Collaborative LLM Numerical Reasoning with Local Data Protection

Min Zhang[1]*, Yuzhe Lu[2], Yun Zhou[2], Panpan Xu[2], Lin Lee Cheong[2], Chang-Tien Lu[1], Haozhu Wang[2]

[1]Virginia Tech
[2]AWS AI

Abstract

Numerical reasoning over documents, which demands both contextual understanding and logical inference, is challenging for low-capacity local models deployed on computation-constrained devices. Although such complex reasoning queries could be routed to powerful remote models like GPT-4, exposing local data raises significant data leakage concerns. Existing mitigation methods generate problem descriptions or examples for remote assistance. However, the inherent complexity of numerical reasoning hinders the local model from generating logically equivalent queries and accurately inferring answers with remote guidance. In this paper, we present a model collaboration framework with two key innovations: (1) a context-aware synthesis strategy that shifts the query topics while preserving reasoning patterns; and (2) a tool-based answer reconstruction approach that reuses the remote-generated plug-and-play solution with code snippets. Experimental results demonstrate that our method achieves better reasoning accuracy than solely using local models while providing stronger data protection than fully relying on remote models. Furthermore, our method improves accuracy by 16.2% - 43.6% while reducing data leakage by 2.3% - 44.6% compared to existing data protection approaches.

Introduction

Numerical reasoning over documents is a practical yet complex task that often requires powerful black-box models like GPT-4 for problem-solving (Akhtar et al. 2023). This task demands a deep understanding of documents, the ability to identify relationships from scattered evidence, and the capability to derive answers through quantitative calculations. In real-life scenarios, numerical reasoning is essential in tasks like analyzing financial reports (Chen et al. 2021; Zhao et al. 2022; Ma et al. 2025), research papers (Wu, Zhu, and Liu 2025), medical documents (Mahendra et al. 2024), and contracts with numerical conditions (Huang et al. 2021). Due to the demanding requirements for contextual understanding and logical reasoning, on-device or in-house small models often struggle to solve such problems effectively. Consequently, remote black-box models with strong problem-solving capabilities are frequently accessed via API calls to address these challenges.

*Work done during Min's internship at AWS AI.

However, the direct exposure of local data to remote models introduces significant risks of information leakage (Wang et al. 2024). Following prior studies (Zhou et al. 2023; Tong et al. 2023; Hartmann et al. 2024), we define local privacy information to pertain to every word, excluding non-sensitive stop words (Yue et al. 2021). Sensitive data can be presented explicitly or embedded implicitly within various contexts and formats, including company details, operational values, and strategic analyses, as illustrated in Fig. 1 (a). While some works (Siyan et al. 2024; Chen et al. 2023; Aahill 2023) detect and remove explicit Personally Identifiable Information (PII), models like GPT-4 can infer personal attributes from residual context (Staab et al. 2024). Similarly, the sentence *"our current policy is not to enter into transactions to hedge our fuel consumption..."* in the example reveals confidential policy decisions without containing any explicit PII. Therefore, in this work, we aim to thoroughly protect the document-level local text before the black-box model inference.

The inherent complexity of numerical reasoning over documents exacerbates the trade-off between data leakage and model utility. On one hand, local data must be concealed from the remote model to minimize information leakage. On the other hand, the remote model often requires detailed contextual information to have a deep understanding of the problem and to provide accurate help for effective reasoning. We identify two primary challenges for reasoning in the typical data protection workflow as shown in Fig. 1 (b):

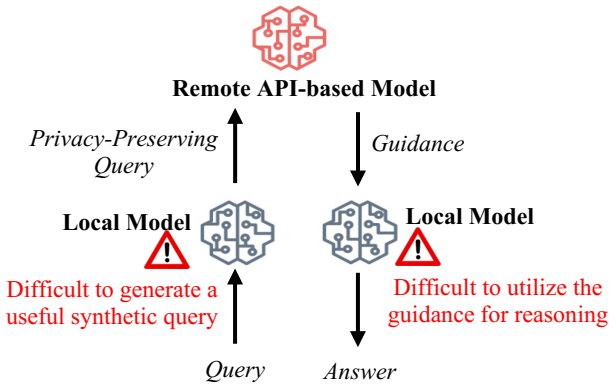
Difficulty in generating logically coherent synthesized queries. Existing methods often locally synthesize various queries for data protection, such as high-level descriptions (Zhang et al. 2024), analogous examples (Hartmann et al. 2024), dp-based permutation or paraphrased documents (Utpala, Hooker, and Chen 2023). However, the complex logical reasoning within documents hampers the synthesis of accurate descriptions or logical-coherent contexts while keeping local data secret.

