Agentic Reinforcement Learning for Local Data Protection

Anonymous ACL submission

Abstract

The growing use of proprietary LLMs via APIs has dramatically increased the rate at which users are forced to expose sensitive text and documents to black-box services. However, existing inference-time defenses for minimizing such privacy leakage either break logical coherence, produce unfaithful surrogates, or restrict outputs to rigid formats—none of which suit reasoning tasks. We consider the case of a smaller local model querying a remote proprietary model for assistance and we argue that the primary challenge comes from balancing two competing demands: queries to the proprietary model must hide sensitive content while still being informative enough for the remote model to provide useful information to the local model, and the smaller local model must be capable of leveraging remote guidance into correct actions and answers. We address both demands through **Privacy-R1**, our proposed reinforcement learning framework that uses an outcome-based reward to train a local model to interleave reasoning on sensitive data with creating privacy-preserving auxiliary queries to a remote model and then integrating the remote guidance to generate a final answer. We use MedQA as our primary testbed and find that Privacy-R1 improves final accuracy of the local model by up to 8.4% on MedQA while maintaining a high privacy preservation score of over 95%, measured using ROUGE-L and 2-gram recall. We also test the model on unseen datasets to test for generalizability and find that accuracy improves by 9.6% on MedXpertQA.

1 Introduction

Many of the most powerful LLMs available for public use are proprietary and only accessible through API calls. Some large open-source models are difficult to deploy locally but can be accessed via APIs from model providers. Since these proprietary models have more knowledge and better reasoning capabilities, pretrained on vast amounts or domain-specific corpora, users often access these LLMs via APIs for their tasks, leaving open the risk of unintentionally sharing sensitive information. Furthermore, users typically expose all their data to these remote black-box models, raising concerns about data leakage (Wang et al., 2024). In many applications such as healthcare, finance, and customer data, it's not safe or legal to share text containing sensitive information through APIs.

In line with previous work (Zhou et al., 2023; Tong et al., 2023; Hartmann et al., 2024), we treat every token in the user input as having the potential to carry private or sensitive information, with the exception of common stop words that are generally regarded as non-sensitive (Yue et al., 2021). Such confidential content may appear in an overt form, like personally identifiable information (PII), or it could be woven implicitly into diverse contexts and formats, ranging from corporate profiles and internal operating metrics to strategic assessments. While a number of approaches (Siyan et al., 2024; Chen et al., 2023; Aahill, 2023) concentrate on masking PII—for instance age, name, or geographic location—recent evidence shows that powerful models such as GPT-4 are still capable of recovering personal attributes from the leftover context (Staab et al., 2024). Therefore, our work seeks to provide comprehensive protection of document-level local text prior to its submission to a black-box model for inference.

Stripping too much information from the local data before querying a proprietary remote model compromises task performance, particularly for reasoning tasks requiring text understanding, domain knowledge, and logical reasoning. Our goal is to minimize privacy leakage while maximizing reasoning accuracy, thereby striking a balance between privacy and accuracy in API-based inference-time model usage.

Prior inference-time protection methods fall into three categories, each ill-suited to reasoning. *DP-*

based perturbation (Utpala et al., 2023; Tong et al., 2023; Xie et al., 2024) substitutes tokens with noise, yielding text that preserves enough semantic meaning for classification but breaks the logical chains that reasoning tasks require. *Task-specific protocols* (Zhang et al., 2026) exchange predefined templates for numerical reasoning and recover answers via a local interpreter, but the rigid format excludes free-form questions and leaves the local model's reasoning ability untouched. *Abstraction-based* methods (Zhang et al., 2024; Hartmann et al., 2024) are most similar to our work, in these approaches a local model sends high-level descriptions or analogous examples to a remote model for assistance. However, these methods rely solely on prompting and do not train the local model to interact with the remote model more effectively.

These shortcomings reflect two core challenges. First, privacy and informativeness are inherently at odds: concealing content limits leakage but also withholds the contextual detail the remote model needs to reason correctly. Second, local capacity bottlenecks the use of remote guidance: even given a correct hint, a small local model often fails to derive the right answer (Zhang et al., 2024; Hartmann et al., 2024).

To address these challenges, we propose an agentic framework that jointly optimizes two objectives: shielding sensitive local data from the proprietary model, and equipping the local model to seek and leverage external guidance to solve the problem. We design a reinforcement learning framework called **Privacy-R1**, that supports LLM rollouts and direct optimization of a local model with a proprietary model in the loop. The reward function is designed to optimize privacy-preservation while querying a remote model, and using the remote guidance for to improve final accuracy on the task. Using GPT-5.4-nano (OpenAI, 2026) as the remote model and Qwen2.5-3B-Instruct (Team et al., 2024) and Qwen3-4B-Instruct (Yang et al., 2025) as local models, we note an improved accuracy score of 5.58% and 8.4% respectively on MedQA (Jin et al., 2021). We track privacy preservation using 2-gram recall and ROUGE-L and find that privacy scores are over 95%, successfully safeguarding sensitive local data. We further verify the effectiveness of our method on the unseen dataset MedXpertQA (Zuo et al., 2025), where it achieves a 9.6% improvement in accuracy. We also discuss the impact of full agent instruction versus staged separate instruction.

