BEYOND SURFACE SEMANTICS: LOGIC-GUIDED PRIVACY-PRESERVING CLOUD-EDGE LLM REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) excel at complex reasoning, but their large resource demands make them difficult to deploy on edge devices. Thus, many users rely on APIs to access cloud-side LLMs for complex reasoning tasks, raising privacy concerns when sending queries to the cloud. Existing methods protect user queries on the edge before cloud reasoning. Recent work further explores cloud-edge collaboration, where the cloud-side LLM provides guidance and the edge-side small language model (SLM) completes reasoning with private data. However, to balance utility and privacy, these methods mainly preserve surface semantics and often fail to retain the logical structure needed for solving the query, making cloud guidance unstable. In this paper, we propose **LoGic**, a logic-guided analogical reasoning framework that shifts the focus from preserving surface semantics to preserving reasoning-critical logical structure under privacy constraints. This enables the cloud-side LLM to provide more effective guidance for improving edge-side SLM reasoning. Specifically, LoGic decouples each private query on the edge side into a logic skeleton and an entity mapping, applying sensitivity-aware differentiated LDP protection and a privacy-aware GRPO strategy to preserve useful logical structure under privacy guarantees. The cloud-side LLM then generates a structure-consistent query from the protected structure and derives its step-by-step solution, serving as a query-specific analogical example for the edge side. Finally, the edge-side SLM transfers the cloud-side LLM's reasoning pattern from the structurally similar query to solve the original query. Experiments on complex reasoning tasks show that LoGic delivers stronger reasoning performance with lower leakage than existing baselines. Code is available at: `https://anonymous.4open.science/r/LoGic-46DC`.

1 INTRODUCTION

Recent large language models (LLMs), such as GPT-4 (Achiam et al., 2023) and Gemini 3 (Pichai et al., 2025), exhibit strong capabilities in complex reasoning tasks like mathematical problem solving and multi-step logical deduction. Despite this, real-world user interactions often occur on edge-side devices (e.g., cellphones). Constrained by limited computation and memory, these devices cannot meet the massive resource demands of LLMs. Consequently, they are typically restricted to deploying small language models (SLMs), which struggle to deliver high-quality solutions for complex reasoning tasks independently. A common solution is to access cloud-side LLMs via API calls (Yu et al., 2024). However, directly sending user queries to cloud servers may expose sensitive information, raising serious privacy concerns in high-risk domains such as healthcare and finance (Huang et al., 2023). Therefore, it is essential to protect user queries before cloud reasoning while retaining sufficient information for effective reasoning.

To balance privacy protection and reasoning quality, most existing studies follow an edge-protected cloud reasoning paradigm: the edge device anonymizes the user query (Yang et al., 2025) or applies differential privacy (DP) perturbation (Chen et al., 2023), after which a cloud-side LLM directly reasons over the protected query and returns the final answer. Although traditional token-level DP perturbation provides formal privacy guarantees (Dwork et al., 2014), it often severely damages contextual semantics and leaves the cloud-side LLM with corrupted inputs (Wu et al., 2025),

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.

substantially reducing the stability and accuracy of downstream reasoning. Recent DP-based methods, such as DYNTEXT (Zhang et al., 2025a) and CAPE (Wu et al., 2025), introduce sensitivity-aware or context-aware mechanisms to mitigate this issue. By adapting perturbation strength or incorporating contextual semantics, they better preserve the topic information and surface semantics, thereby improving cloud-side reasoning performance. However, complex reasoning requires precise variables and constraint relations, which these methods still struggle to preserve. As a result, edge-protected cloud reasoning remains insufficient for high-quality complex reasoning.

This limitation motivates a shift toward privacy-preserving cloud-edge collaborative reasoning (Jin & Wu, 2025; Zhang et al., 2025c). In this paradigm, the edge device protects the user query, the cloud-side LLM generates reasoning guidance from the protected input, and the edge-side SLM combines this guidance with private information to produce the final answer (Tong et al., 2025). This shifts the cloud from final answer generation to reasoning assistance, while the edge remains responsible for final reasoning. The goal is to leverage the cloud-side LLM's reasoning ability while using the edge-side SLM's access to private data to compensate for information loss from privacy protection (Zhan et al., 2026). However, existing cloud-edge collaboration frameworks still face a key weakness in complex reasoning settings. The guidance generated by the cloud-side LLM is often too unreliable to consistently improve the reasoning capability of the edge-side SLM (Zhang et al., 2026). This is because the cloud-side LLM receives protected inputs, such as high-level descriptions (Zhang et al., 2024), DP-perturbed text (Chen et al., 2023), or other sanitized query forms (Zambare et al., 2026), that mainly preserve surface semantics but fail to retain the logical structure required to solve the user query. Without such logical constraints, the cloud-side LLM may produce unstable guidance with logical gaps or factual deviations. Consequently, cloud guidance may fail to effectively enhance the edge-side SLM's reasoning capability and may even disrupt its solving process.

This challenge raises an important unresolved question in privacy-preserving cloud-edge collaboration: how can we provide the cloud-side LLM with an input that supports stable reasoning without exposing sensitive information, so that it can deliver reliable guidance to the edge-side SLM? Our key observation is that complex reasoning depends primarily on the logical structure that determines the solution process, rather than on the surface semantics of the query. If the cloud-side LLM can access a logical structure consistent with the original user query, it can generate accurate reasoning guidance even without private context. The edge-side SLM then only needs to execute this guidance using the true private information, instead of bearing the full burden of complex logical planning.

In this paper, we propose **LoGic**, a logic-guided analogical reasoning framework for privacy-preserving cloud-edge collaboration on complex reasoning tasks. Unlike existing cloud-edge methods that mainly preserve surface semantics, LoGic shifts the design focus toward preserving reasoning-critical logical structure under privacy constraints, while redesigning the information interface between the edge and cloud. The edge transmits a protected logic-entity structure rather than corrupted surface text, enabling the cloud-side LLM to generate a structure-consistent analogical example that enhances the edge-side SLM's complex reasoning ability. Specifically, LoGic consists of two key stages. (1) To provide the cloud-side LLM with reasoning-critical logical information under privacy guarantees, we decouple each private query on the edge side into a logic skeleton and an entity mapping, and apply sensitivity-aware differentiated LDP protection to adapt protection strength to their different privacy-utility roles. We further optimize the extractor with a privacy-aware dual-reward GRPO strategy, which jointly rewards structural quality and downstream utility on the protected representation to preserve its usefulness after perturbation. (2) We introduce structure-consistent analogical reasoning to help the edge-side SLM reuse the cloud-side LLM's reasoning pattern on the structurally similar query. The cloud-side LLM first generates a query-specific analogical example, including a logically consistent analogical query and its step-by-step solution. The edge-side SLM then transfers this reasoning pattern to the original query for final reasoning.

Our contributions are: (1) We propose LoGic, a privacy-preserving cloud-edge collaboration framework that redesigns the edge-cloud interface around protected logic-entity structure, improving edge-side SLM reasoning under privacy constraints. (2) We shift the preservation focus from surface semantics toward logical structure under privacy constraints. By combining differentiated LDP with dual-reward GRPO, LoGic preserves reasoning-critical structure while rigorously protecting private information. (3) We introduce structure-consistent analogical reasoning, which turns cloud-side reasoning ability into reusable scaffolds for edge-side execution. (4) Experiments across complex reasoning benchmarks show that LoGic achieves state-of-the-art (SOTA) privacy-utility trade-offs.

2 Related Work

2.1 Edge-side Privacy Protection for LLM Reasoning

Existing privacy-preserving LLM reasoning methods mainly protect user queries before LLM reasoning, and can be broadly grouped into three categories. (1) Entity sanitization methods, such as anonymization (Yang et al., 2025; Kim et al., 2025; Han et al., 2025) and de-identification (Liu et al., 2023; Aghakasiri et al., 2025; Zambare et al., 2026), remove sensitive entities but rely on accurate entity detection and may discard important context. (2) Cryptography-based solutions (Wu et al., 2024; Hou et al., 2026), such as secure multi-party computation (MPC) (Yao, 1986; Zhang et al., 2025b; Yuan et al., 2026) and homomorphic encryption (HE) (Gentry, 2009; Yu et al., 2026), provide strong security and near-lossless inference but incur high overhead. (3) LDP-based methods offer a more efficient alternative with formal privacy guarantees. SanText (Yue et al., 2021) protects the full vocabulary with metric-LDP (Feyisetan et al., 2020; Qu et al., 2021), while CusText (Chen et al., 2023) improves utility by sampling from truncated neighbor lists at the cost of weaker protection. Recent methods further refine the privacy-utility trade-off (Liu & Zhao, 2026). DYNTEXT (Zhang et al., 2025a) adaptively adjusts perturbation strength based on token sensitivity, and CAPE (Wu et al., 2025) introduces context-aware protection with a bucketing mechanism. However, these methods mainly preserve surface semantics and often fail to retain the key constraint relations required for complex reasoning, making them unreliable for cloud-side reasoning.

