KnowledgeSG: Privacy-Preserving Synthetic Text Generation with Knowledge Distillation from Server

Wenhao Wang[1,3,4], **Xiaoyu Liang**[1], **Rui Ye**[2,4], **Jingyi Chai**[2,4],
Siheng Chen[2,3,4 *], **Yanfeng Wang**[2,3 *],
[1]Zhejiang University, [2]Shanghai Jiao Tong University,
[3]Shanghai AI Laboratory,
[4]Multi-Agent Governance & Intelligence Crew (MAGIC)
12321254@zju.edu.cn

Abstract

The success of large language models (LLMs) facilitate many parties to fine-tune LLMs on their own private data. However, this practice raises privacy concerns due to the memorization of LLMs. Existing solutions, such as utilizing synthetic data for substitution, struggle to simultaneously improve performance and preserve privacy. They either rely on a local model for generation, resulting in a performance decline, or take advantage of APIs, directly exposing the data to API servers. To address this issue, we propose *KnowledgeSG*, a novel client-server framework which enhances synthetic data quality and improves model performance while ensuring privacy. We achieve this by learning local knowledge from the private data with differential privacy (DP) and distilling professional knowledge from the server. Additionally, inspired by federated learning, we transmit models rather than data between the client and server to prevent privacy leakage. Extensive experiments in medical and financial domains demonstrate the effectiveness of *KnowledgeSG*. Our code is now publicly available at https://github.com/wwh0411/KnowledgeSG.

1 Introduction

The world has witnessed the tremendous success of large language models (LLMs) across a variety of tasks (Touvron et al., 2023b; OpenAI, 2023). Such success has attracted numerous parties to fine-tune their customized LLMs by leveraging their local private data (Wu et al., 2023; Xue et al., 2023; Zhou et al., 2024; Singhal et al., 2023). Nonetheless, training such LLMs on private data could cause significant privacy concerns, since LLMs are shown to memorize sensitive information from the training data (Carlini et al., 2021; Lukas et al., 2023).

To address this privacy issue, a series of methods have been proposed to circumvent the direct usage of private data by using synthetic data for substitution (Xie et al., 2024; Yue et al., 2023; Li et al., 2024a). Specifically, some methods use Application Programming Interface (APIs) to generate diverse instructions, directly exposing private data to the API server (Wang et al., 2022). While others rely solely on a local base model, which leads to a quality degradation in synthetic data and eventually lower model performance (Kurakin et al., 2024). Therefore, existing methods suffer from the tradeoff between privacy risk and model performance.

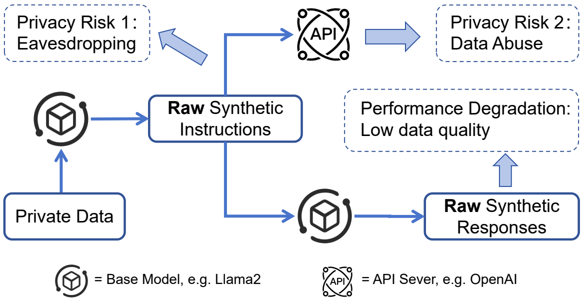

Privacy Risk 1: Eavesdropping
API
Privacy Risk 2: Data Abuse
Raw Synthetic Instructions
Performance Degradation: Low data quality
Private Data
Raw Synthetic Responses
= Base Model, e.g. Llama2
= API Sever, e.g. OpenAI

Figure 1: The dilemma of current synthetic data methods. API-based methods involve more privacy risks while methods based on local models face performance degradation due to lower synthetic data quality.

In this work, we aim to efficiently enhance synthetic data quality while maintaining strict privacy protection. To achieve this goal, we propose *KnowledgeSG* (**Knowledge**-based **S**ynthetic data **G**eneration), a novel client-server framework which leverages a professional server to assist the local client in data generation under theoretical privacy guarantee. Our framework compensates the quality gap between synthetic and original data observed in previous works (Jordon et al., 2022; Arnold and Neunhoeffer, 2021) by efficiently distilling knowledge from the professional model deployed on the server, rather than relying merely on the local model. Additionally, unlike API-based methods, we draw inspiration from federated learning (McMahan et al., 2017) by transmitting model

[*]Corresponding author.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 7677–7695
November 12-16, 2024 ©2024 Association for Computational Linguistics

weights instead of data for knowledge exchange, thereby improving privacy protection.

Specifically, on the client side, we fine-tune the local model with differentially privacy (DP) to learn local knowledge from private data within a privacy budget. For convenient and secure communication between the client and server, we transmit only the LoRA (Hu et al., 2021) adapter of the DP-finetuned model instead of directly transmitting private data. On the server side, raw synthetic instructions are first generated using the uploaded local model. These instructions are then judged by the professional model for quality filtration in an efficient manner (Jiang et al., 2023). Once filtered, the top instructions are fed directly into the professional model to generate accurate responses, bypassing the need to generate potentially incorrect responses from the local model. Finally, the DP-finetuned local model is further optimized by fine-tuning it with the top instructions and corresponding responses to boost its performance. Upon completion, the optimized model is transmitted back to the client, concluding the entire process.

We conduct a series of experiments on two privacy-sensitive domains: medicine and finance. The results prove the effectiveness of our proposed framework on both privacy and performance benchmarks. It is worth mentioning that our method gains a relative improvement of 120.39% than Non-Private training measured by medical free-form evaluation, even surpassing AlpaCare (Zhang et al., 2023), the professional model we deploy. To conclude, our main contributions are:

1. We propose a novel privacy-preserving client-server framework called *KnowledgeSG*, which enhances synthetic data quality by leveraging server-side knowledge distillation to assist the client in data generation.
2. We propose a novel server-side synthetic data generation method that employs a professional model to distill knowledge by providing both judgments and corrections for the raw synthetic data.
3. Extensive experiments validate the effectiveness of our proposed framework.

2 Related Work

2.1 Privacy Concerns with Fine-tuning on Private Data

Fine-tuning large language models is crucial for enhancing their instruction following ability and improving performance on certain downstream tasks (Conover et al., 2023; Wang et al., 2023; Jang et al., 2023). In order to deliver a satisfactory user experience (Zhao et al., 2024) or achieve professional-level expertise (Chaudhary, 2023; Xu et al., 2023), it is inevitable to fine-tune LLMs on user-related private data or proprietary data owned by institutions. However, recent studies (Kandpal et al., 2023; Carlini et al., 2021) have experimentally demonstrated that LLMs can memorize their training datasets, leaving possibilities of leaking private information through either simple prompts (Carlini et al., 2021; Nasr et al., 2023) or delicately designed attacks (Lukas et al., 2023; Gupta et al., 2022).

Continuing to improve the quality and coverage of fine-tuned large language models necessitates the development of alternative approaches to utilizing private data without memorizing it. To mitigate this issue, two mainstream solutions have emerged. The first involves fine-tuning LLMs with differential privacy techniques (Abadi et al., 2016; Yu et al., 2022), while the second focuses on substituting original private data with high-fidelity synthetic ones for fine-tuning (Yue et al., 2023; Xie et al., 2024).

2.2 Synthetic Text Generation

Two widely adopted approaches for generating private synthetic text in practice are In-Context Learning (ICL) (Dong et al., 2022; Chen et al., 2024; Ye et al., 2024a) and Self-Instruction (Wang et al., 2022). Largely relying on prompt design and the base model's comprehension, they suffer from either low data fidelity yielded by the base model, or privacy concerns requesting API servers. What makes it worse, with private data included directly in prompts, these methods pose an additional risk of revealing sensitive information.

Recently, researchers have recognized the feasibility and effectiveness of the DP generator method (Yu et al., 2024; Yue et al., 2023; Kurakin et al., 2024). This approach first trains an LLM on private data with DP, and then repeatedly samples the DP-finetuned model to generate synthetic text sequences. Although proved to gain improvements in distribution similarity, previous works primarily concentrate on generating diverse synthetic instructions. They ignore or skip the practical scenarios where responses are equally crucial for instruction tuning of LLMs. Moreover, current DP generator methods only focus on general knowledge, lead-

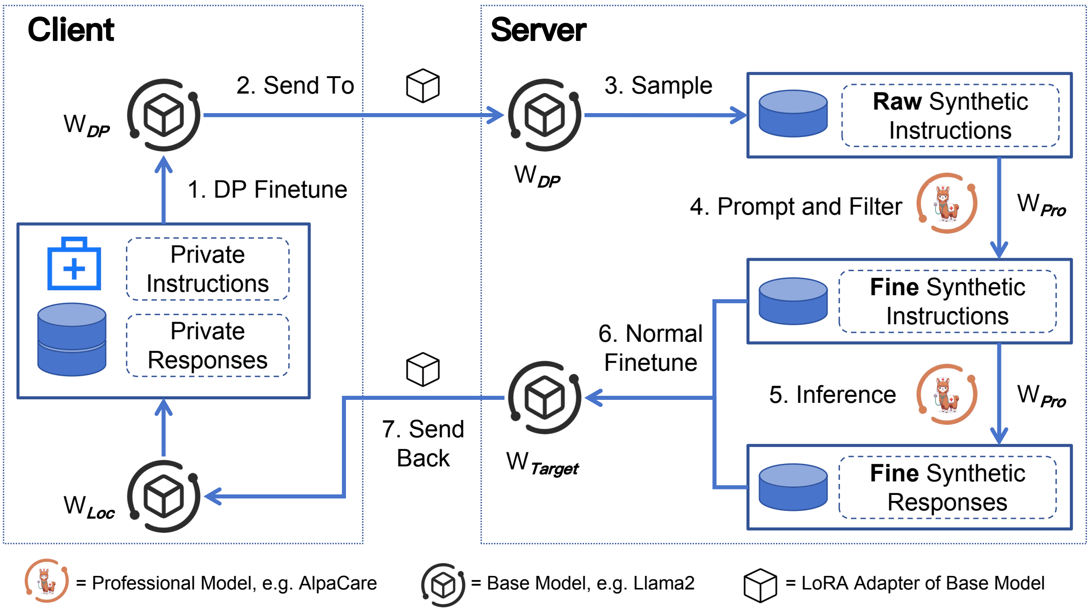


Figure 2: Overview of *KnowledgeSG*'s system architecture. $\mathbb{W}_{Loc}$: the local base model; $\mathbb{W}_{DP}$: DP-finetuned $\mathbb{W}_{Loc}$; $\mathbb{W}_{Target}$: the final target model; $\mathbb{W}_{Pro}$: the professional model. From left to right, $\mathbb{W}_{Loc}$ learns knowledge from private data on the client side and acquires knowledge distillation from $\mathbb{W}_{Pro}$ on the server side.

ing to significantly poorer performance in domain-specific scenarios such as finance and medicine where privacy draws considerable attention. Therefore, *KnowledgeSG* intends to improve the quality of both synthetic instructions and responses by distilling the professional model, especially on domain-specific tasks.