2 Method

Privacy-R1 studies how a comparatively small local model can use a stronger remote model without revealing the original private input. The central design choice is to decouple problem ownership from external knowledge access. The local model is the only problem-solving model that observes the full user question, including all sensitive details and answer options. The remote model is used only as an external expert: it receives a sanitized auxiliary query generated by the local model and returns general guidance. The local model then integrates that guidance with the original private question to produce the final answer. This section first describes the inference workflow independently of learning, then explains how we construct supervised fine-tuning data, and finally presents the reinforcement-learning objective and reward.

2.1 Problem Setting and Design Goals

We consider tasks in which the input question q may contain sensitive details. The user would like to benefit from a strong remote model M_{remote}, but does not want to send q directly to it. Privacy-R1 therefore introduces a local policy M_θ that remains on the trusted side and mediates all access to the remote model. The local policy has two responsibilities: it must formulate a privacy-preserving auxiliary query $\tilde{q}$ that contains enough abstract information for the remote expert to be helpful while avoiding enough task-specific detail that the original input cannot be reconstructed; and it must solve the original task.

This setting creates a three-way tension. First, $\tilde{q}$ must be useful: a query such as "what should I do for this patient?" is private but not self-contained, so the remote model cannot answer it meaningfully. Second, $\tilde{q}$ must be private: copying age, laboratory values, or long phrases from the original question may allow the remote model to infer the original case. Third, the final answer must remain accurate: after receiving remote guidance, the local model must map the abstract advice back to the exact private question and options. Privacy-R1 is designed to learn this balance rather than simply forbidding remote assistance.

2.2 Privacy-R1 Workflow

Privacy-R1 orchestrates three components with clearly separated roles. The **local model** M_θ is the only party with access to the original input and

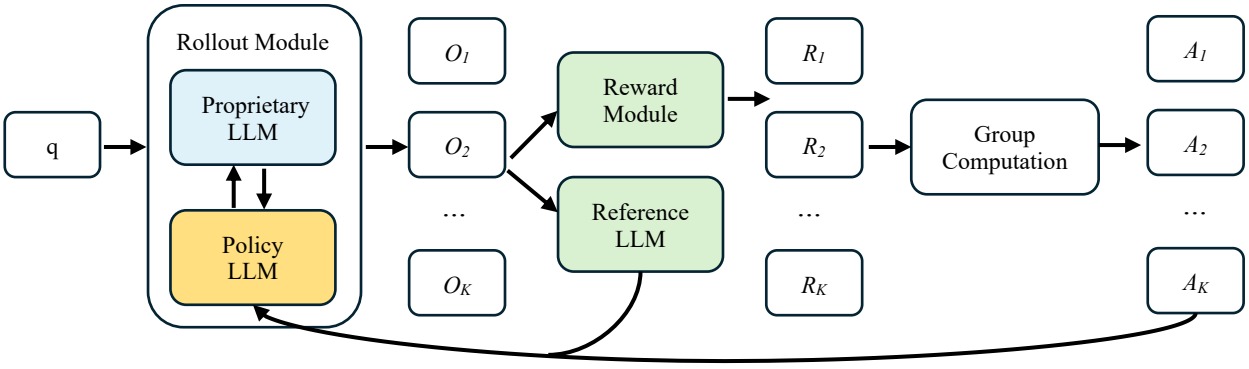

Figure 1: Illustration of GRPO training with the exposure controller agent and the proprietary LLM API.

dynamically switches between a *proposer* and a *solver* role. The **remote expert** M_{remote} is stateless with respect to the task and only sees what the local model chooses to expose. The **privacy monitor**, deployed on the trusted training side, inspects each outgoing query against the original input.

Information flow. Given an original question q, the workflow proceeds as follows: **(1) Query proposal.** M_θ reads q in full and reasons over it, emits an auxiliary query $\tilde{q}$ inside an `<auxiliary_query>` tag that is self-contained and targets the underlying concept rather than the private instance. **(2) Privacy check.** The monitor compares $\tilde{q}$ against q. We use deterministic lexical checks (2-gram recall and ROUGE-L recall, both thresholded at 0.3). If any leakage threshold is exceeded, the rollout is marked as privacy-failing. **(3) Remote consultation.** The remote model is triggered as a tool call: $\tilde{q}$ is parsed from M_θ's output and dispatched as the query to M_{remote}. Only $\tilde{q}$ is sent to M_{remote}, which returns general guidance g (e.g., domain knowledge, common pitfalls, analogous solutions). This guidance is intentionally abstract, providing useful signal to M_θ. **(4) Final answering.** M_θ consumes g together with the original q and emits the final answer inside an `<answer>` tag, performing all instance-specific reasoning and anwering locally.

The entire protocol including reasoning, conditional querying, guidance consumption, and final answering is conveyed through a single end-to-end prompt, so the local model learns the full collaborative behavior from one unified interface. The full prompt is provided in the Appendix A.1.

2.3 Supervised Fine-Tuning Data Collection

The supervised stage teaches the local model the interaction format before RL. Our SFT data is built from existing teacher rollouts. A trajectory is accepted only if it has the expected role pattern, contains exactly one auxiliary query without privacy leakage, contains a final answer tag, and the final answer exactly matches the ground truth.

During multi-turn SFT, loss is applied only to assistant tokens, so both the first assistant turn that proposes the auxiliary query and the final assistant turn that solves the problem are supervised. System, user, and tool messages provide context but are not prediction targets.

