FedCoT: Federated Chain-of-Thought Distillation for Large Language Models

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have emerged as a transformative force in artificial intelligence, demonstrating exceptional proficiency across various tasks. However, their deployment in resource-constrained environments and concerns over user data privacy pose significant challenges. In contrast, Small Language Models (SLMs) offer computational efficiency but often lag in performance. To address these issues, we propose FedCoT, a federated framework designed for the Chain-of-Thought (CoT) distillation of knowledge from LLMs to SLMs, while ensuring the preservation of clients' data privacy. FedCoT ensures secure and efficient knowledge transfer from an LLM on a high-powered server to an SLM on a resource-constrained client, while adhering to privacy requirements. Leveraging perturbed prompts and rationales generated through the CoT approach, the framework enhances the performance of the client's SLM without compromising user data privacy within a multi-task learning framework. We propose two privacy protection strategies: the Exponential Mechanism Strategy and the Adaptive Exponential Mechanism Strategy, which balance user prompt privacy and the usability of rationales. Empirical evaluation on various text generation tasks demonstrates the effectiveness of FedCoT in training task-specific SLMs with enhanced performance while prioritizing data privacy protection.

1 Introduction

Large Language Models (LLMs) have risen as a revolutionary force in artificial intelligence. Prominent LLMs, such as GPT-4 (OpenAI, 2023), LLaMA (Touvron et al., 2023), and Qwen (Bai et al., 2023), have garnered the attention of researchers and practitioners alike, demonstrating unparalleled proficiency across numerous tasks. Nevertheless, the sheer size of these models presents significant obstacles for real-world deployment, particularly in environments with limited resources (Kang et al., 2023; Fan et al., 2025). Meanwhile, as LLMs gain escalating popularity and widespread utilization, privacy concerns have moved to the forefront, especially when it comes to user data and LLMs inference. In contrast, Small Language Models (SLMs) often exhibit superior computational efficiency and faster convergence rates, rendering them perfectly suited for real-time applications or resource-constrained environments. Nonetheless, SLMs also possess certain drawbacks stemming from their performance limitations. The question then arises: *How can we effectively combine the predictive prowess of LLMs with the nimbleness of SLMs, all while adhering to privacy requirements?*

To address these challenges, we propose FedCoT, a federated framework designed for the *Chain-of-Thought (CoT)* (Wei et al., 2022) distillation of knowledge from LLMs to SLMs, while ensuring the preservation of clients' data privacy. FedCoT ensures secure and efficient knowledge transfer from an LLM on a high-powered server to an SLM on a resource-constrained client. The challenge lies in maintaining the privacy of client data while leveraging the server's LLM to aid in training the client's SLM for text generation tasks, thereby elevating its performance. FedCoT aims to bridge this gap, enabling secure and efficient knowledge transfer between LLM and SLM, and ultimately enhancing the capabilities of the SLM without compromising privacy.

As illustrated in Figure 1(a), within our framework, the process works as follows. Initially, the client transmits perturbed prompts to the server's LLM. These prompts are protected by the FedCoT prompt encoder, which employs Differential Privacy (DP) principles (Dwork, 2006; McSherry and Talwar, 2007), ensuring privacy protection. Subsequently, the server's LLM generates perturbed rationales from these prompts through the CoT approach and relays them back to the client. Upon

receiving these perturbed rationales, the client's rationales decoder reconstructs them into their original, aligned form corresponding to the raw prompt. Ultimately, the client utilizes CoT knowledge distillation (Hsieh et al., 2023; Li et al., 2023) to train its *Task-Specific SLM*. This process leverages both label data and rationales within a multi-task learning paradigm (Wei et al., 2022; Hsieh et al., 2023; Zhang and Yang, 2021). These rationales justify the predicted labels and serve as insightful guidance for training smaller and domain-specific models.

Previous endeavors to incorporate DP into language models, specifically through DP-SGD (Song et al., 2013), have primarily centered on navigating the delicate balance between utility and privacy. This is achieved by introducing calibrated noise into gradients or text representations during the model training process. Nonetheless, these methods inherently rely on a trusted server to gather data from data owners for model training (Chen et al., 2023), significantly limiting their applicability in scenarios where such trusted servers are not available, as is the case in our research context.

Within the FedCoT framework, to achieve a balance between preserving the privacy of user prompts and enhancing the usability of rationales, we introduce two privacy protection strategies: the *Exponential Mechanism Strategy* and the *Adaptive Exponential Mechanism Strategy*. In the *Exponential Mechanism Strategy*, we utilize an exponential mechanism to obfuscate the prompts (McSherry and Talwar, 2007; Yue et al., 2021; Chen et al., 2023), followed by decoding the perturbed rationales through In-Context Learning (ICL) (Dong et al., 2024; Tong et al., 2025). In the *Adaptive Exponential Mechanism Strategy*, we utilize an Encoder-Decoder SLM specifically designed to encode original prompts into perturbed prompts and subsequently decode perturbed rationales back into their original form. To effectively train this unified Encoder-Decoder SLM, we utilize a multi-task learning paradigm (Zhang and Yang, 2021), encompassing both the encoding and decoding training processes.

