Towards Practical Few-shot Federated NLP

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

Beiyou Shenzhen Institude[1]
University of Virginia[2]

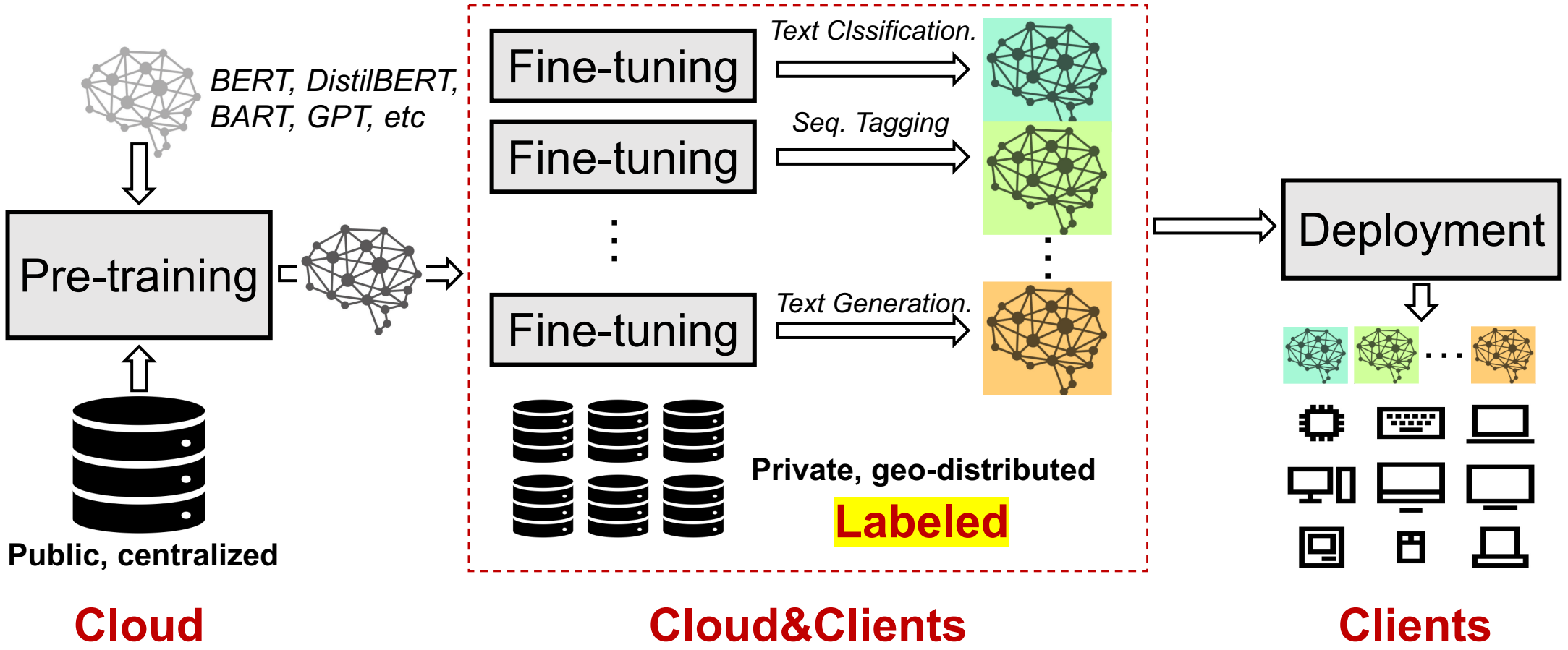
EuroMLSys

The 3rd Workshop on Machine Learning and Systems, co-located with EuroSys 2023.