2.4 Reinforcement Learning

After SFT, we optimize the local policy with multi-turn RL. The remote expert M_{remote} and privacy monitor are part of the rollout environment, while only the local policy M_θ's parameters are updated. As illustrated in Fig. 1, we use GRPO to optimize the policy network. For each input question, the rollout worker samples multiple candidate trajectories from the local model, executes any auxiliary-query tool calls, records privacy-monitor results, and computes a rule-based reward from the completed transcript.

We do not use a learned reward model; the reward is computed directly from answer correctness and privacy metadata.

Accuracy component. The final answer is extracted from the last `<answer>...</answer>` span in the local model response. The accuracy score $a \in \{0, 1\}$ is exact match against the ground truth answer string.

Privacy component. The privacy monitor produces a binary privacy score $p \in \{0, 1\}$. If a generated auxiliary query passes the enabled leakage checks, $p = 1$; if any enabled check flags leakage, $p = 0$. If no auxiliary query is generated, no information is exposed to the remote expert, so the

privacy score defaults to pass.

Scalar reward. The Privacy-R1 reward combines task accuracy with a privacy gate. The reward is defined as

$$R(x, \tau, y) = \begin{cases} a, & \text{if } p = 1, \\ 0.5\,a, & \text{if } p = 0, \end{cases} \tag{1}$$

which can be written equivalently in closed form as $R = a\,(0.5 + 0.5\,p)$ for binary p.

This formulation yields three distinct outcomes: (i) an incorrect answer receives zero reward regardless of privacy behavior, since $a = 0$ nullifies the reward; (ii) a correct answer produced through a privacy-preserving trajectory receives the full reward of 1; and (iii) a correct answer obtained at the cost of leaking private information receives only half credit (0.5). The rationale behind this reward design is empirical: after prompting and supervised fine-tuning, the model already produces auxiliary queries with strong privacy properties, so reinforcement learning primarily needs to optimize accuracy while discouraging residual privacy violations. This design keeps task success as the primary objective while making privacy violations costly even when the model obtains the right answer.

3 Experiment Settings

Datasets We evaluate Privacy-R1 on the multiple-choice medical question answering task used throughout our development runs. Each example consists of a question stem and a fixed set of answer options. The ground truth is the option letter and option content. We use two datasets, MedQA (Jin et al., 2021) and MedXpertQA (Zuo et al., 2025). These data include sensitive information such as patient medical records and examination results that warrant protection, while also requiring strong reasoning capabilities from the model. The MedQA training set contains 10,178 examples, and the test set contains 1,273 examples. We randomly sampled 500 questions from the 2,450 text-only test cases in MedXpertQA as our test set. We train the models only on the MedQA training set, and test on both the MedQA and MedXpertQA test sets.

Metrics. We report three metrics. *Accuracy* is the percentage of test examples for which the extracted final answer matches the gold option. This is the main task metric. *Privacy score* is the mean privacy pass rate emitted by the Privacy-R1 evaluation pipeline; higher is better, and 1.0 indicates that the model's external queries pass the privacy monitor on all evaluated examples. *Tool-call rate* is the fraction of examples on which the local model emits a valid external-expert query. This metric measures whether the model actually uses the remote collaboration mechanism rather than only solving locally under the workflow prompt.

Backbones and remote experts. We evaluate two local backbones: Qwen2.5-3B-Instruct (Team et al., 2024) and Qwen3-4B-Instruct (Yang et al., 2025). For the remote expert, we use GPT-5.4-nano (OpenAI, 2026).

Compared methods. We compare four systems for each local backbone. First, *Local-only* is a direct local inference baseline. The model is given only the question and answer options and must answer without any remote model, tool interface, privacy monitor, SFT policy, or RL policy. This baseline measures the standalone capability of the local model. Second, *Privacy-R1 prompting workflow* uses the Privacy-R1 workflow prompt and the configured remote expert interface, but it does not update the local model parameters. Third, *Privacy-R1 SFT* evaluates a supervised fine-tuned checkpoint trained on collected Privacy-R1-style trajectories. This setting tests whether behavioral cloning alone can teach the local model to follow the private collaboration format. Fourth, *Privacy-R1 GRPO* evaluates the reinforcement-learned policy initialized from an SFT checkpoint and optimized with the Privacy-R1 reward.

Supervised data collection. We use gemini-2.5-flash as the teacher to collect Privacy-R1-like trajectories as the SFT data. After data quality filtering, the trajectory corpus contains 5,813 examples, split into 5,696 training examples and 117 validation examples. Each supervised example stores the multi-turn message trace rather than only the final answer. Therefore, SFT teaches the local model the complete interaction pattern: how to inspect the original medical question, how to form a privacy-preserving expert query when it needs outside help, how to incorporate the returned expert response, and how to produce the final answer in the expected answer format. This design makes SFT a behavior imitation stage rather than a pure answer-only distillation stage. It exposes the model to the Privacy-R1 protocol but does not directly optimize downstream accuracy or privacy-constrained tool-use decisions.

SFT training setup. The SFT model is trained with the Verl (Sheng et al., 2024) FSDP SFT trainer, multi-turn mode enabled. All SFT runs are trained with a maximum sequence length of 8,192 tokens, left truncation, bfloat16 precision, FSDP2 sharding, gradient checkpointing, and remove-padding optimization. We use Adam-style optimization with learning rate 5×10^{-6} and warmup ratio 0.03.