2.2 Privacy-Preserving Cloud-Edge Collaborative Reasoning

Recent studies explore cloud-edge collaboration to combine the reasoning strength of cloud-side LLMs with the privacy advantage of edge-side SLMs (Jin & Wu, 2025). (1) Typically, the cloud directly answers the protected query, while the edge SLM refines this response using private information (Li et al., 2025; Luo et al., 2026; Gu et al., 2026). SnD (Mai et al., 2024) perturbs the query with LDP and denoises the cloud's response on the edge, while InferDPT (Tong et al., 2025) prompts the edge-side SLM to answer the original query by loosely referencing the cloud response. Since the cloud still reasons over corrupted inputs, its direct answers often suffer from severe factual deviations and hallucinations. (2) To mitigate this, more recent works (Niu et al., 2026; Zhan et al., 2025) instruct the cloud to provide reusable guidance, such as semantic sketches or high-level plans, rather than final answers. CoGenesis (Zhang et al., 2024) uses cloud-generated sketches to improve edge-side SLM reasoning. Ding et al. (2024) leverage historical cloud-side LLM interactions, while PRISM (Zhan et al., 2026) generates semantic sketches under adaptive two-layer LDP. Zhang et al. (2026) further let the cloud produce executable reasoning code, but this is mainly limited to numerical or programmatic tasks. Despite these advances, existing methods overlook the logical structure needed for solving the query, making cloud guidance weakly aligned with the original problem. In contrast, we treat logical structure as the core cloud-edge interface, enabling reliable reasoning.

3 Preliminaries

Local Differential Privacy (LDP). Differential Privacy (DP) (Dwork, 2006; Dwork et al., 2014) is widely regarded as the gold standard for data privacy. LDP (Duchi et al., 2013) is a special case of DP in which the server is untrusted, and data privatization is performed on the edge side. Given a privacy budget $\varepsilon \geq 0$, a randomized mechanism M is said to satisfy ε-LDP if for all pairs of adjacent inputs $x, x' \in \mathcal{X}$ and every possible output $y \in \mathcal{Y}$, the following condition holds:

$$Pr[M(x) = y] \leq e^{\varepsilon} Pr[M(x') = y]. \tag{1}$$

In the context of NLP, following CusText (Chen et al., 2023), we consider any pair of input tokens that share the same output set $\mathcal{Y}$ to be adjacent to each other.

Exponential Mechanism (EM). EM (McSherry & Talwar, 2007) is commonly used for DP item selection from discrete domains. Given a privacy budget $\varepsilon \geq 0$, a randomized Exponential mechanism $M(\cdot)$ is ε-LDP compliant if, for any input $x \in X$ and possible output $y \in Y$, it satisfies:

$$\Pr[M(x) = y] \propto \exp(\frac{\varepsilon \cdot u(x, y)}{2\Delta u}), \tag{2}$$

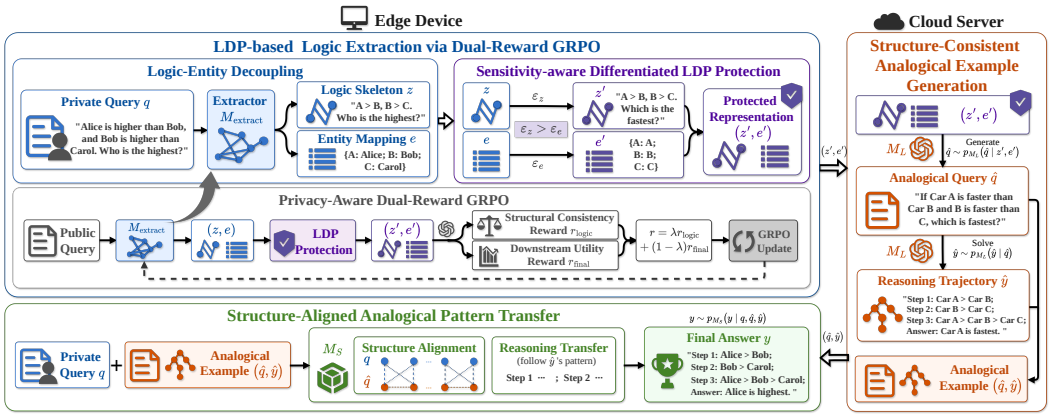

Figure 1: Overview of **LoGic**. LoGic first uses a dual-reward GRPO extractor to decouple each private query into a logic-entity representation, and protects it with differentiated LDP on the edge device. The cloud-side LLM then generates a structure-consistent analogical query and its reasoning trajectory. Finally, the edge-side SLM transfers the reasoning pattern to solve the original query.

where $u(x, y)$ denotes the score of output token y for input token x, the sensitivity Δu is defined as $\Delta u = \max_{x,x' \in X, y \in Y} |u(x, y) - u(x', y)|$. The sampling probability depends only on $u(x, y)$, with larger utility yielding higher probability since ε and Δu are fixed.

4 METHOD

4.1 OVERVIEW

We propose **LoGic** (as shown in Fig. 1) to improve the reasoning capability of edge-side SLMs under formal privacy guarantees, thereby balancing privacy protection and reasoning quality. LoGic consists of two core stages. (1) **LDP-based Logic Extraction via Dual-Reward GRPO (§4.2)** shifts the preservation focus from surface semantics toward logical structure on the edge side. It employs a logic extraction model to transform each private query into a logic-entity representation and applies sensitivity-aware differentiated LDP perturbation. The extractor is optimized with a privacy-aware dual-reward GRPO strategy. (2) **Structure-Consistent Analogical Reasoning (§4.3)** first performs structure-consistent analogical example generation on the cloud side, guiding the LLM to generate an analogical example that includes an analogical query and its step-by-step solution. On the edge side, the SLM then performs structure-aligned analogical pattern transfer by applying the example's reasoning pattern to the private query and producing the final solution. Overall, LoGic follows a paradigm of edge-side privacy protection and cloud-edge collaborative reasoning. The edge device first extracts and protects the logic-entity representation of the private query (§4.2). The cloud-side LLM then generates a structure-consistent analogical reasoning example, and the edge-side SLM transfers its reasoning pattern for local execution (§4.3).

4.2 LDP-BASED LOGIC EXTRACTION VIA DUAL-REWARD GRPO

To provide the cloud-side LLM with reasoning-critical logical structure under privacy guarantees, we decouple the private query on the edge side into a logic skeleton and an entity mapping, and apply sensitivity-aware differentiated LDP protection to adapt protection strength to different privacy-utility roles (§4.2.1). We further use a privacy-aware dual-reward GRPO strategy to improve utility (§4.2.2).

This design is motivated by the limitation of existing privacy-preserving methods, which mainly operate on surface text and aim to preserve semantics after perturbation. However, for complex reasoning tasks, the logical structure that governs the solution process is often more important than the surface semantics. If the cloud-side LLM can access a logical structure consistent with the original query, it can generate accurate reasoning guidance even without private context. Moreover, the abstract logical structure tends to expose less sensitive information.

4.2.1 LOGIC-ENTITY DECOUPLING WITH DIFFERENTIATED LDP PROTECTION

We decouple each private query into a logic-entity representation, and then apply a sensitivity-aware differentiated LDP protection mechanism to obtain a protected representation for transmission.

Logic-Entity Decoupling. To preserve the reasoning structure while reducing privacy exposure, we decouple a private query into two disjoint components: a logical skeleton that determines the reasoning process, and an entity mapping that instantiates this skeleton into a specific problem. Formally, given a private query q, the extractor M_{extract} learns a mapping $f_\theta : q \mapsto (z, e)$, where z is the logic skeleton and e is the entity mapping, with $z \cap e = \emptyset$. This logic-entity representation separates reasoning-critical structure from privacy-sensitive information: z captures variable relations and constraint patterns, while e records mappings from abstract variables to concrete entities. Specifically, we represent z using structured templates with placeholders, such as masked natural language forms or simple pseudo-code expressions (e.g., "If $[\text{VAR}_1] > [\text{VAR}_2]$, then ...", or "$[\text{VAR}_1] \rightarrow [\text{VAR}_2] \rightarrow$ conclusion"). These placeholders represent abstract variables involved in the reasoning process while removing instance-specific details. e is instantiated as a key-value dictionary that binds each abstract variable to its corresponding concrete entity in the original query (e.g., $\{[\text{VAR}_1]$: "Alice's salary", $[\text{VAR}_2]$: "Bob's salary"$\}$). We provide the prompt and example in App. C.

Sensitivity-aware Differentiated LDP Protection. To better balance privacy and downstream utility, we apply sensitivity-aware differentiated LDP perturbation to the logic skeleton z and entity mapping e, producing the protected representation (z', e') for cloud-side reasoning. This is motivated by their different privacy-utility roles. The logic skeleton mainly encodes abstract variable relations and constraint patterns, which are crucial for cloud-side reasoning and usually contain less sensitive information. In contrast, the entity mapping contains concrete instance-specific semantics and therefore requires stronger protection. We thus assign different privacy budgets $\varepsilon_z > \varepsilon_e$, allowing the logical structure to be better preserved while applying stronger perturbation to private entities.