Background

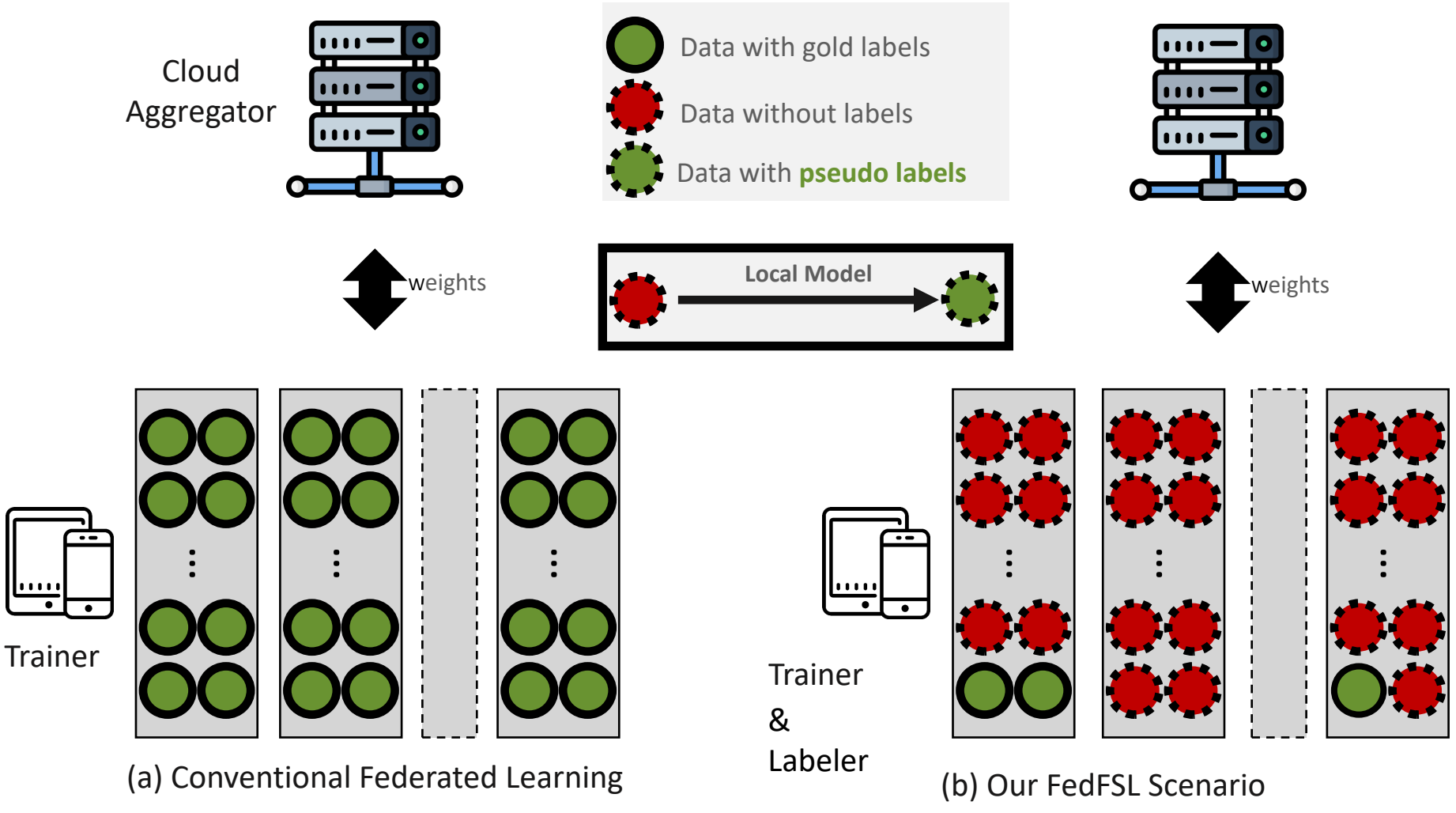
Few-shot Federated Natural Language Processing (NLP)

Federated NLP (FedNLP) [1]



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

Federated Few-shot Learning (FedFSL)



(a) Conventional Federated Learning

(b) Our FedFSL Scenario

Challenges:

1. Lack of labels
2. Error pseudo-label hurts

Solutions:

1. Pseudo labeling
2. Prompt learning

Our system:

