PPC-GPT: Federated Task-Specific Compression of Large Language Models via Pruning and Chain-of-Thought Distillation

Anonymous ACL submission

Abstract

Compressing Large Language Models (LLMs) into task-specific Small Language Models (SLMs) encounters two significant challenges: safeguarding domain-specific knowledge privacy and managing limited resources. To tackle these challenges, we propose PPC-GPT, a innovative privacy-preserving federated framework specifically designed for compressing LLMs into task-specific SLMs via pruning and Chain-of-Thought (COT) distillation. PPC-GPT works on a server-client federated architecture, where the client sends differentially private (DP) perturbed task-specific data to the server's LLM. The LLM then generates synthetic data along with their corresponding rationales. This synthetic data is subsequently used for both LLM pruning and retraining processes. Additionally, we harness COT knowledge distillation, leveraging the synthetic data to further improve the retraining of structurally-pruned SLMs. Our experimental results demonstrate the effectiveness of PPC-GPT across various text generation tasks. By compressing LLMs into task-specific SLMs, PPC-GPT not only achieves competitive performance but also prioritizes data privacy protection.

1 Introduction

Large Language Models (LLMs), such as GPT-4 (OpenAI, 2023a) and LLaMA3-70B (Touvron et al., 2023a), boasting billions of parameters and remarkable text generation capabilities, have emerged as a transformative force in the realm of artificial intelligence. However, their training demands substantial computational resources (OpenAI, 2023b), and their colossal size poses significant hurdles for practical deployment, especially in resource-limited environments. Conversely, Small Language Models (SLMs), such as OPT-1.3B (Zhang et al., 2022) and Pythia-1.4B (Biderman et al., 2023), frequently demonstrate superior computational efficiency and accelerated response rates, making them ideally suited for real-time applications with constrained resources. Enterprises with constrained resources typically prefer deploying SLMs, as they can do so without the concern of potential data leaks, a risk that is heightened when utilizing remote LLMs. Yet, training an SLM from scratch, even the smallest billion-parameter models, entails considerable computational expenses that are financially prohibitive for most enterprises. Furthermore, SLMs exhibit inherent limitations that stem from their performance constraints.

In this work, we aim to tackle the following question: *Is it feasible to develop a task-specific and competitive SLM by harnessing an existing pretrained LLM for enterprises with limited resources, while ensuring compliance with privacy requirements?* To achieve this objective, we delve into structured pruning (Xia et al., 2023; Men et al., 2024; Kim et al., 2024), as a viable approach. Pruning is generally regarded as a strategy for compressing task-specific models by eliminating redundant parameters and expediting inference, all while maintaining task performance.

We identify two crucial technical challenges associated with this problem: Firstly, how can we ensure the privacy of task-specific data when enterprises with limited resources are unable to prune an LLM into an SLM independently? In such cases, the need to transmit task-specific data to a remote server equipped with powerful computing resources arises, a practice that is frequently unacceptable to most enterprises due to privacy concerns. Secondly, how can we ensure that the performance of the SLM remains comparable to that of the LLM? Structured pruning inevitably leads to some degree of performance degradation. To overcome these challenges, we introduce PPC-GPT, a privacy-preserving federated framework designed for compressing LLMs into task-specific SLMs via pruning and Chain-of-Thought (COT) distillation.

As depicted in Figure 1(a), the envisioned architecture of PPC-GPT comprises a high-performance server adept at deploying LLMs and facilitating their pruning into SLMs, coupled with a client endowed with more constrained computational capabilities for running SLMs. Within the confines of our framework, the workflow unfolds as detailed below. Initially, the client sends task-specific data, perturbed to ensure privacy, to the server. These data are protected by the Exponential Mechanism of Differential Privacy (Dwork, 2006; McSherry and Talwar, 2007; Tong et al., 2023), thereby guaranteeing privacy protection. Subsequently, the server-side LLM_{syn} generates synthetic data along with their corresponding rationales, based on these perturbed inputs. The server-side LLM_o, which represents the original model, undergoes pruning by PPC-GPT to yield the target SLM_t. This pruning process is informed by both the synthetic data and their associated rationales. Following the pruning of the LLM_o, the server retrains the target SLM_t through COT (Wei et al., 2022; Hsieh et al., 2023; Li et al., 2023) knowledge distillation, leveraging the same synthetic data and rationales. Lastly, the server dispatches the refined target SLM_t to the client, who then proceeds to retrain the SLM_t utilizing its locally private data.

Our contributions can be summarized as follows:

- **Privacy-Preserving Federated LLM Compression Framework.** We propose PPC-GPT, an innovative privacy-preserving federated framework tailored for compressing LLMs into task-specific SLMs. We use LLM to create synthetic data from DP-perturbed task data. This data helps evaluate layer importance in LLM, guiding the pruning process.

- **COT Knowledge Distillation Leveraging Synthetic Data.** We incorporate COT knowledge distillation, utilizing synthetic data generated from the LLM. This approach enhances the retraining of structurally-pruned SLMs, thereby improving their overall performance.

- **Empirical Assessment of LLM Compressing to Task-Specific SLM.** Through comprehensive compressing experiments conducted on the LLaMA and OPT models across a range of text generation tasks, PPC-GPT has demonstrated its efficacy in compressing LLMs into task-specific SLMs, achieving competitive performance.