RL data and GRPO setup. The GRPO stage uses the MedQA `whole` training split with 10,178 examples. For both Qwen2.5-3B and Qwen3-4B GRPO runs, each training prompt samples 8 rollouts with temperature 0.7 and top-$p = 0.8$; validation uses a single rollout per example. The maximum prompt length is 5,120 tokens, the maximum response length is 4,096 tokens, and the rollout engine uses a maximum model length of 10,240 tokens. The multi-turn agent is allowed up to 3 assistant turns, which is sufficient for one or more privacy-screened expert queries followed by a final answer. Optimization uses GRPO with actor learning rate 1×10^{-5}, PPO mini-batch size 8, per-GPU micro-batch size 4, KL regularization coefficient 0.001, and a low-variance KL estimator. All experiments are conducted on a workstation equipped with 4 NVIDIA A100 GPUs (40GB each). Results are reported from a single run.

4 Results

4.1 Main Result

Main accuracy results. Table 1 shows that GRPO is the only training stage that consistently improves over the local-only baselines on both backbones. On Qwen2.5-3B, the local-only model reaches 44.15% accuracy. The prompt-only Privacy-R1 workflow decreases accuracy to 37.86%, and SFT alone reaches 39.04%. In contrast, Privacy-R1 GRPO reaches 49.73%, improving by 5.58 percentage points over local-only, 11.87 points over the prompt-only workflow, and 10.69 points over the SFT checkpoint. This result indicates that simply exposing a 3B model to a remote-expert interface is not enough: without reward optimization, the model can overuse or misuse the tool and still fail to convert remote information into correct final answers. GRPO changes this behavior by directly selecting for answer correctness under the privacy-constrained workflow.

The same pattern holds more strongly for Qwen3-4B. The local-only Qwen3-4B baseline obtains 58.76% accuracy. The prompt-only Privacy-R1 workflow reaches 61.43%, a 2.67-point improvement over local-only. However, its tool-call rate is 0.00%, so this row should be interpreted as prompt-conditioned direct answering rather than successful local-remote collaboration. The configured remote expert is available in the workflow, but the model does not emit valid auxiliary queries during evaluation. The SFT-only RL-template checkpoint drops to 23.41% accuracy despite a high privacy score, showing that strict imitation of the interaction format can be brittle when it is not reinforced by answer-level reward. Privacy-R1 GRPO gives the best Qwen3-4B result at 67.16%, improving by 8.40 points over local-only and 5.73 points over the prompt-only workflow while activating the tool on 95.05% of examples.

Privacy and tool-use behavior. Privacy-R1 maintains high privacy scores across all workflow-based runs. Qwen2.5-3B prompting achieves a 94.50% privacy score, while Qwen2.5-3B SFT and GRPO reach 98.27% and 97.64%, respectively. Qwen3-4B obtains privacy scores of 100.00%, 99.84%, and 99.92% for prompting, SFT, and GRPO. These numbers show that the privacy monitor and the learned workflow can keep external queries within the allowed privacy boundary even when the tool is used frequently. The Qwen2.5-3B GRPO model calls the tool on 99.84% of examples, and the Qwen3-4B GRPO model calls the tool on 95.05% of examples, while both retain privacy scores above 97.6%.

The tool-call metric also clarifies why accuracy alone is not sufficient to evaluate this setting. For Qwen3-4B prompting, the privacy score is perfect because the model never issues valid tool calls; no private information is leaked, but the remote expert also never contributes. Conversely, Qwen2.5-3B SFT calls the tool on 92.62% of examples but remains below the local-only baseline in accuracy. High tool-use frequency therefore does not guarantee useful collaboration. The desired behavior is not to call the tool as often as possible; it is to issue queries that satisfy the privacy constraint and improve the final medical answer. The GRPO results are the closest to this target behavior: they combine high tool-call rates, high privacy scores, and the best accuracy for both local backbones.

Effect of SFT versus GRPO. The SFT results suggest that supervised imitation is useful as an initialization mechanism but insufficient as the final optimization objective. The SFT data demon-

Table 1: Full-test results on the 1,273-example MedQA test split. Accuracy, privacy, tool-call rate, and accuracy improvement are reported in percentage points. "N/A" means the metric is not defined for local-only inference.

Backbone	Method	Accuracy	Privacy	Tool call	Δ Acc.
Qwen2.5-3B	Local-only	44.15	N/A	N/A	0.00
Qwen2.5-3B	Prompting workflow	37.86	94.50	78.87	-6.29
Qwen2.5-3B	Privacy-R1 SFT	39.04	98.27	92.62	-5.11
Qwen2.5-3B	Privacy-R1 GRPO	**49.73**	97.64	**99.84**	**+5.58**
Qwen3-4B	Local-only	58.76	N/A	N/A	0.00
Qwen3-4B	Prompting workflow	61.43	100.00	0.00	+2.67
Qwen3-4B	Privacy-R1 SFT	23.41	99.84	39.67	-35.35
Qwen3-4B	Privacy-R1 GRPO	**67.16**	99.92	**95.05**	**+8.40**

strates the Privacy-R1 protocol and teaches the local model how a valid trajectory should look, but it cannot fully distinguish trajectories that merely match the format from trajectories that produce correct answers under privacy constraints. This is especially visible in the Qwen3-4B RL-template SFT row: the model largely preserves privacy, but answer accuracy collapses, indicating that format compliance can dominate task solving when the model is evaluated directly after imitation. Qwen2.5-3B SFT is less severe but still underperforms local-only by 5.11 points.

