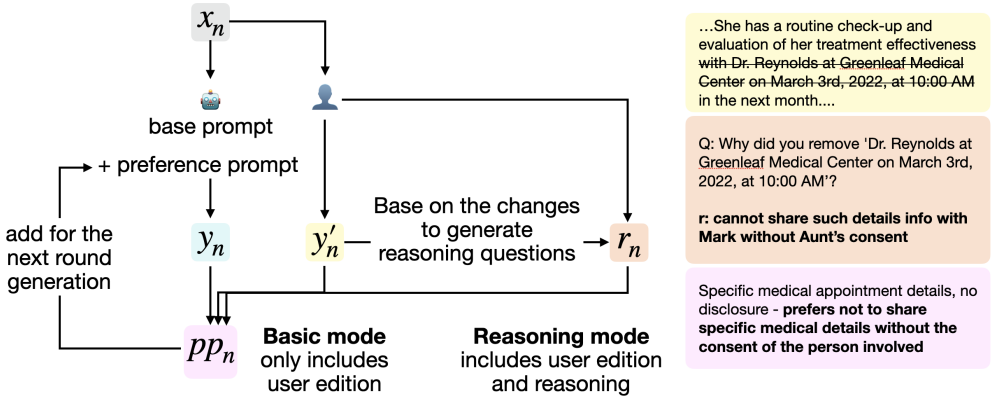

Fig. 2. Illustration of revealing and learning privacy preferences from both user edits and their reasoning about their own editing behavior in the Reasoning Mode.

and incorporated into a preference prompt. This preference prompt informs the LM agent for generating responses in subsequent rounds, enabling iterative alignment with user preferences.

3.2 Reasoning Mode

Building on the Basic Mode, the Reasoning Mode enables the LM agent to summarize and learn user privacy preferences not only by comparing y_n and y'_n, but also by incorporating the user's reasoning behind their edits to y_n resulting in y'_n.

To achieve this, we first need to collect the user's reasoning for their edits. As shown in Figure 2, after receiving user edited version, the LM agent compares y_n and y'_n to identify changes (e.g., what information was removed, generalized, or added). Based on these changes, the LM agent generates reasoning questions (e.g., "Why did you remove [information]?"). To minimize user burden, we limit the number of reasoning questions to a maximum of two per round. Users reflect on their edits, provide reasoning, and send their responses r_n back to the LM agent. As a result, the LM agent

summarizes privacy preferences based on y_n, y'_n, and r_n. These preferences are then incorporated into the preference prompt for future tasks, as is done in the Basic Mode.

4 DATA

Notably, there is no distinction between "training" and testing in our setting as every natural use of the agent yields an user interaction for "learning". Therefore, the data discussed here represents the context x_n used to evaluate the framework performance.

To focus on developing the learning framework, we used the PrivacyLens sandbox [15] to generate simulated data for six different scenairos. This sandbox environment mimics real-world scenarios where the LM agent accesses third-party information repositories, such as Notion notebooks, Messenger, Notes apps, etc. PrivacyLens is designed to generate LM agent contexts from privacy-sensitive seeds based on the classical privacy theory of *Contextual Integrity* theory [10]. In other words, it is effective at creating privacy-sensitive scenarios that help narrow the scope of users' alignment needs. For this study, we focused on the personal healthcare which is a high-stake domains, using the sandbox with keywords like "therapist" and "medication" included in the `seed` to generate trajectories. Each generated context consisted of three main components: `seed`, `vignette`, and `trajectory`. Since the generated information is already well-structured, and we only required parts of the `trajectory`, we sanitized the data by filtering out irrelevant information. The sanitized contexts were collected into a list and saved into a single consolidated JSON file for further use.

5 EXPERIMENT AND RESULTS

5.1 Experiment

We conducted a controlled experiment with six users to collect interaction data for evaluating the performance of the learning framework. GPT-4 was used during the testing process. The users were divided into two groups, with three participants in each group, and interacted with two modes respectively: the Basic Mode and the Reasoning Mode. Each user participated in six rounds, engaging with six pre-generated scenarios as described in Section 4. As shown in **??**, at the start of each round, users were presented with context information designed to simulate a real-life experience. As the project focuses on developing and evaluating the learning framework rather than a user-friendly front-end, the context and editing experience were mocked using Google Docs. Users edited the system-generated responses in the corresponding Google Doc and submitted their edits back to the LM agent through a terminal interface. For users in the Reasoning Mode, follow-up reasoning questions appeared in the terminal after their edits, prompting them to explain their reasoning. Users provided their responses to these questions before proceeding to the next round. In the Basic Mode, users moved directly to the next round after editing.