We implement the perturbation using the exponential mechanism $M(\cdot)$ with a CusText-style customized mapping (Chen et al., 2023). Specifically, we first construct a mapping function that partitions the token vocabulary into disjoint groups. Tokens in the same group $\mathcal{X}_m$ share the same output set $\mathcal{Y}_m$, which consists of semantically close candidate tokens, and are treated as adjacent tokens. We design a scoring function $u(\cdot)$ for $M(\cdot)$ that assigns higher scores to candidate tokens semantically closer to the input token, thereby increasing their probability of being sampled. Given a token $x \in \mathcal{X}_m$ and a candidate token $y \in \mathcal{Y}_m$, we first compute their Euclidean distance $d(x, y)$ in the embedding space to measure semantic distance. The scoring function is then defined as: $u_m(x, y) = 1 - \frac{d(x,y)}{d_m^{\max}}$, where $d_m^{\max} = \max_{x \in \mathcal{X}_m, y \in \mathcal{Y}_m} d(x, y)$. Since $0 \leq d(x, y) \leq d_m^{\max}$, we have $u_m(x, y) \in [0, 1]$ and the sensitivity satisfies $\Delta u_m \leq 1$. Given a privacy budget ε, $M(\cdot)$ samples a candidate token $y \in \mathcal{Y}_m$ according to the exponential mechanism:

$$Pr[M(x) = y] = \text{softmax}(\frac{\varepsilon \cdot u_m(x, y)}{2\Delta u_m}) = \frac{\exp(\frac{\varepsilon \cdot u_m(x,y)}{2\Delta u_m})}{\sum_{y' \in \mathcal{Y}_m} \exp(\frac{\varepsilon \cdot u_m(x,y')}{2\Delta u_m})}, \tag{3}$$

where larger ε favors semantically similar tokens for utility, while smaller ε increases randomness for stronger privacy. Based on the different privacy-utility roles of z and e, we apply Eq. 3 to the two components with separate budgets: ε_z for the logic skeleton and ε_e for the entity mapping. Under the CusText-style customized adjacency relation, where tokens sharing the same output set are adjacent, perturbing tokens in z satisfies ε_z-LDP and perturbing tokens in e satisfies ε_e-LDP. Since z and e are disjoint components of the private query and are perturbed independently, the joint release (z', e') satisfies $\max(\varepsilon_z, \varepsilon_e)$-LDP at the token level by the parallel composition (Dwork et al., 2014). As $\varepsilon_z > \varepsilon_e$, the overall privacy budget is ε_z. The proof of this LDP guarantee is given in App. B.

4.2.2 PRIVACY-AWARE DUAL-REWARD GRPO STRATEGY

To make the edge-side extraction model M_{extract} produce a stable and useful logic-entity representation, we introduce a privacy-aware dual-reward GRPO strategy. Standard supervised fine-tuning (SFT) can teach the model the basic format of logic extraction, but it does not ensure that the extracted skeleton is useful for downstream reasoning, especially after LDP perturbation. Therefore, we first perform supervised fine-tuning on M_{extract} for cold start, and then apply GRPO with a task-oriented dual-reward objective that balances logical structure quality and downstream reasoning performance.

Standard GRPO rewards sampled outputs according to their relative quality within a group (Shao et al., 2024). Applying it directly to logic extraction can optimize the clean output (z, e), while the cloud observes only the perturbed representation (z', e') at deployment. To eliminate this gap, we make GRPO privacy-aware by evaluating rewards on (z', e') after differentiated LDP protection. This aligns optimization with the actual cloud-side input and encourages representations that remain useful after perturbation. Specifically, given an original query q with ground-truth answer g, let y denote the output of M_{extract}, from which we extract the logic skeleton and entity mapping (z, e) and obtain the protected representation (z', e'). We design two complementary reward signals to evaluate its quality.

a. **Structural consistency reward** $r_{\textbf{logic}}$ measures whether the protected logic skeleton z' preserves the key constraints, variable relations, and solving goal of the original query. We use an LLM-as-a-judge evaluator that takes q and z' as input, judges whether they express the same core constraint relations, and normalizes the score to $[0, 1]$.

b. **Downstream utility reward** $r_{\textbf{final}}$ measures whether the protected representation supports downstream reasoning. Specifically, we feed z' into the cloud model M_L and combine it with the protected entity mapping e' to obtain the final answer $a = M_L(z', e')$. We treat this as a proxy for the reasoning utility of (z', e'). If $a = g$, we set $r_{\text{final}}(y) = 1$; otherwise, we set it to 0.

The total reward is defined as: $r(y) = \lambda r_{\text{logic}}(y) + (1 - \lambda) r_{\text{final}}(y)$, where $\lambda \in [0, 1]$ controls the trade-off between structural consistency and downstream performance. We then use these rewards in the GRPO objective (details provided in App. A) to update M_{extract}. This aligns training with the deployment-time cloud input and encourages extractions that remain structurally consistent and useful after LDP perturbation.

4.3 Structure-Consistent Analogical Reasoning

We use analogical reasoning to transfer the cloud-side LLM's reasoning pattern on structurally similar queries to the edge-side SLM, improving reasoning performance on the edge side. This design is inspired by analogical reasoning in human cognition (Cuccio, 2026; Webb et al., 2023), where unfamiliar problems are solved by mapping relational structures from a known case to a new one. The key idea of analogy is not surface similarity between entities, but the preservation of the underlying relations that determine how a solution should proceed. In our setting, the structurally similar query serves as the known case and the private query as the new case, allowing the edge-side SLM to derive the solution from the known case's reasoning pattern instead of open-ended reasoning from scratch.

4.3.1 Cloud-Side Structure-Consistent Analogical Example Generation

To obtain an example structurally similar to the private query, we guide the cloud-side LLM to generate a structure-consistent query from the protected representation and derive its step-by-step solution, serving as a query-specific analogical example for edge-side execution.

Specifically, given the protected representation (z', e') uploaded by the edge device, our goal is to construct a structurally consistent analogical query $\hat{q}$ and its corresponding reasoning trajectory $\hat{y}$, so that they can serve as a reusable analogical example for the edge-side SLM. We use the protected logic skeleton z' as the core structural constraint and the protected entity mapping e' as auxiliary domain information, prompting the cloud-side LLM to generate a natural-language analogical query $\hat{q}$ that is structurally consistent but re-instantiated in content:

$$\hat{q} \sim p_{M_L}(\hat{q} | z', e'), \tag{4}$$

where p_{M_L} denotes the conditional generation distribution of the cloud-side LLM. Here, $\hat{q}$ is not a simple rewrite or sanitized version of the original query, but an isomorphic problem reconstructed in a privacy-safe content space. After generating $\hat{q}$, we further ask the cloud-side LLM to solve it with an existing reasoning prompting strategy, such as CoT (Wei et al., 2022), to obtain a complete reasoning trajectory and final answer:

$$\hat{y} \sim p_{M_L}(\hat{y} | \hat{q}). \tag{5}$$

The resulting $\hat{y}$ provides the final answer and an explicit reasoning trajectory produced by the stronger cloud-side LLM. We return $(\hat{q}, \hat{y})$ to the edge device as a complete structure-consistent analogical example. Through this process, the cloud-side LLM converts its reasoning ability into a transferable reasoning pattern without accessing private data. We provide the prompt and example in App. D.

4.3.2 EDGE-SIDE STRUCTURE-ALIGNED ANALOGICAL PATTERN TRANSFER

Based on the structure-consistent analogical example, we guide the edge-side SLM to transfer the cloud-side LLM's reasoning pattern to the private query, thereby reliably reusing the cloud-side LLM's reasoning capability on the edge side.

Given the private query q and the analogical example $(\hat{q}, \hat{y})$ returned by the cloud, the edge-side SLM generates the final answer y based on their structural correspondence:

$$y \sim p_{M_S}(y|q, \hat{q}, \hat{y}), \tag{6}$$

where p_{M_S} denotes the conditional generation distribution of the edge-side SLM. Specifically, the edge-side model first aligns the shared logical structure between q and $\hat{q}$, and then follows the reasoning trajectory $\hat{y}$ to transfer its reasoning pattern to the private entity information. In this way, complex reasoning is transformed into a structure-constrained execution process. The edge-side SLM can more stably reuse the reasoning capability explicitly demonstrated by the cloud-side LLM on the analogical problem. This allows even a weaker edge-side SLM to solve complex problems that would otherwise exceed its capability, without exposing the true private entities or the original query to the cloud. We provide the prompt and example in App. E.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Datasets. We train the extraction model on GSM8K (Cobbe et al., 2021), a standard benchmark for multi-step mathematical reasoning. For evaluation, we consider two domains, mathematical reasoning and commonsense reasoning, using MathQA [1] and StrategyQA (Geva et al., 2021), respectively.

Models. We select two lightweight edge-side SLMs for resource-constrained settings, Llama3.2-1B-Instruct (Grattafiori et al., 2024) and Qwen2.5-1.5B-Instruct (Hui et al., 2024), and adopt GPT-4.1 (Achiam et al., 2023) as the cloud-side LLM.

Baselines. We compare `LoGic` with three categories of baselines: (1) *Non-private*, including `Cloud-Only` and `Edge-Only`; (2) *Cloud Reasoning*, including `CusText` (Chen et al., 2023) and `CAPE` (Wu et al., 2025); (3) *Cloud-edge Collaborative Reasoning*, including `InferDPT` (Tong et al., 2025), `PRISM` (Zhan et al., 2026), and `CollabLLM` (Zhang et al., 2026).

Evaluation Metrics. Following prior works (Zhan et al., 2026; Zhang et al., 2026), we evaluate each method using three metrics: (1) `Accuracy` (Acc); (2) `Reasoning quality` (RQ); (3) `Leakage` (Leak). More details about datasets, baselines, metrics, and implementation are in App. F.

