ASK WITHOUT TELLING: LOCAL SLMS CONSULT CLOUD LLMS WITHOUT REVEALING TASK INTENT

Anonymous authors
Paper under double-blind review

ABSTRACT

As local small language models (SLMs) increasingly collaborate with more capable cloud large language models (LLMs), a natural privacy question arises: *Can a local SLM obtain cloud LLM guidance while protecting user privacy?* Existing privacy-preserving SLM-LLM frameworks primarily hide *sensitive values* while preserving task semantics, which can still expose what the user is trying to accomplish. For example, allocating scarce medical supplies across hospitals may signal an emerging public-health emergency, while rebalancing an investment portfolio may reveal a private investment strategy, even when names and numerical values are hidden. Recent decoy-based methods further obscure task intent by hiding the real request among alternatives, but stronger protection relies on more decoys or semantic abstraction, increasing overhead or risking utility loss. More fundamentally, existing work does not systematically characterize the components of private task intent or how each should be protected. We therefore introduce *task-private consultation*, which characterizes task intent through two components: *task context* and *task operation*. To the best of our knowledge, this is the first systematic study of these components and their individual and joint protection in local-cloud SLM-LLM consultation. To realize this setting, we propose `PriCon`, an end-to-end framework that transforms the task itself through recoverable mathematical reformulation rather than hiding it among alternatives. A local closed-loop refinement mechanism further maintains privacy and recoverability throughout consultation. Experiments on 100 tasks show that `PriCon` reduces cloud-side task-intent inference Hit@1 to nearly 0%, versus 93–99% under sensitive-value removal and 3–30% under decoy-based protection, while preserving cloud-assisted utility.

1 INTRODUCTION

Imagine a local small language model (SLM) assistant deployed by a regional public-health authority. Following a sudden increase in infectious-disease cases, the authority must allocate a limited stock of critical medical supplies across hospitals. Different hospitals face different supply demands and urgency levels, while the total regional stock is limited. The local SLM may not have sufficient reasoning capability to reliably coordinate the allocation across all hospitals, and therefore turns to a more capable cloud-hosted large language model (LLM) for assistance.

Before consulting the cloud, the SLM removes or replaces sensitive values, including hospital names, priority weights, inventory levels, and demand values. The new request appears sanitized:

> ‘‘Given **hospitals** [H] with different priority levels [W], multiple types of **medical supplies** [S] with limited available stocks [B], and hospital-specific **supply demands** [D], **allocate the available supplies** across hospitals to minimize the overall **impact of medical-supply shortages**.’’

No hospital name, exact demand, inventory level, or priority value is revealed. Yet the request still tells a revealing story. It exposes both the *context*: a public-health authority distributing scarce medical resources across hospitals, and the *operation*: a constrained allocation problem prioritizing hospitals while minimizing shortage impact. From this alone, an observer may infer unusual pressure on regional medical resources, potentially signaling an emerging public-health emergency.

Such task-intent leakage extends well beyond public-health settings. For example, rescheduling a production line may reveal an upcoming product launch or a capacity bottleneck; restructuring a small business's debt may signal severe cash-flow pressure; and rebalancing an investment portfolio may expose a confidential investment strategy. In each case, hiding names, identifiers, and numerical values is not enough. *What the user aims to accomplish may itself be private*.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

This risk is particularly relevant to the emerging paradigm of local SLM-cloud LLM collaboration, where local models work together with more capable cloud models to solve tasks that cannot be reliably completed on-device alone. Recent systems therefore split responsibilities between local SLMs that handle private data and local execution, and stronger cloud LLMs that provide reasoning and planning guidance for complex tasks (Zhang et al., 2024; Narayan et al., 2025; Ni et al., 2026). However, obtaining such guidance still requires the local SLM to send task-related information to the cloud, making every consultation message visible to the cloud provider.

Existing privacy-aware collaboration primarily protects *sensitive values* by removing, replacing, or obscuring private information before cloud consultation (Siyan et al., 2025; Yuan et al., 2026a; Zhan et al., 2026). However, as the medical-supply example illustrates, preserving task semantics for cloud assistance can still expose the application context, requested operation, and underlying objective. Recent decoy-based methods further obscure user intent by hiding the true request among plausible alternatives (Huang et al., 2026a; Zhou et al., 2026). Yet the true request remains cloud-visible, while stronger obfuscation requires more decoys or semantic abstraction, increasing inference overhead or potentially compromising task utility. More fundamentally, existing approaches do not systematically characterize the components of private task intent or how each should be protected.

This gap motivates a fundamental question:

Can a local SLM consult a cloud LLM without revealing the user's private task intent?

We take a more structured view and refer to this setting as **task-private consultation**. We view task intent as containing two components: *task context*, which captures the real-world setting from which the task arises, and *task operation*, which captures the computation or decision the user wishes to perform. Either component may be private. Importantly, the two are not independent sources of leakage: contextual details can themselves reveal the underlying operation.

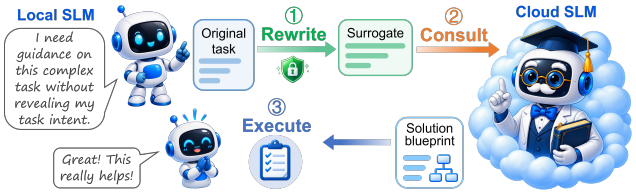


Fig. 1: Task-private consultation.