3 Method

3.1 Problem Setup

Let $\mathbb{D}_{Pri}$ represent the private dataset possessed by the client, which contains privacy from patients. $\mathbb{W}_{Loc}$ is the local base model pre-trained on general data that needs to acquire medical knowledge from $\mathbb{D}_{Pri}$. $\mathbb{W}_{Pro}$ refers to the professional model hosted by the server which is relatively larger than $\mathbb{W}_{Loc}$ and is assumed to have extensive knowledge of the medical domain. To formalize our problem setup, we assume that $\mathbb{D}_{Pri}$ used for instruction tuning consists of two components: *Instruction* and *Response*, both of which contain Personal Identifiable Information (PII), e.g. patients' names. Therefore, $\mathbb{D}_{Pri}$ can not be directly transmitted over networks due to privacy concerns. We present a detailed definition of PII in Appendix D.

Our ultimate objective is to generate a synthetic dataset $\mathbb{D}_{Syn}$ that maintains high data quality while containing no trace of PIIs. This allows us to fine-tune $\mathbb{W}_{Loc}$ on $\mathbb{D}_{Syn}$ to facilitate improvements in privacy-performance trade-off.

3.2 System Overview

We introduce a novel client-server framework called *KnowledgeSG* (**Knowledge**-based **S**ynthetic data **G**eneration), which aims to improve synthetic data quality and further promote model performance without violating privacy.

We attribute the quality gap between synthetic data and original private data to the comprehension deficiency of the local model $\mathbb{W}_{Loc}$ used for generation. Due to privacy concern, previous works place all generation on the client side without involving the server. To compensate for the aforementioned comprehension deficiency, we further extend previous setting into a client-server framework to leverage the knowledge from the server-side professional model $\mathbb{W}_{Pro}$. We give further elaboration of the quality gap in Appendix E.

The client-server framework of *KnowledgeSG* involves learning local knowledge from private data on the client side and acquiring knowledge distillation from the professional model on the server side. We also design a convenient transmitting unit to mitigate potential eavesdropping. In this way, we manage to achieve superior performance results while preventing memorization or leakage of the private dataset $\mathbb{D}_{Pri}$.

3.3 Client Side

On the client side, our framework is primarily designed to extract knowledge from the private data $\mathbb{D}_{Pri}$ without memorization and subordinately designed to be lightweight.

DP-based Local Learning. Due to its direct access to $\mathbb{D}_{Pri}$, the client side must comply with strict privacy constraint while still enabling effective knowledge learning from the private dataset $\mathbb{D}_{Pri}$. To achieve this primary goal, we adopt Differentially Private SGD (DP-SGD) (Abadi et al., 2016).

DP-SGD is a privacy-preserving optimization algorithm that improves upon traditional Stochastic Gradient Descend (SGD) by adding noise to the gradients during training. This noise ensures that the inclusion or exclusion of any individual data sample has a minimal impact on the resulting fine-tuned model $\mathbb{W}_{DP}$, offering strong privacy guarantees. We follow the first step of previous works (Yu et al., 2022; Kurakin et al., 2024; Yue et al., 2023) and adopt DP-SGD as our local training approach. The local base model $\mathbb{W}_{Loc}$ pre-trained on general corpora, is fine-tuned through DP-SGD, i.e. DP-finetuned on $\mathbb{D}_{Pri}$ to gain local knowledge under a privacy budget $(\epsilon, \delta) - DP$. This budget theoretically guarantees the process of DP-finetuning without any leakage of private information, providing the basis for us to transmit the fine-tuned model $\mathbb{W}_{DP}$ to the server later.

LoRA Adaptation. The second characteristic of the client side in *Knowledge* is lightweight, since we do not expect the client to have substantial hardware resources compared to the server. Therefore, we minimize the workload on the client by shifting the resource-intensive data generation process to the server.

Besides, we apply Low-Rank Adaptation (LoRA) (Hu et al., 2021) using the implementation of Wutschitz et al. (2022), as our training approach. LoRA is an efficient fine-tuning technique for large language models. It reduces the number of trainable parameters by introducing low-rank decomposition into the weight matrices of the model, allowing for faster and more resource-efficient adaptation to new tasks.

Even when considered relatively "small", the full size of the base model such as Llama2-7B, still occupies a significant amount of storage. The resulting inconvenience for transmitting the full model weights of $\mathbb{W}_{DP}$ is plain to see. In contrast, LoRA adaptation significantly reduces the transmission burden by allowing us to send only the LoRA adapter $\mathbb{A}_{DP}$, resulting in a far more manageable model size. Detailed comparison of model sizes is shown in Table 1.

Model Type	Params	Size
Base Model	6738 M	26 GB
LoRA Adapter	4.2 M	33 MB

Table 1: The parameter numbers and model sizes for Llama2-7B with & without LoRA rank of 16.

3.4 Server Side

The server side of *KnowledgeSG* is designed to improve data quality beyond what can be achieved by relying solely on the client. It operates through three stages: raw synthetic data generation, refinement of raw synthetic data and normal fine-tuning of local model.

Synthetic Instructions Generation. The first step on the server side is to recover the full model $\mathbb{W}_{DP}$ from $\mathbb{A}_{DP}$, assuming the server has the same base model $\mathbb{W}_{Loc}$ as the client prior to communication. Afterward, we prompt the DP-finetuned model $\mathbb{W}_{DP}$, which has knowledge of the private data $\mathbb{D}_{Pri}$, to generate raw synthetic instructions.

The post-processing property of DP (Dwork and Roth, 2014) ensures that once the model $\mathbb{W}_{Loc}$ has been fine-tuned with DP, sampling from the fine-tuned model $\mathbb{W}_{DP}$ incurs no extra privacy loss. As a result, when the LoRA adapter $\mathbb{A}_{DP}$ is uploaded to the server, it can generate synthetic data without exceeding the privacy budget $(\epsilon, \delta) - DP$.

Synthetic Instruction Filtration. During the second stage, to realize optimal results, we apply two compatible filtration methods distinguished by whether assistance from the professional model $\mathbb{W}_{Pro}$ is required.

Filtration without $\mathbb{W}_{Pro}$ uses similarity de-duplication via the BLEU score (Papineni et al., 2002). Bilingual Evaluation Understudy (BLEU) is a widely used automated evaluation metric for measuring the similarity between machine translation outputs and reference translations to assess translation quality. We adopt it to determine if an synthetic instruction is too similar to any example from the private dataset $\mathbb{D}_{Pri}$ to raise possibilities of leaking privacy. This method is much faster compared with the other model-based method.

For the filtration method involving $\mathbb{W}_{Pro}$, we prompt the raw instructions into $\mathbb{W}_{Pro}$ for judgements. If the instruction is domain-specific, $\mathbb{W}_{Pro}$ assesses whether it is relevant to its domain. If it is domain-specific, $\mathbb{W}_{Pro}$ judges an instructions based on whether this instruction is related to its domain. The detailed prompt we use is provided in Appendix H.

Efficient Knowledge Distillation. Without the need to derive loss from $\mathbb{W}_{Pro}$ (Flemings and Annavaram, 2024), we use a convenient method of knowledge distillation by feeding top instructions into $\mathbb{W}_{Pro}$ to generate preferable responses corresponding to these instructions after filtration (Xu et al., 2023; Wang et al., 2022; Jiang et al., 2023). This step is crucial as the knowledge is embedded in these responses which are subsequently distilled into the local model $\mathbb{W}_{DP}$ through fine-tuning.

Finally, we use the generated instructions and responses sorted by the IFD score (Li et al., 2023a) to normally (non-DP) fine-tune $\mathbb{W}_{DP}$ and obtain the desired model $\mathbb{W}_{Target}$. Further details and results regarding the IFD score are presented in Section 4.5. At this stage, DP-finetuning is not needed, as we assume the refined synthetic data contains no sensitive information.

3.5 Communication between Client & Server

Federated Model Transmission. Although synthetic data is supposed to contain no privacy, i.e. PIIs, two non-negligible concerns remain: (1) The size of the data prepared for fine-tuning are relatively larger than that of the LoRA adapter $\mathbb{A}_{DP}$. (2) Leakage of synthetic data can potentially reveal approximate data distribution or other sensitive information.

Therefore, inspired by federated fine-tuning of language models (Wei et al., 2020; Ye et al., 2024b), we propose to apply transmitting the fine-tuned version of model into our new setting which only has one client and one server, rather than directly transmitting data.

Proposed Transmitting Unit. Moreover, to reduce the potential risk of eavesdropping, i.e. an unauthorized party intercepts and steals the transmitted model, we introduce an efficient transmitting unit. Note that this unit is compatible and optional if the client using *KnowledgeSG* has no concerns about eavesdropping.