Our contributions are summarized as follows:

- **Federated Framework for CoT Distillation in LLMs**. We propose FedCoT, a novel federated framework that facilitates secure and efficient knowledge transfer from LLM to SLM in resource-constrained environments. FedCoT leverages CoT knowledge distillation to enhance Task-Specific SLM within the client. This process leverages rationales produced by the LLM on the server, thereby enriching the client-side SLMs with valuable task-related knowledge.
- **Privacy as a Priority.** FedCoT leverages an *Adaptive Exponential Mechanism Strategy* tailored for encoding prompt to ensure their obfuscation and decoding perturbed rationales. The strategies effectively balance user prompt privacy and the usability of rationales.
- **Empirical Evaluation and Enhanced Performance of Task-Specific SLM**. Through experiments on various text generation tasks, FedCoT demonstrates the effectiveness of its framework in training task-specific SLM with enhanced performance. By harnessing the rationales generated by the server-side LLM, FedCoT provides valuable task-specific knowledge to the SLM.

2 Related Work

2.1 Differential Privacy

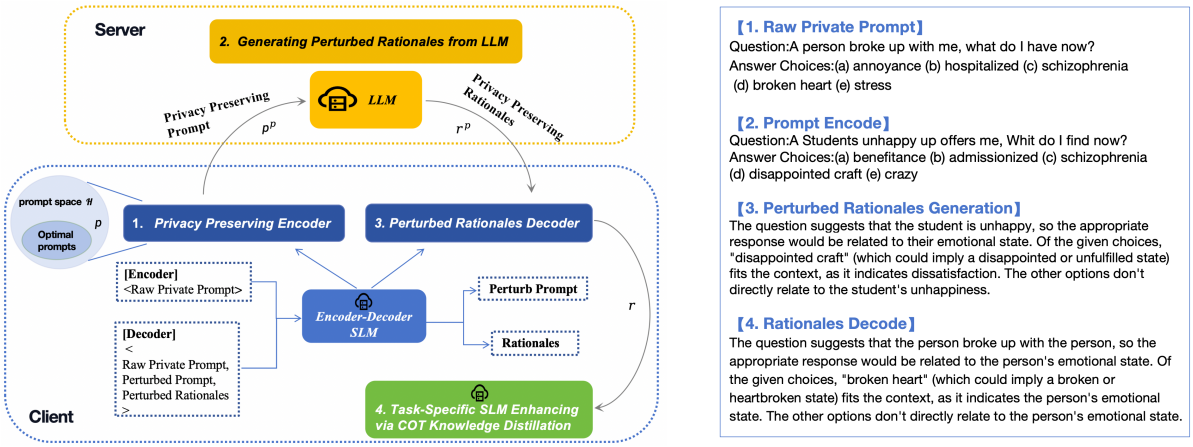
In this section, We briefly revisit two important definitions of differential privacy: ϵ-Differential Privacy and Exponential Mechanism (EM).

ϵ-Differential Privacy. Differential privacy (DP) (Dwork, 2006) is a rigorous mathematical framework that provides strong privacy guarantees for data analysis. It ensures that the output of an algorithm remains statistically indistinguishable whether a particular individual's data is included or excluded from the dataset. Formally, a randomized mechanism M provides ϵ-differential privacy if for all neighboring datasets D and D' (differing in at most one record) and for all sets S of possible outputs:

$$\Pr[M(D) \in S] \leq e^{\epsilon} \Pr[M(D') \in S] \quad (1)$$

where ϵ is the privacy budget that controls the level of indistinguishability.

Exponential Mechanism. The Exponential Mechanism (McSherry and Talwar, 2007) allows for the selection of an outcome from a set of possible outcomes with probabilities proportional to the exponential of their utility scores. Formally, given a utility function $u : D \times R \to \mathbb{R}$ that maps each dataset D and possible outcome r to a real-valued



(a) Overview of our proposed **FedCoT** framework.

(b) Privacy-Preserving Rationals Generation.

Figure 1: The overview of our proposed **FedCoT**. The FedCoT comprises four key components: (1) The *Prompt Encoder*, which perturbs user prompts to ensure privacy; (2) The *LLM*, generating perturbed rationales based on the perturbed prompts; (3) The *Perturbed Rationales Decoder*, which decodes the perturbed rationales back into a usable form; (4) The *Task-Specific SLM Enhancing via CoT Knowledge Distillation*, utilizing both original labeled data and filtered rationales data for multi-task learning.

score, the Exponential Mechanism $M(D, u, R)$ satisfies ϵ-differential privacy if it selects and outputs an $r \in R$ with probability:

$$\Pr[M(D) = r] \propto \exp\left(\frac{\epsilon u(D, r)}{2\Delta u}\right) \quad (2)$$

where Δu is the sensitivity of the utility function (in our work, we use cosine similarity as the utility function), defined as the maximum change in utility score when a single record is added or removed from the dataset:

$$\Delta u = \max_{D, D', r} |u(D, r) - u(D', r)| \quad (3)$$

2.2 Chain of Thought in Large Language Models

The Chain of Thought (CoT) approach has recently garnered significant attention in the realm of LLMs, thanks primarily to its remarkable ability to enhance the reasoning capabilities of these models. This innovative concept was first introduced by (Wei et al., 2022). Their research demonstrated that by prompting LLMs to produce a sequence of intermediary reasoning steps (rationales), the models' performance in handling intricate reasoning tasks could be notably boosted. Since the introduction of CoT, several studies have delved into its extensions and variations. For example, (Kojima et al., 2022) proposed the use of zero-shot CoT, where the model is prompted to generate rationales without relying on prior examples. CoT has also been applied to various domains, including arithmetic reasoning (Cobbe et al., 2021), commonsense reasoning (Klein and Nabi, 2020). Recent studies by (Hsieh et al., 2023; Ho et al., 2023; Li et al., 2023), have capitalized on the generated rationales as a form of insightful supervision to train smaller and domain-specific models. ***However, previous studies have not addressed the domain-specific data privacy issue that arises when LLMs and domain-specific smaller models are deployed across different parties. In our work, we endeavor to address this significant challenge.***