As illustrated in Fig. 1, task-private consultation follows a *rewrite, consult, and execute* workflow: *1) Rewrite*: the local SLM transforms the original task into an intent-obscuring surrogate; *2) Consult*: the cloud LLM reasons only over the surrogate and returns a transferable solution blueprint; *3) Execute*: the local SLM maps the blueprint back to the original task and executes the resulting solution locally. If further assistance is needed, local difficulties are converted into surrogate-level diagnostic requests and the consultation is repeated without exposing the original task or local execution details. We formalize this workflow, its protection scopes, and the cloud-visible transcript in Section 3.

Realizing this workflow presents two central challenges:

- *How can the local SLM construct an intent-obscuring surrogate?* The surrogate must hide the protected intent while preserving the solution-relevant structure needed for cloud guidance. Insufficient transformation leaves the intent recognizable, whereas excessive transformation may weaken the structure needed for useful guidance.
- *How can privacy and recoverability be maintained throughout the consultation?* A safe initial rewrite is insufficient, as local execution failures may trigger follow-up requests that reveal additional task-specific information. Each consultation round must therefore preserve both privacy and local recoverability.

To address these challenges, we propose `PriCon`, an end-to-end framework for task-private consultation. Instead of hiding the true task among alternatives, `PriCon` transforms the task itself through recoverable mathematical reformulation. For context protection, it extracts a domain-neutral mathematical representation that preserves solution-relevant structure and then uses semantic rewriting to recast it as a different application story. For operation or joint protection, it additionally changes the apparent computation through an alternative mathematical formulation before semantic rewriting. A local review and *closed-loop refinement* mechanism further preserves privacy and recoverability throughout the consultation while reusing successful experience across related tasks.

In summary, our main contributions are as follows:

- **Task-private consultation.** We systematically characterize task intent through *task context* and *task operation*, reveal their coupled leakage, and define context, operation, and joint protection over the complete cloud-visible transcript.
- **PriCon framework.** We propose PriCon, which transforms the original task into a recoverable, intent-obscuring surrogate through domain-neutral mathematical reformulation and semantic rewriting, while keeping the recovery mapping local. To the best of our knowledge, PriCon is the first local-cloud LLM consultation framework to use recoverable mathematical reformulation for task-intent protection without relying on additional decoy queries.
- **Transcript-level intent evaluation.** We evaluate task-intent inference by an honest-but-curious cloud LLM from the complete cloud-visible transcript. Experiments on 100 tasks show that PriCon strongly protects task intent while preserving the utility of cloud LLM assistance.

2 Related Work

We organize prior work around the evolving goals of local-cloud language-model collaboration, from efficient workload sharing to privacy-aware delegation and task-intent protection.

Local-cloud language-model collaboration. Existing local SLM-cloud LLM systems divide work according to query difficulty, reasoning complexity, data locality, or system cost. Hybrid LLM (Ding et al., 2024) routes queries between local SLMs and cloud LLMs according to predicted difficulty and desired quality, while AdaSwitch (Sun et al., 2024) lets a local SLM handle simpler reasoning steps and escalates harder ones to the cloud LLM. Minions (Narayan et al., 2025) keeps long contexts local while using the cloud LLM to decompose tasks into simpler subtasks for local execution. CE-CoLLM (Jin & Wu, 2025) and TMO (Yuan et al., 2026b) further optimize local-cloud execution for latency, communication, and inference cost. Overall, these approaches primarily treat local-cloud collaboration as a capability- and efficiency-allocation problem. As such collaboration involves user-side context and intermediate task information, privacy becomes an important concern.

Privacy-preserving SLM-LLM collaboration. Recent privacy-aware frameworks use local SLMs to limit private information exposed to the cloud, either by keeping private context local or sanitizing transmitted information. In its sketch-based variant, CoGenesis (Zhang et al., 2024) keeps personalized context local and sends only a general task instruction to the cloud LLM, which returns a sketch for local generation. Need to Know (Huang et al., 2026b) suppresses task-unnecessary private information while preserving task-essential information for cloud reasoning. Other approaches sanitize private information before transmission. PAPILLON (Siyan et al., 2025) rewrites private queries into privacy-preserving prompts, PAAC (Yuan et al., 2026a) replaces sensitive values with typed placeholders, and PRISM (Zhan et al., 2026) perturbs sensitive entities using local differential privacy. Despite their different mechanisms, these approaches primarily reduce the exposure of private information while preserving the task semantics needed for cloud assistance. Such semantic preservation can still leave the task context and requested operation recognizable, allowing the cloud to infer private activities or objectives even when sensitive values are protected.

Task-intent protection. Recent work has begun to protect task intent through decoy-based obfuscation. PrivShield (Huang et al., 2026a) hides a real query among adversarial queries, while GS-Chaff (Zhou et al., 2026) combines query-adaptive semantic abstraction with plausible decoy queries. Despite their different designs, both rely on ambiguity, with a query representing the real request remaining in a cloud-visible candidate set alongside decoys. Achieving stronger obfuscation may require more decoys or semantic abstraction, increasing inference overhead or potentially compromising task utility. In contrast, PriCon replaces the original task with a recoverable surrogate through mathematical reformulation, without sending the original task or decoys to the cloud.

