AUG-FEDPROMPT: PRACTICAL FEW-SHOT FEDERATED NLP WITH DATA-AUGMENTED PROMPTS

Dongqi Cai[1] *Yaozong Wu*[1] *Haitao Yuan*[1] *Shangguang Wang*[1] *Felix Xiaozhu Lin*[2] *Mengwei Xu*[1*]

[1] Beiyou Shenzhen Institute, China [2] University of Virginia, USA
{cdq, yzwu, sgwang, mwx}@bupt.edu.cn yht19@tsinghua.org.cn felixlin@virginia.edu

ABSTRACT

Transformer-based pre-trained models have become the defacto solution for NLP tasks. Fine-tuning such pre-trained models for downstream tasks often requires tremendous amount of data that is both private and labeled. However, in reality: 1) such private data cannot be collected and is distributed across mobile devices, and 2) well-curated labeled data is scarce. To tackle those issues, we first define a data generator for federated few-shot learning tasks, which encompasses the quantity and distribution of scarce labeled data in a realistic setting. Then we propose AUG-FedPrompt, a **prompt**-based **fed**erated learning algorithm that carefully annotates abundant unlabeled data for data **aug**mentation. AUG-FedPrompt can perform on par with full-set fine-tuning with very few initial labeled data.

1. INTRODUCTION

Federated NLP The development of pre-trained models is overwhelming with the rise of BERT [1]. Their deployment [2, 3, 4, 5] is commonly composed of two-step training: pre-training and fine-tuning. Unlike self-supervised pre-training, fine-tuning is supervised, requiring task-specific tremendous labeled data. However, the exploitation of private user data is restricted and even prohibited in some cases by several data protection regulations such as GDPR [6] and CCPA [7]. Recently, federate learning (FL) becomes the de-facto approach to train a model with privacy preserved. As such, federated NLP (FedNLP) [8, 9] is now an important topic towards practical NLP applications.

Problem and challenge A key obstacle to practical FedNLP is data labeling. It's much more difficult to label data on client devices than on centrally collected data [10]. Lack of sufficient labeled data severely limits the practicality and scalability of FedNLP in real-world NLP applications. Therefore, it is important to address the issue of few-shot or even zero-shot FedNLP tasks. There are very few efforts on this topic [11, 12], which still assume a fairly large number (typically >1000 in total) of labels that are uniformly distributed across clients. However, in practice, the labeled data distribution could be skewed across clients, and such skewness would result in a significant drop in the accuracy according to our experiments in § 2.

Our solution and contribution
(1) To address the lack of labeled data, we first design a comprehensive data generator to simulate the labeled data distribution for few-shot FedNLP tasks. In particular, the generator has two meta-parameters: data quantity and sparsity, encompassing all possibilities of scarce labeled data in real-world few-shot learning.
(2) To boost its accuracy based on the generated data, we design a data-augmented prompt method, namely AUG-FedPrompt, for practical few-shot FedNLP. AUG-FedPrompt orchestrates prompt learning and soft label-based data augmentation. Prompt learning [13] introduces a task description in natural language. It helps task-specific fine-tuning achieve high accuracy with very few labeled data samples in FedNLP. Furthermore, to tackle performance degradation caused by skewed distribution, AUG-FedPrompt leverages enormous and easily accessible unlabeled data. AUG-FedPrompt annotates those unlabeled data and identifies those high-quality samples with the help of model capacity and annotation confidence. Those samples are then used for FL training in the next rounds.
(3) Our extensive experiments on four English datasets demonstrate that AUG-FedPrompt can achieve a substantial performance gain (25%–55% higher accuracy) over the state-of-the-art FedNLP approaches under various few-shot settings. Augmentation with unlabeled data enhances AUG-FedPrompt to perform well with highly skewed labeled distribution across clients. Overall, AUG-FedPrompt can achieve a comparable performance with the state-of-the-art FedNLP approaches with less than 0.1% labeled data.