5.2 OVERALL PERFORMANCE

Tab. 1 reports the results of all baselines across two reasoning datasets and two edge-side SLMs, where all LDP-based methods use $\varepsilon = 3$. Overall, our method `LoGic` achieves the best utility-privacy trade-off among privacy-preserving baselines. It improves accuracy and reasoning quality while keeping the lowest leakage. Notably, under formal privacy guarantees, `LoGic` substantially outperforms `Edge-Only`, enhancing the complex reasoning ability of edge-side SLMs and narrowing the gap to `Cloud-Only`. Specifically, (1) `Cloud-Only` provides the performance upper bound but incurs full leakage, as the cloud directly accesses the private query. (2) Cloud reasoning methods perform poorly because they mainly preserve coarse semantics while disrupting the logical structure required for reasoning. As a result, they cannot reliably support independent cloud-side reasoning, especially on structure-sensitive numerical tasks such as MathQA. (3) Existing cloud-edge collaborative methods also struggle to balance utility and privacy. `InferDPT` severely degrades reasoning utility, `PRISM` improves utility at the cost of higher leakage, and `CollabLLM` is mainly limited to numerical reasoning and lacks formal privacy guarantees. In contrast, our method `LoGic` consistently achieves stronger reasoning performance with low leakage. We further validate the robustness of `LoGic` across independent judge models and different cloud LLMs (see App. H).

Table 1: Comparison of all methods on mathematical and commonsense reasoning tasks under different edge-side SLMs ($\varepsilon = 3$ for LDP-based methods), evaluated using Accuracy, Reasoning quality, and Leakage metrics. The best results among privacy-preserving methods are highlighted in **bold**. ↑ indicates higher is better; ↓ indicates lower is better. "–" indicates the baseline is inapplicable. Our improvements are significant under the t-test with $p < 0.05$ (see details in App. I).

Methods		**MathQA** (Math Reasoning)			**StrategyQA** (Commonsense Reasoning)		
		Accuracy ↑	**Reasoning Q.** ↑	**Leakage** ↓	**Accuracy** ↑	**Reasoning Q.** ↑	**Leakage** ↓
Cloud-Only (upper bound)		0.913	0.885	1.000	0.910	0.928	1.000
Cloud Reasoning	CusText	0.303	0.228	0.132	0.635	0.444	0.207
	CAPE	0.298	0.264	0.079	0.376	0.134	0.127
Model I: Llama3.2-1B-Instruct							
Edge-Only		0.447	0.502	0.000	0.341	0.569	0.000
Cloud-edge Collaborative Reasoning	InferDPT	0.246	0.251	0.094	0.436	0.029	0.132
	PRISM	0.350	0.368	0.566	0.608	0.290	0.113
	CollabLLM	0.453	0.489	0.189	–	–	–
	LoGic (ours)	**0.474**	**0.577**	**0.059**	**0.689**	**0.652**	**0.077**
Model II: Qwen2.5-1.5B-Instruct							
Edge-Only		0.579	0.626	0.000	0.739	0.750	0.000
Cloud-edge Collaborative Reasoning	InferDPT	0.254	0.293	0.094	0.524	0.192	0.132
	PRISM	0.511	0.474	0.566	0.611	0.509	0.113
	CollabLLM	0.609	0.570	0.189	–	–	–
	LoGic (ours)	**0.661**	**0.631**	**0.059**	**0.835**	**0.802**	**0.077**

Table 2: Ablation results on LoGic ($\varepsilon = 3$). *w/o* indicates removal of a specific module or approach from the full method. Best results are in **bold**. ↑ indicates higher is better; ↓ indicates lower is better.

Methods	**MathQA** (Math Reasoning)					**StrategyQA** (Commonsense Reasoning)				
	Acc. ↑	**RQ** ↑	**Leak.** ↓	**Util.** Δ (%) ↓	**Leak.** Δ (%) ↓	**Acc.** ↑	**RQ** ↑	**Leak.** ↓	**Util.** Δ (%) ↓	**Leak.** Δ (%) ↓
Model I: Llama3.2-1B-Instruct										
LoGic (ours)	**0.474**	**0.577**	**0.059**	0.00	0.00	**0.689**	**0.652**	**0.077**	0.00	0.00
w/o Logic Extraction	0.388	0.429	0.097	21.90	64.41	0.485	0.574	0.126	20.79	63.64
w/o Decoupling	0.400	0.281	0.320	33.46	442.37	0.612	0.649	0.185	5.82	140.26
w/o Dual-Reward GRPO	0.310	0.570	0.136	17.91	130.51	0.544	0.582	0.092	15.89	19.48
w/o Analogical Examples	0.398	0.364	0.059	26.48	0.00	0.534	0.509	0.077	22.22	0.00
w/o Analogical Transfer	0.390	0.311	0.059	31.91	0.00	0.553	0.545	0.077	18.08	0.00
Model II: Qwen2.5-1.5B-Instruct										
LoGic (ours)	**0.661**	**0.631**	**0.059**	0.00	0.00	**0.835**	**0.802**	**0.077**	0.00	0.00
w/o Logic Extraction	0.563	0.592	0.087	10.50	47.46	0.767	0.795	0.097	4.51	25.97
w/o Decoupling	0.641	0.618	0.349	2.54	491.53	0.777	0.796	0.175	3.85	127.27
w/o Dual-Reward GRPO	0.534	0.569	0.185	14.52	213.56	0.699	0.761	0.097	10.70	25.97
w/o Analogical Examples	0.583	0.606	0.059	7.88	0.00	0.738	0.768	0.077	7.93	0.00
w/o Analogical Transfer	0.515	0.540	0.059	18.25	0.00	0.757	0.752	0.077	7.79	0.00

5.3 Ablation Studies

We conduct ablation studies to validate each component of LoGic, with results summarized in Tab. 2. We report Accuracy, Reasoning Quality, and Leakage, together with relative changes over the full LoGic, including Utility Drop and Leakage Increase (formulas in App. F). The ablation results show that the full LoGic consistently outperforms all variants. (1) **w/o Logic Extraction (§ 4.2)** bypasses the LDP-based dual-reward logic extraction module entirely, applying token-level LDP directly to the private query. This causes a large utility drop, empirically confirming that direct perturbation destroys the reasoning structure needed for complex tasks. (2) **w/o Decoupling (§ 4.2.1)** abstracts the query without separating the logic skeleton from entity information, leading to a sharp increase in leakage. This confirms that logic-entity decoupling is crucial for isolating private content. (3) **w/o Dual-Reward GRPO (§ 4.2.2)** trains the extraction model only with SFT instead of a privacy-aware dual-reward GRPO strategy. Its performance degradation indicates that SFT alone struggles to output representations robust enough for downstream reasoning under LDP noise. (4) **w/o Analogical Examples (§ 4.3.1)** replaces structure-consistent analogical examples with answer-oriented cloud drafts, resulting in lower performance. This suggests that analogical examples provide a more stable and reusable reasoning scaffold. (5) **w/o Analogical Transfer (§ 4.3.2)** removes

[1] https://huggingface.co/datasets/aledm03/math_qa

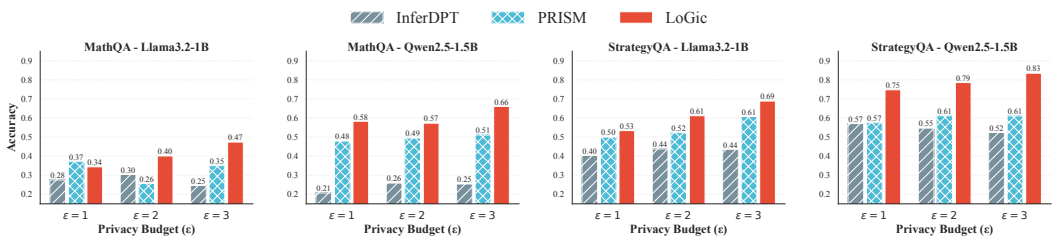

Figure 2: Accuracy across LDP budgets $\varepsilon \in \{1, 2, 3\}$, datasets, and edge-side SLMs.

the explicit analogical query and only returns reasoning trajectories, preventing structure-aligned transfer and increasing the burden on the edge-side SLM.

5.4 Analysis Studies

Analysis of Privacy-Utility Trade-off. To evaluate the downstream performance of the LDP-based privacy-preserving cloud-edge methods under different privacy budgets, we plot downstream accuracy across datasets and edge-side SLMs under strict privacy settings $\varepsilon \in \{1, 2, 3\}$, as shown in Fig. 2. We observe that: (1) `LoGic` achieves the best performance in most settings across all privacy budgets, demonstrating a strong privacy-utility trade-off. (2) `LoGic` shows a steady increase in accuracy on downstream tasks as ε increases, while other methods are less stable and sometimes even degrade, indicating that `LoGic` has stronger robustness. (3) `InferDPT` performs poorly on numerical reasoning tasks that require strong logical dependencies, indicating that methods focusing on surface semantics are insufficient for complex reasoning. In contrast, `LoGic` preserves logical structure more effectively, leading to better generalization and practical utility.

Table 3: Accuracy analysis across different scales of SLMs ($\varepsilon = 3$).