3 Task-Private Consultation

We now formalize task-private consultation by specifying the consultation setting, the information to be protected, and the threat model considered in this work.

3.1 Consultation Setting and Workflow

We consider a trusted local device hosting an SLM and a more capable cloud-hosted LLM. The local SLM receives an original task $\mathcal{T}$, sensitive values $\mathcal{V}$, and access to a local execution environment.

Here, $\mathcal{V}$ denotes task-specific sensitive inputs, such as names, exact locations, identifiers, and measurements, that are required to instantiate the task.

We focus on tasks that exceed the SLM's end-to-end reasoning capability but can be reformulated into a recoverable surrogate whose solution can be instantiated locally. For such tasks, the SLM consults the cloud without transmitting the original task. As illustrated in Fig. 1, task-private consultation follows a *rewrite, consult, and execute* workflow:

$$\underbrace{\mathcal{T}}_{\text{Original Task}} \xrightarrow[\mathcal{M}]{\text{(1) SLM: Rewrite}} \underbrace{\widetilde{\mathcal{T}}}_{\text{Intent-Obscuring Surrogate}} \xrightarrow{\text{(2) LLM: Consult}} \underbrace{\mathcal{B}}_{\text{Solution Blueprint}} \xrightarrow[\mathcal{M},\mathcal{V}]{\text{(3) SLM: Execute}} \underbrace{\widehat{\mathcal{Y}}}_{\text{Local Solution}} .$$

1) **Rewrite.** The local SLM transforms the original task $\mathcal{T}$ into an intent-obscuring surrogate $\widetilde{\mathcal{T}}$ while replacing sensitive values $\mathcal{V}$ with placeholders. It also establishes a local correspondence $\mathcal{M}$ recording how solution-relevant elements of the surrogate relate to the original task.
2) **Consult.** The cloud LLM receives only $\widetilde{\mathcal{T}}$ and returns a transferable solution blueprint $\mathcal{B}$, such as an algorithm, decision procedure, program skeleton, or reasoning steps. The blueprint is expressed entirely in terms of the surrogate because the cloud does not observe $\mathcal{T}$, $\mathcal{M}$, or $\mathcal{V}$.
3) **Execute.** The local SLM uses the correspondence $\mathcal{M}$ to map the cloud-provided blueprint $\mathcal{B}$ back to the original task and instantiates it with the sensitive values $\mathcal{V}$. The resulting solution $\widehat{\mathcal{Y}}$ is verified against the original task requirements and executed locally. If further assistance is needed during local interpretation or execution, the SLM constructs a surrogate-level diagnostic request and repeats the consultation without exposing the original task or execution trace.

A consultation may contain an initial request followed by multiple diagnostic rounds. We define the *cloud-visible transcript* as the complete sequence of surrogate requests and cloud responses exchanged throughout the interaction. Crucially, task-intent privacy is defined over this complete transcript rather than over any single sanitized request. The original task, sensitive values, correspondence, and task-specific execution traces are excluded from this transcript and remain on the local device.

3.2 Task Intent and Protection Scope

Beyond protecting sensitive values $\mathcal{V}$, task-private consultation aims to protect the task intent revealed through cloud interaction. We use the medical-supply allocation example from Section 1 as a running example to formalize this notion. We characterize task intent through two components:

1) **Task context** $\mathcal{C}$ captures the application domain, entities, and situational background. In our example, it corresponds to a public-health setting involving hospitals, medical-supply demands, and scarce regional supplies.
2) **Task operation** $\mathcal{O}$ captures the computation or decision requested by the user, including its objective and major constraints. Here, it corresponds to allocating limited medical supplies across hospitals to minimize priority-weighted shortage impact.

The user or application policy may designate task context $\mathcal{C}$, task operation $\mathcal{O}$, or both as the privacy target. Accordingly, we consider three protection scopes:

- **Context protection** protects task context $\mathcal{C}$ without requiring task operation $\mathcal{O}$ itself to be hidden. In the public-health scenario, the authority may wish to conceal the underlying medical emergency while still allowing the cloud to recognize that it is solving a generic resource-allocation problem.
- **Operation protection** protects task operation $\mathcal{O}$, while task context $\mathcal{C}$ need not itself be private. For the medical-supply allocation example, if the public-health situation is already known, the privacy target may instead be how scarce supplies are prioritized and allocated across hospitals.
- **Joint protection** protects both $\mathcal{C}$ and $\mathcal{O}$. This applies when revealing either the underlying healthcare situation or the resource-allocation strategy would disclose private information.

3.3 Threat Model

Trusted local device. We assume that the local SLM and all local components, including storage, rewriting, review, and execution procedures, behave as intended. The original task $\mathcal{T}$, sensitive values $\mathcal{V}$, local correspondence $\mathcal{M}$, and local execution results remain under local control. Compromise of the local device is outside the scope of this work.

Honest-but-curious cloud. The cloud follows the consultation protocol and provides guidance, but may store and jointly analyze the complete cloud-visible transcript to infer the protected task intent. Under this threat model, task-private consultation aims to make the protected task intent difficult to infer from the cloud-visible transcript.