2. PRELIMINARYIES AND PROBLEM SETUP

Federated NLP Training Procedure The two NLP training phases, i.e., pre-training and fine-tuning, require data of disparate natures. While pre-training is typically done on public text corpora such as Wikipedia articles, fine-tuning requires domain-specific samples, such as user reviews, messages, or emails. Especially for mobile computing, the domain-specific

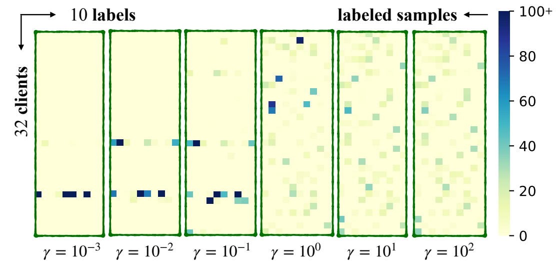


Fig. 1: Visualizing the labeled data sparsity on YAHOO [14] with n=1024, ξ=32, γ being 10^n, n=-3,-2,..,2. Each subfigure is a 32×10 matrix, where 32 is the number of clients and 10 is the number of labels. The intensity of each cell represents the number of labeled samples for a specific label in the client-side local data.

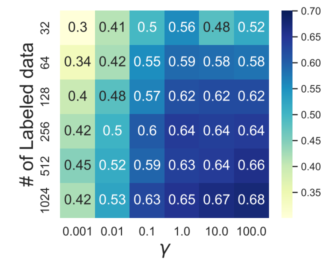


Fig. 2: Average accuracy of federated few-shot learning under different data quantity and sparsity. When sparsity γ grows larger, labeled data will be more uniformly distributed, and vice versa. Dataset: YAHOO [14].

samples are collected from end users, and distributed over mobile devices. In addition, protecting privacy also should be considered. To fine-tune models on such private, distributed data, federated learning is the de-facto approach [8, 9]. Ahead of time, a cloud service distributes a pre-trained model to all client devices. In a training session targeting a specific NLP task and domain, a cloud service selects multiple mobile devices to participate in training. A device trains a local copy of the model with its private data and sends the model updates to the cloud. Having aggregated model updates from multiple devices, the cloud sends an updated model to the devices. The training procedure repeats many rounds until the model converges.

Federated few-shot data generator Apart from data privacy, lack of sufficient labeled data is also a crucial question and inherent feature for mobile scenarios, i.e., few-shot learning. Alike full-set dataset could be non-independent and identically distributed (non-iid), the scarce labeled data is not always uniformly distributed in the real world. Based on the definition of full-set non-iid partition strategies [8], we further define the sparsity of labeled data under federated few shot learning scenario.

We define a new triple tuple (n, γ) to represent the practical few-shot learning training data distribution, where n represents the total numbers of labeled data, γ represents the sparsity of labeled data.

The quantity of labeled data in selected clients follows a Dirichlet allocation $z \sim \text{Dir}_{\xi}\ (\gamma)$ where ξ is the number of clients owing labeled data[1]. We can then assign labeled data from the global labeled dataset to selected clients based on the distribution z, i.e., client$_i$ owns labeled dataset $|\mathcal{T}_i| = z_i n$. For example, as shown in Fig. 1, we visualize the labeled data sparsity on Yahoo [14] with n=1024, ξ=32, γ being 10^n, n=-3,-2,..,2. Each sub-figure is a 32×10 matrix, the intensity of which represents the labeled samples of a particular label. When γ is small (10^{-3}, 10^{-2}, 10^{-1}), the labeled data will be skewed distributed, i.e., only few clients own labeled data; when γ=10^2, labeled data is nearly uniformly distributed on all clients.