GRPO addresses this limitation by optimizing the policy with a scalar reward tied to the final outcome of the workflow. The reward encourages the local model to produce the correct final answer while respecting the privacy monitor and the interaction protocol. Empirically, this turns SFT from a behavioral warm start into a trainable policy: after GRPO, Qwen2.5-3B moves from 39.04% to 49.73%, and Qwen3-4B moves from the supervised initialization regime to 67.16%. The improvement is not only an accuracy gain but also a change in interaction quality. The GRPO policies learn to invoke the external expert reliably, keep the queries privacy-compliant, and integrate the returned information into the final answer.

Backbone-level comparison. Qwen3-4B is stronger than Qwen2.5-3B in both local-only and GRPO settings. The local-only gap is 14.61 percentage points, and the final GRPO gap is 17.43 points. This indicates that Privacy-R1 benefits from a more capable local reasoner: the local model must decide what information is missing, formulate a safe query, interpret the expert response, and still perform final answer selection. At the same time, the relative gains show that both model scales benefit from reinforcement learning. Qwen2.5-3B gains 5.58 points over local-only, and Qwen3-4B gains 8.40 points. Thus, Privacy-R1 does not merely expose the advantage of the larger backbone; it improves both tested local models by training the collaboration policy directly.

Method	Accuracy	Privacy	Tool Call
Prompting	6.8	**99.4**	73.2
Privacy-R1 GRPO	**16.4**	96.6	**98.8**

Table 2: Unseen-dataset evaluation on MedXpertQA. The RL model is trained only on MedQA and evaluated directly on MedXpertQA without any additional tuning. Accuracy, privacy, and tool-call rates are reported as percentages.

Takeaway. Across both backbones, the full Privacy-R1 GRPO pipeline gives the best accuracy while preserving high privacy. Prompting alone is unstable: it can hurt a weaker local model and, for Qwen3-4B, can improve accuracy without actually using the remote expert. SFT alone teaches the protocol but does not reliably optimize final-answer correctness. GRPO is the stage that aligns the local model's tool-use decisions with the end task.

4.2 Generalization to an Unseen Dataset

We further evaluate whether the learned policy generalizes beyond the training distribution. The RL model is trained only on MedQA and never observes MedXpertQA during training. Therefore, evaluating on MedXpertQA provides a direct test of unseen-dataset generalization: improvements on this benchmark indicate that the learned privacy-preserving tool-use policy is not merely overfitting to MedQA, but can transfer to a different medical QA distribution.

Table 2 shows that RL substantially improves performance on the unseen MedXpertQA dataset. Compared with prompting, the RL-trained model improves the overall reward from 6.7 to 16.1, corresponding to a 9.4-point absolute gain and approx-

imately a 2.4× relative improvement. Accuracy also increases from 6.8 to 16.4, showing that the model learns a policy that transfers to a new medical QA benchmark rather than only improving on the MedQA training distribution.

The gain is closely related to tool-use behavior. The prompting baseline calls the external expert in only 73.2% of examples, while the RL model calls it in 98.8% of examples. This suggests that RL teaches the model to more reliably seek external assistance when needed. Although privacy decreases slightly from 99.4 to 96.6, it remains high, indicating that the improvement in task performance is not achieved by abandoning privacy preservation. Overall, the unseen-dataset results support the effectiveness of RL: training on MedQA yields a more robust privacy-preserving reasoning policy that generalizes to MedXpertQA and significantly outperforms prompting.

4.3 Agentic Full-Instruction versus Staged Separate-Instruction

We further study how the prompt interface affects Privacy-R1. We compare two instruction designs as shown in Table 3. The first is the *Agentic Full-Instruction* design, which corresponds to the implementation-level `whole` prompt mode. In this design, the local model receives the full multiple-choice question, including the answer options, and decides whether to issue a privacy-screened auxiliary query to the remote expert. The local model therefore keeps the complete decision context throughout the interaction: it can inspect the original question, formulate a sanitized expert query, interpret the expert response, and select the final option. The second is the *Staged Separate-Instruction* design, which corresponds to the implementation-level `separate` prompt mode. In this design, the instruction explicitly separates remote consultation from final answering. The local model is encouraged to construct an auxiliary query for the expert as a separate subproblem, while the final answer is still produced locally.

Tool activation. The clearest advantage of the Staged Separate-Instruction design is that it makes tool use much easier to elicit. For Qwen2.5-3B, Staged Separate-Instruction raises the tool-call rate from 78.87% to 99.69%, while also improving prompt-only accuracy from 37.86% to 42.34%. This suggests that the staged instruction gives the smaller model a simpler action structure: instead of jointly deciding how to reason over the full question and when to consult the expert, the model is guided toward an explicit expert-query step. A similar pattern appears for Qwen3-4B in the available initialization runs. Agentic Full-Instruction with the base Qwen3-4B model obtains 61.43% accuracy but makes no valid tool calls, so the result is best interpreted as direct local answering under the Privacy-R1 prompt. By contrast, the Staged Separate-Instruction initialization run calls the tool on 83.19% of examples. Thus, Staged Separate-Instruction is effective at inducing collaboration behavior at the surface level.