4 PRICON: REALIZING TASK-PRIVATE CONSULTATION

We now present PriCon, our end-to-end framework for realizing task-private consultation. It centers on two local mechanisms: *intent-obscuring rewriting*, which constructs the cloud-visible surrogate, and *closed-loop refinement*, which maintains privacy and recoverability throughout consultation.

4.1 INTENT-OBSCURING REWRITING

As illustrated in Fig. 2, PriCon selects a rewriting path according to the protection scope defined in Section 3.2. A natural first step is to abstract the task into a domain-neutral mathematical representation. Although this removes much of the original semantics, the resulting representation may still contain recognizable structural cues about the original task intent, as reflected by the *Math-only* results in Table 2. Rather than bypassing this step, PriCon retains the mathematical representation as a local intermediate, since it provides the *experience bank* in Section 4.2 with a shared structure for identifying similar tasks across domains and reusing successful experience. It then applies semantic rewriting to recast this representation as a different application story before cloud consultation. For context protection, PriCon follows this mathematical-to-semantic path; for operation or joint protection, it additionally introduces an *alternative mathematical formulation* to change the apparent computation before semantic rewriting. Throughout the process, the correspondence $\mathcal{M}$ remains local and maps cloud guidance back to the original task. We illustrate these rewriting paths using our running medical-supply example.

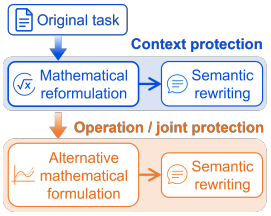


Fig. 2: Rewriting paths.

4.1.1 CONTEXT PROTECTION

We first consider context protection, where the goal is to hide the original task context while leaving the underlying task operation $\mathcal{O}$ unchanged.

Mathematical reformulation. PriCon first expresses the task in a domain-neutral mathematical form, replacing domain-specific entities with abstract roles while preserving their relations and underlying operation. Applying this abstraction to our running medical-supply example yields the following intermediate representation:

> ‘‘Given **demand groups** [G] with different priority weights [W], multiple **resource types** [R] with limited capacities [B], and group-specific resource requirements [D], **allocate available resources** across demand groups to minimize the overall **impact of unmet resource requirements**.’’

Hospitals are abstracted as demand groups and medical supplies as resource types, while sensitive values remain local in $\mathcal{V}$. This abstraction removes most healthcare-specific cues and substantially reduces task-intent inference, but does not eliminate it. The remaining concepts, such as demand groups, limited capacities, and unmet requirements, still reveal a recognizable resource-allocation operation, leaving the cloud LLM with a few clues about the task context. PriCon therefore treats the mathematical form as an intermediate representation rather than the final cloud-facing request.

Semantic rewriting. PriCon further embeds the mathematical structure in a coherent but semantically distant application story. Intuitively, this alternative story gives the cloud LLM a coherent surrogate-domain interpretation, making it less likely to associate the request with original task. For example, the generic resource-allocation problem can be rewritten as GPU allocation in a data-center setting:

> ‘‘Given **model-training jobs** [J] with different priority levels [W], multiple types of **GPU resources** [G] with limited available capacities [B], and job-specific compute demands [D], **allocate the available GPU resources** across training jobs to minimize the overall **impact of unmet compute demand**.’’

Here, demand groups and resource types become model-training jobs and GPU resources. The cloud now sees a GPU-allocation task rather than the original healthcare task, while the relationships needed for local recovery remain available through the correspondence $\mathcal{M}$.

4.1.2 Operation Protection and Joint Protection

Protecting task operation is more challenging because the surrounding context may itself reveal the protected operation. For example, hospitals, medical-supply demands, and limited stocks already suggest a resource-allocation problem even if the operation is not explicitly stated. `PriCon` therefore adopts a conservative strategy: whenever task operation $\mathcal{O}$ is protected, it also transforms contextual cues that could help infer $\mathcal{O}$. Operation protection and joint protection consequently follow the same rewriting path, although their requested protection scopes remain conceptually distinct.

Alternative mathematical formulation. Starting from the mathematical representation, `PriCon` constructs an alternative formulation with a different apparent computation and a recoverable solution mapping. In our example, the generic resource-allocation problem is recast as shortest-path search on layered directed acyclic graphs (DAGs):

> ``Given a set of **layered directed graphs** [G], each with **states** [V], **feasible transitions** [E], and **transition costs** [C], find a **minimum-cost path** from the source to a terminal node in each graph.''

The correspondence between the original and alternative formulations can be made explicit. Since supply types are independent in both the objective and stock constraints, the task can be decomposed into separate allocation subproblems, one for each supply type. For supply type s, the subproblem is:

$$\underbrace{\min_{\{x_h^s\}} \sum_{h=1}^{H} w_h \cdot \ell_h^s(d_h^s - x_h^s)}_{\text{minimize shortage impact}}, \quad \text{s.t.} \quad \underbrace{0 \le x_h^s \le d_h^s,\ \ x_h^s \in \mathbb{Z}_{\ge 0},\ \forall h}_{\text{allocation within demand, discrete units}}; \quad \underbrace{\sum_{h=1}^{H} x_h^s \le B^s}_{\text{stock constraint}}. \tag{1}$$

Here, d_h^s and x_h^s denote the demand and allocation of supply s at hospital h, w_h is its priority weight, B^s is the available stock, and the function ℓ_h^s measures shortage impact.