Local answer reconstruction with limited reasoning ability. Existing methods leverage a local model to integrate local data and the remote model's response to generate the final response. Although they are effective for semantic-focused tasks such as creative writing, translation, and summarization (Tong et al. 2023; Zhang et al. 2024; Hartmann et al. 2024), their performance drops dramatically in reasoning tasks which demand both deep contextual understanding

Context: *Company/Project Information*
… shows company X's annual aircraft fuel consumption and costs, including taxes, for our mainline and regional operations for 2018, 2017 and 2016 (gallons and aircraft fuel expense in millions) . *Important Values*
In 2018, the aircraft fuel expense was $ 9896. The average price per gallon was $ 2.23. The percent of total operating expenses was 23.6%. In 2017… *Policy/Financial Analysis*
…, we did not have any fuel hedging contracts outstanding to hedge our fuel consumption … we will continue to be fully exposed to fluctuations in fuel prices . our current policy is not to enter into transactions to hedge our fuel consumption …

Question: What was the total operating expenses in 2018?
Golden Answer: 41932.2 Explanation: div(9896, 23.6%)

(a) An example of information leakage



(b) The typical workflow for privacy preservation

Figure 1: (a) An example from FinQA with highlighted sensitive information. (b) The typical workflow and challenges for preserving privacy in interactions with remote API-based models.

and complex logic. Even with accurate hints or examples, small models often struggle to reconstruct correct answers due to their limited reasoning abilities.

To address these challenges, we propose a novel method that transforms reasoning queries into a different domain while preserving the linguistic and logical patterns. As illustrated in (Fig. 2), by translating reasoning requests from one domain (e.g., *aircraft fuel consumption*) to another (e.g., *advertising revenue*) while preserving their inferential skeleton (e.g., *deriving the total amount from the individual count and percentage*), we enable secure delegation of reasoning to remote models without compromising sensitive information. This pattern-preserving transformation not only protects privacy but also retains the epistemic scaffolding necessary for solution generalization.

Building on this abstraction, we further introduce a tool-based answer reconstruction strategy where the remote model returns a reusable problem-solving tool with executable code snippets. This plug-and-play construct enables precise local answer recovery through direct numerical substitution, without local model dependence.

Our main contributions are summarized as follows:

- We design a topic-shifted and pattern-preserving data synthesis approach that replaces the semantic surface while keeping the reasoning structure intact as a whole, enabling privacy-preserving delegation to remote models.
- We propose a plug-and-play answer reconstruction method that leverages tool-centric reusable solution paradigm from the remote model, facilitating precise answer recovery via numerical substitution.
- We achieve a superior accuracy-privacy trade-off than existing collaborative inference methods with data protection measures, improving accuracy by 16.2% - 43.6% while reducing data leakage by 2.3% - 44.6%.

Related Work

Prior works address data protection for training data (Papernot et al. 2017; Yue et al. 2021; Tian et al. 2022; Kurakin et al. 2023; Xie et al. 2024; Yu et al. 2024) or for demonstrations (Hong et al. 2024; Carey et al. 2024), but these approaches do not protect user queries during inference.

To protect privacy at inference time, some methods add differential privacy (DP) noise to text embeddings in white-box settings (Du et al. 2023; Zhou et al. 2023). However, these techniques are incompatible with black-box models, which typically require plain-text inputs rather than embeddings.

Some studies preserve privacy by replacing PII like names or locations using hider models (Chen et al. 2023) or prompt optimization (Siyan et al. 2024), but residual context often still leaks information (Staab et al. 2024).

For document-level privacy protection, some approaches apply DP-based perturbations to generate semantically similar but altered contexts (Tong et al. 2023; Xie et al. 2024), or prompt LLMs to paraphrase documents for downstream tasks like sentiment classification (Utpala, Hooker, and Chen 2023). Yet, these methods generally preserve semantics without guaranteeing logical consistency with the original content. Other works prompt local models to generate high-level descriptions (Zhang et al. 2024) or analogous examples (Hartmann et al. 2024). Still, generating queries that are both privacy-preserving and informative remains challenging, and local models often struggle to effectively utilize remote guidance due to limited capacity.

In contrast, our method enhances both privacy and logical fidelity by introducing a pattern-preserving topic shifter, and further employs a plug-and-play answer reconstruction strategy that avoiding local model dependence.

Method

To maximize the utility of remote models while minimizing local data leakage in numerical reasoning tasks, we propose an effective collaboration protocol based on the observation that sensitive queries can be transformed into a different domain without their underlying mathematical structure altered. Our approach involves translating queries from one domain to another, ensuring that the remote model processes semantically transformed but structurally equivalent data. This strat-