3 The Proposed FedCoT Framework

In this section, we introduce FedCoT, a federated framework designed for the *CoT* distillation of knowledge from LLMs hosted on a high-powered server to SLMs deployed on a resource-constrained client. The FedCoT framework can enhance the performance of SLMs while maintaining client data's privacy, leveraging the capabilities of LLM. We assume the server to be *semi-honest*, implying that it may attempt to recover the private data of the client from the information it observes. We illustrate the FedCoT in Figure 1(a), outline its training algorithm in Algorithm 1 (Appendix A), and detail its resource requirements in Appendix B.

3.1 Privacy Preserving Prompt Encoder

Before the client transmits its raw prompts to the server-side LLM, we need the privacy protection strategy to protect the raw prompts. In this section, we propose two privacy protection strategies:

1. **Exponential Mechanism Encoder Strategy**. In the first strategy, we utilize an exponential mechanism (McSherry and Talwar, 2007), which satisfies the criteria for the ϵ-DP. For detailed information about the exponential mechanism, please refer to Section 2.1.

   Let us consider an Exponential Mechanism $M(\cdot)$. Given a input prompt $p = \{x_i\}_{i=1}^{S}$ comprising S tokens, a set X encompassing all possible input tokens, and a set Y of all potential output tokens, *the mechanism* $M(\cdot)$ *is applied to each input token* $x_i \in p$. If x_i belongs to X, it is replaced with an output token y_i from Y. Through this process, we obtain a perturbed prompt $p^p = \{y_i\}_{i=1}^{S}$.

2. **Adaptive Exponential Mechanism Encoder Strategy**. *The tokens within a prompt differ significantly in terms of their importance and degree of privacy.* Applying a uniform privacy budget ϵ across all tokens may not lead to the most optimal solution. To further optimize the privacy-utility balance, we propose an *Adaptive Exponential Mechanism Encoder* strategy. This strategy is built upon the first exponential mechanism. In the *Adaptive Exponential Mechanism Encoder* strategy, we utilize an Encoder-Decoder SLM specifically designed to encode raw prompts into perturbed prompts and subsequently decode perturbed rationales back into their original form. This strategy involves two training process: encoding training process and decoding training process. In this section, we mainly focus on encoding training process.

   Initially, an encoding training process is required for the Encoder-Decoder SLM. Formally, let's denote a public dataset as $P = \{(p_i, p_i^{\epsilon}))\}_{i=1}^{N}$, where p_i represents raw private prompt, p_i^{ϵ} represents perturbed prompt generated using the first exponential mechanism with a privacy budget of ϵ. In the encoding training process, we train the Encoder-Decoder SLM: $g_{\phi}(p_i) \rightarrow p_i^{\epsilon}$. The details of encoding training process is illustrated in Algorithm 1.

   The Prompt Encoder objective can be formulated as follows:

$$\mathcal{L}_{\text{Encoder}}(\phi; \mathcal{P}) = \mathbb{E}_{(p,p^{\epsilon})\sim\mathcal{P}} \ell_{\text{CE}}(g_{\phi}(p), p^{\epsilon}) \tag{4}$$

   where ℓ_{CE} *is the cross-entropy loss.*

As illustrated in Figure1(b), we can observe an exemplary comparison between the original prompt and its perturbed prompt in Step 1 and Step 2. This perturbed prompt serves as the new, privacy-enhanced input for further processing.

3.2 Generating Perturbed Rationales from LLM

When the server-side LLM receives the perturbed prompt, we leverage the Chain-of-Thought (CoT) prompting technique introduced by (Wei et al., 2022) to generate rationales from the LLM using this perturbed prompt. These generated rationales, which are also perturbed, are then transmitted to the client. For instance, as illustrated in Figure 1(b), given a perturbed prompt in the Step 2, the LLM generates perturbed rationales in the Step 3.

3.3 Perturbed Rationales Decoder

Once the client receives the perturbed rationales from the server-side LLM, it must initiate a "decoder" process to decode the rationales. In this section, we also propose two strategies correspond to the two protection strategy of the prompt encoder module:

1. **Exponential Mechanism Decoder Strategy**. In the first decoding strategy, which corresponds to Exponential Mechanism Encoder strategy. Here, we utilize In-Context Learning (ICL) (Dong et al., 2024; Tong et al., 2025) with the Encoder-Decoder SLM to decode the perturbed rationales. we can input a sample $x_i = (p, p^p, r^p)_i$ into the Encoder-Decoder SLM to prompt the generation of rationales, where p represents raw private prompt, p^p represents perturbed prompt and r^p represents perturbed rationales generated from LLM. $(p^p, r^p)_i$ can be viewed as an example for Encoder-Decoder SLM in ICL. This allows the Encoder-Decoder SLM to generate rationales r_i that are aligned with the original, unperturbed prompt.