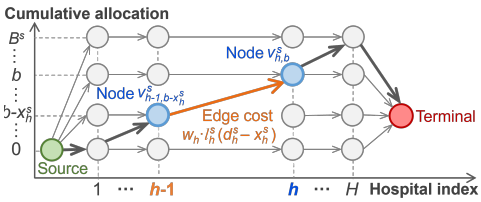


Fig. 3: Layered-DAG reformulation.

As illustrated in Fig. 3, each supply-specific subproblem is represented by a layered DAG. For supply type s, layer h corresponds to hospital h, while node $v_{h,b}^s$ represents a cumulative allocation of b units after the first h hospitals. An edge from $v_{h-1,b-x_h^s}^s$ to $v_{h,b}^s$ represents assigning x_h^s units to hospital h and incurs cost $w_h \cdot \ell_h^s(d_h^s - x_h^s)$. Thus, each source-to-terminal path corresponds to a feasible allocation, and the minimum-cost path yields the optimal solution to formulation (1). Repeating this construction across supply types yields the complete allocation $\{x_h^s\}$. The transformation changes the apparent operation from resource allocation to layered-DAG shortest-path search while preserving a reliable mapping back to the original solution. Compared with the initial mathematical reformulation, this alternative formulation further obscures task intent, but its mathematical structure may still leave some clues, as reflected by the *Math-only* results in Table 2.

Semantic rewriting. `PriCon` therefore recasts the path-search structure as a network-routing story before cloud consultation:

> ``Given multiple **layered communication networks** [N], each with **network states** [V], **available links** [E], and associated **transmission costs** [C], find a **minimum-cost route through each network** from its source gateway to a destination gateway.''

Under this interpretation, graph states, transitions, and costs become network states, communication links, and transmission costs. The cloud therefore reasons over a network-routing task rather than the original medical-supply allocation task, further obscuring its context and operation.

Appendix A provides detailed local correspondences $\mathcal{M}$ and representative cloud outputs for the medical-supply example. Appendix B further presents an interesting `PriCon` case transforming logic-grid reasoning into cooking-procedure planning.

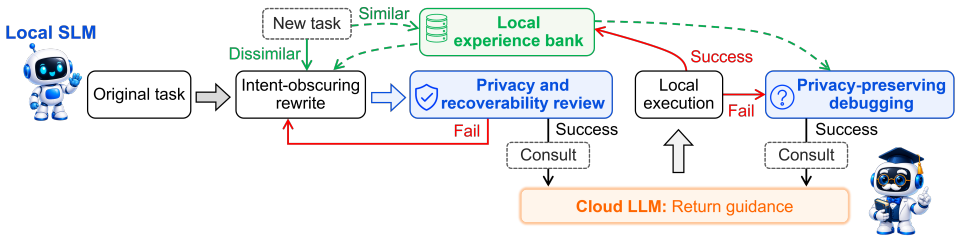


Fig. 4: Closed-loop consultation and refinement in PriCon.

4.2 Local Review and Closed-Loop Refinement

Stronger transformations can better obscure task intent but make cloud guidance harder to map back for local execution. Preserving more of the original structure improves recoverability but may retain intent-revealing cues. As shown in Fig. 4, PriCon therefore forms a closed refinement loop: failed reviews trigger further rewriting, execution failures trigger privacy-preserving debugging, and successful executions are stored in a local experience bank.

1) **Privacy and recoverability review.** Before transmission, a local reviewer checks whether sensitive values $\mathcal{V}$ are exposed, whether the protected intent is inferable from semantic or structural cues, and whether $\mathcal{M}$ can map cloud guidance back to the original task. If any check fails, the SLM revises the surrogate and repeats the review within a refinement budget.
2) **Privacy-preserving debugging.** Cloud guidance may still be too abstract or complex for the local SLM to execute reliably. Instead of returning task-specific failures to the cloud, the SLM abstracts them into surrogate-level diagnostic requests and subjects them to the same privacy and recoverability review. For example, it may ask the cloud to decompose a difficult surrogate reasoning step without revealing the original task or local execution trace.
3) **Local experience bank.** Successful consultations are stored locally as reusable experiences, including the mathematical representation, optional alternative formulation, semantic rewrite, correspondence $\mathcal{M}$, and validated refinement or debugging trajectories. When a new task arrives, PriCon retrieves prior experiences by comparing their mathematical structures rather than rewritten stories, using four fields: problem type, decision variables, objective, and constraints. Prior experiences are ranked by embedding similarity across these fields; see Appendix C for details. If a sufficiently similar experience is found, it is used to initialize surrogate construction, reducing LLM consultation rounds and token usage; otherwise, PriCon constructs a new surrogate and stores its validated trajectory as a new experience.