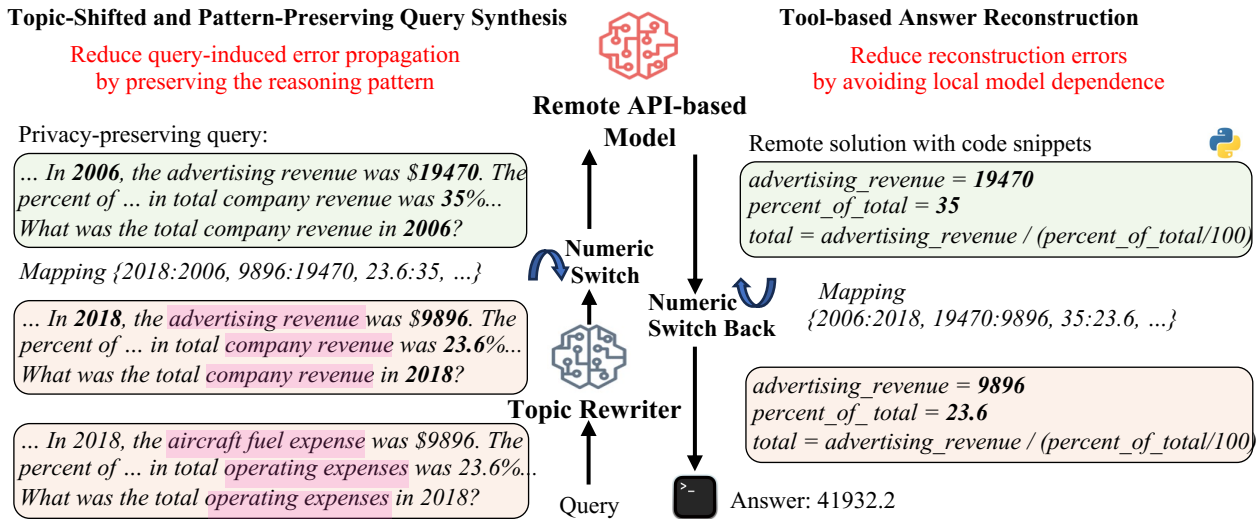


Figure 2: The proposed method illustrated with examples for each step. The original query is transformed from one topic (e.g., *aircraft fuel consumption*) to another (e.g., *advertising revenue*) by a distilled topic rewriter, while preserving its reasoning pattern (e.g., *deriving the total amount from the individual count and percentage*). This enables secure delegation to a remote API-based model to elicit a tool-centric, plug-and-play solution for local answer reconstruction through direct numerical switch, without requiring local model re-inference.

egy mitigates the risk of exposing sensitive information while offering the local model a reusable solution. To achieve this, we introduce two key components of our protocol (as shown in Fig. 2), topic-shifted and pattern-preserving query synthesis and tool-based answer reconstruction, in the following subsections.

Topic-Shifted and Pattern-Preserving Query Synthesis

In this section, we introduce the module leveraging specialized local models to synthesize requests with shifted topics but equivalent mathematical abstractions. Since hiding the local information and maintaining the underlying logic remains challenging for small local models, we fine-tune a dedicated request synthesis model and subsequently apply a numerical replacement strategy. By decoupling semantic protection and numerical protection, maintaining numerical values intact in the topic shifter not only facilitates mathematical abstraction consistency verification but also paves the way for local answer reconstruction (as explained in the next section). Our data synthesis approach is detailed below.

Topic Shifter To protect the overall local information instead of detecting specific sensitive words, we instruct the synthesis model to shift the topic while maintaining the original format, logic and numerical values. By preserving numeric values, we establish a clear mapping between the original and synthesized objects (e.g., *fuel expense* $\leftrightarrow$ *advertising revenue* in Fig. 2).

We formally characterize the input and output of the proposed request synthesis model, denoted as $\mathcal{M}_{\mathcal{S}}$, using the following equation:

$$\tilde{C}, \tilde{q} = \mathcal{M}_{\mathcal{S}}(C, q) \tag{1}$$

where $\tilde{C}, \tilde{q}$ represent the transformed context and query derived from the original inputs C, q.

To deploy an efficient local synthesizer, we distill the capabilities of a large remote model into a smaller local model. The remote model $\mathcal{M}_{\mathcal{R}}$ generates synthetic data for training. Since the numerical values remain unchanged during the topic shift, a logically consistent rewritten question should yield the same answer as the original question. Through such post-hoc analysis, we confirm its proficiency in instruction-following and generating pattern-preserving requests. In contrast, small local models often struggle to meet these requirements, especially in keeping numerical consistency and logical coherence. Via instruction tuning, we enhance the local synthesizer's rewriting and instruction-following capabilities.

Data Switch Since the topic shifter focuses on semantic protection, leaving the numerical values unchanged, we further obfuscate numerical values in the synthesized request to ensure complete anonymization. Using regular expressions, we extract all numbers $\mathcal{N} = \{n_1, n_2, \ldots, n_k\}$ from the request and apply a transformation

$$h : n_i \mapsto \tilde{n_i} \tag{2}$$

We employ three strategies to ensure data transformation quality: special number handling, offset transformation for year-related values, and order-preserving transformation. Special numbers (e.g., 28–31 for month-end dates) remain unchanged to preserve their semantic meaning. Integers between the year-related range undergo offset-based transformation that maintains relative differences between years. All other numbers are sorted into intervals and mapped to randomly sampled values from a target range while preserving their original order relationships. The final synthesized request with transformed

numerical values, denoted as $(\tilde{C}_h, \tilde{q_h})$, is then forwarded to the remote model for assistance.

The data switch ensures numerical security, complementing the topic shifter's for comprehensive data protection. Decoupling numerical protection during local data synthesis, we simplify subsequent local reconstruction. Serving as a bridge between the topic shifter and local reconstruction, the data switch ensures a seamless transition for accurate and secure data processing.

By fine-tuning the specialized local synthesizer $\mathcal{M}_{\mathcal{S}}$, we ensure the synthetic request retains the original problem-solving logic while achieving a complete topic shift. Additionally, the numeric transformation step guarantees that shared values do not expose sensitive information. The transformation h is stored locally as a dictionary, and its role in enhancing local inference accuracy is further elaborated in the subsequent section.