However, higher tool use does not imply a higher RL ceiling. After reinforcement learning, Agentic Full-Instruction gives stronger final policies. On Qwen2.5-3B, Agentic Full-Instruction GRPO reaches 49.73% accuracy, whereas Staged Separate-Instruction GRPO reaches only 37.08%. This is a 12.65-point gap in favor of Agentic Full-Instruction. The comparison is especially revealing because Staged Separate-Instruction starts with a higher prompt-only score and a much higher tool-call rate, yet its GRPO policy does not preserve this advantage. Its tool-call rate drops from 99.69% at initialization to 73.21% after RL evaluation, and accuracy falls below both its own prompt-only result and the Agentic Full-Instruction GRPO result. This indicates that merely forcing or encouraging expert calls is not sufficient; the policy must learn queries that preserve the information needed for final answer selection.

The Qwen3-4B results show the same qualitative conclusion, although the gap is smaller. Staged Separate-Instruction GRPO reaches 65.04% accuracy with a 94.19% tool-call rate, which is a strong result and confirms that the staged instruction can be optimized by RL. Nevertheless, the Agentic Full-Instruction GRPO policy reaches 67.16% accuracy, with a slightly higher tool-call rate. Agentic Full-Instruction therefore provides the better final ceiling even though its prompt-only behavior is less collaborative. This difference matters because the goal of Privacy-R1 is not only to call the expert, but to call the expert in a way that improves the final answer under the privacy constraint.

Why Agentic Full-Instruction works better after RL. We attribute the stronger RL ceiling of Agentic Full-Instruction to the fact that the local model retains the complete decision context. Agentic Full-Instruction lets the local model inspect

Table 3: Comparison between *Agentic Full-Instruction* and *Staged Separate-Instruction* on the full MedQA test split. Accuracy, privacy, and tool-call rate are percentages.

Backbone	Instruction design	Stage / checkpoint	Acc.	Privacy	Tool call
Qwen2.5-3B	Staged Separate	Prompting	42.34	93.95	99.69
Qwen2.5-3B	Agentic Full	Prompting	37.86	94.50	78.87
Qwen2.5-3B	Staged Separate	GRPO	37.08	97.09	73.21
Qwen2.5-3B	Agentic Full	GRPO	**49.73**	97.64	**99.84**
Qwen3-4B	Staged Separate	Prompting	47.68	100.00	83.19
Qwen3-4B	Agentic Full	Prompting	61.43	100.00	0.00
Qwen3-4B	Staged Separate	GRPO	65.04	99.45	94.19
Qwen3-4B	Agentic Full	GRPO	**67.16**	99.92	**95.05**

the answer choices while deciding what knowledge to request, but still requires the query itself to pass the privacy monitor. This creates a more informative optimization problem for GRPO: the reward can shape both the privacy-preserving abstraction sent to the expert and the final integration step. Staged Separate-Instruction, in contrast, can over-emphasize the act of querying. It makes the expert-call action easy, but the separated query may omit option-level distinctions that are necessary for the final answer. As a result, high tool-call rate becomes a weak proxy for useful collaboration. Overall, these results suggest that Staged Separate-Instruction is useful for encouraging tool exploration and making the privacy boundary explicit, but Agentic Full-Instruction is the better interface for the final Privacy-R1 policy because it gives RL enough context to optimize the entire local-remote reasoning trajectory.

5 Related Work

Existing research has largely focused on safeguarding training corpora (Papernot et al., 2017; Yue et al., 2021; Tian et al., 2022; Kurakin et al., 2023; Xie et al., 2024; Yu et al., 2024) or in-context demonstrations (Hong et al., 2024; Carey et al., 2024), leaving user queries submitted at inference time largely unprotected. A line of work targeting inference-time privacy injects differential privacy (DP) noise directly into text embeddings under white-box assumptions (Du et al., 2023; Zhou et al., 2023). Such strategies, however, cannot be applied to black-box APIs, since these services accept raw text rather than continuous embedding vectors as input. Another direction focuses on scrubbing personally identifiable information either through dedicated hider modules (Chen et al., 2023) or via prompt-level optimization (Siyan et al., 2024). Even after such substitutions, the surrounding context frequently retains enough signal to enable re-identification (Staab et al., 2024). When the unit of protection is an entire document, one family of methods perturbs the text with DP mechanisms to produce semantically close yet modified variants (Tong et al., 2023; Xie et al., 2024), while another instructs LLMs to rephrase the document for downstream applications such as sentiment analysis (Utpala et al., 2023). A common limitation of these techniques is that, although meaning is roughly retained, the rewritten text is not guaranteed to remain logically faithful to the source. A separate thread leverages on-device models to produce abstract summaries (Zhang et al., 2024) or to construct analogous surrogate examples (Hartmann et al., 2024). Nevertheless, crafting queries that simultaneously hide sensitive information and remain useful for the task is still an open problem, and small local models often fail to make full use of signals provided by a stronger remote model owing to their constrained capability.

6 Conclusion

We tackle the problem of preserving document-level privacy when reasoning through API-based proprietary LLMs, where prior defenses sacrifice either logical coherence, query faithfulness, or output flexibility. Our key insight is that a capable local agent is needed to balance privacy against informativeness and to make effective use of remote guidance. Built on this insight, Privacy-R1 trains such an agent via reinforcement learning with outcome-based rewards, yielding 5.58%–8.40% accuracy gains while concealing over 95% of local content, with consistent benefits on unseen data. We hope this perspective encourages further work on agentic approaches to privacy-aware LLM usage to preserve sensitive user data even when querying a remote proprietary model.