Performance degradation under skewed distribution We show the impact of data sparsity on federated few-shot learning in Fig. 2. The convergence performance degrades when sparsity γ is turning smaller, i.e., labeled data is more and more skewed distributed. For example, when labeled data is 1024 in total, uniform distribution (γ=100) will be 26% better than thick distribution (γ=0.001). The rationale behind this is that under the common non-iid labeled data distribution, single client tends to own more specific kinds of data labels. Such a skewed distribution of labeled data will lead to unfairness, i.e., aggregated model prefers those specific labels. After all, skewed distribution of labeled data could result in significant accuracy drop. More detailed analysis will be presented in Section 4.3.

3. PROPOSED METHOD

To address the challenge of labeled data privacy concern and scarcity, this paper proposes AUG-FedPrompt that can leverage massive unlabeled data via federated prompt learning. The key rationale is fine-tuning the pre-trained model via prompt learning on client devices. After local training and federated aggregation, cross-device soft label inferring procedure helps carefully annotating clients' enormous unlabeled data.

We describe the training workflow of AUG-FedPrompt in Fig. 3. A public pre-trained transformer-based language model M is transferred to chosen clients. We assume that each client has access to a tiny training set $\mathcal{T}$ and a much larger set of unlabeled examples $\mathcal{D}$.

For local prompt training, we annotate T as the vocabulary of model M, $\_\_ \in T$ as the mask token and T^* as the set of all token sequences. The sequence of input phrases is $\mathbf{x} = (s_1, \ldots, s_k)$ where $s_i \in T^*$. The pattern-verbalizer pair $\mathbf{p}$ includes: 1) a *pattern* $P : X \rightarrow T^*$ maps inputs x to a cloze question containing a single mask; 2) a *verbalizer* $v : Y \rightarrow T$ maps each output y to a single token representing

[1] ξ could be an optional hyper-parameter to strict the maximum of clients owing labeled data. In this manuscript, we fix ξ as 32 for simplicity.

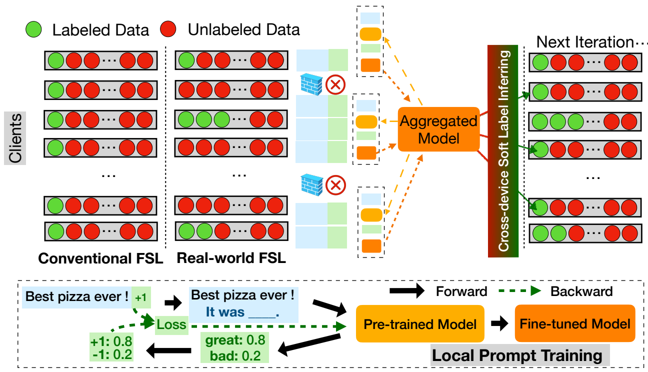


Fig. 3: Workflow of AUG-FedPrompt .

its task-specific meaning in the pattern.

The purpose of local prompt training is to derive the probability that y is the correct output of x from the probability that v(y) is the most likely token at the masked position in $P(x)$. Based on this rationale, we define the conditional probability distribution $s_{\mathbf{p}}$ of y given x as:

$$s_{\mathbf{p}}(y \mid x) = \frac{\exp q_{\mathbf{p}}(y \mid x)}{\sum_{y' \in Y} \exp q_{\mathbf{p}}(y' \mid x)} \tag{1}$$

where $q_{\mathbf{p}}(y \mid x) = M(v(y) \mid P(x))$ is the probability that M assigns to $v(y)$ in sequence $P(x)$.

For client-side fine-tuning, M is fine-tuned on local labeled data (x, y) by minimizing the cross-entropy between $s_{\mathbf{p}}(y \mid x)$ and y. For server-side aggregation, in iteration i, client k sends the updated model M_k^i to the cloud for aggregation:

$$M^i = \frac{\sum_{k=0}^{\xi-1} |\mathcal{T}| M_k^i}{\sum_{k=0}^{\xi-1} |\mathcal{T}|} \tag{2}$$

where M^i is the aggregated model and $|\mathcal{T}|$ is the size of training set.