Tool-based Local Answer Reconstruction

In addition to protecting local data, another key aspect of privacy-preserving collaborative inference is how to best leverage remote assistance. Due to privacy constraints, the solution from the remote model to a proxy request cannot be used directly. Naively, one could simply add the remote guidance to local model's context to elicit a better response. However, we found that this strategy does not lead to satisfactory performance for the local model on numerical reasoning tasks owing to its limited capabilities. Thus, we propose a more structured scheme for the local model to generate its answers, which leads to dramatic performance improvements.

Since the synthesized request maintains the same logic as the original one, they share the same problem-solving pattern. To best preserve and communicate this pattern, we instruct the remote model to generate Python code (Chen et al. 2022) as a tool for the local model. During answer reconstruction, the local model will simply perform data substitution: since intermediate steps are represented using Python variables, substituting the input data is sufficient. The final answer is obtained by executing the code in an interpreter. We elaborate each of these steps below.

Remote Assistance After performing topic rewriting and numerical anonymization, we provide the synthesized context $\tilde{C}_h$ and synthesized question $\tilde{q_h}$ to the remote model $\mathcal{M}_{\mathcal{R}}$:

$$f(m_1, m_2 \ldots, m_t) = \mathcal{M}_{\mathcal{R}}(\tilde{C}_h, \tilde{q_h}) \tag{3}$$

where $m_i \in \{\tilde{n_1}, \tilde{n_2}, \ldots, \tilde{n_k}\}$. The remote model returns the Python code snippets $f(\cdot)$ for problem solving with transformed numeric inputs from the synthesized request.

Local Reconstruction Since the local model often fails to recognize the logical connection between the original and synthesized request and thus struggles to write Python codes for the local problem, we implement a plug-and-play approach to reuse the Python solution f from the remote model. Specifically, we compute the answer to the local problem by directly executing the following with Python interpreter:

$$f(h^{-1}(m_1), h^{-1}(m_2), \ldots, h^{-1}(m_t)) \tag{4}$$

Recall that h is the mapping between original and transformed numeric values. Simply by swapping the input values, we can obtain the answer for the local problem thanks to the logical consistency between original and synthesized requests. With the collaboration scheme above, we maximize the remote model utility while relieving the reasoning model from reasoning burdens. Before we move on to showcase the superior performance of our method, we would like to emphasize the unique contribution of our approach. While our method might seem to be an instantiation of Program-of-Thought (PoT)(Chen et al. 2022), our focus is mainly on how to reuse the Python solution with quite different contexts and accurately transfer to a logically equivalent problem instead of leveraging code to enhance reasoning. It's important to note that PoT is merely a vehicle for our problem-solving pattern, and its success is deeply rooted in the logical consistency of queries as well as the decoupling of semantics and data protection within our approach.

Experiment Settings

Datasets We conduct experiments using two question-answering datasets: FinQA (Chen et al. 2021) and MultiHiertt (Zhao et al. 2022). Both datasets involve questions that require numerical reasoning based on provided documents. They contain both explicit and implicit sensitive information, including financial data analysis, project details, and decision-making content, making them well-suited for evaluating our document-level local data protection approach.

Data Processing For privacy-preserving collaboration methods, we shorten the context by retrieving relevant parts of the full document. This retrieval process is a natural choice for long-document processing and reduces the data synthesis burden compared to using the entire document. To build it, we leverage the local self-consistency inference process by prompting the local model to surface supporting sentences during its reasoning. These are then combined with results from a BM25 retriever to form the final context. Details of the local retriever can be found in the Appendix, and it can also be replaced with other retrievers.

Models We employed two lightweight local models designed for resource-constrained environments. Specifically, we used Phi-3-mini-128k-instruct (3.8B params), and Llama-3.2-3B-Instruct. For collaborative reasoning, we used GPT-4o as the remote model.

Model Distillation Settings For the topic rewriter model, we adopted a distillation setup where Llama-3.2-3B-Instruct served as the student model and GPT-4o as the teacher model. After applying a data quality filtering process, via leakage evaluation, conflict evidence detection, and answer consistency verification (see Appendix), we retained 5762 training samples out of 6360 samples from MultiHiertt's training set. We only trained a single synthesis model and used it for different datasets and local models.

Evaluation Metrics Our evaluation comprises two key aspects: accuracy and local data leakage. Accuracy assesses

Datasets		MultiHiertt		FinQA	
Metric		Acc.(%)↑	Leakage(%)↓	Acc.(%)↑	Leakage(%)↓
Local Model: Phi-3-mini-128k-instruct (3.8B)					
Local Methods	*Single-Inference*	54.4	0	71.1	0
	Self-Consistency	64.6	0	80.0	0
Collaboration Methods	*Hint*	42.7	6.0	63.9	28.8
	Example	57.0	16.1	71.4	38.9
	Ours	**80.1**	**3.7**	**87.6**	**6.4**
Local Model: Llama-3.2-3B-Instruct					
Local Methods	*Single-Inference*	33.8	0	54.0	0
	Self-Consistency	44.5	0	68.6	0
Collaboration Methods	*Hint*	27.0	20.9	55.3	50.5
	Example	31.1	18.8	55.5	20.0
	Ours	**70.6**	**5.5**	**90.1**	**5.9**