7 Limitations

While Privacy-R1 demonstrates a favorable privacy–accuracy trade-off, our study has several limitations that we leave to future work. First, our experiments are conducted primarily on medical reasoning datasets; although the domain is well-suited to studying privacy-sensitive inference, the generalization of our framework to other high-stakes areas such as legal, financial, and scientific reasoning remains to be examined.

8 Potential Risks

Our framework reduces but does not eliminate privacy leakage. During inference, the local model may still expose sensitive user information to the remote model when generating auxiliary queries, especially on inputs that differ from the training distribution. We therefore advise caution when deploying our method in privacy-sensitive applications, and recommend complementing it with post-processing measures such as output filtering, query sanitization, and human-in-the-loop review.

References

Aahill. 2023. What is azure ai language - azure ai services.

Alycia N Carey, Karuna Bhaila, Kennedy Edemacu, and Xintao Wu. 2024. Dp-tabicl: In-context learning with differentially private tabular data. In *2024 IEEE International Conference on Big Data (BigData)*, pages 1552–1557. IEEE Computer Society.

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

Minxin Du, Xiang Yue, Sherman SM Chow, Tianhao Wang, Chenyu Huang, and Huan Sun. 2023. Dp-forward: Fine-tuning and inference on language models with differential privacy in forward pass. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 2665–2679.

Florian Hartmann, Duc-Hieu Tran, Peter Kairouz, Victor Cărbune, et al. 2024. Can llms get help from other llms without revealing private information? *arXiv preprint arXiv:2404.01041*.

Junyuan Hong, Jiachen T Wang, Chenhui Zhang, LI Zhangheng, Bo Li, and Zhangyang Wang. 2024. Dp-opt: Make large language model your privacy-preserving prompt engineer. In *The Twelfth International Conference on Learning Representations*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Alexey Kurakin, Natalia Ponomareva, Umar Syed, Liam MacDermed, and Andreas Terzis. 2023. Harnessing large-language models to generate private synthetic text. *arXiv preprint arXiv:2306.01684*.

OpenAI. 2026. GPT-5.4-nano. https://platform.openai.com/docs/models/gpt-5.4-nano.

Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. 2017. Semi-supervised knowledge transfer for deep learning from private training data. In *International Conference on Learning Representations*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*.

Li Siyan, Vethavikashini Chithrra Raghuram, Omar Khattab, Julia Hirschberg, and Zhou Yu. 2024. Papillon: Privacy preservation from internet-based and local language model ensembles. *arXiv preprint arXiv:2410.17127*.

Robin Staab, Mark Vero, Mislav Balunović, and Martin Vechev. 2024. Beyond memorization: Violating privacy via inference with large language models. In *International Conference on Learning Representations 2024*.

Qwen Team, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Zhiliang Tian, Yingxiu Zhao, Ziyue Huang, Yu-Xiang Wang, Nevin L Zhang, and He He. 2022. Seqpate: Differentially private text generation via knowledge distillation. *Advances in Neural Information Processing Systems*, 35:11117–11130.

Meng Tong, Kejiang Chen, Yuang Qi, Jie Zhang, Weiming Zhang, and Nenghai Yu. 2023. Inferdpt: Privacy-preserving inference for black-box large language model. *arXiv preprint arXiv:2310.12214*.

Saiteja Utpala, Sara Hooker, and Pin-Yu Chen. 2023. Locally differentially private document generation using zero shot prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8442–8457.

Jeffrey G Wang, Jason Wang, Marvin Li, and Seth Neel. 2024. Pandora's white-box: Increased training data leakage in open llms. *arXiv preprint arXiv:2402.17012*.

Chulin Xie, Zinan Lin, Arturs Backurs, Sivakanth Gopi, Da Yu, Huseyin Inan, Harsha Nori, Haotian Jiang, Huishuai Zhang, Yin Tat Lee, et al. 2024. Differentially private synthetic data via foundation model apis 2: text. In *Proceedings of the 41st International Conference on Machine Learning*, pages 54531–54560.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Da Yu, Peter Kairouz, Sewoong Oh, and Zheng Xu. 2024. Privacy-preserving instructions for aligning large language models. In *International Conference on Machine Learning*, pages 57480–57506. PMLR.

Xiang Yue, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman SM Chow. 2021. Differential privacy for text analytics via natural text sanitization. *arXiv preprint arXiv:2106.01221*.

Kaiyan Zhang, Jianyu Wang, Ermo Hua, Biqing Qi, Ning Ding, and Bowen Zhou. 2024. Cogenesis: A framework collaborating large and small language models for secure context-aware instruction following. *arXiv preprint arXiv:2403.03129*.

Min Zhang, Yuzhe Lu, Yun Zhou, Panpan Xu, Lin Lee Cheong, Chang-Tien Lu, and Haozhu Wang. 2026. Collaborative llm numerical reasoning with local data protection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 34772–34780.

Xin Zhou, Yi Lu, Ruotian Ma, Tao Gui, Yuran Wang, Yong Ding, Yibo Zhang, Qi Zhang, and Xuan-Jing Huang. 2023. Textobfuscator: Making pre-trained language model a privacy protector via obfuscating word representations. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5459–5473.

Yuxin Zuo, Shang Qu, Yifei Li, Zhang-Ren Chen, Xuekai Zhu, Ermo Hua, Kaiyan Zhang, Ning Ding, and Bowen Zhou. 2025. Medxpertqa: Benchmarking expert-level medical reasoning and understanding. In *International Conference on Machine Learning*, pages 80961–80990. PMLR.