Dataset	Method	Models			
		Llama-3.2-1B-Instruct	Qwen2.5-1.5B-Instruct	Llama-3.2-3B-Instruct	Qwen2.5-3B-Instruct
MathQA	Edge-Only	0.447	0.579	0.666	0.822
	LoGic (ours)	**0.474**	**0.661**	**0.673**	**0.831**
StrategyQA	Edge-Only	0.341	0.739	0.815	0.820
	LoGic (ours)	**0.689**	**0.835**	**0.824**	**0.823**

Analysis of Different SLM Scales. To evaluate the effectiveness of `LoGic` across edge-side SLMs of different scales, we test four edge models on two datasets, including two Llama3.2 models (Grattafiori et al., 2024) and two Qwen2.5 models (Hui et al., 2024) with different parameter sizes. As shown in Tab. 3, `LoGic` consistently improves edge-side SLMs, with larger gains for smaller models such as Llama3.2-1B-Instruct and Qwen2.5-1.5B-Instruct. The below-0.5 accuracy of Edge-Only on StrategyQA with Llama3.2-1B-Instruct mainly results from zero-shot instruction-following failures. These results show that `LoGic` can remain effective under limited edge-side reasoning capability. As the edge model becomes stronger, the gain over Edge-Only diminishes, since the marginal benefit of cloud assistance decreases when local reasoning is already sufficient. This further highlights the applicability of `LoGic` to resource-constrained edge settings, where edge-side SLMs may struggle with complex reasoning while raw queries cannot be directly exposed to the cloud.

6 Conclusion

We proposed LoGic to improve edge-side SLM reasoning in privacy-preserving cloud-edge collaboration. LoGic shifts privacy protection from surface semantics to reasoning-critical logical structure on the edge side, constructing an LDP-protected logic-entity representation through LDP-based logic extraction with dual-reward GRPO. Based on this representation, the cloud-side LLM generates a structure-consistent analogical example, which serves as a transferable reasoning scaffold for edge-side execution. The edge-side SLM then reuses the reasoning pattern to solve the original private query. Experiments on mathematical and commonsense reasoning tasks show that LoGic achieves SOTA privacy-utility trade-offs.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing and grammar checking of the manuscript. We have not used generative AI tools for generating research ideas, designing experiments, collecting or analyzing data, writing or analyzing code, or producing any figures, and the remaining required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work. Specifically, all AI-polished text was manually checked and, where necessary, revised by the authors to ensure accuracy, clarity, and faithfulness to the original meaning, and no AI-generated content appears in the paper without author verification. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.

Kiana Aghakasiri, Noopur Zambare, JoAnn Thai, Carrie Ye, Mayur Mehta, J Ross Mitchell, and Mohamed Abdalla. Not what the doctor ordered: Surveying llm-based de-identification and quantifying clinical information loss. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pp. 32175–32191, 2025.

Maurice Stevenson Bartlett. Properties of sufficiency and statistical tests. Proceedings of the royal society of london. series a-mathematical and physical sciences, 160(901):268–282, 1937.

Sai Chen, Fengran Mo, Yanhao Wang, Cen Chen, Jian-Yun Nie, Chengyu Wang, and Jamie Cui. A customized text sanitization mechanism with differential privacy. In Findings of the Association for Computational Linguistics: ACL 2023, pp. 5747–5758, 2023.

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

Valentina Cuccio. Abduction and analogical reasoning in human cognition and metaphorical thought. Review of Philosophy and Psychology, pp. 1–16, 2026.

Yucheng Ding, Chaoyue Niu, Fan Wu, Shaojie Tang, Chengfei Lyu, and Guihai Chen. Enhancing on-device llm inference with historical cloud-based llm interactions. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 597–608, 2024.

John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In 2013 IEEE 54th annual symposium on foundations of computer science, pp. 429–438. IEEE, 2013.

Cynthia Dwork. Differential privacy. In International colloquium on automata, languages, and programming, pp. 1–12. Springer, 2006.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science, 9(3–4):211–407, 2014.

Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. Privacy-and utility-preserving textual analysis via calibrated multivariate perturbations. In Proceedings of the 13th international conference on web search and data mining, pp. 178–186, 2020.

Craig Gentry. A fully homomorphic encryption scheme. Stanford university, 2009.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. Transactions of the Association for Computational Linguistics (TACL), 2021.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. arXiv preprint arXiv:2407.21783, 2024.

Yujie Gu, Richeng Jin, Xiaoyu Ji, Yier Jin, and Wenyuan Xu. Differentially private and communication efficient large language model split inference via stochastic quantization and soft prompt. arXiv preprint arXiv:2602.11513, 2026.

Changzhou Han, Zehang Deng, Wanlun Ma, Xiaogang Zhu, Minhui Xue, Tianqing Zhu, Sheng Wen, and Yang Xiang. Codebreaker: Dynamic extraction attacks on code language models. In 2025 IEEE Symposium on Security and Privacy (SP), pp. 559–575. IEEE, 2025.

Florian Hartmann, Duc-Hieu Tran, Peter Kairouz, Victor Cărbune, and Blaise Aguera Y Arcas. Can llms get help from other llms without revealing private information? In Proceedings of the Fifth Workshop on Privacy in Natural Language Processing, pp. 107–122, 2024.

Xiaoyang Hou, Jian Liu, Jingyu Li, Yuhan Li, Jiawen Zhang, Wen-jie Lu, Cheng Hong, and Kui Ren. Ciphergpt: Secure two-party gpt inference. IEEE Transactions on Dependable and Secure Computing, 2026.

Ken Huang, Fan Zhang, Yale Li, Sean Wright, Vasan Kidambi, and Vishwas Manral. Security and privacy concerns in chatgpt. In Beyond AI: ChatGPT, Web3, and the business landscape of tomorrow, pp. 297–328. Springer, 2023.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. arXiv preprint arXiv:2409.12186, 2024.

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

Kyuyoung Kim, Hyunjun Jeon, and Jinwoo Shin. Self-refining language model anonymizers via adversarial distillation. In The Thirty-ninth Annual Conference on Neural Information Processing Systems, 2025.

Senyao Li, Haozhao Wang, Wenchao Xu, Rui Zhang, Song Guo, Jingling Yuan, Xian Zhong, Tianwei Zhang, and Ruixuan Li. Collaborative inference and learning between edge slms and cloud llms: A survey of algorithms, execution, and open challenges. arXiv preprint arXiv:2507.16731, 2025.

Chang Liu and Junzhou Zhao. Mtsp-ldp: A framework for multi-task streaming data publication under local differential privacy. Proceedings of the ACM on Management of Data, 4(1 (SIGMOD): 1–27, 2026.

Zhengliang Liu, Yue Huang, Xiaowei Yu, Lu Zhang, Zihao Wu, Chao Cao, Haixing Dai, Lin Zhao, Yiwei Li, Peng Shu, et al. Deid-gpt: Zero-shot medical text de-identification by gpt-4. arXiv preprint arXiv:2303.11032, 2023.

Mingshun Luo, Haolei He, Wenti Yang, Shuai Yuan, and Zhitao Guan. An efficient privacy-preserving transformer inference scheme for cloud-based intelligent decision-making in aiot. Journal of Systems Architecture, pp. 103687, 2026.

Peihua Mai, Ran Yan, Zhe Huang, Youjia Yang, and Yan Pang. Split-and-denoise: Protect large language model inference with local differential privacy. In Forty-first International Conference on Machine Learning, 2024.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07), pp. 94–103. IEEE, 2007.

Wujia Niu, Lan Zhang, Haoran Cheng, and Shen Li. Privsplit: A lossless method for prompt privacy in distributed parameter-efficient fine-tuning. In Proceedings of the ACM Web Conference 2026, pp. 2557–2568, 2026.

Sundar Pichai, Demis Hassabis, and Koray Kavukcuoglu. A new era of intelligence with gemini 3. Google. URL: https://blog. google/products-and-platforms/products/gemini/gemini-3/(: 16.01. 2026), 2025.

Chen Qu, Weize Kong, Liu Yang, Mingyang Zhang, Michael Bendersky, and Marc Najork. Natural language understanding with privacy-preserving bert. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 1488–1497, 2021.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024.

Meng Tong, Kejiang Chen, Jie Zhang, Yuang Qi, Weiming Zhang, Nenghai Yu, Tianwei Zhang, and Zhikun Zhang. Inferdpt: Privacy-preserving inference for black-box large language models. IEEE Transactions on Dependable and Secure Computing, 2025.

Taylor Webb, Keith J Holyoak, and Hongjing Lu. Emergent analogical reasoning in large language models. Nature Human Behaviour, 7(9):1526–1541, 2023.

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

Haoqi Wu, Wenjing Fang, Yancheng Zheng, Junming Ma, Jin Tan, Yinggui Wang, and Lei Wang. Ditto: Quantization-aware secure inference of transformers upon mpc. arXiv preprint arXiv:2405.05525, 2024.

Haoqi Wu, Wei Dai, Wang Li, and Qiang Yan. Cape: Context-aware prompt perturbation mechanism with differential privacy. In International Conference on Machine Learning, pp. 67184–67201, 2025.

Tianyu Yang, Xiaodan Zhu, and Iryna Gurevych. Robust utility-preserving text anonymization based on large language models. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 28922–28941, 2025.