We start by sampling a small amount of data from public datasets, e.g. Alpaca (Taori et al., 2023), as the seed dataset $\mathbb{D}_{Seed}$, which is agreed and shared by the client and server at the beginning. Then we fine-tune the original base model $\mathbb{W}_{Loc}$ on $\mathbb{D}_{Seed}$ to create a full adaption of model weights and replace original $\mathbb{W}_{Loc}$ with the new model $\mathbb{W}'_{Loc}$. The local learning process described in Section 3.3 is based on $\mathbb{W}'_{Loc}$ afterwards. In this way, we make sure that, even if an adversarial eavesdropper intercepts the LoRA adapter $\mathbb{A}_{DP}$, he cannot recover our entire model with the old version of base model $\mathbb{W}_{Loc}$ instead of $\mathbb{W}'_{Loc}$.

4 Experiments

4.1 Basic Setups

Models and Datasets. If not otherwise mentioned, our base model is pre-trained Llama2-7B (Touvron et al., 2023b). We choose FinGPT (Yang, 2023) and AlpaCare (Zhang et al., 2023) as our professional models for financial and medical domains respectively. The dataset sample is kept to 500 for any comparison except the ablation study in Section 4.6. We use the name substitution technique in Appendix B.2 to pre-process datasets, preventing inaccurate evaluation on privacy.

Baselines. Our baselines comprise one None-Private approach, one private approach with DP-SGD (Abadi et al., 2016), and six private approaches using synthetic data generation, i.e. ICL (Dong et al., 2022), Self-Instruct (Wang et al., 2022), Self-Instruct-ICL, DP-Gene (Kurakin et al., 2024), DP-Instruct (Yu et al., 2024) and DP-Instruct-ICL. The detailed comparison of baselines is shown in Table 14 in Appendix F.3.

4.2 Privacy Evaluation

Setups. We study the privacy leakage of LLM by measuring the reconstruction rates following Lukas et al. (2023)[1]. In this approach, the attacker is given a sentence with multiple masked pieces of PII and asked to reconstruct the target PII from given candidates. The reconstruction rate is then calculated as the success ratio over attempt times.

In practice, for each sample in our training dataset, we mask all individual names and randomly choose one as the target. Then we use the PII reconstruction attack (Lukas et al., 2023) to predict the targeted individual name from a list of candidates and report the average prediction accuracy. Concretely, each time we sample 64 names as candidates from our datasets, making sure one of them is correct, and decode from the model using top-k sampling with k set to 40. We employ Flair[2] models (Akbik et al., 2018) to tag individual names in the datasets.

[1] https://github.com/microsoft/analysing_pii_leakage

[2] https://github.com/flairNLP/flair

Baselines	Medical	Inc	Financial	Inc
Random	1.56	0	1.56	0
Non-Private	97.13	95.57	96.23	94.67
ICL	5.47	3.91	7.40	5.84
Self-Instruct	1.46	-0.10	1.89	0.33
Self-Instruct-ICL	3.33	1.77	3.77	1.81
DP-Gene	2.26	0.70	2.52	0.96
DP-Instruct	1.07	-0.49	3.14	1.58
DP-Instruct-ICL	3.60	2.04	5.03	3.47
KnowledgeSG	0.87	-0.69	1.89	0.33

Table 2: Reconstruction rate comparison between different baselines on the medical and financial domains. *Inc* represents the increase of reconstruction rate between certain baseline and random guessing. Higher reconstruction rate indicates more memorization of the private data. Results in both domains demonstrate that synthetic data methods, including *KnowledgeSG*, achieve significantly better privacy protection than non-private methods.

Results. From Table 2, we can see that: (1) Using synthetic data instead of original data successfully reduces the PII reconstruction rate by a tremendous margin, demonstrating superior privacy protection over Non-Private method. (2) Differentially private training can preserve data privacy to a great content, but is still not on par with synthetic data approaches. (3) The privacy protection capabilities of different baselines exploiting synthetic data are closely aligned, with *KnowledgeSG* ranking first and ICL lagging behind, which validates the effectiveness of our method. This is reasonable in that ICL-related methods require few-shot examples from the original dataset to generate responses, thus introducing greater privacy risks.

4.3 Financial Benchmarks

Setups. We use the financial sentiment analysis dataset[3] as the training dataset (Yang et al., 2023). During the evaluation, we employ the code from Yang et al. (2023)[4] and consider four financial sentiment analysis benchmarks, including FPB (Malo et al., 2014), FIQA-SA (Maia et al., 2018), TFNS (Magic, 2022), and NWGI (Yang, 2023), where both accuracy and F1 score are measured. Besides, we also report the performance of GPT-3.5 (Ouyang et al., 2022) and GPT-4 (OpenAI, 2023) for reference. Since NWGI cannot be measured using GPT-3.5/4, we report the average metric of the first three and four evaluation datasets for an overall comparison.

Results. Table 3 demonstrates the results of our method and six other baselines using synthetic data generation on financial benchmarks. From the table, we can conclude that: (1) *KnowledgeSG* outperforms all other baselines on average, even better than using original private data, proving the effectiveness of knowledge distillation from professional model through our framework, not to mention our privacy-preserving nature. (2) For the FiQA-SA dataset, a large portion of the evaluation sample labels are *Neutral*. Following the evaluation benchmarks (Yang, 2023), we treat responses with no predictions (Positive/Negative/Neutral) as *Neutral*. This situation rarely happens except for pre-trained models that struggle with instruction following. Most of LLaMA2-7B's responses are classified as *Neutral*, thus explaining its unexpectedly strong performance on FiQA-SA. (3) Ignoring FiQA-SA, some synthetic generation baselines still perform even worse than the pre-trained Llama2 on FPB and TFNS. This phenomenon shows evidence for the quality issue we found for domain-specific data after generation. The *Gap Ratio*, as introduced in Appendix E.2 is 0.4682 for FPB and 0.3663 for TFNS, both below the heuristically drawn datum line of 0.5.

4.4 Medical Free-Form Evaluation

Setups. We utilize the HealthCareMagic-100k dataset[5] (Li et al., 2023c) as our training dataset, since it contains many individual names (e.g. see Fig 4). This dataset consists of real conversations between patients and doctors collected from the HealthCareMagic website.

Following Zhang et al. (2023), we conduct free-form evaluation by employing GPT-3.5-turbo (Zheng et al., 2023) to serve as a judge. For each instruction in the test dataset, the judge pairwise compares two responses resulting from the target model and THE reference model, respectively. We

[3] https://huggingface.co/datasets/FinGPT/fingpt-sentiment-train

[4] https://github.com/AI4Finance-Foundation/FinGPT

[5] https://huggingface.co/datasets/lavita/ChatDoctor-HealthCareMagic-100k

Evaluation	FPB		FiQA-SA		TFNS		NWGI		Avg:3		Avg:4	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
GPT-3.5	0.781	0.781	0.662	0.730	0.731	0.736	-	-	0.725	0.749	-	-
GPT-4	0.834	0.833	0.545	0.630	0.813	0.808	-	-	0.731	0.757	-	-
Llama2-7B	0.462	0.390	0.822	0.800	0.386	0.296	0.583	0.503	0.557	0.495	0.563	0.497
FinGPT v3.3	0.882	0.882	0.858	0.874	0.903	0.903	0.643	0.643	0.881	0.886	0.822	0.826
Non-Private	0.753	0.752	0.724	0.767	0.622	0.639	0.657	0.656	0.699	0.719	0.689	0.703
ICL	0.366	0.251	0.724	0.725	0.418	0.421	0.563	0.532	0.502	0.466	0.517	0.482
Self-Instruct	0.317	0.185	0.695	0.661	0.304	0.257	0.489	0.404	0.439	0.368	0.451	0.377
Self-Instruct-ICL	0.295	0.153	0.644	0.561	0.483	0.483	0.461	0.347	0.474	0.399	0.470	0.386
DP-Gene	0.308	0.181	0.618	0.519	0.397	0.371	0.453	0.366	0.441	0.357	0.444	0.359
DP-Instruct	0.296	0.285	0.615	0.489	0.439	0.439	0.421	0.300	0.450	0.404	0.443	0.378
DP-Instruct-ICL	0.332	0.299	0.666	0.588	0.399	0.345	0.472	0.382	0.465	0.410	0.467	0.403
KnowledgeSG	**0.779**	**0.775**	**0.791**	**0.806**	**0.782**	**0.743**	**0.658**	**0.658**	**0.784**	**0.775**	**0.752**	**0.745**

Table 3: Comparison with baselines on the financial benchmarks, where the sentiment analysis dataset from FinGPT (Yang et al., 2023) is used. Four evaluation datasets are considered, including FPB, FIQA-SA, TFNS, and NWGI. We also show results of GPT-3.5/4, Llama2-7B and FinGPT v3.3 for reference. We leverage Llama2-7B as the base model and FinGPT v3.3 as the professional model. The results demonstrate that *KnowledgeSG* outperforms all other baselines and is on par with the performance of GPT3.5/4.

employ text-davinci-003, GPT-3.5-turbo, GPT-4 and Claude-2 as reference models. To avoid positional bias, we evaluate each sample twice with exchanged positions of different responses generated by the test and reference models. We follow Li et al. (2023b) to score the models by calculating the win rate. Additional experiments on medical benchmarks are attached in Appendix C.1.