A Appendix

A.1 Agentic Full Instruction Prompt Template

The complete prompt template used by the Agentic Full-Instruction workflow is shown below. In implementation, the placeholder {question} is replaced by the full MedQA question together with all answer options.

Whole-prompt template used by PrivacyR1

System: You are a helpful and harmless assistant that collaborates with an external expert for problem solving and prioritizes privacy protection.

User: Answer the following question.
Follow these rules strictly:
1. Always perform reasoning inside <think> and </think> first every time you get new information, including the problem or the external guidance.
2. Request guidance from an external expert if you are uncertain about the correct answer, the underlying mechanism, the neccessary knowledge, or the reasoning approach. Propose a privacy-preserving auxiliary query to seek external expert's guidance by outputting EXACTLY ONE auxiliary query in the following format:
<think> write your step-by-step thinking here </think> <auxiliary_query> write your privacy-preserving query here </auxiliary_query>
The auxiliary query must be privacy-preserving and self-contained:
Privacy Guardrail: Avoid including details that could lead the external advisor to guess, reconstruct, or infer key aspects of the original input.
Privacy is considered compromised if any of the following conditions are met: (1) The inclusion of important information such as names, ages, or other values; (2) Substantial token-level overlap with the original content; (3) Content that allows the remote model to infer or reconstruct sensitive elements of the original input.
Note that general or commonly used words, even if present in the source material, are not considered privacy violations when they lack contextual specificity or reidentification power.
Self-Containment: Note that the external expert cannot see the original question. The auxiliary query must be self-contained and independently understandable, so that an external expert can interpret and answer it without access to the original problem.
For example, this bad, not self-contained auxiliary query "Could you provide insight into the most likely causes for the patient's symptoms?" cannot stand on its own. Because it refers to "for the patient's symptoms," but the external expert does not have access to any original information of the patient symptoms or options. As a result, the query cannot be understood or answered independently based on its own content, which violates the self-containment requirement.
Tips: Focus the query on concepts, uncertainties, or domain-specific questions rather than specific individuals or cases.
Here is a good example of the auxiliary query: "In emergency medicine, what are the standard ethical and legal guidelines for providing life-saving treatment to a minor who cannot obtain parental consent, particularly in cases of suspected severe allergic reaction with airway involvement and hypotension?" This auxiliary query is good because it is self-contained and general—it asks about standard medical ethics and emergency guidelines without referencing any specific case details. It also preserves privacy by avoiding identifiable or unique information, while still clearly conveying the clinical scenario (life-threatening allergic reaction in a minor) needed for an expert to provide useful guidance.
3. The external expert will return the guidance.
You may repeat the above process for multiple rounds if needed, but only one auxiliary query per turn.
4. Once you have sufficient information to solve the problem, you need to integrate the external guidance and reason step by step inside <think> and </think> and to provide the final answer inside <answer> and </answer>.
You must analyze the external guidance and solve the problem step by step inside <think> and </think>. Explicitly identify key hints and strategies in the external expert's guidance and how these might help solve the original question.
Then, output the answer inside <answer> and </answer>. The answer should be in the exact format including both the option letter and the full text of that option. For example, <answer>A: Paris</answer>.
You must put all reasoning, re-checking, and correction inside <think>...</think>.
You must output exactly one final answer tag: <answer>...</answer>.
The <answer>...</answer> tag must appear after </think>.
Do not output anything after </answer>.
If you revise your answer, revise it inside <think>, then output only the final answer

```
once.
Question:
[{question}]
```

A.2 Licenses of Artifacts

We use the following pre-trained models and datasets in this work: Qwen2.5-3B-Instruct and Qwen3-4B-Instruct (Team et al., 2024; Yang et al., 2025) (Apache 2.0 license), GPT-5.4-nano (accessed via the OpenAI API under OpenAI's terms of use), MedQA (Jin et al., 2021) and MedXpertQA (Zuo et al., 2025) (released under [check license]). Our use of these artifacts is consistent with their intended research purposes and respective licenses. We will release our code and trained models under the Apache 2.0 license to facilitate reproducibility and future research.

A.3 Data Statement and PII Considerations

Datasets Used. Both datasets are derived from standardized medical licensing and specialty board examinations, and consist of multiple-choice questions designed to evaluate professional medical knowledge and clinical reasoning rather than to record real patient encounters.

Personally Identifying Information. To the best of our knowledge, neither dataset contains personally identifying information (PII) of real individuals. MedQA is constructed from publicly released examination materials (USMLE, MCMLE, and TWMLE), in which patient descriptions are hypothetical clinical vignettes authored by examiners for pedagogical purposes; they do not correspond to identifiable patients. MedXpertQA is similarly built from examination-style questions and further applies data synthesis and expert review to mitigate data leakage and ensure that questions reflect generic clinical scenarios rather than specific patient records. Consistent with the original dataset documentation, the authors of MedXpertQA explicitly state that the dataset "consists of medical examination questions rather than real patient data" and is not intended to contain personally identifiable information.

Offensive Content. Both datasets focus on clinical reasoning and medical knowledge, and we did not encounter offensive content during our manual inspection of randomly sampled subsets.

A.4 Implementation Details.

We use the `rouge-score` Python package for computing ROUGE-L recall. 2-gram recall is computed via standard whitespace tokenization after lowercasing. For training, we use the `transformers` library and `verl` for reinforcement learning. Inference is performed with `vLLM`. Specific package versions are documented in our released code.