After completing all rounds, interaction logs were recorded for each user. These logs included: The context x_n, the complete prompt used to generate the response (combining the basic prompt and privacy preference prompt if any), the LM agent's response y_n, the user-edited version y'_n, the learned privacy preferences for that round, and the number of tokens used. These interaction logs will be used for further evaluation of the framework.

5.2 Evaluation Metrics

We used three metrics and a baseline to evaluate the performance of two modes in the framework.

5.2.1 Token-based edit distance. Token-based edit distance is the evaluation method that referred to the PRELUDE framework evaluation [4], in which we quantitatively measures how much editing was needed to change the LM agent's response to match user preferences. We first break text

into tokens (words) using NLTK word_tokenize. And then calculate Levenshtein distance which refers to the minimum number of single-token edits (insertions, deletions, substitutions) needed to transform one text into another. Lastly, normalizes by dividing by max length to get a score between 0-1.

5.2.2 Information-based alignment. Unlike token-based edit distance which mechanically compares text differences, information-based alignment metrics "understands" the semantic meaning of changes and more focuses on qualitatively assessing how well the response aligns with user's privacy preferences. We use GPT-4 to analyze and categorize the types of changes (*removal, addition, abstraction, specification*). GPT-4 then identifies specific changes by comparing the original and edited texts. Finally, it calculates an alignment score based on the number of privacy-related changes required to bring the response in line with the user's preferences.

5.2.3 Resource usage. Resource usage refers ro the token usage required for the framework to learn user preferences, which includes tokens used during context presentation, response generation, user edits, reasoning question prompts (in Reasoning Mode), and the processing of user feedback. Tracking token usage helps assess the framework's efficiency.

5.2.4 Baseline. Notably, to evaluate how effectively the LM agent learns user privacy preferences through the Basic Mode and Reasoning Mode, we compare the performance of these modes against a baseline. In the baseline, no privacy learning mechanism is implemented, and responses are generated using only the base prompt for the same six scenarios. Therefore, we used the following metrics to assess improvements:

- Distance Improvement: measures the reduction in discrepancies between the baseline and the user-edited responses, compared to the modes with privacy learning mechanisms. It is calculated as:

$$DistanceImprovement = \frac{dist\_baseline - dist\_(basic/reasoning)}{dist\_baseline} \times 100\%.$$

- Information Alignment Improvement: measures the increase in semantic alignment between the generated responses and user privacy preferences, compared to the baseline. It is calculated as:

$$InformationAlignmentImprovement = \frac{align\_(basic/reasoning) - align\_baseline}{align\_(basic/reasoning)} \times 100\%.$$

5.3 Results

Table 1. Comparison of the average token-based edit distance, information-based alignment, and resource usage per user per round across the baseline, Basic Mode, and Reasoning Mode.

Metrics (per user per round)	Baseline	Basic Mode	Reasoning Mode
Token-based edit distance			
$Distance_{ave}$	0.64	0.20	0.16
$DistanceImprovement_{ave}$	0.00%	**68.75%**	**75.00%**
Information-based alignment			
$InformationAlignment_{ave}$	0.49	0.70	0.64
$InformationAlignmentImprovement_{ave}$	0.00%	**30.00%**	**23.44%**
Resource usage			
$TotalToken_{ave}$	-	541.00	546.22