Andrew Chi-Chih Yao. How to generate and exchange secrets. In 27th annual symposium on foundations of computer science (Sfcs 1986), pp. 162–167. IEEE, 1986.

Da Yu, Peter Kairouz, Sewoong Oh, and Zheng Xu. Privacy-preserving instructions for aligning large language models. arXiv preprint arXiv:2402.13659, 2024.

Ye Yu, Yifan Zhou, Yi Chen, Pedro Soto, Wenjie Xiong, and Meng Li. Cachemir: Fully homomorphic encrypted inference of generative large language model with kv cache. arXiv preprint arXiv:2602.11470, 2026.

Mu Yuan, Lan Zhang, Yihang Cheng, Miao-Hui Song, Guoliang Xing, and Xiang-Yang Li. Stip: Three-party privacy-preserving and lossless inference for large transformers in production. In Network and Distributed System Security (NDSS) Symposium, 2026.

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

Noopur Zambare, Kiana Aghakasiri, Carissa Lin, Carrie Ye, J Ross Mitchell, and Mohamed Abdalla. Towards fair and efficient de-identification: Quantifying the efficiency and generalizability of de-identification approaches. In Findings of the Association for Computational Linguistics: EACL 2026, pp. 4242–4257, 2026.

Huiyou Zhan, Xuan Zhang, Haisheng Tan, Han Tian, Dongping Yong, Junyang Zhang, and Xiang-Yang Li. Pice: A semantic-driven progressive inference system for llm serving in cloud-edge networks. arXiv preprint arXiv:2501.09367, 2025.

Junfei Zhan, Haoxun Shen, Zheng Lin, and Tengjiao He. Prism: Privacy-aware routing for adaptive cloud–edge llm inference via semantic sketch collaboration. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 40, pp. 28150–28158, 2026.

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

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 Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 4295–4312, 2024.

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

Wang Zhang, Yu Guo, Luoxun Liu, Yuxin Xi, Yue Wang, and Fangda Guo. Seclmm: Towards secure and lightweight inference service for large multimodal models. In 2025 IEEE/ACM 33rd International Symposium on Quality of Service (IWQoS), pp. 1–10. IEEE, 2025b.

Yingyi Zhang, Pengyue Jia, Xianneng Li, Derong Xu, Maolin Wang, Yichao Wang, Zhaocheng Du, Huifeng Guo, Yong Liu, Ruiming Tang, et al. Lsrp: A leader-subordinate retrieval framework for privacy-preserving cloud-device collaboration. In Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2, pp. 3889–3900, 2025c.

A Group Relative Policy Optimization Details

Group Relative Policy Optimization (GRPO) (Shao et al., 2024) optimizes a policy without an explicit value model by comparing multiple sampled outputs for the same input. For each query q, GRPO samples a group of G responses $\{o_i\}_{i=1}^{G}$ from the old policy $\pi_{\theta_{old}}$. Given their rewards $\{r_i\}_{i=1}^{G}$, the normalized group-relative advantage is

$$A_i = \frac{r_i - \text{mean}(\{r_i\}_{i=1}^{G})}{\text{std}(\{r_i\}_{i=1}^{G}) + v_{stab}}, \tag{7}$$

where v_{stab} is a small constant for numerical stability. Let $\rho_i = \frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}$ denote the importance ratio. The GRPO objective is

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}_{q\sim\mathcal{D}}\left[\frac{1}{G}\sum_{i=1}^{G}\left(\min\left(\rho_i A_i, \text{clip}(\rho_i, 1-v, 1+v)A_i\right) - \beta D_{KL}(\pi_\theta \| \pi_{ref})\right)\right], \tag{8}$$

where the clipping term stabilizes policy updates and the KL term prevents the updated policy from drifting too far from the reference policy.

B Proof of Privacy Guarantee

We formally prove that the protected representation (z', e') satisfies $\max(\varepsilon_z, \varepsilon_e)$-LDP at the token level under the CusText-style customized adjacency relation.

Following CusText (Chen et al., 2023), we first construct a mapping function that partitions the token vocabulary into disjoint input groups. Tokens in the same group $\mathcal{X}_m$ share the same output set $\mathcal{Y}_m$ and are treated as adjacent tokens. Therefore, the LDP guarantee is defined over any two tokens $x, x' \in \mathcal{X}_m$ that share the same output support $\mathcal{Y}_m$.

First, we establish that the exponential replacement mechanism $M(\cdot)$ defined in Eq. 3 satisfies ε-LDP within each token group. For any two adjacent input tokens $x, x' \in \mathcal{X}_m$ and any output token $y \in \mathcal{Y}_m$,

we expand the probability ratio:

$$\begin{aligned}\frac{\Pr[M(x)=y]}{\Pr[M(x')=y]} &= \frac{\exp\left(\frac{\varepsilon\cdot u_m(x,y)}{2\Delta u_m}\right)\Big/\sum_{\tilde{y}\in\mathcal{Y}_m}\exp\left(\frac{\varepsilon\cdot u_m(x,\tilde{y})}{2\Delta u_m}\right)}{\exp\left(\frac{\varepsilon\cdot u_m(x',y)}{2\Delta u_m}\right)\Big/\sum_{\tilde{y}\in\mathcal{Y}_m}\exp\left(\frac{\varepsilon\cdot u_m(x',\tilde{y})}{2\Delta u_m}\right)} \\ &= \exp\left(\frac{\varepsilon\left(u_m(x,y)-u_m(x',y)\right)}{2\Delta u_m}\right)\cdot\frac{\sum_{\tilde{y}\in\mathcal{Y}_m}\exp\left(\frac{\varepsilon\cdot u_m(x',\tilde{y})}{2\Delta u_m}\right)}{\sum_{\tilde{y}\in\mathcal{Y}_m}\exp\left(\frac{\varepsilon\cdot u_m(x,\tilde{y})}{2\Delta u_m}\right)}.\end{aligned} \tag{9}$$

By the definition of sensitivity,

$$\Delta u_m = \max_{y\in\mathcal{Y}_m}\max_{x,x'\in\mathcal{X}_m}|u_m(x,y)-u_m(x',y)|,$$

we have $u_m(x,y)-u_m(x',y)\le\Delta u_m$ for any $y\in\mathcal{Y}_m$. Thus, the first term is bounded by:

$$\exp\left(\frac{\varepsilon\left(u_m(x,y)-u_m(x',y)\right)}{2\Delta u_m}\right)\le\exp\left(\frac{\varepsilon\cdot\Delta u_m}{2\Delta u_m}\right)=e^{\frac{\varepsilon}{2}}. \tag{10}$$

For the second term, applying $u_m(x',\tilde{y})-u_m(x,\tilde{y})\le\Delta u_m$ to each candidate $\tilde{y}\in\mathcal{Y}_m$ yields:

$$\begin{aligned}\sum_{\tilde{y}\in\mathcal{Y}_m}\exp\left(\frac{\varepsilon\cdot u_m(x',\tilde{y})}{2\Delta u_m}\right) &\le \sum_{\tilde{y}\in\mathcal{Y}_m}\exp\left(\frac{\varepsilon\cdot\left(u_m(x,\tilde{y})+\Delta u_m\right)}{2\Delta u_m}\right) \\ &= e^{\frac{\varepsilon}{2}}\sum_{\tilde{y}\in\mathcal{Y}_m}\exp\left(\frac{\varepsilon\cdot u_m(x,\tilde{y})}{2\Delta u_m}\right).\end{aligned} \tag{11}$$

Substituting these two bounds back into Eq. 9, we obtain:

$$\begin{aligned}\frac{\Pr[M(x)=y]}{\Pr[M(x')=y]} &\le e^{\frac{\varepsilon}{2}}\cdot e^{\frac{\varepsilon}{2}}\cdot\frac{\sum_{\tilde{y}\in\mathcal{Y}_m}\exp\left(\frac{\varepsilon\cdot u_m(x,\tilde{y})}{2\Delta u_m}\right)}{\sum_{\tilde{y}\in\mathcal{Y}_m}\exp\left(\frac{\varepsilon\cdot u_m(x,\tilde{y})}{2\Delta u_m}\right)} \\ &= e^{\varepsilon}.\end{aligned} \tag{12}$$

Therefore, the replacement mechanism $M(\cdot)$ satisfies ε-LDP for any two adjacent tokens that share the same customized output set.

Next, given a private query q, LoGic decouples it into two disjoint components $q\to(z,e)$, where z denotes the logic skeleton and e denotes the entity mapping. We apply the replacement mechanism independently to these two components, with $\varepsilon=\varepsilon_z$ for tokens in z and $\varepsilon=\varepsilon_e$ for tokens in e. Therefore, perturbing tokens in z satisfies ε_z-LDP, and perturbing tokens in e satisfies ε_e-LDP under the customized adjacency relation.

Since the logic skeleton z and the entity mapping e are disjoint components of the private query and are perturbed independently, the parallel composition property of DP (Dwork et al., 2014) implies that their joint release (z',e') satisfies

$$\max(\varepsilon_z,\varepsilon_e)\text{-LDP}$$

at the token level. Since $\varepsilon_z>\varepsilon_e$, the overall privacy budget is ε_z.

C Prompt and examples for Logic-Entity Decoupling

We provide the prompt along with a concrete example to illustrate how the proposed logic-entity decoupling (§4.2) transforms private queries into structured logic skeletons and entity mappings.