5 Experimental Results

5.1 Experimental Setup

Models and tasks. We evaluate PriCon with three models serving as local-side SLMs: GPT-4o-mini, DeepSeek-V4-Flash, and Qwen3.5-Flash, each paired with DeepSeek-V4 Pro as the cloud LLM (OpenAI, 2024; DeepSeek-AI, 2026; Qwen Team, 2026). Our benchmark contains 10 task families spanning structured and long-context reasoning, decision and optimization, coding, and cross-domain composite tasks, as summarized in Table 1. We retain only tasks that local SLMs cannot reliably solve end-to-end, thus requiring cloud LLM guidance; see Appendix D for the screening procedure. After screening, we retain 10 diverse instances per family, yielding *100 tasks in total*. All retained task instances are available in our GitHub repository.[1]

Comparison settings. We compare direct cloud consultation, sensitive-value removal, a decoy-based obfuscation baseline (Huang et al., 2026a), and PriCon. For PriCon, we evaluate the complete semantic rewriting process under context and joint protection. We additionally report a *Math-only* ablation that stops before semantic rewriting, isolating its benefit. Operation-only protection is evaluated separately to study context-operation coupling. For the decoy baseline, we use 5 and 20 decoys to examine its privacy-overhead trade-off.

Metrics. As each benchmark task may contain multiple subtasks, we evaluate privacy and utility at the subtask level. For privacy, we ask the cloud LLM to infer three targets from the complete

[1] https://github.com/guantian111/Pricon_tasks

Table 1: Task families in our evaluation benchmark.

Category	Task families
Structured and long-context reasoning	Alibi verification, logic grid, counterfactual reasoning, recursive expansion, relation graph
Decision and optimization	Financial modeling, budget allocation, office setup
Coding	Python coding
Cross-domain composite	Composite cross-domain

Table 2: Task-intent inference by the cloud LLM under different consultation settings and local SLMs. Entries report task-averaged Hit@1 / Hit@5 (%) over 100 benchmark tasks.

Local SLM	Inference Target	Direct cloud consultation	Sensitive-value removal	Decoy		PriCon-Context		PriCon-Joint	
				5	20	Math-only	**Semantic**	Math-only	**Semantic**
GPT-4o-mini	Context	94 / 98	94 / 98	10 / 32	6 / 21	1 / 3	**0 / 0**	3 / 10	**0 / 0**
	Operation	98 / 100	99 / 100	30 / 65	18 / 50	49 / 63	31 / 42	1 / 5	**0 / 0**
	Joint	95 / 98	94 / 98	7 / 31	3 / 13	0 / 0	**0 / 0**	0 / 1	**0 / 0**
DeepSeek-V4-Flash	Context	98 / 100	96 / 99	9 / 34	8 / 20	3 / 3	**0 / 0**	1 / 4	**0 / 0**
	Operation	95 / 99	95 / 99	26 / 63	30 / 68	48 / 62	59 / 75	1 / 4	**0 / 0**
	Joint	98 / 100	96 / 99	6 / 30	4 / 15	0 / 1	**0 / 0**	0 / 2	**0 / 0**
Qwen3.5-Flash	Context	95 / 97	93 / 99	10 / 38	3 / 18	1 / 3	**0 / 0**	2 / 6	**0 / 0**
	Operation	99 / 100	96 / 98	29 / 66	18 / 46	48 / 64	56 / 69	2 / 7	**0 / 2**
	Joint	96 / 97	93 / 99	7 / 30	4 / 14	0 / 0	**0 / 0**	0 / 2	**0 / 0**

Table 3: Cloud-assisted utility under different consultation settings. Entries report task-averaged subtask success rate (%) over 100 benchmark tasks.

Local SLM	Direct cloud consultation	Sensitive-value removal	Decoy		PriCon-Context		PriCon-Joint	
			5	20	Math-only	**Semantic**	Math-only	**Semantic**
GPT-4o-mini	100	100	100	100	100	100	100	100
DeepSeek-V4-Flash	100	100	100	100	100	100	100	100
Qwen3.5-Flash	100	100	100	100	100	100	100	100

cloud-visible transcript: task context, task operation, and their pair (joint inference). For each task, *Hit@1* and *Hit@5* measure the proportions of its subtasks whose ground-truth targets appear among the cloud LLM's top-1 and top-5 guesses, respectively; joint inference requires the correct context-operation pair. For utility, we compute each task's subtask success rate as the proportion of its subtasks yielding functionally correct final solutions, with direct cloud consultation as the reference. In the main results, we average these per-task privacy and utility scores across all 100 benchmark tasks. We additionally report LLM token usage and consultation rounds as system overhead.

5.2 Main Results

Task-intent protection. We first evaluate task-intent protection across the three local SLMs, with the local experience bank disabled to isolate the effect of rewriting. Table 2 compares the cloud LLM's task-intent inference under different consultation settings, with detailed category-wise results in Appendix E. Direct cloud consultation yields Hit@1 scores of 94–99%, while sensitive-value removal yields similarly high scores of 93–99%, showing that hiding sensitive values alone provides little protection against task-intent inference. The decoy baseline reduces inference, but operation Hit@1 remains 18–30% even with 20 decoys. In contrast, `PriCon-Context` with semantic rewriting reduces context Hit@1 to 0% across all three local SLMs. Under joint protection, `PriCon-Joint` with semantic rewriting achieves 0% Hit@1 for all three inference targets, while Hit@5 remains at 0% in nearly all cases.

Effect of semantic rewriting. The *Math-only* ablation already substantially reduces inference of the protected intent, showing that the mathematical stages provide strong protection. Under context protection, its context Hit@1 is 1–3%, whereas semantic rewriting reduces it to 0% across all three local SLMs. Under joint protection, some inference remains with *Math-only*, whereas semantic rewriting reduces Hit@1 to 0% across all inference targets. These results support using mathematical reformulation as an intermediate stage and semantic rewriting as the final cloud-facing form.