2. **Adaptive Exponential Mechanism Decoder Strategy**. In the second decoding strategy,

which corresponds to Adaptive Exponential Mechanism Encoder strategy. The rationales decoder module also use the same the Encoder-Decoder SLM with Section 3.1.

Initially, a decoding training process is required for the Encoder-Decoder SLM. Formally, let's denote a public dataset as $R = \{(x_i, r_i))\}_{i=1}^{N}$, where x_i represents an input, where $x_i = (p, p^p, r^p)_i$, p represents raw private prompt, p^p represents perturbed prompt generated from Encoder-Decoder SLM, r^p represents perturbed rationales generated from LLM. r_i represents the raw rationale of raw prompt p generated from LLM. In the decoding training process, we train the Encoder-Decoder SLM: $g_\phi(x_i) \rightarrow r_i$. The details of decoding training process is illustrated in Algorithm 1.

The Rationales Decoder objective can be formulated as follows:

$$\mathcal{L}_{\text{Decoder}}(\phi; \mathcal{R}) = \mathbb{E}_{(x,r)\sim\mathcal{R}} \ell_{\text{CE}}(g_\phi(x), r) \tag{5}$$

Subsequently, once the decoding training process of Encoder-Decoder SLM is finished, we can input a sample $x_i = (p, p^p, r^p)_i$ into the SLM, where r^p represents perturbed rationales generated from LLM. This allows the SLM to generate rationales r_i that are aligned with the original, unperturbed prompt.

We approach the training of the Encoder-Decoder SLM as a multi-task learning problem encompassing both the encoding and decoding training processes. *The multi-task learning objective for the Encoder-Decoder SLM can be formulated as follows:*

$$\mathcal{L}_1 = \mathcal{L}_{\text{Encoder}} + \mathcal{L}_{\text{Decoder}} \tag{6}$$

As illustrated in Figure1(b), we can observe an exemplary comparison between the perturbed rationales from LLM and its decoded rationales from SLM in Step 3 and Step 4. It's worth noting that although the SLM has the ability to generate aligned rationales independently, the quality often falls short due to its limited capabilities. By leveraging the perturbed rationales, we effectively transfer the powerful capabilities of the server-side LLM to enhance the Encoder-Decoder SLM, thereby improving the overall quality of the generated rationales.

3.4 Enhancing Task-Specific SLM via CoT Knowledge Distillation

In our work, we undertake the training of the client's Task-Specific SLM tailored for text generation tasks. Initially, we elaborate on the prevalent framework for learning task-specific models. Leveraging this established framework, we enhance it by integrating rationales produced from the rationales decoder module into the training process. Formally, let's denote a dataset as $D = \{(x_i, (y_i, r_i))\}_{i=1}^{N}$, where x_i represents an input, y_i represents the associated expected output label, and r_i is the corresponding desired rationale.

We conceptualize learning with rationales as a *multi-task learning* problem. Specifically, we train the model $f_\omega(x_i) \rightarrow (y_i, r_i)$ to accomplish not just the prediction of task labels but also the generation of the corresponding rationales based on textual inputs. This multi-task training ensures that our model not only produces accurate predictions but also provides insightful justifications for its decisions. By doing so, we enhance the transparency and explainability of the model. *The multi-task learning objective for the Task-Specific SLM can be formulated as follows:*

$$\mathcal{L}_2 = \mathcal{L}_{\text{Label}} + \mathcal{L}_{\text{Rationale}} \tag{7}$$

where $\mathcal{L}_{Label}$ is the label prediction loss:

$$\mathcal{L}_{\text{Label}}(\omega; \mathcal{D}) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \ell_{\text{CE}}(f_\omega(x), y) \tag{8}$$

and $\mathcal{L}_{Rationale}$ is the rationale generation loss:

$$\mathcal{L}_{\text{Rationale}}(\omega; \mathcal{D}) = \mathbb{E}_{(x,r)\sim\mathcal{D}} \ell_{\text{CE}}(f_\omega(x), r) \tag{9}$$

where ℓ_{CE} is the cross-entropy loss, $f_\omega(.)$ is the Task-Specific SLM model.

3.5 Privacy Analysis of FedCoT

The privacy-protection strategies in FedCoT implement a token-level Exponential Mechanism in feature space, adhering to the ϵ-DP principles. This mechanism provides mathematically provable privacy guarantees at the token-level granularity, as extensively validated in privacy-preserving NLP research (Yue et al., 2021; Chen et al., 2023; Tong et al., 2025). Our experimental results further validate this approach: when privacy budget is low, the rationales generated from perturbed prompts show significantly lower similarity to those from original prompts, demonstrating the effectiveness of our privacy protection while acknowledging the inherent privacy-utility trade-off.