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 (DP). *The Definition of ϵ-Differential Privacy (DP)* (Dwork, 2006). A randomized algorithm $M : D \rightarrow S$ is ϵ-Differential Privacy if for any two neighboring datasets $D_1, D_2 \in D$ that differ exactly in a single data sample, and for any output $O \subseteq S$:

$$P_r[M(D_1) \in O] \leq e^{\epsilon} P_r[M(D_2) \in O] \quad (1)$$

where ϵ is a privacy parameter. Smaller values of ϵ imply stronger privacy guarantees.

Exponential Mechanism. *The Definition of Exponential Mechanism* (McSherry and Talwar, 2007; Tong et al., 2023). For a given scoring function $u : X \times Y \rightarrow R$, a randomized mechanism $M(X, u, Y)$ is ϵ-DP compliant if it satisfies:

$$P_r[y|x] \propto exp(\frac{\epsilon \cdot u(x,y)}{2 \triangle u}) \quad (2)$$

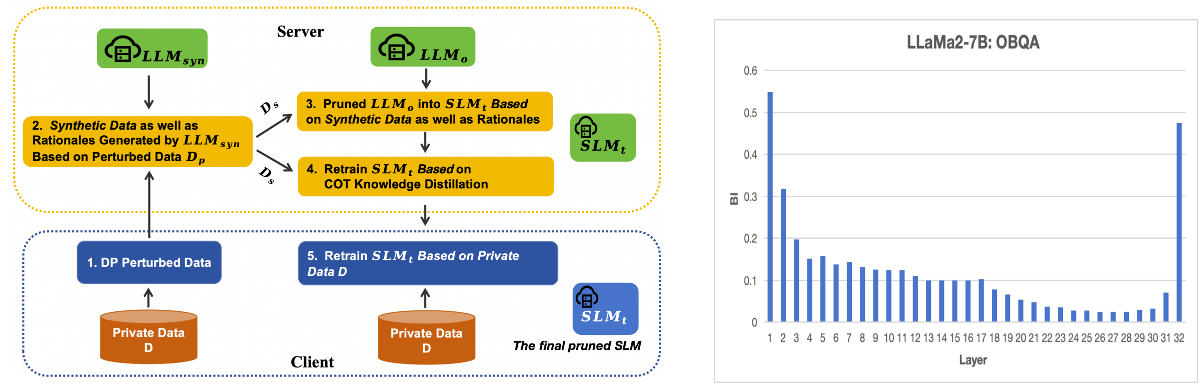
where the sensitivity $\triangle u$ is defined as:

$$\triangle u = \max_{x,x' \in X, y \in Y} |u(x,y) - u(x',y)| \quad (3)$$

2.2 Differential Privacy Synthetic Data

A practical approach to generating private synthetic data involves training a language model, such as LLaMa2-7B (Touvron et al., 2023a), on private data using DP through DP-SGD (Song et al., 2013; Bassily et al., 2014; Abadi et al., 2016). Subsequently, the DP model is sampled repeatedly to produce synthetic data (Mattern et al., 2022; Yue et al., 2022; Kurakin et al., 2023). Research conducted by (Mattern et al., 2022; Yue et al., 2022; Kurakin et al., 2023) demonstrates that training downstream models on DP synthetic data achieves performance comparable to training directly on real data with DP, thereby underscoring the high quality of the synthetic data.

However, a significant challenge arises because cutting-edge LLMs, like GPT-4, do not offer model weights, making DP fine-tuning impractical. Even for open-source LLMs, such as LLaMa3-70B (Touvron et al., 2023a), the process is resource-intensive. Meanwhile, these DP fine-tuning methods inherently rely on a trusted server to gather data from


Server
LLM_{syn}
LLM_o
2. *Synthetic Data* as well as Rationales Generated by LLM_{syn} Based on Perturbed Data D_p
D_s
D_s
3. Pruned LLM_o into SLM_t *Based* on *Synthetic Data* as well as Rationales
4. Retrain SLM_t *Based* on COT Knowledge Distillation
SLM_t
1. DP Perturbed Data
5. Retrain SLM_t *Based on Private Data D*
Private Data D
Private Data D
SLM_t
The final pruned SLM
Client
LLaMa2-7B: OBQA
0.6
0.5
0.4
0.3
0.2
0.1
0
BI
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
Layer

(a) Overview of our proposed **PPC-GPT** framework.

(b) Layer Importance Example: The significance of each layer, as indicated by the BI (Block Influence) value of LLaMa2-7B on the OBQA dataset, based on the PPC-GPT framework.

Figure 1: The overview of our proposed **PPC-GPT**. The PPC-GPT comprises four key components: (1) The *DP Perturbed Data*, which perturbs the client's data to ensure privacy; (2) The *Synthetic Data Generation*, responsible for creating new synthetic data and rationales based on the perturbed data; (3) The *Layer-Wise Structured Pruning*, a process that prunes original LLM_o to obtain the target smaller SLM_t; (4) The *Retraining SLM*, where the target SLM_t is retrained using both synthetic and original private data to restore accuracy.

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. In the context of this work, we operate within a client-server architecture where fine-tuning the LLM on the server is not an option.