Prompt for Logic–Entity Decoupling

You are a logic skeleton extractor. Given a user query, your task is to convert it into a logical skeleton together with a complete entity mapping from abstract symbols to concrete items. The two components must contain non-overlapping information: the logic skeleton should describe only the abstract reasoning structure, while the mapping should contain the concrete information removed from the skeleton.

Requirements:

- Replace all entities with symbols (A, B, C, ...).
- Replace all exact numbers with symbols (m, n, k, ...).
- End with exactly one abstract question line.
- The skeleton should be sufficient to determine the decision.

Output format:

```
[Logic Skeleton]: ...
[Mapping]: A = ...; B = ...; m = ...; n = ...
```

Specific Example for Logic-Entity Decoupling

Example Input:
Facts: The members of The Police were musicians, not law enforcement officers. Only law enforcement officers can perform lawful arrests.
Question: Could the members of The Police perform lawful arrests?

Example Output:

```
[Logic Skeleton]: A are B. B are not C. Only C can perform D. Can A
perform D?
[Mapping]:
- A = The members of The Police;
- B = musicians;
- C = law enforcement officers;
- D = lawful arrests
```

D Prompt and Examples for Structure-Consistent Analogical Example Generation

We provide the prompt along with a concrete example to illustrate how the proposed structure-consistent analogical example generation (§4.3.1) enables the cloud-side LLM to construct a structurally consistent analogical query and its reasoning trajectory from the protected representation (z', e').

Prompt for Structure-Consistent Analogical Example Generation

You are an expert structure-consistent analogical generator. You will be provided with an abstract logic skeleton and a corresponding entity mapping. Your task consists of two sequential steps:

- Substitute the placeholders in the skeleton with the provided entities to construct a coherent, natural-language query. You must strictly preserve the original logical topology, operations, and constraints without adding or altering any logic;
- Solve the newly constructed query with a concise step-by-step reasoning trajectory.

Output format:

```
[Analogical Query]: ...
[Reasoning Trajectory]: ...
[Final Answer]: True or False
```

Specific Example for Structure-Consistent Analogical Example Generation

Example Input:
Logic Skeleton: A represent B. B are not C. Exclusively C can execute D. Can A execute D?
Mapping: A = The citizens of The Station; B = dancers; C = statute agents personnel; D = proper raids

Example Output:

```
[Analogical Query]:
The citizens of The Station are dancers. Dancers are not statute agents
personnel. Exclusively statute agents personnel can execute proper
raids. Can the citizens of The Station execute proper raids?

[Reasoning Trajectory]:
Step 1: The citizens of The Station are dancers.
Step 2: Dancers are not statute agents personnel.
Step 3: Since only statute agents personnel can execute proper raids,
the citizens of The Station cannot execute proper raids.

[Final Answer]:
False.
```

E Prompt and Examples for Structure-Aligned Analogical Pattern Transfer

We provide the prompt along with a concrete example to illustrate how the proposed structure-aligned analogical pattern transfer (§4.3.2) enables the edge-side model to transfer the reasoning pattern from the analogical example $(\hat{q}, \hat{y})$ to the original private query q.

Prompt for Structure-Aligned Analogical Pattern Transfer

You are a structure-aligned reasoning executor. You will be given a private query and a structure-consistent analogical example (including its reasoning trajectory). Your task is to solve the private query by transferring the reasoning pattern from the analogical example.
Requirements:

- Follow the reasoning pattern in the analogical reasoning trajectory.
- Apply the same reasoning steps to the private query using its own entities.
- Do not copy the analogical query or its entities.

Output format:

```
[Reasoning Trajectory]: ...
[Final Answer]: True or False
```

Specific Example for Structure-Aligned Analogical Pattern Transfer

Example Input:
Private Query: The members of The Police were musicians, not law enforcement officers. Only law enforcement officers can perform lawful arrests. Could the members of The Police perform lawful arrests?

Analogical Query: The citizens of The Station are dancers. Dancers are not statute agents personnel. Exclusively statute agents personnel can execute proper raids. Can the citizens of The Station execute proper raids?

Analogical Reasoning:
Step 1: The citizens of The Station are dancers.
Step 2: Dancers are not statute agents personnel.
Step 3: Since only statute agents personnel can execute proper raids, the citizens of The Station cannot execute proper raids.

Example Output:

```
[Reasoning Trajectory]:
Step 1: The members of The Police are musicians.
Step 2: Musicians are not law enforcement officers.
Step 3: Since only law enforcement officers can perform lawful arrests,
the members of The Police cannot perform lawful arrests.

[Final Answer]:
False.
```

F Experimental Setting Details

F.1 Details of the Benchmark Datasets

For training the edge-side extraction model, we use GSM8K (Grade School Math 8K) (Cobbe et al., 2021), a standard mathematical reasoning dataset consisting of grade-school-level problems that require multi-step reasoning. We randomly sample 2,000 examples and use GPT-4.1 to extract the corresponding logic skeletons and entity mappings following our logic–entity decoupling scheme (§ 4.2.1). This yields a synthetic dataset for small-scale training of the edge-side extraction model. For evaluation, we consider two domains: mathematical reasoning and commonsense reasoning. For mathematical reasoning, we use MathQA [2], a multiple-choice benchmark for mathematical problem solving. It evaluates the model's ability to understand math problems, reason over candidate options, and perform operation-based reasoning, which involves strong structural sensitivity and numerical dependencies. We randomly sample 2,000 examples from MathQA. For commonsense reasoning, we use StrategyQA (Geva et al., 2021), a binary question-answering benchmark with 2,290 examples that requires implicit multi-hop reasoning and strong knowledge dependencies, where the reasoning path is not explicitly stated in the question.

F.2 Baselines

We compare LoGic against three categories of baselines in our experiments:

Non-private methods. (1) `Cloud-Only` directly queries the cloud-side LLM through an API with task prompts. It does not provide privacy protection and serves as the performance upper bound. (2) `Edge-Only` uses only the edge-side SLM for reasoning with task prompts. Since all computation is performed locally, it does not leak private information.

Cloud Reasoning methods. These methods first sanitize the private query on the edge side and then send the protected query to the cloud-side LLM for direct reasoning. (1) `CusText` (Chen et al., 2023) performs LDP-based token-level replacement using fixed-size neighboring candidate sets and the exponential mechanism. (2) `CAPE` (Wu et al., 2025) performs LDP-based token-level replacement with context-aware protection and a bucket-based sampling mechanism.

[2] https://huggingface.co/datasets/aledm03/math_qa

Cloud-edge Collaborative Reasoning methods. These methods first sanitize the private query on the edge side, then use the cloud-side LLM to generate intermediate reasoning guidance, and finally combine the cloud guidance with private information on the edge side to produce the final answer. (1) `InferDPT` (Tong et al., 2025) applies dynamic LDP-based token-level replacement on the edge side, uses the cloud-side response as a reference, and lets the edge-side SLM answer the original query. (2) `PRISM` (Zhan et al., 2026) applies adaptive two-layer LDP based on entity risks, asks the cloud-side LLM to generate semantic sketches from the perturbed prompt, and lets the edge-side SLM refine the answer with the original query. (3) `CollabLLM` (Zhang et al., 2026) shifts the query topic while preserving its reasoning pattern, and lets the cloud-side LLM generate executable code for edge-side value substitution. It does not provide LDP guarantees and is mainly limited to numerical reasoning tasks. For clarity, we abbreviate Collaborative LLM Numerical Reasoning with Local Data Protection as `CollabLLM`.

F.3 Evaluation Metrics

Following prior work on privacy-preserving cloud-edge reasoning (Zhan et al., 2026; Zhang et al., 2026), we evaluate each method using three metrics:

(1) **Accuracy** measures whether the predicted answer matches the ground truth:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}(a_i = g_i), \tag{13}$$

where a_i and g_i denote the predicted and ground-truth answers for the i-th example, and $\mathbf{1}(\cdot)$ is the indicator function.

(2) **Reasoning quality** evaluates the verifiability and consistency of the generated reasoning trajectory, including whether the reasoning proceeds coherently and whether the final answer is supported by the reasoning. Following Zhan et al. (2026), we adopt an LLM-as-a-judge approach and prompt GPT-4.1 to score the reasoning trajectory in $[0, 1]$.

(3) **Leakage** measures the probability that the content transmitted to the cloud reveals private information and is used as an empirical indicator of privacy leakage. Following Zhang et al. (2026); Hartmann et al. (2024), we define private information as word-level sensitive content, excluding non-sensitive stop words. We prompt GPT-4.1 to make a binary judgment, where an upload is considered leaked if it shares key nouns with the original query, such as specific entities, facts, or proprietary details. A lower leakage rate indicates stronger privacy protection.