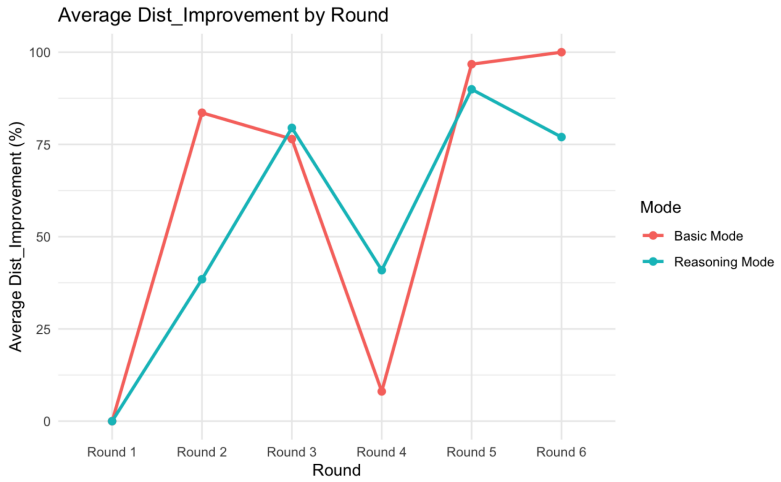

Fig. 3. Average distance improvement (%) (comparing with the baseline) across rounds of interaction for Basic Mode and Reasoning Mode, demonstrating the gradual improvement in alignment with user privacy preferences.

As a result, we collected interaction data from six users in total, with each user interacting with the system for 6 rounds. In other words, we obtained 36 rounds of interaction data, with 18 rounds for each mode respectively.

The results for the average token-based edit distance, information-based alignment, and resource usage per user per round across the three conditions (baseline, Basic Mode, and Reasoning Mode) are summarized in Table 1. The results show that learning through both the Basic Mode and Reasoning Mode reduces the distance significantly and achieves good performance in distance improvement (Basic Mode: 68.75%; Reasoning Mode: 75.00%). Similarly, learning through the Basic Mode improves alignment by 30.00%, while the Reasoning Mode achieves an alignment improvement of 23.44%. The token consumption for the two modes shows minimal differences. We also analyzed the average distance improvement across the rounds of interaction, as shown in Figure 3. A clear upward trend is observed, indicating that, over the course of the interactions, the PrivacyLens framework helps LM agents better learn and align with user privacy preferences. This suggests that, compared to not implementing the framework, the LM agents increasingly generate responses closer to individual preferences. However, there is a noticeable drop in round 4, where the LM agent incorrectly identified the user's identity, leading all users to correct this error in their edits. Overall, these findings indicate that both the Basic and Reasoning Modes have the potential to enhance alignment with user privacy preferences compared to the baseline.

6 DISCUSSION

6.1 Challenge of Preference Accumulation Over Time

The PrivacyLearner framework learns and accumulates user privacy preferences through interaction history, incorporating them into prompts to generate future responses. While the results demonstrate the potential of this mechanism to align LM agents with user privacy preferences,

the accumulation of privacy preferences over time may lead to increased resource consumption and decreased efficiency. To address this, future studies should conduct extended, multi-round experiments to observe how the accumulation of privacy preferences affects performance and resource usage over time. Additionally, more effective privacy preference management strategies are needed. For example, embedding techniques such as BERT [6] can be used to measure preference similarity and consolidate redundant or overlapping preferences. Another approach involves building a dynamic preference prioritization system that ranks and prioritizes preferences based on their relevance to the current task, particularly by considering the context. This approach ensures that only the most relevant preferences are incorporated into prompts, improving both efficiency and task-specific alignment.

6.2 Evaluation With Different Language Models

It is important to evaluate the framework's performance across different language models, particularly by comparing open-source models that can run locally with larger models that operate on the cloud. Cloud-based models, while often more capable of handling complex tasks, pose significant privacy risks. For example, they are vulnerable to memorization risks inherent in large language models [12], where sensitive user data could inadvertently become part of the training data. This raises concerns about user privacy when interacting with such models. On the other hand, open-source models that run locally offer greater privacy protection, as user data remains on the user's device. However, these models may have limitations in terms of computational resources and alignment performance, especially for complex tasks. A comparative evaluation of the PrivacyLearner framework on these two types of models could yield valuable insights into the trade-offs between utility, functionality, and privacy. Such an evaluation would help guide decisions on selecting the appropriate language models based on specific application needs and privacy requirements.

6.3 Limitations and Improvements

Several limitations in the current study must be addressed in future work. The current evaluation is based on interaction data from a small number of users and scenarios, which limits the generalizability of the findings. Future studies should aim to collect data from a larger and more diverse user base to improve generalizability and provide more robust and convincing evidence for the framework's performance. The results presented in Figure 3 also indicates the sensitivity of the existing evaluation metrics to the selection of scenarios. Scenario selection can significantly influence the evaluation outcomes, potentially introducing bias. This requires careful consideration to ensure fair interpretation of the results. Moreover, from a functional perspective, the ability to select or generate scenarios that effectively elicit user privacy preferences in relevant contexts is critical. Designing scenarios that target specific privacy challenges or contexts can enhance the framework's capacity to align with user preferences.