2.3 Model Pruning

Model pruning, initially proposed by (LeCun et al., 1989) and subsequently enhanced by (Han et al., 2015), stands as a resilient and efficient strategy for mitigating model redundancy and attaining compression. This methodology branches into two primary techniques: *unstructured pruning and structured pruning*.

Unstructured pruning (Dong et al., 2017; Lee et al., 2019; Wang et al., 2020; Sun et al., 2023; Frantar and Alistarh, 2023) can obtain highly compressed models by directly pruning neurons, disregarding the model's internal architecture, which also causes unstructured sparsity and hard deployment. A more pragmatic and structured option is structured pruning. *Structured pruning* targets organized patterns for removal, encompassing entire layers (Jha et al., 2023), attention heads within Multi-Head Attention (MHA) mechanisms (Michel et al., 2019), hidden sizes in Feedforward Neural Networks (FFN) (Nova et al., 2023), as well as hybrid configurations (Kurtić et al., 2024). In recent times, there has been a surge in structured pruning research tailored specifically for LLMs. For example, ShortGPT (Men et al., 2024), LaCo (Yang et al., 2024), and Shortened LLaMa (Kim et al., 2024) concentrate solely on pruning depth (i.e., layer-wise). LLM-Pruner (Ma et al., 2023) eliminates coupled structures in relation to network width while preserving the layer count. Sheared-LLaMA (Xia et al., 2023) introduces a mask learning phase that is designed to pinpoint prunable components in both network width and depth. Our work falls in the category of structured pruning of LLMs.

3 The Proposed PPC-GPT Framework

In this section, we introduce PPC-GPT, an innovative privacy-preserving federated framework specifically designed for compressing LLMs into task-specific SLMs via pruning and COT distillation. We illustrate the PPC-GPT in Figure 1(a). The PPC-GPT comprises four key modules: *DP Perturbed Data*, *Synthetic Data Generation*, *Layer-Wise Structured Pruning*, and *Retraining*. We will elaborate on these modules in Section 3.2, Section 3.3, Section 3.4 and Section 3.5, respectively, following presenting the problem formulation we try to address in Section 3.1. Our aim is to provide a comprehensive understanding of how PPC-GPT addresses the intricate challenges associated with compressing LLMs while maintaining the client's data privacy and optimizing task-specific performance.

3.1 Problem Formulation

Given an LLM f_{θ_o} with parameters θ_o, which represents the original LLM that requires pruning, and a task-specific dataset $\mathcal{D}$ containing private data, our objective is to develop a target smaller, task-specific compressed SLM f_{θ_t} parameterized by θ_t. To acheive this, we seek to find the optimal pruning strategy $\mathcal{P}$ and retraining approach $\mathcal{R}$. The objective can be formulated as follows:

$$\begin{aligned} &\min_{\mathcal{P},\mathcal{R}} \mathcal{L}(\theta_t; \theta_o, \mathcal{D}) \\ s.t. \quad &|\theta_t| \ll |\theta_o| \quad \text{and} \quad \mathcal{L}_p(\mathcal{D}) < \delta \end{aligned} \tag{4}$$

where $\mathcal{L}(\theta_t; \theta_o, \mathcal{D})$ is the loss function measuring the performance of the compressed SLM on the task-specific dataset. $|\theta_t|$ and $|\theta_o|$ denote the number of parameters in the compressed and original models, respectively. $\mathcal{L}_p(\mathcal{D})$ is the privacy loss incurred due to the perturbation of the data to ensure differential privacy.

Our goal is to find the optimal pruning strategy $\mathcal{P}$ and retraining approach $\mathcal{R}$ that minimizes the overall loss, taking into account both the performance of the compressed SLM and the privacy protection of the task-specific data in the client. We assume the server to be *semi-honest*, meaning it may attempt to extract the client's private data from the information it receives.

3.2 DP Perturbed Data

We utilize an exponential mechanism (McSherry and Talwar, 2007; Yue et al., 2021; Chen et al., 2023) to perturb the local private data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, which satisfies the criteria for the ϵ-DP. For detailed information about the exponential mechanism, please refer to Section 2.1. We denote the perturbed dataset as $\mathcal{D}_p = \{(x_i^p)\}_{i=1}^{N}$, where x_i^p signifies an perturbed input based on the original local private dataset $\mathcal{D}$.

The Exponential Mechanism $\mathcal{M}$ is defined as a randomized algorithm that, given the original local private dataset $\mathcal{D}$, outputs the perturbed dataset $\mathcal{D}_p$ with probability proportional to the exponential of the utility score:

$$\mathcal{M}(\mathcal{D}) = \mathcal{D}_p \quad \text{with prob} \propto exp(\frac{\epsilon \cdot u(\mathcal{D}, \mathcal{D}_p)}{2 \triangle u}) \tag{5}$$

3.3 Synthetic Data Generation

When the server-side LLM_{syn} receives the perturbed data $\mathcal{D}_p$, the server initiates a procedure where LLM_{syn} generates fresh synthetic data along with their corresponding rationales based on these perturbed data. We denote the synthetic dataset as $\mathcal{D}_s = \{(x_i, (y_i, r_i))\}_{i=1}^{N_s}$, where x_i signifies an input, y_i signifies the corresponding expected output label, r_i signifies the desired rationale, and N_s represents the sample size of synthetic data.