Results. From Table 4 and Table 10, we can conclude that: (1) Considering both benchmark and free-form results, *KnowledgeSG* consistently and significantly surpasses all other baselines in the medical domain. Particularly in the free-from evaluation, our method outperforms all other synthetic text generation baselines to a large margin, even doubling the performance of the None-private approach using original private data. (2) DP-based generation methods achieve much higher win rate scores than that of Self-instruction-based methods. This is expected because DP-based methods require additionally differentially private fine-tuning of the base model on private data. (3) The free-form results of *KnowledgeSG* surpassing AlpaCare (underlined in Table 4) highlight the immense potential of synthetic generation approaches which acquire knowledge distillation from a professional model, inspiring future research to further explore this area.

4.5 Data Quality Measurement.

Embedding Distribution Similarity. As shown in Yue et al. (2023), the similarity of synthetic data to the original data implicitly indicates its quality. Unlike typical natural language generation (NLG) tasks such as machine translation, which have ground truth references for evaluation, quantifying the similarity between synthetic and original private samples is non-trivial due to the absence of one-to-one mapping between them.

To measure the embedding distribution distance between synthetic and original data, we use sentence-transformers[6] library (Reimers and Gurevych, 2019) to embed both datasets. After that, we calculate the distance between these two embeddings using two widely-adopted metrics as Yue et al. (2023) does: (1) MAUVE[7] (Pillutla et al., 2023, 2021): MAUVE first clusters the samples in each dataset into a histogram (i.e. two histograms for two datasets), and then uses divergence frontiers (Liu et al., 2021) to calculate the divergence between the two histograms. (2) Fréchet Inception Distance (FID) (Heusel et al., 2018): FID calculates the feature-wise mean and covariance matrices of the embedding vectors and then measures the Fréchet distance between the two sets.

Note that the experiments in Section 4.5 are based on the same datasets we generated in Section 4.4. For paraphrase-MiniLM-L6-v2, its FID score is about 10 times the absolute value of other embedding models. Therefore for an unbiased comparison, we scale its score to match the magnitude of others.

Instruction Following Difficulty. Instruction following difficulty (IFD) introduced by (Li et al., 2023a), evaluates how much help the instruction provides for the generation of corresponding re-

[6] https://huggingface.co/sentence-transformers

[7] https://github.com/krishnap25/mauve

Evaluation	Text-davinci-003	GPT-3.5-turbo	GPT-4	Claude-2	Avg
AlpaCare (Zhang et al., 2023)	0.666	0.506	0.474	0.497	0.536
Llama2-7B	0.135	0.104	0.038	0.046	0.081
Non-Private	0.389	0.303	0.151	0.179	0.255
ICL (Dong et al., 2022)	0.380	0.280	0.141	0.166	0.241
Self-Instruct (Wang et al., 2022)	0.208	0.152	0.054	0.054	0.117
Self-Instruct-ICL	0.247	0.167	0.064	0.089	0.142
DP-Gene (Kurakin et al., 2024)	0.307	0.235	0.097	0.121	0.190
DP-Instruct (Yu et al., 2024)	0.255	0.184	0.076	0.097	0.153
DP-Instruct-ICL	0.382	0.295	0.187	0.199	0.266
KnowledgeSG	**0.776**	**0.530**	**0.457**	**0.488**	**0.562**

Table 4: Performance results and comparative analysis of free-form instruction evaluation in the medical domain. *KnowledgeSG* outperforms all other baselines and has a relative improvement of 120.39% than Non-Private method. Numbers with underlines represent performance surpassing the professional model AlpaCare (Zhang et al., 2023).

Baselines	Paraphrase-MiniLM-L6-V2		All-Mpnet-Base-V2		All-MiniLM-L6-V2		Avg	
	MAUVE (↑)	FID (↓)	MAUVE (↑)	FID (↓)	MAUVE (↑)	FID (↓)	MAUVE (↑)	FID (↓)
ICL	69.83	59.96	71.73	52.33	85.00	53.76	75.52	55.35
Self-Instruct	72.26	61.27	91.72	50.05	67.72	52.82	77.07	54.21
Self-Instruct-ICL	71.77	59.75	77.61	53.49	78.55	53.06	76.14	55.94
DP-Gene	83.23	59.41	89.58	51.42	84.47	53.58	85.76	54.80
DP-Instruct	81.29	**58.92**	83.18	50.10	89.14	51.95	84.54	53.66
DP-Instruct-ICL	81.97	60.00	92.20	**49.45**	82.06	52.36	85.41	53.94
KnowledgeSG	**90.77**	59.01	**96.48**	50.04	**92.82**	**51.75**	**93.36**	**53.60**

Table 5: Embedding distribution distance between the synthetic and original data measured by the MAUVE and FID score. Better similarity indicates better quality of the synthetic data. The results on average reaffirm that *KnowledgeSG* has best data quality compared to other baselines.

sponse. It compares the change of losses in model responses with and without the instructional context, and outputs a ratio as the final score. A lower IFD score indicates better quality of the evaluated sample. Thus we apply IFD score to measure the utility and quality of the generated instruction tuning datasets. The average IFD scores of dataset samples before filtering are presented in Table 3, exhibiting the disparity in the generation capabilities across various baselines. In practice, we deploy IFD score as the data filtering measure (Li et al., 2024b; Zhang et al., 2024) in our framework. However, in consideration of fair comparison with other baselines, we exclude it from the experiments in Sections 4.3 and 4.4.

Results. From Table 5 and Fig 3, We can conclude that: (1) Although the absolute values of MAUVE and FID are influenced by the specific settings used in its calculation, e.g. scalar scaling constants, the relative rankings of different synthetic datasets remain consistent. Still, *KnowledgeSG* achieves the best similarity measured by the MAUVE score. For the FID score, our method is only second to DP-Instruct-ICL, an improved version we adopt from Yu et al. (2024). (2) The leading performance of *KnowledgeSG* indicates better quality of synthetic data compared to other baselines. This is consistent with the performance results in Section 4.4 (3) For instruction following difficulty, the results conform to those of embedding distribution similarity, further proving the effectiveness of our proposed method.

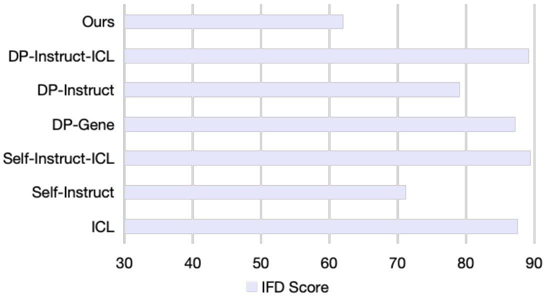


Figure 3: Instruction following difficulty of different baselines exploiting Llama2-7B as the base model. Lower IFD score indicates better quality of synthetic data. We evaluate on the synthetic datasets which are generated during the experiments in Section 4.4.

4.6 Ablation on Dataset Size

Setups. We perform an ablation study on dataset size to investigate its impact on the model's final performance through synthetic data generation. The training and evaluation setups are the

Dataset Size	500	1000	2000	3000
Non-Private	0.325	0.371	0.379	0.391
ICL	0.329	0.335	0.364	0.368
KnowledgeSG	0.708	0.724	0.747	0.757

Table 6: Ablations on dataset size. With more data involved, the model performance improves as expected.

same as Section 4.4. For a fair comparison, we make sure that each data sample is iterated 5 times by training the models for corresponding rounds wile keeping other parameters fixed (e.g., the 500-sample dataset is trained for 50 rounds, and the 1000-sample dataset for 100 rounds).

Results. For all methods shown in Table 6, the results indicate that as the amount of involved data increases, the performance of the trained model improves correspondingly. However, the last row of *KnowledgeSG* suggests that the improvement from accumulating additional data may reach a potential threshold. We leave further exploration of this for future work.

4.7 Transmitting Unit

Setups. We employ alpaca (Peng et al., 2023) and randomly select 50 samples to form our seed dataset $\mathbb{D}_{Seed}$. We first fine-tune Llama2-7B on $\mathbb{D}_{Seed}$, then replace the original model with its fine-tuned version. We assume the attacker only has access to the transmitting process, meaning he can intercept the LoRA adapter fine-tuned on the new base model. Without access to $\mathbb{D}_{Seed}$, the attacker can only attempt to merge the adapter with the original base model, i.e. open-sourced Llama2-7B, thus unable to reproduce the full performance of our model *Relative Drop* is calculated by $Relative\ Drop = \frac{(KnowledgeSG - Attacker)}{KnowledgeSG}$.

Results. Results in Table 7 show that the performance of model stolen by the attacker drops significantly compared to *KnowledgeSG*. This demonstrates that our model is not compromised, confirming the efficacy of proposed transmitting unit.

5 Discussions

5.1 Why not Scrubbing

The most intuitive way of privacy-preserving is PII scrubbing. PII scrubbing is a dataset curation technique that removes PII from text, relying on Named Entity Recognition (NER) to tag PII. In practice, using scrubbing to mask or add noise to original data, is flawed and must balance the trade-off between minimizing disclosure and preserving the utility of the dataset. Nonetheless, modern NER has mixed recall of 97% for names and 80% for care unit numbers on medical data (Vakili et al., 2022; Lukas et al., 2023), indicating that many PIIs are still retained after scrubbing.

Evaluation	Avg:3		Avg:4	
	Acc	F1	Acc	F1
Llama2-7B	0.557	0.495	0.563	0.497
KnowledgeSG	0.784	0.775	0.752	0.745
Attacker	0.419	0.343	0.428	0.350
Relative Drop	46.49%	55.76%	43.06%	53.08%

Table 7: Experiments of proposed transmitting unit. The *Relative Drop* in performance suggests that our model is safeguarded against the attacker during transmission.

5.2 Why not DP-SGD only

Fine-tuning models to satisfies DP can only address the risk of memorization. There is no protection during the data collection stage where the user instructions are exposed to human annotators for response generation (Yu et al., 2024). Moreover, using DP-SGD to prevent memorization by adding noise into the training process is destined to sacrifice performance. As proved in our experiments in Table 11, employing DP-SGD alone leads to considerable performance drop.