(4) **Utility Drop** aggregates the relative deterioration in both **Accuracy** (ACC) and **Reasoning quality** (RQ) compared to the full `LoGic`, ensuring a balanced assessment of problem-solving efficacy. It is defined as:

$$\text{Utility Drop} = \frac{1}{2}\left(\frac{\text{Acc}_{\text{ours}} - \text{Acc}_i}{\text{Acc}_{\text{ours}}} + \frac{\text{RQ}_{\text{ours}} - \text{RQ}_i}{\text{RQ}_{\text{ours}}}\right) \times 100\% \tag{14}$$

(5) **Leakage Increase** measures the relative inflation in **Leakage** (Leak) compared to the full `LoGic`, expressed as:

$$\text{Leakage Increase} = \frac{\text{Leak}_i - \text{Leak}_{\text{ours}}}{\text{Leak}_{\text{ours}}} \times 100\% \tag{15}$$

F.4 Implementation Details

We conduct all experiments on NVIDIA A100 GPUs. For the edge-side logic extraction model, we adopt Qwen2.5-1.5B-Instruct (Hui et al., 2024) as the base model. For supervised fine-tuning (SFT), we construct a synthetic dataset of 2,000 examples based on GSM8K (Cobbe et al., 2021), with GPT-4.1 generating the corresponding logic extraction annotations. This stage provides the model with basic logic extraction capability. We use 8 GPUs, a per-device batch size of 2, gradient accumulation steps of 4, a learning rate of 1×10^{-5}, 3 epochs, a warmup ratio of 0.1, and random seed 42. We further optimize the SFT model using our Privacy-Aware Dual-Reward GRPO strategy (see § 4.2.2). We set the dual-reward weight $\lambda = 0.5$, group size to 8, rollout batch size to 8, learning rate to 5×10^{-7}, and train for 1 epoch with a KL coefficient $\beta = 0.001$. The SFT checkpoint is used as the reference policy, with random seed 42. We train a single extraction model and apply it

Table 4: Sensitivity analysis of the privacy allocation ratio ϵ_e/ϵ_z on MathQA.

ϵ_e/ϵ_z	1/15	1/6	1/3	1/2	2/3	1
Accuracy ↑	0.544	0.575	0.661	0.621	0.658	0.663
Reasoning Quality ↑	0.576	0.589	0.651	0.616	0.657	0.652
Leakage ↓	0.018	0.051	0.089	0.083	0.119	0.147

Table 5: Evaluation with independent judge models on MathQA.

Methods	Accuracy ↑	Reasoning Quality ↑			Leakage ↓		
		GPT-4.1	Claude Sonnet 4.6	Gemini 2.5 Pro	GPT-4.1	Claude Sonnet 4.6	Gemini 2.5 Pro
Cloud-Only	0.913	0.885	0.885	0.895	1.000	1.000	1.000
Edge-Only	0.579	0.626	0.615	0.592	0.000	0.000	0.000
CusText	0.303	0.228	0.249	0.239	0.132	0.108	0.237
CAPE	0.298	0.264	0.229	0.221	0.079	0.102	0.136
InferDPT	0.254	0.293	0.278	0.267	0.094	0.094	0.151
PRISM	0.511	0.474	0.502	0.507	0.566	0.447	0.647
CollabLLM	0.609	0.570	0.576	0.576	0.189	0.213	0.332
LoGic (ours)	**0.661**	**0.631**	**0.651**	**0.654**	**0.059**	**0.089**	**0.112**

across all datasets and edge-side reasoning models, as the extraction module and overall framework are designed to be model-agnostic and dataset-agnostic. For differentiated LDP protection, we set $\varepsilon_e = \frac{1}{3}\varepsilon_z$ by default (see App. G), with the total privacy budget defined as $\varepsilon = \max\{\varepsilon_z, \varepsilon_e\} = \varepsilon_z$. Following CusText (Chen et al., 2023), we set the candidate set size K to 20 by default. Unless otherwise specified, all judge evaluations use GPT-4.1 with temperature 0, with each evaluation repeated three times, and manual verification on 100 randomly sampled examples to confirm prompt reliability.

G Sensitivity Analysis of the Privacy Allocation Ratio

The default privacy allocation ratio ϵ_e/ϵ_z is designed to balance two competing objectives: providing stronger protection for entity/value tokens, which are more instance-specific and privacy-sensitive, while preserving sufficient logical information for cloud-side analogical reasoning. Intuitively, a smaller ϵ_e over-perturbs the entity mapping and consequently degrades reasoning utility, whereas a larger ϵ_e weakens privacy protection while providing limited additional utility.

To examine the sensitivity of `LoGic` to this hyperparameter, we conduct an additional sensitivity analysis on MathQA by varying ϵ_e/ϵ_z while keeping $\epsilon_z = 3$. The edge model is Qwen2.5-1.5B-Instruct, the cloud LLM is GPT-4.1, and Claude Sonnet 4.6 is used as the judge model for evaluating `Reasoning Quality` and `Leakage`. The results are presented in Tab. 4. Ratios below $1/3$ noticeably reduce utility; ratios above $1/3$ bring little/no utility gain, while leakage rises as the ratio approaches 1. Thus, $1/3$ is a reasonable privacy-utility knee point.

H Additional Validation with Independent Judges and Cloud LLMs

To further assess the robustness and generalizability of our empirical findings, we conduct two complementary validation experiments on MathQA using Qwen2.5-1.5B-Instruct as the edge model.

Independent Judge Validation. We retain GPT-4.1 as the cloud LLM for all cloud-based baselines, while re-evaluating the outputs of all methods using two independent judge models, Claude Sonnet 4.6 and Gemini 2.5 Pro, for both `Reasoning Quality` (RQ) and `Leakage` (Leak). The results are presented in Tab. 5. The relative performance of the methods remains highly consistent across the three judges. In particular, LoGic maintains the best privacy-utility trade-off across all judges.

Cloud LLM Generalization. We further replace GPT-4.1 with Claude Sonnet 4.6 as the cloud LLM for all cloud-based methods, while evaluating the resulting outputs using GPT-4.1 and Gemini 2.5 Pro. The results are shown in Tab. 6. LoGic consistently outperforms all privacy-preserving baselines,

Table 6: Evaluation with Claude Sonnet 4.6 as the cloud LLM on MathQA.

Methods	**Accuracy ↑**	**Reasoning Quality ↑**		**Leakage ↓**	
		GPT-4.1	**Gemini 2.5 Pro**	**GPT-4.1**	**Gemini 2.5 Pro**
Cloud-Only (Claude Sonnet 4.6)	0.913	0.894	0.895	1.000	1.000
Edge-Only	0.579	0.626	0.592	0.000	0.000
CusText	0.373	0.284	0.236	0.132	0.237
CAPE	0.253	0.260	0.221	0.079	0.136
InferDPT	0.267	0.298	0.283	0.094	0.151
PRISM	0.531	0.547	0.526	0.566	0.647
CollabLLM	0.612	0.620	0.632	0.189	0.332
LoGic (ours)	**0.641**	**0.658**	**0.642**	**0.059**	**0.112**

demonstrating that the proposed framework does not rely on GPT-4.1 and generalizes well across different cloud LLMs.

Overall, these additional experiments provide evidence that the empirical findings of LoGic are robust to both the choice of judge model and the choice of cloud LLM. Across independent judges and different cloud LLM configurations, LoGic consistently maintains a favorable privacy-utility trade-off, supporting the robustness and generalizability of the proposed framework.

I Significance Test Results

Table 7: Statistical significance test results (p-values) across models for Accuracy, Reasoning Quality, and Leakage metrics on MathQA and StrategyQA. All p-values < 0.05 indicate statistically significant improvements over the baselines.

Dataset	Model	Accuracy	Reasoning Quality	Leakage
MathQA	Llama3.2-1B	2.87e-10	9.27e-07	2.78e-12
	Qwen2.5-1.5B	1.81e-03	2.61e-04	2.78e-12
StrategyQA	Llama3.2-1B	2.88e-05	2.50e-13	1.79e-11
	Qwen2.5-1.5B	1.98e-10	3.41e-12	1.79e-11

We conduct the t-test (Bartlett, 1937) to evaluate whether the improvements of our method are statistically significant. The statistical unit is an individual test example from MathQA or StrategyQA. All tests are paired, since every method is evaluated on the same examples. The comparison is LoGic versus the strongest corresponding privacy-preserving baseline under the same dataset, edge model, and evaluation metric. The p values reported in Tab. 7 are all smaller than 0.05, indicating that the improvements are significant.

J Efficiency and Resource Overhead

Table 8: Reasoning-time efficiency comparison on the entire MathQA dataset. We report the total edge-side time, cloud-side time, and overall inference time in hours.

Methods	Edge Time (h) ↓	Cloud Time (h) ↓	Total Time (h) ↓
InferDPT	0.6	0.3	0.9
PRISM	0.1	0.3	0.4
CollabLLM	0.2	0.3	0.5
LoGic	0.1	0.3	0.4

We further compare the reasoning-time overhead of privacy-preserving cloud-edge collaborative methods on the MathQA dataset. We report the edge-side time, cloud-side time, and total inference time in Tab. 8. Since the training cost of LoGic's logic extractor and CollabLLM's local topic

converter is a one-time offline cost, the trained modules can be reused across evaluation datasets and edge-side reasoning models. Therefore, we exclude these offline training costs and focus on deployment-time reasoning overhead. As shown in Tab. 8, LoGic achieves the lowest total inference time, matching PRISM with a total cost of 0.4 hours on MathQA. More importantly, under the same inference-time budget as PRISM, LoGic achieves stronger reasoning performance and lower leakage, demonstrating a more favorable efficiency-utility-privacy trade-off for privacy-preserving cloud-edge reasoning.