We introduce a simple and efficient method for generating synthetic data, utilizing prompt engineering techniques and CoT technology:

1. **Question Generation.** We prompt LLM_{syn} to create a new question, starting from a perturbed question. To enhance the validity of these new created questions, we enforce three guidelines within the prompt: (1) the new question needs to conform to common knowledge, (2) it must be solvable on its own, independent of the original question, and (3) it should not contain any answer responses. Furthermore, we establish specific formatting standards for both questions and answers, customized to suit the needs of various datasets (Li et al., 2024).

2. **Answer Generation.** We instruct LLM_{syn} to generate a COT response for every newly created question. For consistency, we request LLM_{syn} to generate answers to the same question three times and check for agreement. If the answers differ, we reject the synthetic data.

3. **Rationale Generation.** We request LLM_{syn} to generate rationales for each synthetic data using the COT prompting technique.

Detailed prompt designs are presented in Appendix B. The generated synthetic data and their rationales are then employed for model pruning and retraining on the server-side.

3.4 Layer-Wise Structured Pruning

In neural networks exhibiting high redundancy, it is plausible that certain layers contribute minimally to the ultimate performance of the model. This phenomenon can be attributed to the homogeneous functionalities of these layers relative to others within the network architecture. LLMs are no exception, with varying levels of redundancy observed across their layers, particularly more pronounced in deeper layers. To eliminate these redundant layers, we require a metric that is inherent to the model itself for assessing the significance of each layer. This metric should be capable of quantifying the essentiality of each layer in relation to the model's overall functionality and performance.

To quantify the impact of each layer, we use a novel metric termed "Block Influence"(BI), which is proposed in the ShortGPT (Men et al., 2024). This metric is grounded in the hypothesis that a transformer block's significance is directly proportional to the extent it modifies the hidden states. Mathematically, the BI score for the i^{th} block is computed as:

$$\text{BI}_i = 1 - \mathbb{E}_{X,t}\left[\frac{X_{i,t}^T X_{i+1,t}}{||X_{i,t}||_2||X_{i+1,t}||_2}\right], \quad (6)$$

where X_i denotes the input to the i^{th} layer, and $X_{i,t}$ represents the t^{th} row of X_i.

On the server, we utilize the synthetic dataset $\mathcal{D}_s$, as described in Section 3.3, to compute the BI score for each layer of the LLM_o model, denoted as f_{θ_o}. This model represents the original LLM that requires pruning.

The original BI method (Men et al., 2024) relies solely on input and task label information, processed through a single forward pass: $f_{\theta_o}(x_i) \rightarrow y_i$. ***We further extend the BI computation to encompass two distinct facets of influence:*** $f_{\theta_o}(x_i) \rightarrow y_i$ and $f_{\theta_o}(x_i) \rightarrow r_i$. This enhancement not only facilitates the prediction of task labels but also enables the generation of corresponding rationales based on the inputs. ***Our novel BI score is determined as follows:***

$$\text{BI}_i = \text{BI}_{\text{Label},i} + \text{BI}_{\text{Rationale},i} \quad (7)$$

where $\text{BI}_{\text{Label},i}$ and $\text{BI}_{\text{Rationale},i}$ signify the influences pertaining to label predictions and rationale generation, respectively.

A higher BI score indicates a more crucial layer. An example is shown in the Figure 1(b). We arrange the layers in ascending order based on their BI scores and proceed to eliminate those with the lower BI scores.

3.5 Retraining

We employ the term "retraining" to designate the process of performance recovery subsequent to pruning. In this section, retraining is divided into two stages: (1) *Server-side Retraining*, and (2) *Client-side Retraining*.

Server-side Retraining. On the server side, we utilize the synthetic dataset $\mathcal{D}_s$, as described in Section 3.3, to retrain the pruned model, SLM_t. ***We propose COT knowledge distillation, guided by rationales generated by LLM_{syn}, to enhance the performance of SLM_t.*** Formally, we conceptualize the learning process with rationales as a *multi-task learning* problem (Zhang and Yang, 2021; Wei et al., 2022; Hsieh et al., 2023). Specifically, we train the model $f_{\theta_t}(x_i) \rightarrow (y_i, r_i)$ to achieve not only the prediction of task labels but also the generation of corresponding rationales based on textual inputs. This multi-task training ensures that our model produces not only accurate predictions but also insightful justifications for its decisions, thereby enhancing the model's transparency and explainability. The multi-task learning objective can be formulated as follows:

$$\mathcal{L} = \mathcal{L}_{\text{Label}} + \mathcal{L}_{\text{Rationale}} \quad (8)$$

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

$$\mathcal{L}_{\text{Label}}(\theta_t; \mathcal{D}_s) = \mathbb{E}_{(x,y)\sim\mathcal{D}_s}\ell_{\text{CE}}(f_{\theta_t}(x), y) \quad (9)$$

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

$$\mathcal{L}_{\text{Rationale}}(\theta_t; \mathcal{D}_s) = \mathbb{E}_{(x,r)\sim\mathcal{D}_s}\ell_{\text{CE}}(f_{\theta_t}(x), r) \quad (10)$$

where ℓ_{CE} denotes the cross-entropy loss, $f_{\theta_t}(\cdot)$ denotes the SLM_t model.