Table 1: Main results of the proposed method and baselines, where accuracies are normalized to the remote model's single-inference performance (66.4% on MultiHiertt and 73.7% on FinQA).

whether the predicted answer matches the ground truth. Following (Hartmann et al. 2024), we report normalized accuracy scores (actual accuracy divided by remote-only inference accuracy) to show how much of the remote model's performance different methods can achieve. For data leakage assessment, we follow prior studies (Zhou et al. 2023; Tong et al. 2023), which define local privacy information as pertaining to every word, excluding non-sensitive stop words (Yue et al. 2021). Similar to recent LLM-as-a-judge approaches (Hartmann et al. 2024; Siyan et al. 2024) for sensitive data leakage detection, we prompt the LLM to provide a binary judgment on whether the synthesized data contains original information. Specifically, we instruct a strong model to evaluate the presence of local information in remote model interactions. The evaluation prompt is as follows: *Given context A and context B, determine whether context B uses information from context A. Ignore table formats and sentence structures; If they share some similar important nouns, it can be considered that context B uses information from context A. Respond directly with Yes or No.* Here, context A denotes the original input, while context B represents the transmitted text during interactions with the remote model. We utilized GPT-4o-mini as the judge for data leakage evaluation. To validate the evaluation's reliability, we conducted a manual review of 200 randomly selected leakage mapping results from the training set in both datasets. The agreement rate with human annotations was 96%, demonstrating a high alignment between the automated evaluation and human judgment.

Baselines We compare our method with baselines involving local-only approaches, vanilla cascading without data protection, and model collaboration with different protection strategies. All methods utilize 3-shot prompting. We employ top-p sampling (Holtzman et al. 2019) with $p = 0.9$ for local models and greedy sampling for the remote model.

Single-Inference: We use in-context learning with few-shot demonstrations (Brown et al. 2020) and Program-of-Thought (PoT) (Chen et al. 2022) for local solution generation. Results are averaged over seven runs.

Self-Consistency: To enhance local inference, we adopt self-consistency (Wang et al. 2022), selecting the answer with the highest voting consistency from seven executions.

Vanilla Model Cascading: Following (Yue et al. 2024), if local answer consistency falls below a threshold, the request is directly sent to a remote black-box model without data protection.

Hint: Following (Zhang et al. 2024; Hartmann et al. 2024), the local model generates a problem description, and the remote model provides a high-level hint. The hint is then integrated with local data for local inference.

Example: Following Hartmann et al. (2024); Utpala, Hooker, and Chen (2023), we use a local model to rephrase queries by concealing sensitive information (including both topics and numerical values) before sending to a remote model. The local model then combines local data with the remote solution for final inference.

Results

We first present the main results, comparing our method with local-only and existing privacy-preserving baselines in terms of accuracy and leakage relative to directly sending queries to the remote model. We then evaluate different privacy strategies under a model cascade framework, where a decision maker routes only harder queries to the remote model, to assess effectiveness on queries of different difficulty. Next, we conduct an ablation study to assess the contribution of each component. We further examine the impact of training data size on model performance. Finally, we perform an error analysis on incorrect cases.

Main Results

Table 1 shows the accuracy and leakage results for various methods with different local models on two datasets. Compared to local inference methods, our method provides an alternative solution that trades minimal data leakage for substantial accuracy improvements. Compared to collaborative inference methods with other data protection strategies, our method presents a superior improvement in both accuracy and data protection.

Our method significantly improves local accuracy while approaching the upper accuracy bound of fully utilizing the remote LLM, with minimal data leakage. *(1) Improved local accuracy approaching remote accuracy*. Compared to single-inference and self-consistency approaches, our method achieves a notable accuracy boost. For instance, on MultiHiertt, Phi-3-mini-128k-instruct improves from 54.4% to 80.1%, and on FinQA, Llama-3.2-3B-Instruct increases from 54.0% to 90.1%, closely approaching remote model performance. Our method effectively leverages the remote model to compensate for the limited reasoning capabilities of the local model. *(2) Minimal data leakage*. In addition to substantial accuracy improvements, our method maintains low data leakage. On MultiHiertt, leakage is limited to 3.7% and 5.5% for Phi-3-mini-128k-instruct and Llama-3.2-3B-Instruct, respectively. On FinQA, leakage remains at 6.4% and 5.9%, reflecting a balanced trade-off between privacy and accuracy. These values are markedly lower compared to complete leakage in remote-only approaches.