4 Experiments

4.1 Setup

We have established a scenario to evaluate the performance of the FedCoT framework across a range of text generation tasks. This setup involves a client-server architecture, where the client holds two downstream SLMs: an *Encoder-Decoder SLM*, which specializes in encoder-decoder functionalities and a *Task-Specific SLM*, tailored for specific tasks. On the server-side, we host a LLM for more general and powerful text generation capabilities. Specifically, Table 1 outlines the detailed configurations of both the LLM and the SLMs. In our experimental setup, the *Encoder-Decoder SLM* and *Task-Specific SLM* are the identical architecture.

		SLM	
Setting	**LLM**	**Encoder-Decoder**	**Task-Specific**
Setting 1	LLaMa3 70B	Pythia-1.4B	Pythia-1.4B
Setting 2	Qwen-14B	Qwen-0.5B	Qwen-0.5B

Table 1: LLM and SLMs Setting of FedCoT.

Datasets and Evaluation Metrics. We conduct an evaluation of FedCoT on 4 QA datasets. Specifically, we include CommonsenseQA (CQA) (Talmor et al., 2019), OpenBookQA (OBQA) (Mihaylov et al., 2018), BoolQ (Clark et al., 2019), ArcE (Clark et al., 2018). For these datasets, we primarily use **Accuracy** as the evaluation metric. It's worth noting that in our experiments, all methods undergo zero-shot evaluation except FewShot(1-shot), and we use the *lm-evaluation-harness* package (Gao et al., 2023).

Baselines. Since we incorporate two distinct strategies in the prompt encoder and perturbed rationales decoder, we denote FedCoT method with the Exponential Mechanism Strategy as *FedCoT-E* and FedCoT method with the Adaptive Exponential Mechanism Strategy as *FedCoT-A*. We conduct a comparative analysis to evaluate the performance of our FedCoT framework, which comprises both *FedCoT-E* and *FedCoT-A*.

These baselines included:

- FewShot-LLM, which represents the few-shot capabilities of LLM on the server;
- FewShot-SLM, which represents the few-shot performance of SLM on the client;
- Standalone, where the client fine-tunes its local model using its own private dataset;
- Non-Private, where the client send its raw local prompt to server, get rationales from LLM and fine-tunes its local model like FedCoT, but without privacy-preserving.

4.2 Main Results

In this section, we undertake a comparative analysis of the task performance of FedCoT. We assess both the FedCoT-E and FedCoT-A methods against other baselines on Task-Specific SLM under the privacy budget $\epsilon = 3$. Our experiments encompass two model configurations: ***Setting 1*** (LLM: LLaMa3-70B, Encoder-Decoder SLM & Task-Specific SLM: Pythia-1.4B) and ***Setting 2*** (LLM: Qwen1.5-14B, Encoder-Decoder SLM & Task-Specific SLM: Qwen1.5-0.5B).

The results, as presented in Table 2, clearly illustrate that both FedCoT-E and FedCoT-A exhibit significantly better performance when compared to FewShot-SLM and Standalone methods. Furthermore, FedCoT-A demonstrates notably superior performance compared to FedCoT-E. Specifically, take the model Setting 1 as an example, FedCoT-E surpasses the Standalone method by 4.3%, 3.2%, 7.1%, and 5.1% in the CQA, OBQA, BoolQ, and ArcE datasets, respectively. Meanwhile, FedCoT-A demonstrates even greater superiority, exceeding the Standalone method by 5.7%, 4.6%, 6.7%, and 6% across the same datasets.

4.3 Performance Evaluation on various SLMs

In this section, we extend the evaluation of FedCoT's effectiveness to encompass various client-side SLMs. These SLMs include LLaMa2-1.3B (Xia et al., 2024), Qwen1.5-1.8B (Bai et al., 2023), and OPT-1.3B (Zhang et al., 2022). We have chosen LLaMa3-70B (Dubey et al., 2024) as LLM. Table 3 provides a clear illustration of how FedCoT(with ϵ = 3) consistently outperforms the Standalone method across various SLMs.

4.4 Ablation Study

Influence of Privacy Budgets. We delve into the influence of privacy budgets on the performance of FedCoT. To ensure experimental consistency, we fix the model configuration to *Setting 1* (as detailed in Table 1) for all subsequent ablation experiments. Table 4 presents an overview of FedCoT's performance across a range of privacy budgets ($\epsilon = 1, 3, 5, 10$).

As the privacy budget ϵ increases, the performance of both FedCoT-E and FedCoT-A exhibits

Model	Method	CQA	OBQA	BoolQ	ArcE
Setting 1	FewShot-LLM	70.29	80.66	90.08	82.69
	FewShot-SLM	21.19	26.60	52.11	28.91
	Standalone	42.43	38.73	73.07	40.33
	Non-Private	49.22	46.07	80.61	48.01
	FedCoT-E	46.70	41.93	80.02	45.42
	FedCoT-A	48.10	43.30	79.77	46.34
Setting 2	FewShot-LLM	80.9	82.8	85.2	80.3
	FewShot-SLM	25.7	28.6	59.7	40.7
	Standalone	55.7	43.4	78.4	50.3
	Non-Private	59.3	55.1	80.5	57.6
	FedCoT-E	57.6	50.8	79	52.6
	FedCoT-A	58.6	53.1	80.2	56.5