Client-side Retraining. On the client side, we utilize local private data $\mathcal{D}$ to further retrain the pruned model, SLM_t, once it has been received from the server. ***Our work encompasses conventional training, leveraging ground truth labels to further enhance the performance of SLM_t.*** Formally, the label prediction loss for this dataset $\mathcal{D}$ is formulated as follows:

$$\mathcal{L}_{\text{Label}}(\theta_t; \mathcal{D}) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\ell_{\text{CE}}(f_{\theta_t}(x), y) \quad (11)$$

4 Experiments

4.1 Setup

We have devised a scenario to assess the performance of the PPC-GPT framework across various text generation tasks. This setup employs a client-server architecture, where the server hosts a LLM for synthetic data generation, denoted as LLM_{syn}. Specifically, we have selected LLaMa3-70B (Dubey et al., 2024) for this purpose. For model pruning, we utilize LLaMA2-7B (Touvron et al., 2023b) and OPT-6.7B (Zhang et al., 2022) as the source models, denoted as LLM_o. In the default setting, the privacy budget $\epsilon = 3$, and the synthetic data ratio is 8.

Datasets and Evaluation Metrics. We conduct a comparative evaluation of PPC-GPT on QA datasets. Specifically, we include CommonsenseQA (CQA) (Talmor et al., 2018), OpenBookQA (OBQA) (Mihaylov et al., 2018), ARC-C (Clark et al., 2018), ARC-E (Clark et al., 2018), FiQA-SA (Maia 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 and we use the *lm-evaluation-harness* package (Gao et al., 2023).

Baselines. To evaluate the performance of our PPC-GPT framework, we conducted a comparative analysis against the following baselines:

- DenseSFT, where the client independently fine-tunes LLM_o using its private dataset.
- Plain-C, where the client independently prunes LLM_o using its private dataset (suppose the client can deploy LLM_o) and subsequently fine-tunes the pruned model.
- DP-Instruct-C (Yu et al., 2024), where the client finetunes generator (e.g., LLaMa2-1.3B) with DP-SGD and using synthetic datasets generated from generator to prune LLM_o and subsequently fine-tunes the pruned model with the private dataset.

4.2 Main Results

In our experiments, we extensively evaluated the performance of the proposed PPC-GPT framework across various text generation tasks. Notably, given that current structured pruning methods typically reduce parameters by no more than 30%, we conducted experiments with approximately 30% of the parameters pruned. Additional experiments exploring different parameter reduction proportions will be discussed in Section 5.5.

As shown in Table 1, the results highlight the effectiveness of PPC-GPT in compressing LLMs into task-specific SLMs while prioritizing data privacy protection, when compared to other baseline approaches. PPC-GPT outperforms the DP-Instruct-C method, which utilizes DP-SGD for privacy protection during model compression. Furthermore, PPC-GPT even surpasses the Plain-C method, which directly compresses the model using private data. Additionally, when compared to DenseSFT, the compressed model in PPC-GPT even outperforms the raw model on some datasets. Specifically, taking LLaMa2-7B for an example, in the LLaMa2-7B model, PPC-GPT outperforms the DP-Instruct-C method by 0.4%, 5.2%, 5%, 15.1%, and 1.8% on the CQA, OBQA, ARC-E, ARC-C, and FiQA-SA datasets, respectively. Similarly, PPC-GPT exceeds the Plain-C method by 0.7%, 2%, 4.5%, 8.2%, and 1.6% on the respective datasets.

5 Ablation Study

5.1 Impact of Different Privacy Budgets

In this section, we explore the impact of privacy budgets on the performance of PPC-GPT. Table 2 presents PPC-GPT's performance across a range of privacy budgets ($\epsilon = 1, 3, 5, 10$). Notably, when juxtaposed with Table 1, it becomes apparent that even with a privacy budget of $\epsilon = 1$, PPC-GPT outperforms the Plain-C method by 1.7% and 3.4% on the OBQA and ARC-E datasets, respectively, within the LLaMa2-7B model. Similarly, PPC-GPT exceeds it by 14% and 14.4% in the OPT-6.7B model. As the privacy budget ϵ increases, PPC-GPT's performance demonstrates a significant improvement, highlighting its proficiency and adaptability in achieving a balance between privacy and utility.

5.2 Impact of Different Synthetic Data

In this section, we explore the impact of synthetic data on PPC-GPT's performance, considering two dimensions: the synthetic data ratio and the inclusion of rationales in synthetic data.

Synthetic Data Ratio. Table 3 presents the performance of PPC-GPT across various synthetic data ratios (ratio = 1, 2, 4, 8). As the ratio of synthetic data increases, PPC-GPT's performance exhibits a substantial improvement, highlighting the crucial role of the synthetic data ratio and indicating that a higher amount of synthetic data results in further improvements. Specifically, PPC-GPT with the synthetic data ratio of 8 outperforms the ratio of 1 by 1.7% and 4.1% on the OBQA and ARC-E datasets, respectively, within the LLaMa2-7B model. Similarly, with the OPT-6.7B model, it exceeds the ratio of 1 by 4.2% and 7.6%.