Our method achieves a new frontier in both the accuracy and the data protection over data protection baselines. *(1) Accuracy improvement.* Our method demonstrates a substantial performance improvement over existing approaches, with notable gaps observed across various datasets and models. Our method outperforms the Hint-based method by 23.7% - 43.6% in accuracy, and the Example-based method by 16.2% - 39.5%. Notably, the Hint and Example methods result in even lower accuracy than solely using the local model. This is because the unreliable, logic-incoherent query fails to trigger the remote large model for tailored assistance, and it does not effectively invoke the weaker local model's reasoning capabilities. In contrast, our method generates logically consistent queries to elicit the remote model to produce problem-solving patterns, and it easily recovers answers through data replacement. *(2) Leakage reduction.* Our method achieves a significant reduction in data leakage compared to existing approaches. On the MultiHiertt dataset, leakage decreases by 2.3% - 15.4%. The reduction is even more pronounced on the FinQA dataset (14.1% - 44.6%). This is because the Hint method tends to leak contextual information when generating problem descriptions, and the Example method struggles to hide information effectively on its own. In contrast, our approach achieves effective information hiding and consistency maintenance by decoupling topic shifting and data replacement, along with distillation from the strong model. Examples of the synthesized query for different methods can be found in Appendix.

Our local synthesis model and overall framework are general, maintaining both model-agnostic and dataset-agnostic properties. Finally, our method demonstrates strong generalization across different datasets and local retrievers. Specifically, the local synthesis model was only trained on the MultiHiertt dataset using context shortened by the local retriever model Phi3. As shown in Table 1, our approach consistently achieves the highest accuracy and the lowest local data leakage rate, highlighting its effectiveness across unseen datasets (FinQA) and local retriever models (Llama3.2-3B-Instruct).

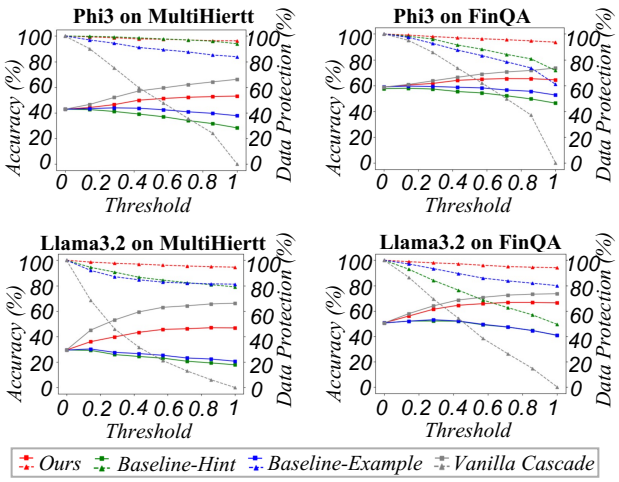


Figure 3: Results with a referral module in model cascade under varying thresholds. A higher threshold leads to more instances of seeking remote model collaboration. Solid lines represent actual accuracy, while the dashed lines show local data protection.

Results with a Referral Module in Model Cascade

In practice, it is unnecessary to forward all local instances to the remote model, as the local model can handle certain simple instances. Building on the vanilla model cascade (Yue et al. 2024), we use the local model's answer consistency rate to determine whether the remote model should be involved. If the consistency rate is below the threshold, the remote model is activated to assist in the decision-making process. When the threshold is set to 0, all queries are handled locally; when set to 1, all queries involve the remote model.

Fig. 3 illustrates the actual accuracy and local data protection of various model collaboration methods at different thresholds. The solid lines represent actual accuracy, while the dashed lines indicate local data protection, with the local data protection rate calculated as 1 minus the leakage rate. The upper-right corner represents a combination of high accuracy and high local data protection. Our results show that while the vanilla model cascade method achieves higher accuracy by directly exposing the full original context, it leads to significant local data leakage. In contrast, baseline privacy-preserving methods improve local data protection but at the cost of dramatically reduced accuracy. Our approach strikes a balance, achieving comparable accuracy while maintaining high local data protection.

Ablation Study

To evaluate the effectiveness of the main components of our method, synthesizer distillation and answer reconstruction, we perform an ablation study and present the results in Table 2.

In the "w/o Tool" setting, we replace the tool-based answer construction with local inference using the same synthesized example and remote solution as before ablation. The request synthesis step remains unchanged, but the local model per-

	Distil.	Tool	Acc	Leak.
Ours	✓	✓	**90.1**	**5.9**
w/o Tool	✓	✗	65.0	5.9
w/o Distil.	✗	✓	42.2	15.1
w/o Distil. and Tool	✗	✗	65.7	15.1

Table 2: Ablation study conducted on the FinQA dataset with Llama3.2-3B-Instruct as both the local inference model and local hider model.

Training Data Size	Acc. (%)	Leakage (%)
800	74.3	9.2
2000	73.2	8.2
5762	70.6	5.5

Table 3: Sensitivity to training data size on MultiHiertt dataset with Llama3.2-3B-Instruct model.

forms inference by combining the synthesized query, remote solution, and original query. This results in a significant accuracy drop, as the local model, with its limited reasoning capability, struggles to comprehend and replicate the example for effective reasoning despite receiving identical remote information.

In the "w/o Distillation" setting, we remove the distillation process and directly prompt the unmodified Llama3.2-3B-Instruct as the local synthesizer using the same data protection instructions. Other components, including tool-based answer reconstruction, remain unchanged. We observe a significant drop in accuracy, primarily because the local model alone fails to generate logically consistent queries and preserve the original numerical values. Firstly, this results in unreliable queries, preventing the remote model from producing a valid problem-solving pattern. Consequently, the tool-based answer reconstruction, which relies heavily on this pattern, fails to generate correct answers. Secondly, the unpreserved numbers in the context after topic rewriting cause the local model to switch back to the wrong numbers even if using the correct problem-solving pattern.