Table 2: We compare the performance of Task-Specific SLM trained with FedCoT-E ($\epsilon = 3$) and FedCoT-A ($\epsilon = 3$) against the Task-Specific SLM trained using baseline methods. We consider two model settings: **Setting 1** (LLM: LLaMa3-70B, Encoder-Decoder SLM & Task-Specific SLM: Pythia-1.4B) and **Setting 2** (LLM: Qwen1.5-14B, Encoder-Decoder SLM & Task-Specific SLM: Qwen1.5-0.5B)

Dataset	Method	LLaMa2	Qwen1.5	OPT
CQA	Standalone	61.5	57.8	56.42
	FedCoT-E	63.03	60.30	57.55
	FedCoT-A	64.27	62.21	60.18
OBQA	Standalone	47.53	52.60	40.93
	FedCoT-E	51.73	56.40	49.13
	FedCoT-A	49.8	57.20	48.4
BoolQ	Standalone	81.65	81.41	72.84
	FedCoT-E	83.94	82.59	82.46
	FedCoT-A	82.99	82.90	82.68
ArcE	Standalone	40.33	55.58	45.92
	FedCoT-E	54.11	61.07	49.67
	FedCoT-A	54.66	62.43	50.69

Table 3: We compare the performance of Task-Specific SLMs, which have been trained with FedCoT-E($\epsilon = 3$) and FedCoT-A($\epsilon = 3$), against Standalone across various SLMs, including LLaMa2-1.3B, Qwen1.5-1.8B and OPT-1.3B.

a notable uptick. Moreover, FedCoT-A consistently outperforms FedCoT-E under identical privacy budget conditions (ϵ). When compared alongside Table 2, it becomes evident that with a privacy budget escalated to $\epsilon = 10$, FedCoT-E surpasses the Standalone method by 5.6%, 6.1%, 6.3%, and 6.8% within the CQA, OBQA, BoolQ, and ArcE datasets, respectively. Similarly, FedCoT-A outperforms it by 4.3%, 7.1%, 6.8%, and 7%. Notably, across all evaluated datasets, at a privacy budget of $\epsilon = 10$, FedCoT attains performance levels comparable to Non-Private approaches, underscoring its proficiency and adaptability in striking a balance between privacy and utility.

Method	ϵ	CQA	OBQA	BoolQ	ArcE
FedCoT-E	1	45.63	42.13	78.91	44.84
	3	46.70	41.93	80.02	45.42
	5	46.50	43.35	80.17	46.70
	10	48.03	44.87	79.37	47.14
FedCoT-A	1	47.31	43.20	79.63	46.65
	3	48.10	43.30	79.77	46.34
	5	47.96	44.20	79.91	48.08
	10	47.74	45.81	79.86	47.30

Table 4: Comparison of the performance of Task-Specific SLM trained with FedCoT-E and FedCoT-A across **different privacy budgets** ϵ.

Influence of Perturbed Rationales Decoding. We undertake an analysis to investigate the effects of perturbed rationales decoding on FedCoT when $\epsilon = 3$. Table 5 offers a comparison of FedCoT's performance, contrasting the results when perturbed rationales decoding is employed (FedCoT-E w/ and FedCoT-A w/) versus when it is not (FedCoT-E w/o and FedCoT-A w/o). Specifically, FedCoT-E w/ surpasses the FedCoT-E w/o by 2%, 1.3%, 1.5%, and 0.6% in the CQA, OBQA, BoolQ, and ArcE datasets, respectively. Meanwhile, FedCoT-A w/ demonstrates even greater superiority, exceeding the FedCoT-A w/o by 1.8%, 1.6%, 0.7%, and 3% across the same datasets. The findings unequivocally demonstrate that FedCoT exhibits superior performance when perturbed rationales decoding is utilized, as compared to when it is absent.

Perturbed Rationales vs Original Rationales. We focus on analyzing the quality of the perturbed rationales (r^p) generated from the perturbed prompt of LLM based on FedCoT-E and FedCoT-A methods and compare them with the rationales (r) generated from raw prompt of the LLM. To evaluate the similarity between r^p and r, we use *TokenRatio* metric. A higher *TokenRatio* indicates a greater degree of similarity between the perturbed and original rationales. For more details about *TokenRatio*, please refer to Appendix E.

As shown in Table 6, with an increase in the privacy budget ϵ and a corresponding decrease in

Method	Dataset	Decoding	
		w/	w/o
FedCoT-E	CQA	46.70	44.79
	OBQA	41.93	40.6
	BoolQ	80.02	78.5
	ArcE	45.42	44.78
FedCoT-A	CQA	48.10	46.26
	OBQA	43.30	41.7
	BoolQ	79.77	79.06
	ArcE	48.08	45.13

Table 5: Comparison of Task-Specific SLM Performance in FedCoT: With vs. Without perturbed rationales decoding.

perturbation, both the *TokenRatio* of FedCoT-E and FedCoT-A have risen notably. Furthermore, in most of tasks, the *TokenRatio* of FedCoT-A is higher than that of FedCoT-E in the same level of privacy budget ϵ. The experimental results confirm that the *TokenRatio* observed in the perturbed rationales produced by both FedCoT-E and FedCoT-A, positively correlate with the privacy budget ϵ. This suggests that as the privacy constraints are relaxed (higher ϵ values), the perturbed rationales become more similar to the original rationales.