Synthetic Data Rationales. We undertake an analysis to investigate the effects of rationales on

			DataSets				
Model	**Method**	**Ratio (%)**	**CQA**	**OBQA**	**ARC-E**	**ARC-C**	**FiQA-SA**
LLaMa2-7B	DenseSFT	0	$81.6_{\pm 0.54}$	$80.3_{\pm 0.50}$	$82.9_{\pm 0.18}$	$60.0_{\pm 0.42}$	$68.9_{\pm 1.66}$
	Plain-C	30	$77.6_{\pm 0.14}$	$77.9_{\pm 0.16}$	$79.7_{\pm 0.29}$	$54.0_{\pm 0.82}$	$71.1_{\pm 1.37}$
	DP-Instruct-C	30	$77.9_{\pm 0.62}$	$74.7_{\pm 1.32}$	$79.2_{\pm 0.33}$	$47.1_{\pm 4.10}$	$70.9_{\pm 0.83}$
	PPC-GPT	30	$78.3_{\pm 0.41}$	$79.9_{\pm 0.57}$	$84.2_{\pm 0.33}$	$62.2_{\pm 0.61}$	$72.7_{\pm 0.54}$
OPT-6.7B	DenseSFT	0	$75.4_{\pm 0.64}$	$60.0_{\pm 0.99}$	$65.8_{\pm 0.70}$	$31.4_{\pm 0.86}$	$70.0_{\pm 1.09}$
	Plain-C	30	$47.4_{\pm 1.12}$	$36.5_{\pm 1.48}$	$40.2_{\pm 0.89}$	$27.6_{\pm 0.37}$	$52.4_{\pm 1.37}$
	DP-Instruct-C	30	$58.7_{\pm 2.04}$	$39.7_{\pm 1.04}$	$44.5_{\pm 2.53}$	$28.6_{\pm 1.72}$	$54.5_{\pm 1.67}$
	PPC-GPT	30	$65.6_{\pm 0.95}$	$52.1_{\pm 0.96}$	$57.3_{\pm 0.16}$	$36.0_{\pm 0.59}$	$64.9_{\pm 1.26}$

Table 1: Performance Comparison of Compression Methods on LLMs.

			Privacy Budget(ϵ)			
Model	**Datasets**	**Stage**	**1**	**3**	**5**	**10**
LLaMa2	OBQA	S	65.4	67.1	67.9	69.4
		C	79.6	79.9	80.1	79.8
	ARC-E	S	78.8	80.4	79.9	79.5
		C	83.1	84.2	84.4	83.4
OPT	OBQA	S	35.7	36.3	36.1	38.8
		C	50.5	52.1	52.4	53.5
	ARC-E	S	49.1	50.4	49.3	50.5
		C	54.6	57.3	55.5	55.3

Table 2: Comparison of PPC-GPT's performance across **different privacy budgets** ϵ. **S** denotes the performance of target SLM_t on the server-side, while **C** represents the performance of target SLM_t on the client-side.

			Synthetic Data Ratio			
Model	**Datasets**	**Stage**	**1**	**2**	**4**	**8**
LLaMa2	OBQA	S	62.3	64.6	64.6	67.1
		C	78.2	78.3	78.5	79.9
	ARC-E	S	73.5	75.5	77.9	80.4
		C	80.1	80.8	82.3	84.2
OPT	OBQA	S	32.9	34.7	36.9	36.3
		C	47.9	50.2	51.5	52.1
	ARC-E	S	40.4	43.9	47.5	50.4
		C	49.7	52.3	54.9	57.3

Table 3: Comparison of PPC-GPT's performance across **different synthetic data ratio**.

PPC-GPT's performance. Table 4 compares PPC-GPT's performance between synthetic data with and without rationales (PPC-GPT w/ rationales and PPC-GPT w/o rationales). The findings demonstrate that PPC-GPT exhibits superior performance when the rationales of synthetic data is utilized, as compared to when it is absent. Specifically, PPC-GPT w/ rationales outperforms PPC-GPT w/o rationales by 0.8% and 0.9% on the OBQA and ARC-E datasets, respectively, within the LLaMa2-7B model. Similarly, with the OPT-6.7B model, PPC-GPT w/ rationales exceeds PPC-GPT w/o rationales by 7% and 9.1%.

5.3 Impact of Server-Side Retraining

In this section, we explore the impact of server-side retraining on the performance of PPC-GPT. Table 5 presents a comparison of PPC-GPT's performance with and without server-side retraining. The findings demonstrate that PPC-GPT exhibits superior performance when server-side retraining is utilized, as compared to when it is absent. Specifically, PPC-GPT w/ server-side retraining outperforms PPC-GPT w/o server-side retraining by 2% and 4.5% on the OBQA and ARC-E datasets, respectively, within the LLaMa2-7B model. Similarly, with the OPT-6.7B model, PPC-GPT w/ server-side retraining exceeds PPC-GPT w/o server-side retraining by 15.1% and 15.7%.

Model	Datasets	Stage	Rationales w/	Rationales w/o
LLaMa2	OBQA	S	67.1	65.9
		C	79.9	79.1
	ARC-E	S	80.4	77.9
		C	84.2	83.3
OPT	OBQA	S	36.3	31.1
		C	52.1	45.1
	ARC-E	S	50.4	43.2
		C	57.3	48.2

Table 4: Comparison of PPC-GPT's performance: with vs. without **rationales**.