6 Conclusions

This paper addresses the challenge of preserving privacy while fine-tuning large language models on sensitive data. To improve the quality of synthetic data, an aspect often overlooked in previous works, we introduce a novel client-server framework called *KnowledgeSG*. Specifically, *KnowledgeSG* leverages knowledge distillation from a professional server, by prompting it to provide judgments and corrections for raw synthetic data generated by the DP-finetuned base model. Inspired by federated learning, *KnowledgeSG* transmits models rather than data through a specially designed transmitting unit to ensure privacy. We conduct extensive experiments, and the results validate the effectiveness of *KnowledgeSG*. The framework achieves a relative improvement of 120.39% compared to the Non-Private training, as measured by medical free-form evaluation. Additionally, *KnowledgeSG* significantly reduces the reconstruction rate from 97.13 to 0.87, demonstrating its strong privacy-preserving capabilities.

7 Limitations

While *KnowledgeSG* offers best privacy and performance trade-off across various domain-specific scenarios, its effectiveness on general tasks remains to be fully explored. Further experiments are needed to test its generalizability in broader contexts.

Also, *KnowledgeSG* involves more communication and computation cost than Non-Private finetuning, as it requires DP-finetuning the base model and leveraging a professional model for knowledge distillation. However, we believe these costs are justified, given the significant reduction in memorization concerns and the substantial performance improvements.

For future directions, we plan to conduct experiments on more general tasks and seek ways to optimize communication and computation costs. Additionally, we aim to make the deployment of *KnowledgeSG* more compatible and lightweight.

Acknowledgments

This research is supported by the National Key R&D Program of China under Grant 2021ZD0112801, NSFC under Grant 62171276 and the Science and Technology Commission of Shanghai Municipal under Grant 21511100900 and 22DZ2229005. We are grateful to Yifei Zhang, Changyu Miu, Huiyao Chen and Mengying Yuan, for their valuable discussions as well as feedback on the manuscript. We also thank TruthAI, for its GPU support.

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, CCS'16. ACM.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In COLING 2018, 27th International Conference on Computational Linguistics, pages 1638–1649.

Christian Arnold and Marcel Neunhoeffer. 2021. Really useful synthetic data – a framework to evaluate the quality of differentially private synthetic data. Preprint, arXiv:2004.07740.

Loubna Ben Allal, Niklas Muennighoff, Logesh Kumar Umapathi, Ben Lipkin, and Leandro von Werra. 2022. A framework for the evaluation of code generation models. https://github.com/bigcode-project/bigcode-evaluation-harness.

Hannah Brown, Katherine Lee, Fatemehsadat Mireshghallah, Reza Shokri, and Florian Tramèr. 2022. What does it mean for a language model to preserve privacy? In Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, pages 2280–2292.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. Preprint, arXiv:2012.07805.

Sahil Chaudhary. 2023. Code alpaca: An instruction-following llama model for code generation. https://github.com/sahil280114/codealpaca.

Huiyao Chen, Yu Zhao, Zulong Chen, Mengjia Wang, Liangyue Li, Meishan Zhang, and Min Zhang. 2024. Retrieval-style in-context learning for few-shot hierarchical text classification. Preprint, arXiv:2406.17534.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: An open platform for evaluating llms by human preference. Preprint, arXiv:2403.04132.

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

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Shizhe Diao, Rui Pan, Hanze Dong, Ka Shun Shum, Jipeng Zhang, Wei Xiong, and Tong Zhang. 2023. Lmflow: An extensible toolkit for finetuning and inference of large foundation models. arXiv preprint arXiv:2306.12420.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. arXiv preprint arXiv:2301.00234.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 320–335.

Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci., 9:211–407.

James Flemings and Murali Annavaram. 2024. Differentially private knowledge distillation via synthetic text generation. Preprint, arXiv:2403.00932.

Samyak Gupta, Yangsibo Huang, Zexuan Zhong, Tianyu Gao, Kai Li, and Danqi Chen. 2022. Recovering Private Text in Federated Learning of Language Models. Advances in Neural Information Processing Systems, 35:8130–8143.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. Proceedings of the International Conference on Learning Representations (ICLR).

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset. NeurIPS.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2018. Gans trained by a two time-scale update rule converge to a local nash equilibrium. Preprint, arXiv:1706.08500.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In ICLR.

Timour Igamberdiev, Thomas Arnold, and Ivan Habernal. 2022. DP-rewrite: Towards reproducibility and transparency in differentially private text rewriting. In Proceedings of the 29th International Conference on Computational Linguistics, pages 2927–2933, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Joel Jang, Seungone Kim, Seonghyeon Ye, Doyoung Kim, Lajanugen Logeswaran, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2023. Exploring the benefits of training expert language models over instruction tuning. In Proceedings of the 40th International Conference on Machine Learning, volume 202 of Proceedings of Machine Learning Research, pages 14702–14729. PMLR.

Yuxin Jiang, Chunkit Chan, Mingyang Chen, and Wei Wang. 2023. Lion: Adversarial distillation of proprietary large language models. Preprint, arXiv:2305.12870.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. Applied Sciences, 11(14):6421.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2567–2577.

James Jordon, Lukasz Szpruch, Florimond Houssiau, Mirko Bottarelli, Giovanni Cherubin, Carsten Maple, Samuel N. Cohen, and Adrian Weller. 2022. Synthetic data – what, why and how? Preprint, arXiv:2205.03257.

Nikhil Kandpal, Krishna Pillutla, Alina Oprea, Peter Kairouz, Christopher A. Choquette-Choo, and Zheng Xu. 2023. User Inference Attacks on Large Language Models. Preprint, arxiv:2310.09266.

Alexey Kurakin, Natalia Ponomareva, Umar Syed, Liam MacDermed, and Andreas Terzis. 2024. Harnessing large-language models to generate private synthetic text. Preprint, arxiv:2306.01684.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles.

Haoran Li, Qingxiu Dong, Zhengyang Tang, Chaojun Wang, Xingxing Zhang, Haoyang Huang, Shaohan Huang, Xiaolong Huang, Zeqiang Huang, Dongdong Zhang, Yuxian Gu, Xin Cheng, Xun Wang, Si-Qing Chen, Li Dong, Wei Lu, Zhifang Sui, Benyou Wang, Wai Lam, and Furu Wei. 2024a. Synthetic Data (Almost) from Scratch: Generalized Instruction Tuning for Language Models. Preprint, arxiv:2402.13064.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. 2024b. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. arXiv preprint arXiv:2402.00530.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2023a. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. ArXiv, abs/2308.12032.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023b. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.

Yunxiang Li, Zihan Li, Kai Zhang, Ruilong Dan, Steve Jiang, and You Zhang. 2023c. Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge. Cureus, 15(6).

Lang Liu, Krishna Pillutla, Sean Welleck, Sewoong Oh, Yejin Choi, and Zaid Harchaoui. 2021. Divergence Frontiers for Generative Models: Sample Complexity, Quantization Effects, and Frontier Integrals. In NeurIPS.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In International Conference on Learning Representations.

Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. 2023. Analyzing Leakage of Personally Identifiable Information in Language Models. In 2023 IEEE Symposium on Security and Privacy (SP), pages 346–363. IEEE Computer Society.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023. Wizardcoder: Empowering code large language models with evol-instruct. Preprint, arXiv:2306.08568.

Neural Magic. 2022. Twitter financial news sentiment. https://huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment.

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. Companion Proceedings of the The Web Conference 2018.

P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. Journal of the Association for Information Science and Technology, 65.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics, pages 1273–1282. PMLR.

Yuta Nakamura, Shouhei Hanaoka, Yukihiro Nomura, Naoto Hayashi, Osamu Abe, Shuntaro Yada, Shoko Wakamiya, and Eiji Aramaki. 2020. Kart: Privacy leakage framework of language models pre-trained with clinical records. arXiv preprint arXiv:2101.00036.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. 2023. Scalable Extraction of Training Data from (Production) Language Models. Preprint, arxiv:2311.17035.

OpenAI. 2023. Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. NIPS, 35:27730–27744.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Proceedings of the Conference on Health, Inference, and Learning, volume 174 of Proceedings of Machine Learning Research, pages 248–260. PMLR.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. arXiv preprint arXiv:2304.03277.

Krishna Pillutla, Lang Liu, John Thickstun, Sean Welleck, Swabha Swayamdipta, Rowan Zellers, Sewoong Oh, Yejin Choi, and Zaid Harchaoui. 2023. MAUVE Scores for Generative Models: Theory and Practice. JMLR.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. In NeurIPS.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. 2023. Large language models encode clinical knowledge. Nature, 620(7972):172–180.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

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.

Thomas Vakili, Anastasios Lamproudis, Aron Henriksson, and Hercules Dalianis. 2022. Downstream task performance of bert models pre-trained using automatically de-identified clinical data. In International Conference on Language Resources and Evaluation.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. 2023. How far can camels go? exploring the state of instruction tuning on open resources. arXiv preprint arXiv:2306.04751.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions. arXiv preprint arXiv:2212.10560.

Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. 2020. Federated learning with differential privacy: Algorithms and performance analysis. IEEE Transactions on Information Forensics and Security, 15:3454–3469.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. arXiv preprint arXiv:2303.17564.

Lukas Wutschitz, Huseyin A. Inan, and Andre Manoel. 2022. dp-transformers: Training transformer models with differential privacy. https://www.microsoft.com/en-us/research/project/dp-transformers.

Chulin Xie, Zinan Lin, Arturs Backurs, Sivakanth Gopi, Da Yu, Huseyin A. Inan, Harsha Nori, Haotian Jiang, Huishuai Zhang, Yin Tat Lee, Bo Li, and Sergey Yekhanin. 2024. Differentially Private Synthetic Data via Foundation Model APIs 2: Text. Preprint, arxiv:2403.01749.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. arXiv preprint arXiv:2304.12244.