Utility. Table 3 shows that the complete `PriCon` with semantic rewriting achieves 100% subtask success across all three local SLMs, matching direct cloud consultation. Thus, task-intent protection does not reduce cloud-assisted utility on our benchmark.

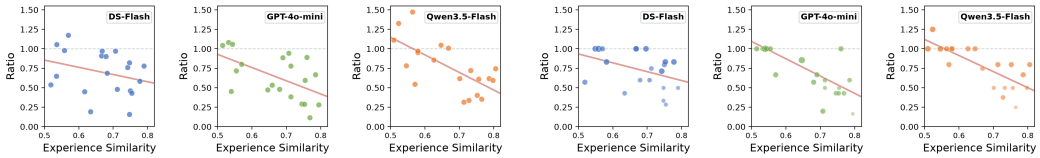
Consultation overhead. Direct cloud consultation and sensitive-value removal each require only one round: the former sends the task directly, while the latter simply replaces sensitive values with placeholders. However, direct consultation sends all task-specific inputs to the cloud LLM, so its token cost can grow substantially with input size. `PriCon` introduces additional rewriting and refinement, requiring 5.6–7.2 consultation rounds and 1.96–2.52M LLM tokens with semantic rewriting. Nevertheless, this overhead remains far below that of the 20-decoy setting, which requires

Table 4: Average consultation overhead across 100 benchmark tasks. LLM token usage is reported in millions (M = 10^6).

Local SLM	Metric	Direct cloud consultation	Sensitive-value removal	Decoy		PriCon-Context		PriCon-Joint	
				5	20	Math-only	**Semantic**	Math-only	**Semantic**
GPT-4o-mini	LLM token usage	0.49 M	0.34 M	2.48 M	11.4 M	2.29 M	1.96 M	2.39 M	2.52 M
	Consultation rounds	1	1	7.8	35.9	7.2	5.6	7.5	7.2
DeepSeek-V4-Flash	LLM token usage	0.65 M	0.35 M	2.86 M	12.1 M	2.29 M	2.03 M	2.39 M	2.31 M
	Consultation rounds	1	1	9.0	38.1	7.2	7.1	5.8	6.6
Qwen3.5-Flash	LLM token usage	0.54 M	0.39 M	2.86 M	12.7 M	2.51 M	2.0 M	2.60 M	2.52 M
	Consultation rounds	1	1	9.0	39.8	7.9	5.7	7.7	7.2

Table 5: Context-operation coupling under operation-only and joint protection. Entries report Hit@1 / Hit@5 (%).

Local SLM	Inference Target	Sensitive-value removal	Operation-only		PriCon-Joint	
			Math-only	Semantic	Math-only	**Semantic**
GPT-4o-mini	Operation	99 / 100	3 / 10	3 / 6	1 / 5	**0 / 0**
	Joint	94 / 98	2 / 5	1 / 5	0 / 1	**0 / 0**
DeepSeek-V4-Flash	Operation	95 / 99	1 / 7	2 / 4	1 / 4	**0 / 0**
	Joint	96 / 99	1 / 4	1 / 4	0 / 2	**0 / 0**
Qwen3.5-Flash	Operation	96 / 98	3 / 9	2 / 7	2 / 7	**0 / 2**
	Joint	93 / 99	0 / 4	0 / 2	0 / 2	**0 / 0**



(a) LLM token usage ratio. (b) Consultation round ratio.

Fig. 5: Effect of experience similarity on consultation overhead under PriCon-Joint with semantic rewriting. Each ratio is computed as the overhead with experience reuse divided by that without experience reuse.

35.9–39.8 rounds and 11.4–12.7M tokens. Together with Table 2, these results show that PriCon achieves stronger task-intent protection at much lower overhead than heavy decoy-based obfuscation.

Context-operation coupling. Although task context and operation are distinct privacy targets, contextual cues may still reveal the protected operation. With operation-only protection, operation Hit@1 remains at 2–3% when the original context is retained, whereas joint protection reduces it to 0% across all local SLMs. These results show that task context can act as a side channel for operation inference, supporting PriCon's strategy of transforming context whenever the operation is protected.

Experience reuse. We finally evaluate experience reuse on held-out tasks. Figure 5 shows that both LLM token usage and consultation rounds generally decrease as experience similarity increases. At low similarity (below approximately 0.55), retrieved experiences may introduce irrelevant information and increase token usage. For consultation rounds, some tasks also show little reduction because their original overhead is already low. For more demanding cases, reuse can reduce consultation rounds to approximately 20% of the baseline. These observations suggest using a similarity threshold around 0.6 to avoid ineffective reuse; for less similar tasks, PriCon constructs surrogates independently and stores them as new experiences for future reuse.

6 Discussion

Applicability boundary. PriCon is most suitable when the task context can be separated from the underlying task operation needed for cloud reasoning. It is less applicable to knowledge queries whose answers are inherently tied to the task context, such as the effects or use of a particular medication. Such queries may instead be handled by a local knowledge base without cloud consultation. Operation protection is also challenging for complex tasks, where finding an alternative formulation with a reliable solution mapping can be difficult. A richer experience bank may expand this boundary.