For data augmentation, M^i is transferred to those clients with enormous unlabeled data for soft label inferring. Each unlabeled example $\hat{x} \in D$ is labeled with soft label $\hat{y}$ based on $s_{\mathbf{p}}(\hat{y} \mid \hat{x})$. The soft-labeled dataset is used to fine-tune the client-side model in the next iteration.

The resulting soft-labeled dataset could consist of enormous samples with wrong labels. Directly involving them in the next training iteration will poison the foundation model, which makes it could be even worse than purely using the limited labeled data. To address this issue, we propose two techniques to filter out those wrong samples and remain the purity of augment dataset: 1) **Filtering by model capacity.** We filter out those models with low model capacity, i.e., the performance on validation datasets is bad. 2) **Filtering by confidence.** We filter out those samples with low confidence, i.e., the probability of the most likely label is less than a threshold. Both capacity and confidence are hyper-parameters. They can be flexibly adjusted according to different scenarios.

Dataset	Prompt	Train	Test
AGNEWS [14]	a (____) b	120,000	7.600
MNLI [15]	"a" ? ‖ ____, "b"	392,702	9,815
YAHOO [14]	[Category:] a ____ b	1,400,000	60,000
YELP [14]	It was ____. a	650,000	50,000

Table 1: Evaluation datasets. Each dataset is distributed to 1000 clients. Label quantity of each class follows the non-iid label distribution in [8] where $\alpha = 1$.

4. EVALUATION

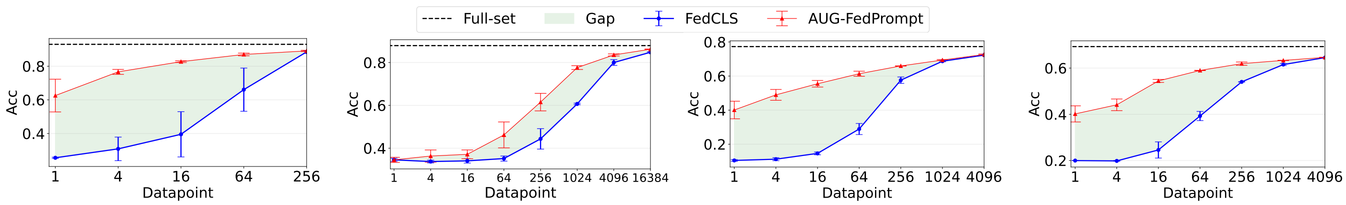
In this section, we evaluate the performance of AUG-FedPrompt across data scales. AUG-FedPrompt significantly outperforms naive federated fine-tuning. It could perform on par with full-set training while saving up to 99.9% labeled data. Apart from data efficiency, AUG-FedPrompt shows great robustness under various practical few-shot scenario regardless of skewed or uniform data distribution.

4.1. Experiment Setup

Dataset and models We perform our evaluation on four English datasets and manually designed prompts[2], detailed information is shown in Table 1. (1) AGNEWS [14] is a news classification dataset. Given headline a and text body b, news needs to be classified as one of the four categories World, Sports, Business or Science/Tech. (2) MNLI [15] is a sentence understanding dataset. Given text pairs x = (a, b), the task is to find out whether a implies b, a and b contradict each other or neither. (3) YELP [14] is a restaurant rating dataset. Given a customer's review, text should be estimated on a 1-5 star scale. (4) YAHOO [14] is a text classification dataset. Given a question a and an answer b, one of ten possible categories needs to be assigned. We run all experiments with the same pre-trained model, roberta-large (355M parameters) from RoBERTa [16], which we load from the transformers [17] library.

Hyper-parameters In line with previous observations [18], few-shot fine-tuning performance varies across chosen labeled data considerably. We run every experiment 3 times in order to reduce variance. Unless otherwise stated, we use the recommended set of hyper-parameters from previous work [13]: mini-batch size as 4; local training iteration as 1; learning rate as 10^{-5}; max sequence length as 256. For careful annotation, we filter out those aggregated models performing worse than the zero-shot model and those soft-labeled data with confidence lower than 0.9. For the FL configurations at the server side, we follow the prior FedNLP literature [9, 8] to select 5 participants by default for each training round. The fine-tuned models will be collected in the central server and aggregated through *FedAvg* algorithm [19].