In the "w/o Distillation and Tool" setting, we remove both the tool-based answer reconstruction and distillation for the local synthesizer. The local model uses the same synthesized query as in the "w/o Distillation" setting, along with the synthesized example and solution from the remote model, to perform re-inference. Accuracy improves compared to "w/o Distillation" because, despite the incorrect number in the example, local inference relies more on the model's understanding and reasoning of the original query. However, accuracy remains lower than our method, as incoherent examples and the limited reasoning ability of the local model hinder effective inference.

Sensitivity to Training Data Size

We vary the training size for the topic rewriter and report the results in Table 3. As the training size decreases, we observe a slight increase in task accuracy accompanied by a consistent rise in leakage. This is because the topic rewriter sometimes makes only minor changes to the original query when trained on less data. Thus, the accuracy increases at the cost of reduced privacy protection. This aligns with the inherent accuracy–leakage trade-off. Remarkably, even with only 800 training examples, our method still significantly outperforms existing baselines in both accuracy and privacy, demonstrating the effectiveness of our approach.

Error Type	Percent (%)
Retrieval Error	20
Rewrite Error	16.7
Numeric Switch Error	6.7
Remote LLM Error	33.3
Answer Reconstruction Error	0
Annotation Error/Ambiguous Question	23.3

Table 4: Error analysis.

Error Analysis

We randomly sample 30 error cases from all incorrect predictions on MultiHiertt with the Phi3 model and manually analyze the error types. The distribution is shown in Table 4.

The most common source of error is remote LLM's error (33.3%), including evidence grounding, logical reasoning, and Python coding errors. This is the imperfect model's inherent error, note that the accuracy for remote-only method on MultiHiertt/FinQA dataset is 66.4% and 73.7%. A significant portion of the errors is attributed to flaws in the dataset itself (23.3%). Either the annotations were incorrect, or the questions were too ambiguous for a precise answer to be generated. The retrieval issues account for 20% of errors. We follow existing methods to shorten the context for long-text reasoning for input length-limited models as the first step in our method. The retrieval is not the main focus or contribution of our paper. It can be replaced with advanced retrieval methods to mitigate retrieval errors. The errors for two core designs in our method — topic rewriter (16.7%) and numeric switch (6.7%) for answer reconstruction (0%) are relatively small. Rewriter errors mainly stem from the rewriter model confusing some phrases that look similar or messing up the calculation units. Although imperfect, our design significantly reduces the errors compared to existing baselines by generating logically consistent synthesis data and reusing the Python tool with the data switch.

Conclusion

In this work, we propose a simple yet effective method for LLM numerical reasoning and protect the local data. We develop a context-aware synthesis strategy that ensures logical consistency while shifting query domains and a tool-based answer reconstruction approach that reuses remote-generated code snippets with local data. Our experimental results confirm the effectiveness of this approach, demonstrating substantial improvements in reasoning accuracy while mitigating data exposure. These findings pave the way for further advancements in privacy-preserving LLM collaboration systems, highlighting the potential for secure and efficient deployment in real-world applications.

References

Aahill. 2023. What is Azure AI Language - Azure AI services.

Akhtar, M.; Shankarampeta, A.; Gupta, V.; Patil, A.; Cocarascu, O.; and Simperl, E. 2023. Exploring the Numerical Reasoning Capabilities of Language Models: A Comprehensive Analysis on Tabular Data. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 15391–15405. Singapore: Association for Computational Linguistics.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Carey, A. N.; Bhaila, K.; Edemacu, K.; and Wu, X. 2024. DP-TabICL: In-Context Learning with Differentially Private Tabular Data. In *2024 IEEE International Conference on Big Data (BigData)*, 1552–1557. IEEE Computer Society.

Chen, W.; Ma, X.; Wang, X.; and Cohen, W. W. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

Chen, Y.; Li, T.; Liu, H.; and Yu, Y. 2023. Hide and seek (has): A lightweight framework for prompt privacy protection. *arXiv preprint arXiv:2309.03057*.

Chen, Z.; Chen, W.; Smiley, C.; Shah, S.; Borova, I.; Langdon, D.; Moussa, R.; Beane, M.; Huang, T.-H.; Routledge, B. R.; et al. 2021. FinQA: A Dataset of Numerical Reasoning over Financial Data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3697–3711.

Du, M.; Yue, X.; Chow, S. S.; Wang, T.; Huang, C.; and Sun, H. 2023. Dp-forward: Fine-tuning and inference on language models with differential privacy in forward pass. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2665–2679.

Hartmann, F.; Tran, D.-H.; Kairouz, P.; Cărbune, V.; et al. 2024. Can LLMs get help from other LLMs without revealing private information? *arXiv preprint arXiv:2404.01041*.

Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; and Choi, Y. 2019. The Curious Case of Neural Text Degeneration. In *International Conference on Learning Representations*.