Model	Dataset	Server:Retraining w/	Server:Retraining w/o
LLaMa2	OBQA	79.9	77.9
	ARC-E	84.2	79.7
OPT	OBQA	52.1	37.0
	ARC-E	57.3	41.6

Table 5: Comparison of PPC-GPT's performance: with vs. without **server-side retraining**.

5.4 Impact of Different Importance Metric

In this section, we explore the impact of different important metrics on PPC-GPT's performance:

Seq: The importance is directly correlated with the sequence order, where the shallower layers hold greater importance.

BI: BI mentioned in previous section 3.4.

Table 6 presents PPC-GPT's performance across different important metrics. The findings demonstrate that PPC-GPT with BI exhibits superior performance than PPC-GPT with Seq.

5.5 Impact of Different Model Pruning Ratio

In this section, we explore the impact of different model pruning ratio on PPC-GPT's performance. Table 7 presents the performance of PPC-GPT across different model pruning ratios (namely, 0%, 30%, 50%, and 70%). As the pruning ratio increases, the performance of PPC-GPT exhibits a decline.

Model	Datasets	Stage	Important BI	Important Seq
LLaMa2	OBQA	S	67.1	66.5
		C	79.9	79.9
	ARC-E	S	80.4	80.0
		C	84.2	83.9
OPT	OBQA	S	36.3	34.7
		C	52.1	48.3
	ARC-E	S	50.4	43.7
		C	57.3	51.7

Table 6: Comparison of PPC-GPT's performance across **different importance metrics**.

Model	Ratio (%)	DataSets OBQA	DataSets ARC-E
LLaMa2	0	80.3	82.9
	30	79.9	84.2
	50	74.4	76.8
	70	35.3	37.4
OPT	0	60.0	65.8
	30	52.1	57.3
	50	36.1	38.3
	70	30.9	33.2

Table 7: Comparison of PPC-GPT's performance across **different pruning ratios**.

6 Conclusions

In this study, we introduce PPC-GPT, a federated framework designed to compress LLMs into task-specific SLMs while preserving privacy. This is achieved through pruning and COT distillation. PPC-GPT employs a server-client architecture, wherein client data perturbed with DP is transmitted to the server for the generation of synthetic data. This synthetic data, along with its associated rationales, is subsequently used for both the pruning of LLMs and the retraining of pruned SLMs. Our experimental findings show that PPC-GPT successfully compresses LLMs into highly competitive SLMs, with a strong emphasis on safeguarding data privacy.

Limitations

While PPC-GPT shows promising results in compressing LLMs into task-specific SLMs while ensuring data privacy, it has several limitations. Firstly, PPC-GPT relies on the LLM's robust chain-of-thought capabilities to generate high-quality synthetic data and rationales. If the LLM lacks such capabilities, the performance of the compressed SLMs may be impacted. Furthermore, as observed in our experiments, the performance of PPC-GPT tends to degrade with higher pruning ratios. This indicates that optimizing the pruning strategy to strike a better balance between model size and performance remains an open challenge.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. 2014. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pages 464–473. IEEE.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

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.

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*.

Xin Dong, Shangyu Chen, and Sinno Pan. 2017. Learning to prune deep neural networks via layer-wise optimal brain surgeon. 30.

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.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pages 10323–10337. PMLR.

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.

Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander 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. *arXiv preprint arXiv:2305.02301*.

Ananya Harsh Jha, Tom Sherborne, Evan Pete Walsh, Dirk Groeneveld, Emma Strubell, and Iz Beltagy. 2023. How to train your (compressed) large language model. *arXiv preprint arXiv:2305.14864*.

Bo-Kyeong Kim, Geonmin Kim, Tae-Ho Kim, Thibault Castells, Shinkook Choi, Junho Shin, and Hyoung-Kyu Song. 2024. Shortened llama: A simple depth pruning for large language models. *arXiv preprint arXiv:2402.02834*.

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

Eldar Kurtić, Elias Frantar, and Dan Alistarh. 2024. Ziplm: Inference-aware structured pruning of language models. *Advances in Neural Information Processing Systems*, 36.

Yann LeCun, John Denker, and Sara Solla. 1989. Optimal brain damage. *Advances in neural information processing systems*, 2.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. 2019. Snip: Single-shot network pruning based on connection sensitivity.

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.

Chen Li, Weiqi Wang, Jingcheng Hu, Yixuan Wei, Nanning Zheng, Han Hu, Zheng Zhang, and Houwen Peng. 2024. Common 7b language models already possess strong math capabilities. *arXiv preprint arXiv:2403.04706*.

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. *arXiv preprint arXiv:2306.14050*.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720.

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. Www'18 open challenge: financial opinion mining and question answering. In *Companion proceedings of the the web conference 2018*, pages 1941–1942.

Justus Mattern, Zhijing Jin, Benjamin Weggenmann, Bernhard Schoelkopf, and Mrinmaya Sachan. 2022. Differentially private language models for secure data sharing. *arXiv preprint arXiv:2210.13918*.

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.

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. 2024. Shortgpt: Layers in large language models are more redundant than you expect. *arXiv preprint arXiv:2403.03853*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32.