[2]We try 6, 2, 6, 4 different prompts for each datasets separately and report the chosen one that performs best. The verbalizers are the same as previous literature [13].



(a) AGNEWS (b) MNLI (c) YAHOO (d) YELP

Fig. 4: Average accuracy and standard deviation for AUG-FedPrompt across data scales. FedCLS stands for the vanilla federated fine-tuning. Full-set stands for fine-tuning on the full labeled data.

Dataset		AGNEWS	MNLI	YAHOO	YELP
Uniform	FedCLS	66.1±12.8	60.1±0.4	57.6±1.9	54.0±0.1
	FedPrompt	**87.0**±0.8	**77.6**±0.8	**66.0**±0.1	**61.9**±0.7
Skew	FedCLS	64.8±3.1	37.7±5.6	24.4±10.3	38.3±8.8
	FedPrompt	68.4±2.4	42.4±5.8	41.8±4.3	51.2±1.8
	w/ augment	**90.2**±0.5	**75.7**±1.2	**66.9**±1.1	**58.2**±2.4

Table 2: AUG-FedPrompt enhances performance under different few-shot learning settings. FedPrompt stands for AUG-FedPrompt without unlabeled data augmentation. Datapoint: 64 for agnews, 1024 for mnli, 256 for yahoo and yelp.

4.2. AUG-FedPrompt Performance across Data Scales

AUG-FedPrompt enjoys a substantial advantage on each task. As shown in Fig. 4, we compare our AUG-FedPrompt performance with FedCLS, i.e., the vanilla federated fine-tuning where a generic classifier layer inserted after pre-trained models is fine-tuned. Highlighted region shows the accuracy gap between AUG-FedPrompt and FedCLS. There are up to 50%, 25%, 55%, 38% accuracy improvement separately for 4 datasets. Both approaches improve with more labeled data, but AUG-FedPrompt remains better by a varying amount. AUG-FedPrompt reaches 99% relative performance of full-set with 90% less training data compared to full-set federated training. AUG-FedPrompt shows a strong zero-shot inferring capability, i.e., without task-specific fine-tuning, expected from MNLI dataset. MNLI dataset may need more labeled data to make the sufficient usage of the prompt to the pre-trained models. As for a usable accuracy, i.e., 90% relative performance of full-set training accuracy, AUG-FedPrompt only needs 64, 256, 256 in total for AGNEWS, YAHOO and YELP,saving up to 99.9% training data compared to full-set federated fine-tuning.

4.3. Impact of Data Augmentation

AUG-FedPrompt enhances FedPrompt performance when labeled data is skewed distributed. As shown in Table 2, FedPrompt without data augment shows competitive performance when labeled data is uniformly distributed on clients. While skew distribution of labeled data will hurt FedPrompt performance significantly. For example, FedPrompt performance degrades to 41.8% on YHAOO when 256 labeled data is skewed distributed on 32 clients. Considering that skewed distribution is common in real-world, we integrate AUG-FedPrompt with data augmentation to mitigate the performance degradation.

The intuition of prompts is that they introduce a task description in natural language, it helps task-specific fine-tuning perform well even with few training points. This rationale paves the way for the efficiency of our cross-device soft label inferring. AUG-FedPrompt helps to label more data correctly at the early stage of training. Together with our samples filter for annotated data, AUG-FedPrompt makes unlabeled data seldom hurt. For example, we annotate 100 unlabeled data on each client involved in per round for AGNEWS. The average annotating accuracy of unlabeled data in the first three rounds is 92.5%. The inference accuracy will further increase along with the FL training moves on, reaching 95.3% at the convergence round. Those 'nail' data, about 5 out of 100 in total, is hard to be correctly annotated and filtered out. Fortunately, we observe that they do not affect the model convergence as shown in Table 2. After cross-device soft label inferring, AUG-FedPrompt performs on par with full-set fine-tuning and greatly outperforms vanilla few-shot fine-tuning, reaching a usable accuracy with scarce labeled data on local.