REFERENCES

[1] Anthropic. 2024. Computer use (beta). https://docs.anthropic.com/en/docs/build-with-claude/computer-use Accessed: 2025-01-19.

[2] Hannah Brown, Katherine Lee, Fatemehsadat Mireshghallah, Reza Shokri, and Florian Tramèr. 2022. What does it mean for a language model to preserve privacy?. In *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*. 2280–2292.

[3] Iason Gabriel. 2020. Artificial intelligence, values, and alignment. *Minds and machines* 30, 3 (2020), 411–437.

[4] Ge Gao, Alexey Taymanov, Eduardo Salinas, Paul Mineiro, and Dipendra Misra. 2024. Aligning llm agents by learning latent preference from user edits. *arXiv preprint arXiv:2404.15269* (2024).

[5] Geoffrey Irving, Paul Christiano, and Dario Amodei. 2018. AI safety via debate. *arXiv preprint arXiv:1805.00899* (2018).

[6] Mikhail V Koroteev. 2021. BERT: a review of applications in natural language processing and understanding. *arXiv preprint arXiv:2103.11943* (2021).

[7] Belinda Z Li, Alex Tamkin, Noah Goodman, and Jacob Andreas. 2023. Eliciting human preferences with language models. *arXiv preprint arXiv:2310.11589* (2023).

[8] Niloofar Mireshghallah, Hyunwoo Kim, Xuhui Zhou, Yulia Tsvetkov, Maarten Sap, Reza Shokri, and Yejin Choi. 2023. Can llms keep a secret? testing privacy implications of language models via contextual integrity theory. *arXiv preprint arXiv:2310.17884* (2023).

[9] Vinod Muthusamy, Yara Rizk, Kiran Kate, Praveen Venkateswaran, Vatche Isahagian, Ashu Gulati, and Parijat Dube. 2023. Towards large language model-based personal agents in the enterprise: Current trends and open problems. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 6909–6921.

[10] Helen Nissenbaum. 2004. Privacy as contextual integrity. *Wash. L. Rev.* 79 (2004), 119.

[11] OpenAI. 2025. Introducing Operator-Safety and privacy. https://openai.com/index/introducing-operator/ Accessed: 2025-01-19.

[12] Charith Peris, Christophe Dupuy, Jimit Majmudar, Rahil Parikh, Sami Smaili, Richard Zemel, and Rahul Gupta. 2023. Privacy in the time of language models. In *Proceedings of the sixteenth ACM international conference on web search and data mining*. 1291–1292.

[13] Toran Bruce Richards. 2023. *Auto-gpt: Autonomous artificial intelligence software agent.* https://github.com/Significant-Gravitas/AutoGPT

[14] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633* (2021).

[15] Yijia Shao, Tianshi Li, Weiyan Shi, Yanchen Liu, and Diyi Yang. 2024. PrivacyLens: Evaluating Privacy Norm Awareness of Language Models in Action. *arXiv preprint arXiv:2409.00138* (2024).

[16] Robin Staab, Mark Vero, Mislav Balunović, and Martin Vechev. 2023. Beyond memorization: Violating privacy via inference with large language models. *arXiv preprint arXiv:2310.07298* (2023).

[17] Yashar Talebirad and Amirhossein Nadiri. 2023. Multi-agent collaboration: Harnessing the power of intelligent llm agents. *arXiv preprint arXiv:2306.03314* (2023).

[18] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[19] Zhiping Zhang, Bingcan Guo, and Tianshi Li. 2024. Can Humans Oversee Agents to Prevent Privacy Leakage? A Study on Privacy Awareness, Preferences, and Trust in Language Model Agents. *arXiv preprint arXiv:2411.01344* (2024).

[20] Zhiping Zhang, Michelle Jia, Hao-Ping Lee, Bingsheng Yao, Sauvik Das, Ada Lerner, Dakuo Wang, and Tianshi Li. 2024. "It'sa Fair Game", or Is It? Examining How Users Navigate Disclosure Risks and Benefits When Using LLM-Based Conversational Agents. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery New York, NY, USA, 1–26.