Siqiao Xue, Caigao Jiang, Wenhui Shi, Fangyin Cheng, Keting Chen, Hongjun Yang, Zhiping Zhang, Jianshan He, Hongyang Zhang, Ganglin Wei, et al. 2023. Db-gpt: Empowering database interactions with private large language models. arXiv preprint arXiv:2312.17449.

Hongyang Yang. 2023. Data-centric fingpt. open-source for open finance. https://github.com/AI4Finance-Foundation/FinGPT.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023. Fingpt: Open-source financial large language models. FinLLM Symposium at IJCAI 2023.

Rui Ye, Rui Ge, Yuchi Fengting, Jingyi Chai, Yanfeng Wang, and Siheng Chen. 2024a. Leveraging unstructured text data for federated instruction tuning of large language models. arXiv preprint arXiv:2409.07136.

Rui Ye, WenHao Wang, Jingyi Chai, Dihan Li, Zexi Li, Yinda Xu, Yaxin Du, Yanfeng Wang, and Siheng Chen. 2024b. OpenFedLLM: Training Large Language Models on Decentralized Private Data via Federated Learning. In ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models.

Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. 2021. Opacus: User-friendly differential privacy library in PyTorch. arXiv preprint arXiv:2109.12298.

Da Yu, Peter Kairouz, Sewoong Oh, and Zheng Xu. 2024. Privacy-Preserving Instructions for Aligning Large Language Models. Preprint, arxiv:2402.13659.

Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A. Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. 2022. Differentially Private Fine-tuning of Language Models. Preprint, arxiv:2110.06500.

Xiang Yue, Huseyin Inan, Xuechen Li, Girish Kumar, Julia McAnallen, Hoda Shajari, Huan Sun, David Levitan, and Robert Sim. 2023. Synthetic Text Generation with Differential Privacy: A Simple and Practical Recipe. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1321–1342, Toronto, Canada. Association for Computational Linguistics.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2024. MAmmoTH: Building math generalist models through hybrid instruction tuning. In The Twelfth International Conference on Learning Representations.

Xinlu Zhang, Chenxin Tian, Xianjun Yang, Lichang Chen, Zekun Li, and Linda Ruth Petzold. 2023. Alpacare:instruction-tuned large language models for medical application. Preprint, arXiv:2310.14558.

Zhuo Zhang, Jingyuan Zhang, Jintao Huang, Lizhen Qu, Hongzhi Zhang, and Zenglin Xu. 2024. FedPIT: Towards Privacy-preserving and Few-shot Federated Instruction Tuning. Preprint, arxiv:2403.06131.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatgpt interaction logs in the wild. Preprint, arXiv:2405.01470.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. Preprint, arXiv:2306.05685.

Zhi Zhou, Jiang-Xin Shi, Peng-Xiao Song, Xiao-Wen Yang, Yi-Xuan Jin, Lan-Zhe Guo, and Yu-Feng Li. 2024. Lawgpt: A chinese legal knowledge-enhanced large language model. Preprint, arXiv:2406.04614.

A Privacy Analysis

A.1 Potential Privacy Risks

There is a potential privacy concern that the base model may have already encountered the private dataset $\mathbb{D}_{Pri}$ during pre-training. If this is the case, synthetic data generated by the base model $\mathbb{W}_{Loc}$ or its DP-finetuned variant $\mathbb{W}_{DP}$ may still violate privacy requirements (Igamberdiev et al., 2022). Additionally, if the professional model $\mathbb{W}_{Pro}$ has been trained on $\mathbb{D}_{Pri}$, it could inadvertently produce sensitive information such as individual names, when we utilize it to distill knowledge and improve the synthetic data generated by $\mathbb{W}_{DP}$.

To address this concern in *KnowledgeSG*, we will provide both theoretical elaborations and experimental results. It is important to note that the likelihood of private datasets being leaked and pretrained by models is minimal in real-world applications. Our work focuses on preventing further memorization when using sensitive data, rather than reversing any memorization that has already occurred.

Evaluation	GPT-3.5-turbo
Llama2-7B	12.96
Non-Private	0.254
ICL	0.133
KnowledgeSG	**0.499**

Table 8: Free-form evaluation results using medical-ai-chatbot as the private dataset.

A.2 Theoretical Privacy Elaborations

Interchangeability of Models. In our framework, both the base model and professional model are interchangeable. *KnowledgeSG* is not dependent on any specified LLM, e.g. Llama2-7B. The clients using *KnowledgeSG* can select any other LLM that has not been pre-trained on their private datasets to mitigate the risk.

Theoretical Guarantee of Differential Privacy. Based on previous works, we assert the privacy-preserving nature of our framework is justified by differential privacy theory. First, on the client side, we follow Abadi et al. (2016); Yue et al. (2023) to DP-fintuned the base model $\mathbb{W}_{Loc}$. This provides us with a strong theoretical guarantee against memorization within the privacy budget $(\epsilon, \delta) - DP$.

Second, on the server side, the post-processing property of DP (Dwork and Roth, 2014) ensures that once the model $\mathbb{W}_{Loc}$ has been fine-tuned with DP, sampling from the fine-tuned model $\mathbb{W}_{DP}$ does not result in extra privacy loss. Therefore, when the LoRA adapter $\mathbb{A}_{DP}$ is uploaded to the server, it can generate synthetic data without exceeding the privacy budget, mitigating associated privacy risks.

A.3 Experimental Results

Setups. To further validate the effectiveness of *KnowledgeSG* and ensure that no private data has been accessed by either the base model or the professional model, we conducted additional experiments using the ai-medical-chatbot dataset[8], which was collected and released six months later than Llama2-7B and AlpaCare. We adhere to the experimental setups described in Section 4.4 and also employ Llama2-7B as the base model.

Results. The results presented in Table 8, reaffirm the effectiveness of *KnowledgeSG*, regardless of whether the models had access to the private dataset. It also shows that *KnowledgeSG* can generalize well across different datasets. Additionally,

[8] https://huggingface.co/datasets/ruslanmv/ai-medical-chatbot

they demonstrate that *KnowledgeSG* generalizes well across different datasets. Llama2 trained on the ai-medical-chatbot dataset yields lower scores compared to its training on HealthCareMagic, indicating that the latter dataset may have higher quality.

Llama2 trained on the ai-medical-chatbot dataset yields lower scores compared to its training on HealthCareMagic, suggesting that the latter dataset may have higher quality.

B Additional Techniques

B.1 Filtration with Models

As mentioned in Section 3, filtration with model means that we prompt the professional model $\mathbb{W}_{Pro}$ with raw instructions for judgments. Then we filter out subpar instructions based on judgements.

For domain-specific settings such as the medical domain, the judgements are mainly based on whether the tested instructions are related to particular medical knowledge. We first prompt AlpaCare using the template written in Figure 10, then extract judgements from the model outputs. In experiments, we also try GPT-3.5-turbo as the domain classifier of instructions and receive acceptable results.

B.2 Name Substitution

In order to discard the possibility that the pre-trained model has already seen those individual names (e.g. *John, Trump*) in our training datasets $\mathbb{D}_{Pri}$, we ask GPT-4 (OpenAI, 2023) to generate hundreds of unique names (e.g. *Anastasija, Melangell*) to substitute the original names. This technique addresses the potential privacy risk discussed in Appendix A and pave the groundwork for accurate experiments in Section 4.2.

To evaluate the name substitution technique, we follow the experimental setups in Section 4.2, and compare reconstruction rates of different baselines before and after name substitution. The results in Table 9 reveal the effectiveness of our approach. Before name substitution, there is no distinguished gap between the different models. After name substitution, as expected, the pre-trained Llama2 exhibits no memorization, while the Non-private approach shows high memorization because of fine-tuning over private data. And the memorization issue is addressed through synthetic text generation.

Reconstruction	Llama2-7B	None-Private	Synthetic
Before	40.23	43.73	42.57
After	1.89	96.23	3.77

Table 9: Reconstruction rate comparison *Before* and *After* name substitution using Flair as the NER extraction tool. The expansion of the gap between Non-Private and Synthetic methods validates our name substitution approach.

Evaluation	PubMedQA	MedQA	MedMCQA	Avg
Non-Private	41	27.57	25.79	31.45
ICL	40.9	28.75	15.31	28.32
Self-Instruct	44.4	24.27	19.85	29.51
Self-Instruct-ICL	48.1	28.91	25.51	34.17
DP-Gene	43.2	26.08	22.53	30.60
DP-Instruct	36.8	26.24	26.46	29.83
DP-Instruct-ICL	54.5	23.88	**27.37**	35.25
KnowledgeSG	**58.3**	**30.24**	26.8	**38.45**

Table 10: Performance results on medical domain. Comparative analysis of free-form instruction evaluation.

C Additional Experiments

C.1 Medical Benchmarks

Setups. We evaluate the same models as Section 4.4 on 3 medical question answering benchmarks including MedQA (Jin et al., 2021), PubMedQA (Jin et al., 2019), and MedMCQA (Pal et al., 2022). We follow the code base of LMflow[9] (Diao et al., 2023) and use the prompt shown in Figure 6 to inference answers.

Results. From Table 10, we can conclude that: (1) Compared to free-form evaluation in Section 4, the results on medical benchmarks are more random. Along with the limit of performance ceiling, the gap between different methods are narrowed especially on MedQA and MedMCQA. (2) Our method still performs the best on average.