Method	ϵ	CQA	OBQA	BoolQ	ArcE
FedCoT-E	1	23.8	33	34.5	26.7
	3	30.8	45.26	48.5	44.7
	5	43.2	66.3	72.8	67.4
	10	48.5	75.8	85.4	74.5
FedCoT-A	1	34.5	37.9	47.1	20.7
	3	34.5	49.5	59.6	30
	5	45.2	69.6	77.4	36.2
	10	48.6	76.12	84.2	38.6

Table 6: We conduct a comparative analysis to assess the **perturbed rationales** produced by FedCoT-E and FedCoT-A methods against the **original rationales** that are directly generated from the raw prompt of the LLM. Metric used: TokenRatio.

Decoded Rationales vs Original Rationales. We delve into the quality analysis of the decoded rationales produced by the rationales decoder module based on FedCoT-E and FedCoT-A methods. We compare these decoded rationales against those generated directly from raw prompt of the LLM. We utilize the *TokenRatio* metric to assess their similarities.

As shown in Table 7, in contrast to FewShot-SLM, it becomes apparent that the decoded rationales' quality based on FedCoT-E and FedCoT-A methods isn't solely reliant on the locally decoded SLM. The perturbed rationales crafted by the LLM indeed fulfill their intended purpose. When juxtaposed with Table 6, it's clear that at comparable ϵ levels, the *TokenRatio* for the decoded rationales surpass those of the perturbed rationales in the FedCoT-E and FedCoT-A methods. This underscores the effectiveness of the rationales decoder module in the FedCoT-E and FedCoT-A methods.

Method	ϵ	CQA	OBQA	BoolQ	ArcE
FewShot-SLM	-	42.9	54.5	35.8	28.6
FedCoT-E	1	36	46.33	44.13	32.7
	3	39	53.77	53.1	46
	5	44.8	67.9	73.9	60.1
	10	48.4	75.1	85.4	66.7
FedCoT-A	1	41.1	60.36	62.8	42.19
	3	45.8	65.35	64.7	42.99
	5	50	75.5	72.9	44.3
	10	53.3	78.9	76.6	45.3

Table 7: We conduct a comparative analysis to assess the **decoded rationales** produced by FedCoT-E and FedCoT-A methods against the **original rationales** that are directly generated from the raw prompt of the LLM. Metric used: TokenRatio.

5 Conclusions

In this study, we introduce FedCoT, a federated framework designed to distill knowledge from LLMs to SLMs in resource-constrained environments. FedCoT facilitates secure knowledge transfer from LLMs to SLMs by leveraging perturbed prompts and rationales, thereby enhancing the performance of SLMs without compromising user privacy. We present two innovative privacy protection strategies, including an Adaptive Exponential Mechanism strategy, which effectively balance privacy preservation and the usability of rationales. Experiments on various text generation tasks demonstrate FedCoT's ability to enhance SLM performance with LLM support while prioritizing data privacy.

Limitations

While our proposed FeCoT framework demonstrates promising results in privacy-preserving knowledge transfer from LLMs to SLMs, it is important to acknowledge several considerations that could be addressed in future work. Firstly, the framework's performance benefits are contingent upon the server-side LLM's CoT reasoning capabilities. Although contemporary LLMs like GPT-4 and LLaMA exhibit strong reasoning skills, frameworks such as FedCoT may encounter limitations when deployed with less sophisticated LLMs. This suggests an opportunity for further research to enhance FedCoT's robustness against variability in LLM reasoning abilities. Secondly, our evaluation primarily focused on LLaMa and Qwen as the server-side LLMs, with client-side SLMs including Pythia, LLaMa, Qwen, and OPT. While these models are representative of current state-of-the-art architectures, extending testing to a more diverse set of LLMs could provide deeper insights into FedCoT's generalizability.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

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

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

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

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, et al. 2024. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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

Tao Fan, Guoqiang Ma, Yan Kang, Hanlin Gu, Yuanfeng Song, Lixin Fan, Kai Chen, and Qiang Yang. 2025. Fedmkt: Federated mutual knowledge transfer for large and small language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 243–255.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14852–14882.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017.

Yan Kang, Tao Fan, Hanlin Gu, Lixin Fan, and Qiang Yang. 2023. Grounding foundation models through federated transfer learning: A general framework. *arXiv preprint arXiv:2311.17431*.

Tassilo Klein and Moin Nabi. 2020. Contrastive self-supervised learning for commonsense reasoning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7517–7523.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. 2023. Symbolic chain-of-thought distillation: Small models can also "think" step-by-step. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2665–2679.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391.

OpenAI. 2023. Gpt-4.

Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. 2013. Stochastic gradient descent with differentially private updates. In *2013 IEEE global conference on signal and information processing*, pages 245–248. IEEE.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2024. Sheared LLaMA: Accelerating language model pre-training via structured pruning. In *The Twelfth International Conference on Learning Representations*.

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

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609.

A FedCoT Algorithm

The algorithm of FedCoT is outlined in Algorithm 1.

B FedCoT's Computational and Communication Overhead

FedCoT is designed to be efficient and scalable in resource-constrained environments. The communication overhead is minimal, with costs comparable to plaintext data transmission. Computational requirements are equivalent to standard SLM fine-tuning (SFT) on local tasks. Our experimental validation was conducted using NVIDIA V100 GPUs, demonstrating practical deployment feasibility.