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. *arXiv preprint arXiv:1809.02789*.

Azade Nova, Hanjun Dai, and Dale Schuurmans. 2023. Gradient-free structured pruning with unlabeled data. In *International Conference on Machine Learning*, pages 26326–26341. PMLR.

OpenAI. 2023a. Gpt-4.

OpenAI. 2023b. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

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.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2023. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

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

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Chaoqi Wang, Guodong Zhang, and Roger Grosse. 2020. Picking winning tickets before training by preserving gradient flow. *arXiv preprint arXiv:2002.07376*.

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. 2023. Sheared llama: Accelerating language model pre-training via structured pruning. *arXiv preprint arXiv:2310.06694*.

Yifei Yang, Zouying Cao, and Hai Zhao. 2024. Laco: Large language model pruning via layer collapse. *arXiv preprint arXiv:2402.11187*.

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

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

Xiang Yue, Huseyin A Inan, Xuechen Li, Girish Kumar, Julia McAnallen, Hoda Shajari, Huan Sun, David Levitan, and Robert Sim. 2022. Synthetic text generation with differential privacy: A simple and practical recipe. *arXiv preprint arXiv:2210.14348*.

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 multitask learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609.

A Implementation Details

A.1 Hyperparameter Settings

During the training process, we specifically configured the parameters. Specifically, we set the batch size to 32 and utilized the AdamW optimizer. The maximum number of training steps varied between 300 and 6400. Additionally, we established a learning rate of 5e-5. For the input and target lengths, we set the maximum question length to 64 and the maximum target length to 128. For the LoRA configuration of LLaMa2, we set the LoRA alpha to 32 and the LoRA rank to 8. In contrast, for the OPT model, we configured the LoRA alpha to 64 and the LoRA rank to 32. The Lora dropout for both models was set to 0.1.

A.2 Data Splitting

For the datasets, all splits (training, validation, and test) were downloaded from HuggingFace (Lhoest et al., 2021).

A.3 Dataset Licenses

All the datasets were downloaded from HuggingFace(Lhoest et al., 2021) and under Apache License, Version 2.0.

A.4 Machine Configuration

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

B Synthetic Prompt Templates

Table 8 and 9 provide prompt templates for question generation, answer generation, and rationale generation.

Tasks	Prompts
Question Generation	Please act as a professional teacher. Your goal is to promote research in advanced question-answering, probing a deeper understanding of both the topic (with salient facts summarized as an open book, also provided with the dataset) and the language it is expressed in. You will be given a multiple-choice question. Please create a new question and multiple choices based on the Given Question And Multiple Choices and following instructions. To achieve the goal, you have two jobs. # Please generate a similar but new question and multiple choices according to the Given Question And Multiple Choices. # Check the question and multiple choices by solving it step-by-step to find out if it adheres to all principles. You have eight principles to do this. # Ensure the new question only asks for one thing, be reasonable, be based on the Given Question And Multiple Choices, and can be answered with only one right choice. # Ensure the new questions requires multi-step reasoning, use of additional common and commonsense knowledge. # Ensure the new question is in line with common sense of life. # Ensure your student can answer the new question without the given question. If you want to use some numbers, conditions or background in the given question, please restate them to ensure no information is omitted in your new question. # Please DO NOT include solution in your question. # Make sure the choices in CREATED QUESTION AND CHOICES are list format, starts with [and ends with]. # Ensure only one choice in CREATED QUESTION AND CHOICES is right. # Ensure your output only has three lines, the first line is "CREATED QUESTION AND CHOICES:", the second line starts with "Question:", and the third line starts with "Choices". Given Question and Multiple Choices: {question}, {choices} Your output should be in the following format: CREATED QUESTION AND CHOICES: Question: <your created question> Choices: <your created choices>
Answer Generation	Please act as a professional teacher. Your goal is to accurately solve a multiple-choice question. To achieve the goal, you have two jobs. # Write detailed solution to a Given Question. # Write the final choice to this question. You have three principles to do this. # Ensure the solution is step-by-step. # Ensure the final answer is just a letter. # Use of additional common and commonsense knowledge. Given Question and Choices: {question}, {choices} Your output should be in the following format: SOLUTION: <your detailed solution to the given question> FINAL ANSWER: <your final choice to the question with only an uppercase letter>

Table 8: The prompt templates are used for generating questions and answers.

Tasks	Prompts
Rationale Generation	You are given the right Answer from Choices, please explain it in "Rationale" with few words. Please refer to the example to write the rationale. Try to generate logically clear and correct rationale. Reply in english only and use '<end>' to finish your rationale. Your reply format must strictly follow the provided example and reply rationale contents only! Example(s): Question: The sun is responsible for Choices: ['puppies learning new tricks', 'children growing up and getting old', 'flowers wilting in a vase', 'plants sprouting, blooming and wilting'] Answer: 'plants sprouting, blooming and wilting'. Rationale: The sun provides light and warmth, essential for the process of photosynthesis in plants, which enables them to grow, bloom, and eventually wilt due to natural life cycles. Please explain: Question: {question} Choices: {choices.text} Answer: {choices.text[choices.label.index(answerKey)]} Rationale:

Table 9: The prompt templates are used for generating rationale.