Distinctions in Medical Evaluations. Compared to the benchmark results in Table 10, the gap between different baselines is much more pronounced and noticeable in the free-form evaluation in Table 4, aligning more closely with expectations. We attribute the reasons as: (1) For MedQA and MedMCQA, the dataset we use is HealthCareMagic, whose purpose is to provide patients with consultant. This may not correspond with the nature of benchmarks to choose the right answer to a medicine-related question. (2) Benchmark results

[9] https://github.com/OptimalScale/LMFlow

Evaluation	Avg:3		Avg:4	
	Acc	F1	Acc	F1
Non-Private	0.699	0.719	0.689	0.703
DP-SGD	0.419	0.343	0.428	0.350
KnowledgeSG	0.784	0.775	0.752	0.745

Table 11: Comparison of Non-Private approach with DP-SGD. The drop in performance validates the limitations of relying on DP-SGD only.

involve more randomness, thus improving the performance of inferior competitors to some extent.

C.2 DP-SGD Performance Evaluation

We follow the details for DP-finetuning in Appendix F.1 and evaluate its performance on the financial domain, same as Section 4.3.

From the results in Table 11, we can conclude that relying on DP-SGD only results in a considerable decline of performance, necessitating our approach of synthetic data generation with knowledge distillation from server.

C.3 Generalizability in Other Domains

Setups. To evaluate the generalizability of *KnowledgeSG*, we conduct additional experiments in the mathematical and code domains.

For the experimental setup of mathematical domain, we utilize 500 samples from the lighteval/MATH dataset[10], employing MAmmoTH-7B (Yue et al., 2024) as the professional model and Llama2-7B as the base model. Following Yue et al. (2024), we evaluate models on the GSM8K dataset (Cobbe et al., 2021) using the public benchmark MAmmoTH[11]. For the code domain, we utilize the PythonCodeInstructions-18k dataset[12], employing Llama3-8B-Instruct[13] as the professional model. We evaluate models on HumanEval dataset (Chen et al., 2021) using the bigcode-evaluation-harness benchmark[14] (Ben Allal et al., 2022).

We compare three representative methods: Non-Private fine-tuning, In-Context Learning (ICL), and a simplified version of *KnowledgeSG* that replaces the synthetic responses in ICL with those generated by the professional model $\mathbb{W}_{Pro}$.

Evaluation Metric	GSM8K Accuracy	HumanEval Pass@10
Llama2-7B	12.96	17.68
Non-Private	21.30	18.90
ICL	14.27	18.29
KnowledgeSG*	**33.83**	**20.73**

Table 12: Performance results on mathematical and code domains. The relative improvement of KnowledgeSG over Non-Private and ICL demonstrates the generalizability of *KnowledgeSG*. We show accuracy and Pass@10 for GSM8K and HumanEval respectively. *: Given that privacy concerns are not the primary issue in the generation of synthetic data for mathematical and code domains, we adopt a simplified version which focuses on knowledge distillation for convenience. This approach excludes differential privacy fine-tuning, instruction filtration, and the transmitting unit.

Results. As shown in Table 12, *KnowledgeSG* outperforms ICL and Non-Private methods. The results confirm the effectiveness of *KnowledgeSG* in the math and code domain, further proving its generalizability. However, in the code domain, the performance gap between different methods is less pronounced compared to other domains. We attribute this to the suboptimal coding performance of pretrained Llama2-7B, which may lack the capacity to generalize effectively on coding tasks. This finding aligns with related studies, where experiments on HumanEval are primarily conducted using the Llama2-13B model or larger variants (Luo et al., 2023; Xu et al., 2023). The reason we prefer financial and medical domain than code and math is that math solving and code writing tasks are not directly related to privacy because there usually is no PIIs in these datasets.

Our preference for the financial and medical domains over the code and math domains in our experiments stems from the fact that datasets involving math solving and code writing are not directly related to privacy concerns, as they typically do not contain personally identifiable information (PII).

D Definition of PII

There are various definitions of **Privacy** catering to different privacy concerns in different scenarios. A LLM can know your preference by digging into your search histories. It can also infer that you have a girlfriend from your recent query of buying flowers on Valentine's day. In this work, we mainly research on one of the definitions of privacy, i.e. PII which is well-studied by the community.

PII is short for Personal Identifiable Information,

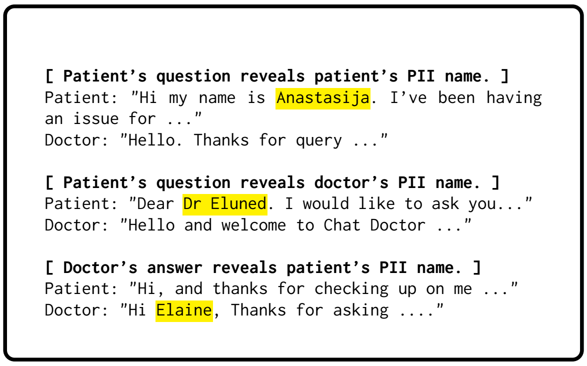
[10] https://huggingface.co/datasets/lighteval/MATH

[11] https://github.com/TIGER-AI-Lab/MAmmoTH

[12] https://huggingface.co/datasets/iamtarun/python_code_instructions_18k_alpaca

[13] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

[14] https://github.com/bigcode-project/bigcode-evaluation-harness



[Patient’s question reveals patient’s PII name.]
Patient: "Hi my name is Anastasija. I’ve been having an issue for ..."
Doctor: "Hello. Thanks for query ..."

[Patient’s question reveals doctor’s PII name.]
Patient: "Dear Dr Eluned. I would like to ask you..."
Doctor: "Hello and welcome to Chat Doctor ..."

[Doctor’s answer reveals patient’s PII name.]
Patient: "Hi, and thanks for checking up on me ..."
Doctor: "Hi Elaine, Thanks for asking ...."

Figure 4: Examples of individual names contained in the ICliniq dataset (Li et al., 2023c). Individual names as one form of PII, can be used to identify corresponding individuals. For anonymity, we substitute the original names with synthetic ones as mentioned in Appendix B.2.

representing data that can identify an individual. As detailed elaborated in Lukas et al. (2023), PII can be a direct identifier when leakage of that data alone is sufficient to re-identify an individual, or quasi-identifier when only an aggregation of many quasi-identifiers can reliably re-identify an individual. Apart from names and addresses, PII could also be ticker symbol, transaction figures and credit securities accounts in financial domain, and health insurance card numbers in medical domain.

We show examples of PII from HealthCareMagic dataset in Fig 4. Since our current focus is not on any specific category of leaked PII, we only evaluate Individual Name in Section 4 for convenience.

E Differences of Domain-Specific Data from General Data

E.1 Illustration

We give additional illustration in this section to explain the performance discrepancies of domain-specific data and general data after synthetic data generation.

Deploying an LLM to generate new synthetic data from the original private data is just like asking a student to read an examination question and try to create a new copy of it. Naturally, the quality of the rewritten question is highly dependent on how the student understands the original question, and how he may generalize. As illustrated in Fig 5, a Ph.D. student will behave well on general questions, e.g. Alpaca[16] (Taori et al., 2023). But if you ask a kindergarten student to create a new calculus test based on several examples, e.g. Math[17] (Hendrycks et al., 2021b), it is highly unlikely that he can fulfil this task.

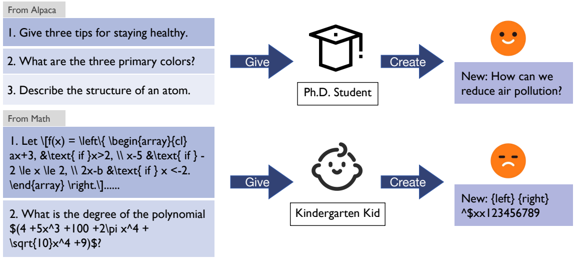

From Alpaca
1. Give three tips for staying healthy.
2. What are the three primary colors?
3. Describe the structure of an atom.
Give
Ph.D. Student
Create
New: How can we reduce air pollution?
From Math
1. Let \[f(x) = \left\{ \begin{array}{cl} ax+3, &\text{ if }x>2, \\ x-5 &\text{ if } -2 \le x \le 2, \\ 2x-b &\text{ if } x <-2. \end{array} \right.\]......
2. What is the degree of the polynomial $(4 +5x^3 +100 +2\pi x^4 + \sqrt{10}x^4 +9)$?
Give
Kindergarten Kid
Create
New: {left} {right} ^$xx123456789

Figure 5: Illustration of our identified gap between model comprehension and data complexity. We make an analogy by describing a situation where a student is asked to create a new question based on given examples.

In practical applications, it is the same nature for LLM-based synthetic data generation where domain-specific data, i.e. the calculus test is more difficult for general foundation models to comprehend. In real-world scenarios when a financial or medical facility tries to train a domain-specific LLM without memorizing its high-value private data (Nakamura et al., 2020; Brown et al., 2022), he is inclined to deploy the synthetic text generation approach. With consideration of resources, he has no choice but to fine-tune a limited-size LLM. However, due to the speciality of original data, small models pre-trained on general data (e.g. Llama2-7B (Touvron et al., 2023a,b) and ChatGlm 6B (Du et al., 2022)) are unable to fully understand the domain knowledge and consequently fail to maintain high utility of original data after synthetic generation.

E.2 Gap Ratio

For the purpose of quantifying the gap between domain-specific data and general data and providing better understanding of the proposed problem, we heuristically define a ratio called *Gap Ratio*.

We choose GPT-4 (OpenAI, 2023) to be the datum model as we assume it is an all-around player that behaves well both on general tasks and domain-specific tasks. And the *Gap Ratio* is calculated by the ratio of target model results and GPT-4 results on the same evaluation benchmark. For example, from Table 13, Llama2-7B’s *Gap Ratio* is 0.8722 on Chatbot Arena and 0.7007 on general benchmarks on average.