C Rationales Generation through CoT

We utilize the rationales data generated by server-side LLM through chain-of-thought (CoT)(Wei et al., 2022)(Hsieh et al., 2023) technique to enhance the performance of the client's task-specific SLM. These rationales justify the predicted labels and serve as insightful guidance for training smaller and domain-specific models. Consider the following example: when asked "Question:A beaver is know for building prowess, their supplies come

Algorithm 1 FedCoT

Input:
T: total number of rounds;
$\mathcal{P}$: encoding training datasets;
$\mathcal{R}$: decoding training datasets;
$\mathcal{D}$: task-specific training datasets;
η_ϕ: learning rate of Encoder-Decoder SLM;
η_ω: learning rate of Task-Specific SLM.
Output: g_ϕ, f_ω.

▷ Multi-Task Training for Encoder-Decoder SLM based on Public Datasets $\mathcal{P}$ and $\mathcal{R}$.
for each epoch $t \in [T]$ **do**
  $\phi^{t+1} \leftarrow \phi^t - \eta_\phi \nabla \mathcal{L}_1$.
end for
▷ Generate p^p using the updated Encoder.
$p^p = \text{SLM}_{\text{Encoder}}(p)$.
▷ Generate perturbed rationales from LLM on the server.
$r^p = \text{LLM}(p^p)$.
▷ Decode perturbed rationales using the updated Encoder-Decoder SLM.
$r = \text{SLM}_{\text{Decoder}}(r^p)$.
▷ Multi-Task Training for Task-Specific SLM based on Datasets $\mathcal{D}$.
for each epoch $t \in [T]$ **do**
  $\omega^{t+1} \leftarrow \omega^t - \eta_\omega \nabla \mathcal{L}_2$.
end for

from where? Answer Choices: (a) british columbia (b) body of water (c) wooded area (d) pay debts (e) zoo". Utilizing the chain-of-thought (CoT) technique, the LLM can generate intermediate rationales like, "The answer must be the place where beavers get their supplies. Of the above choices, only wooded areas have the supplies that beavers need." Such rationales bridge the gap between the input and the final answer, often encapsulating valuable task-related knowledge. This knowledge would traditionally require extensive data for smaller and task-specific models to acquire. Therefore, we harness these rationales as enriched training material for small language models, employing a multi-task training paradigm that encompasses both label prediction task and rationale prediction task.

D More on Experimental Details

D.1 Hyperparameter Settings

SLM Parameters. During the training process for both the Encoder-Decoder SLM and the Task-Specific SLM, we specifically configured the parameters. We set the batch size to 32 and employed the AdamW optimizer. The maximum number of training steps ranged from 400 to 1500. Additionally, we assigned the values of 0.5 to both α and β. Furthermore, the learning rates for η_ϕ and η_ω were established at 5e-5.

D.2 Data Splitting

For the datasets CQA/OBQA/BoolQ//ArcE/, all splits (training, validation, and test) were downloaded from HuggingFace (Lhoest et al., 2021). During the training of the Encoder-Decoder SLM, we randomly divided the training data into two equal parts. One part was designated as the public dataset, while the other part was allocated as the private dataset for the client.

D.3 Dataset Licenses

For the datasets CQA/OBQA/BoolQ//ARC-E/ were downloaded from HuggingFace(Lhoest et al., 2021) and under Apache License, Version 2.0.

D.4 Machine Configuration

The experiments were conducted on machines equipped with 4 and 8 NVIDIA V100 32G.

E The Definition of TokenRatio Metric

TokenRatio$(r^{'}, r)$. This metric calculates the unique words(u) in $r^{'}$ and counts how many of these words are also present in r, denoted as i. The *TokenRatio* is then calculated as i divided by the total number of unique words in $r^{'}$ ($|u|$).

F Outperforming Standalone with 50% Data

In this section, we conduct an in-depth analysis to explore the influence of training data size on model performance. We compare the FedCoT method with the Standalone approach, varying the amount of training data used. Table 8 provides a clear illustration of how FedCoT(with $\epsilon = 3$) achieves superior performance even with significantly fewer training samples compared to Standalone. More specifically, when trained on merely **50%** of the complete CQA, OBQA, BoolQ, and ArcE datasets, both FedCoT-E and FedCoT-A either surpass or closely match the performance of Standalone method.

Dataset	Method	25%	50%	75%	100%
CQA	FedCoT-E	37.74	42.63	44.56	46.7
	FedCoT-A	39.28	44.77	44.00	48.1
	Standalone	-	-	-	42.43
OBQA	FedCoT-E	32.4	38.27	40.67	41.93
	FedCoT-A	34.07	38.08	42.00	43.3
	Standalone	-	-	-	38.73
BoolQ	FedCoT-E	69.96	72.26	77.67	80.02
	FedCoT-A	69.61	73.73	77.82	79.77
	Standalone	-	-	-	73.07
ArcE	FedCoT-E	37.79	41.42	42.22	45.42
	FedCoT-A	37.64	41.86	45.28	46.34
	Standalone	-	-	-	40.33

Table 8: We compare the performance of Task-Specific SLM trained with FedCoT-E($\epsilon = 3$) and FedCoT-A($\epsilon = 3$) against Standalone, across a range of dataset sizes from 25% to 100%. The '-' indicates a method does not apply to the corresponding dataset sizes.