Hong, J.; Wang, J. T.; Zhang, C.; Zhangheng, L.; Li, B.; and Wang, Z. 2024. DP-OPT: Make Large Language Model Your Privacy-Preserving Prompt Engineer. In *The Twelfth International Conference on Learning Representations*.

Huang, J.; Li, Z.; Fountalis, I.; and Naik, M. 2021. Numerical Reasoning over Legal Contracts via Relational Database. In *Workshop on Databases and AI*.

Kurakin, A.; Ponomareva, N.; Syed, U.; MacDermed, L.; and Terzis, A. 2023. Harnessing large-language models to generate private synthetic text. *arXiv preprint arXiv:2306.01684*.

Ma, T.; Du, J.; Huang, W.; Wang, W.; Xie, L.; Zhong, X.; and Zhou, J. T. 2025. LLM Knows Geometry Better than Algebra: Numerical Understanding of LLM-Based Agents in A Trading Arena. *arXiv preprint arXiv:2502.17967*.

Mahendra, R.; Spina, D.; Cavedon, L.; and Verspoor, K. 2024. Do Numbers Matter? Types and Prevalence of Numbers in Clinical Texts. In Demner-Fushman, D.; Ananiadou, S.; Miwa, M.; Roberts, K.; and Tsujii, J., eds., *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, 409–415. Bangkok, Thailand: Association for Computational Linguistics.

Papernot, N.; Abadi, M.; Erlingsson, Ú.; Goodfellow, I.; and Talwar, K. 2017. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. In *International Conference on Learning Representations*.

Siyan, L.; Raghuram, V. C.; Khattab, O.; Hirschberg, J.; and Yu, Z. 2024. PAPILLON: PrivAcy Preservation from Internet-based and Local Language MOdel ENsembles. *arXiv preprint arXiv:2410.17127*.

Staab, R.; Vero, M.; Balunović, M.; and Vechev, M. 2024. Beyond Memorization: Violating Privacy Via Inference with Large Language Models. In *International Conference on Learning Representations 2024*.

Tian, Z.; Zhao, Y.; Huang, Z.; Wang, Y.-X.; Zhang, N. L.; and He, H. 2022. Seqpate: Differentially private text generation via knowledge distillation. *Advances in Neural Information Processing Systems*, 35: 11117–11130.

Tong, M.; Chen, K.; Qi, Y.; Zhang, J.; Zhang, W.; and Yu, N. 2023. InferDPT: Privacy-preserving inference for black-box large language model. *arXiv preprint arXiv:2310.12214*.

Utpala, S.; Hooker, S.; and Chen, P.-Y. 2023. Locally Differentially Private Document Generation Using Zero Shot Prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 8442–8457.

Wang, J. G.; Wang, J.; Li, M.; and Neel, S. 2024. Pandora's White-Box: Increased Training Data Leakage in Open LLMs. *arXiv preprint arXiv:2402.17012*.

Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N. A.; Khashabi, D.; and Hajishirzi, H. 2022. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Wu, J.; Zhu, J.; and Liu, Y. 2025. Agentic Reasoning: Reasoning LLMs with Tools for the Deep Research. *arXiv preprint arXiv:2502.04644*.

Xie, C.; Lin, Z.; Backurs, A.; Gopi, S.; Yu, D.; Inan, H.; Nori, H.; Jiang, H.; Zhang, H.; Lee, Y. T.; et al. 2024. Differentially private synthetic data via foundation model APIs 2: text. In *Proceedings of the 41st International Conference on Machine Learning*, 54531–54560.

Yu, D.; Kairouz, P.; Oh, S.; and Xu, Z. 2024. Privacy-Preserving Instructions for Aligning Large Language Models. In *International Conference on Machine Learning*, 57480–57506. PMLR.

Yue, M.; Zhao, J.; Zhang, M.; Du, L.; and Yao, Z. 2024. Large Language Model Cascades with Mixture of Thought Representations for Cost-Efficient Reasoning. In *The Twelfth International Conference on Learning Representations*.

Yue, X.; Du, M.; Wang, T.; Li, Y.; Sun, H.; and Chow, S. S. 2021. Differential privacy for text analytics via natural text sanitization. *arXiv preprint arXiv:2106.01221*.

Zhang, K.; Wang, J.; Hua, E.; Qi, B.; Ding, N.; and Zhou, B. 2024. Cogenesis: A framework collaborating large and small language models for secure context-aware instruction following. *arXiv preprint arXiv:2403.03129*.

Zhao, Y.; Li, Y.; Li, C.; and Zhang, R. 2022. MULTIHIERTT: Numerical Reasoning over Multi Hierarchical Tabular and Textual Data. In *60th Annual Meeting of the Association for Computational Linguistics, ACL 2022*, 6588–6600. Association for Computational Linguistics (ACL).

Zhou, X.; Lu, Y.; Ma, R.; Gui, T.; Wang, Y.; Ding, Y.; Zhang, Y.; Zhang, Q.; and Huang, X.-J. 2023. TextObfuscator: Making pre-trained language model a privacy protector via obfuscating word representations. In *Findings of the Association for Computational Linguistics: ACL 2023*, 5459–5473.