A moving capability frontier. As local SLMs become more capable, more tasks will be completed entirely on-device. Task-private consultation will therefore shift toward harder reasoning, planning, and decision tasks that still benefit from cloud guidance. Advances in local SLMs and cloud LLMs may also enable more complex transformations, further extending this frontier.

Beyond language models. This work focuses on text-based SLM-LLM consultation, while task intent may also be revealed through images, audio, and other modalities. Extending PriCon to multimodal models may require modality-specific transformations of objects, scenes, or spatial relations while preserving information needed for reasoning. Extending task-private consultation beyond text is an important direction for multimodal local-cloud collaboration.

AI USE STATEMENT

In this work, we used generative AI tools for literature retrieval and discovery (e.g., identifying potentially relevant related work), checking mathematical formulations and notation, identifying potential inconsistencies in technical arguments, and providing feedback on methodology and experimental design. We also used generative AI tools for language polishing, grammar correction, and improving the clarity and readability of the manuscript. All retrieved literature, AI-assisted mathematical formulations, and technical suggestions were independently reviewed and verified by the authors. The core research problem, methodology, experiments, and experimental results were developed and conducted by the authors. We take responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We take several steps to support reproducibility. (i) Appendix A provides detailed rewriting examples across different protection settings, including the local correspondences, representative cloud LLM guidance, and local execution procedures. (ii) Appendix C specifies the representation, similarity computation, and retrieval procedure used by the local experience bank. (iii) Section 5 reports the model and experimental setup, while Appendix D details the task-screening procedure; the complete benchmark task instances are additionally provided in our anonymous GitHub repository. Together, these materials document the key implementation and evaluation details needed to reproduce our experiments.

REFERENCES

DeepSeek-AI. DeepSeek-V4 Preview: Entering the Era of Affordable Million-Token Context, April 2026. Official release announcement.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. Hybrid LLM: Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations (ICLR)*, 2024.

Xinyu Huang, Leming Shen, Zijing Ma, and Yuanqing Zheng. Toward privacy-preserving and personalized smart homes via tailored small language models. *IEEE Transactions on Mobile Computing*, 25(7):9505–9520, 2026a.

Xinyue Huang, Xiaochun Cao, and Wenyuan Yang. Need to know: Contextual-integrity-grounded query rewriting for privacy-conscious LLM delegation. *arXiv preprint arXiv:2606.04067*, 2026b.

Hongpeng Jin and Yanzhao Wu. CE-CoLLM: Efficient and adaptive large language models through cloud-edge collaboration. In *IEEE International Conference on Web Services (ICWS)*, pp. 316–323, 2025.

Avanika Narayan, Dan Biderman, Sabri Eyuboglu, Avner May, Scott Linderman, James Zou, and Christopher Re. Cost-efficient collaboration between on-device and cloud language models. In *International Conference on Machine Learning (ICML)*, pp. 45682–45719, 2025.

Xiaoqi Ni, Jie Wang, Lin Yang, Yiyang Lu, Hanzhu Chen, Rui Liu, and Jianye Hao. Following the navigation: Enhancing small language models contextual reasoning with LLM guidance. In *International Conference on Learning Representations (ICLR)*, pp. 145028–145051, 2026.

OpenAI. GPT-4o mini: Advancing Cost-Efficient Intelligence, July 2024. Official release announcement.

Qwen Team. Qwen3.5: Towards Native Multimodal Agents, February 2026. Official release announcement.

Li Siyan, Vethavikashini Chithrra Raghuram, Omar Khattab, Julia Hirschberg, and Zhou Yu. PAPILLON: Privacy preservation from Internet-based and local language model ensembles. In *Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 3371–3390, 2025.

Hao Sun, Jiayi Wu, Hengyi Cai, Xiaochi Wei, Yue Feng, Bo Wang, Shuaiqiang Wang, Yan Zhang, and Dawei Yin. AdaSwitch: Adaptive switching between small and large agents for effective cloud-local collaborative learning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8052–8062, 2024.

Liangqi Yuan, Wenzhi Fang, Shiqiang Wang, and Christopher G. Brinton. PAAC: Privacy-aware agentic device-cloud collaboration. *arXiv preprint arXiv:2605.08646*, 2026a.

Liangqi Yuan, Dong-Jun Han, Shiqiang Wang, and Christopher G. Brinton. Device-cloud collaborative llm inference with multi-modal, multi-task, and multi-turn conversations. *IEEE Transactions on Networking*, 34:6124–6139, 2026b.

Junfei Zhan, Haoxun Shen, Zheng Lin, and Tengjiao He. PRISM: Privacy-aware routing for adaptive cloud–edge llm inference via semantic sketch collaboration. In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 28150–28158, 2026.

Kaiyan Zhang, Jianyu Wang, Ermo Hua, Biqing Qi, Ning Ding, and Bowen Zhou. CoGenesis: A framework collaborating large and small language models for secure context-aware instruction following. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 4295–4312, 2024.

Quan Zhou, Zhicheng Wang, Zhe Yue, Libin Cai, Kuien Liu, and Caiyan Qin. Gs-chaff: Multi-agent prompt-level semantic chaffing for privacy-preserving llm inference. *Sensors*, 26(17):5385, 2026.