No matter what the absolute value is in different measurements of model performance, we can ap-

[16] https://huggingface.co/datasets/tatsu-lab/alpaca

[17] https://huggingface.co/datasets/lighteval/MATH

	Chatbot Arena[15]	MT-Bench	MMLU	Datum	FPB	PubMedQA
GPT-4	1189	8.96	86.4	-	0.833	-
ChatGPT	-	-	70.0	-	0.781	63.9*
Llama2-7B-Chat	1037	6.27	45.8	-	-	-
Llama2-7B	-	-	-	-	0.39	7.2
Llama-7B	-	-	35.2	-	-	5.2*
Gap Ratio	0.8722$^{\uparrow}$	0.6998$^{\uparrow}$	0.5301$^{\uparrow}$	0.5^{-}	0.4682$^{\downarrow}$	0.1127$^{\downarrow}$

Table 13: Comparison between {Llama2-7B, Llam2-7B-Chat} and {GPT-4, ChatGPT } on general benchmarks including Chatbot Arena Leaderboard, MT-Bench, MMLU (Chiang et al., 2024; Hendrycks et al., 2021a; Zheng et al., 2023) and domain-specific benchmarks including FPB, PubMedQA(Malo et al., 2014; Jin et al., 2019). Results with tagger* is collected from Zhang et al. (2023). Results with $^{\uparrow}$ and $^{\downarrow}$ indicate whether the *Gap Ratio* exceeds the datum line of 0.5 or not.

parently see that the gap between Llama2 and GPT will be greatly widened if changed from general to a specific domain. As in Table 13, we draw a datum line of 0.5, smaller than which indicates a tendency of worse synthetic generation.

F Implementation Details

F.1 Training Details

For normal fine-tuning (not DP), we follow the codebase of (Ye et al., 2024b)[18] and use the local training algorithm to train the model for 100 rounds in total. For each round, we train for 10 steps with batch-size set to 5 using AdamW (Loshchilov and Hutter, 2018) optimizer. This means each sample in the training dataset is iterated for 10 times on average, equal to training the model for 10 epochs without setting max-steps. We apply a cosine learning rate schedule according to the round index. The initial learning rate in the first round is $5e-5$, and the final learning rate in the last round is $1e-6$.

For DP fine-tuning, we follow the codebase of dp-transformers library (Wutschitz et al., 2022)[19], which is a wrapper around Opacus (Yousefpour et al., 2021)[20]. We train the model for 4 epochs for the first stage of generation, and 10 epochs for fair comparison between training on private data with DP and training on synthetic data. The target epsilon is set to 8 and maximum per-sample gradient norm is set to 1.0 for differentially private training. The privacy budget we use is $(\epsilon, \delta) = (8, \frac{1}{N})$. According to (Lukas et al., 2023), these values are close to established DP deployments such as Apple's QuickType and Google's models.

The max sequence length is set to 512 for training in both normal and DP fine-tuning. All the training experiments are conducted on one NVIDIA GeForce RTX 3090.

The rank of LoRA (Hu et al., 2021) is 32 with a scalar $\alpha = 64$. We use the Alpaca (Taori et al., 2023) template to format the instruction.

F.2 Inferencing Details

We use VLLM (Kwon et al., 2023) for faster inferencing and set the max-model-len to as long as 2048 to obtain more information. The inferencing experiments are mostly conducted on A100 40G. We set temperature to 0.7 to encourage diversity. We follow in-context learning (Dong et al., 2022) and self-instruct (Wang et al., 2022) to formulate our prompts. The prompt templates we employ are shown in Figure 7 and 8. To make sure we have enough instructions for subsequent filtering, the generation times are set two times of the original dataset size. To ensure sufficient instructions for subsequent filtering, the generation count is set to twice the size of the original dataset. For instruction extraction and pre-processing, we extract the first instruction the model generates and filter those shorter than 2 tokens.

F.3 Baselines

To give a detailed comparison between different baselines in our experiments, we elaborate on three aspects in Table 14, ranging from the model used for generating instructions, whether the baseline first generates instructions then responses and whether the baseline requires few-shot examples to generate response if it is twp-step. DP-Instruct-ICL and Self-Instruct-ICL are different from DP-Instruct and Self-Instruct in that they require few-shot examples from original dataset to produce better responses during the second stage of generation while the others do not. Theoretically, DP-Instruct performs better than Self-Instruct and DP-Gene

[18] https://github.com/rui-ye/OpenFedLLM

[19] https://github.com/microsoft/dp-transformers

[20] https://github.com/pytorch/opacus

Baselines	Model	Two-Step	ICL
ICL	Pre-trained	✗	-
Self-Instruct	Pre-trained	✓	✗
Self-Instruct-ICL	Pre-trained	✓	✓
DP-Gene	DP-finetuned	✗	-
DP-Instruct	DP-finetuned	✓	✗
DP-Instruct-ICL	DP-finetuned	✓	✓
KnowledgeSG	DP-finetuned	✓	✗

Table 14: Elaboration of baselines. *Model* means the generative model used for generating synthetic instructions. *Twp-Step* means whether the baseline first generates instructions then responses or generates both instructions and responses meanwhile. *ICL* means whether the baseline requires few-shot examples from original dataset to generate response at the second stage.

performs better than ICL because of additional DP-finetuning of base model.

G Deployment Guidance

To facilitate real-world applications and future work, we provide a detailed guidance on the deployment of *KnowledgeSG*. The framework involves three main stages.

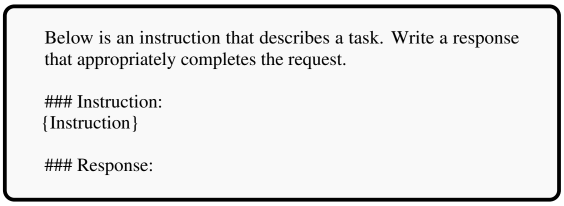
Preparations and Transmitting Unit. (1) Prepare the base model, e.g. Llama2-7B and establish a code base that can do normal-finetuning of LLMs, e.g. LlamaFactory. (2) Establish a communication channel and sample a small amount of data to construct the seed dataset sharing between the client and server. (3) Fine-tune the base model on this seed dataset to obtain a modified base model on both client side and server side.

Client Side. (1) Prepare the private dataset intended for use. (2) Establish a code base that can achieve DP-finetuning of LLMs.

Server Side. (1) Prepare the professional model. Most of open-sourced large language models can be easily downloaded from the HuggingFace website. (2) Write a code that can inference LLMs and design the prompts which are related to the professional model we choose.

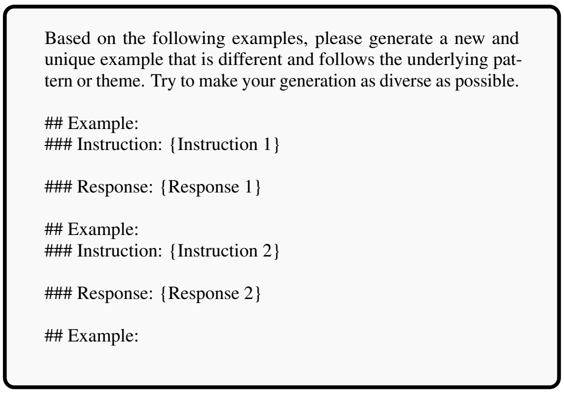
After this deployment, we can apply *KnowledgeSG* in a client-server framework and obtain the desired model.

H Templates



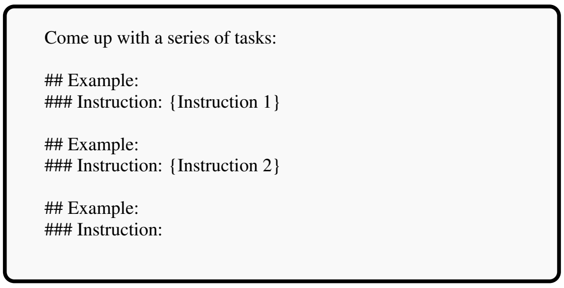
```
Below is an instruction that describes a task. Write a response
that appropriately completes the request.

### Instruction:
{Instruction}

### Response:
```

Figure 6: Templates-1



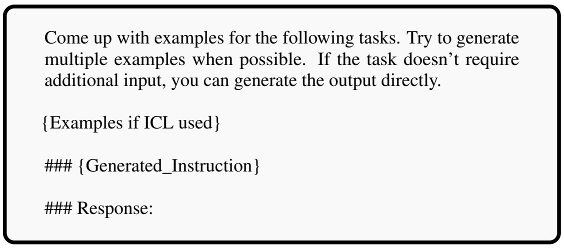
```
Based on the following examples, please generate a new and
unique example that is different and follows the underlying pattern or theme. Try to make your generation as diverse as possible.

## Example:
### Instruction: {Instruction 1}

### Response: {Response 1}

## Example:
### Instruction: {Instruction 2}

### Response: {Response 2}

## Example:
```

Figure 7: Templates-2



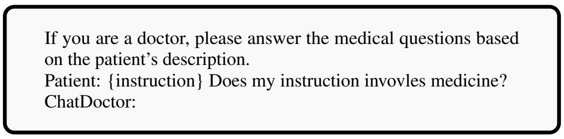
```
Come up with a series of tasks:

## Example:
### Instruction: {Instruction 1}

## Example:
### Instruction: {Instruction 2}

## Example:
### Instruction:
```

Figure 8: Templates-3



```
Come up with examples for the following tasks. Try to generate
multiple examples when possible. If the task doesn't require
additional input, you can generate the output directly.

{Examples if ICL used}

### {Generated_Instruction}

### Response:
```

Figure 9: Templates-4



```
If you are a doctor, please answer the medical questions based
on the patient's description.
Patient: {instruction} Does my instruction invovles medicine?
ChatDoctor:
```

Figure 10: Templates-5