5. CONCLUSIONS AND FUTURE WORK

In this manuscript, we comprehensively define a data generator for federated few-shot learning tasks, encompassing the quantity and sparsity of scarce labeled data. Then we propose AUG-FedPrompt, a federated learning algorithm that orchestrates prompt learning and soft label-based data augmentation. AUG-FedPrompt shows competitive performance under various federated few-shot learning settings, including uniform and skewed data distribution for scarce labeled data.

At last, we find that AUG-FedPrompt demonstrates its strong performance on top of large-scale pre-trained models. Fine-tuning those 'behemoths' is resource-costly. The communication of enormous updated model parameters also consumes huge bandwidth resources. In the future, we will develop a system solution to make our AUG-FedPrompt resource-efficient.

6. REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[2] Betty Van Aken, Benjamin Winter, Alexander Löser, and Felix A Gers, "How does bert answer questions? a layer-wise analysis of transformer representations," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1823–1832.

[3] Seohyun Kim, Jinman Zhao, Yuchi Tian, and Satish Chandra, "Code prediction by feeding trees to transformers," in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 2021, pp. 150–162.

[4] Alexey Svyatkovskiy, Shao Kun Deng, Shengyu Fu, and Neel Sundaresan, "Intellicode compose: Code generation using transformer," in *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2020, pp. 1433–1443.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al., "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[6] Paul Voigt and Axel Von dem Bussche, "The eu general data protection regulation (gdpr)," *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, vol. 10, no. 3152676, pp. 10–5555, 2017.

[7] Stuart L Pardau, "The california consumer privacy act: Towards a european-style privacy regime in the united states," *J. Tech. L. & Pol'y*, vol. 23, pp. 68, 2018.

[8] Bill Yuchen Lin, Chaoyang He, Zihang Ze, Hulin Wang, Yufen Hua, Christophe Dupuy, Rahul Gupta, Mahdi Soltanolkotabi, Xiang Ren, and Salman Avestimehr, "FedNLP: Benchmarking federated learning methods for natural language processing tasks," in *Findings of the Association for Computational Linguistics: NAACL 2022*, Seattle, United States, July 2022, pp. 157–175, Association for Computational Linguistics.

[9] Dongqi Cai, Yaozong Wu, Shangguang Wang, Felix Xiaozhu Lin, and Mengwei Xu, "Autofednlp: An efficient fednlp framework," *arXiv preprint arXiv:2205.10162*, 2022.

[10] Huatao Xu, Pengfei Zhou, Rui Tan, Mo Li, and Guobin Shen, "Limu-bert: Unleashing the potential of unlabeled data for imu sensing applications," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, 2021, pp. 220–233.

[11] Chenyou Fan and Jianwei Huang, "Federated few-shot learning with adversarial learning," in *2021 19th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*. IEEE, 2021, pp. 1–8.

[12] Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He, "Federated meta-learning with fast convergence and efficient communication," *arXiv preprint arXiv:1802.07876*, 2018.

[13] Timo Schick and Hinrich Schütze, "Exploiting cloze questions for few shot text classification and natural language inference," *arXiv preprint arXiv:2001.07676*, 2020.

[14] Xiang Zhang, Junbo Zhao, and Yann LeCun, "Character-level convolutional networks for text classification," *Advances in neural information processing systems*, vol. 28, 2015.

[15] Adina Williams, Nikita Nangia, and Samuel R Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," *arXiv preprint arXiv:1704.05426*, 2017.

[16] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[17] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al., "Huggingface's transformers: State-of-the-art natural language processing," *arXiv preprint arXiv:1910.03771*, 2019.

[18] Teven Le Scao and Alexander M Rush, "How many data points is a prompt worth?," *arXiv preprint arXiv:2103.08493*, 2021.

[19] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.