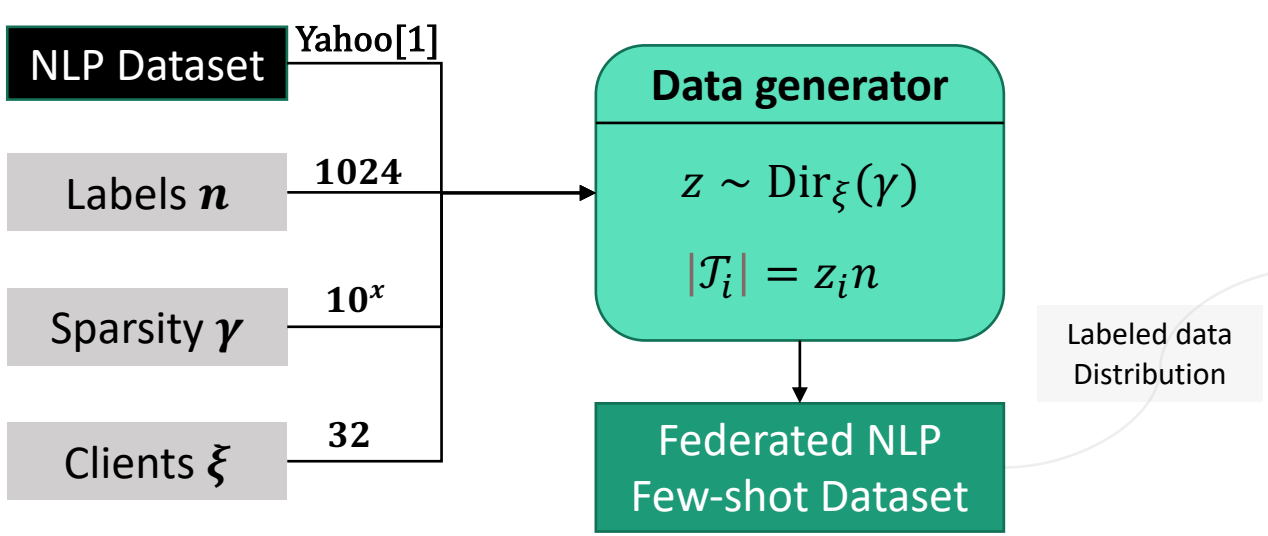
AUG-FedPrompt

Problem setup

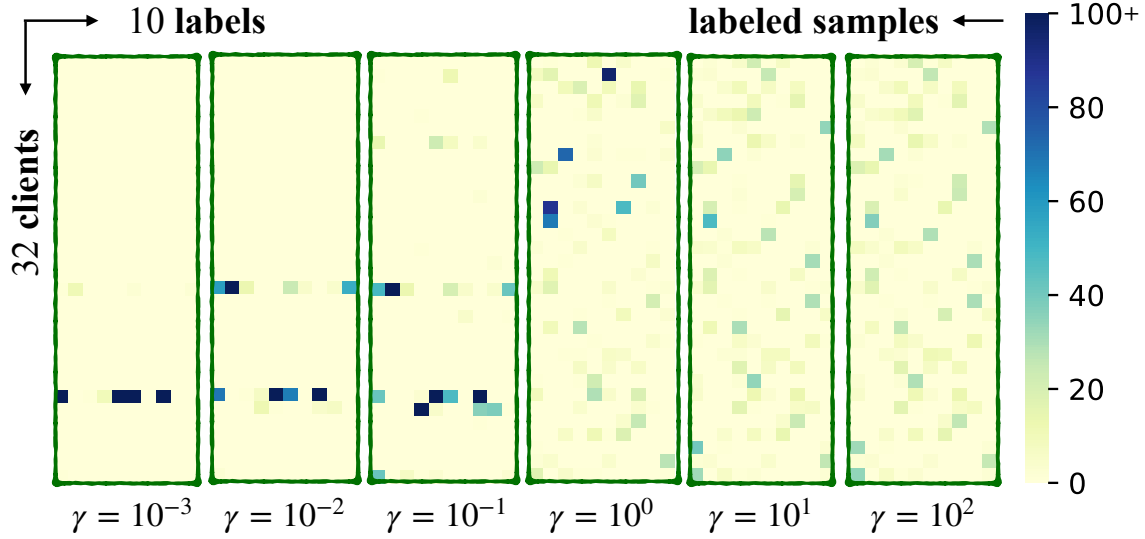
Data generator and preliminary experiments

Problem setup

We propose a data generator to simulate federated few-shot dataset.

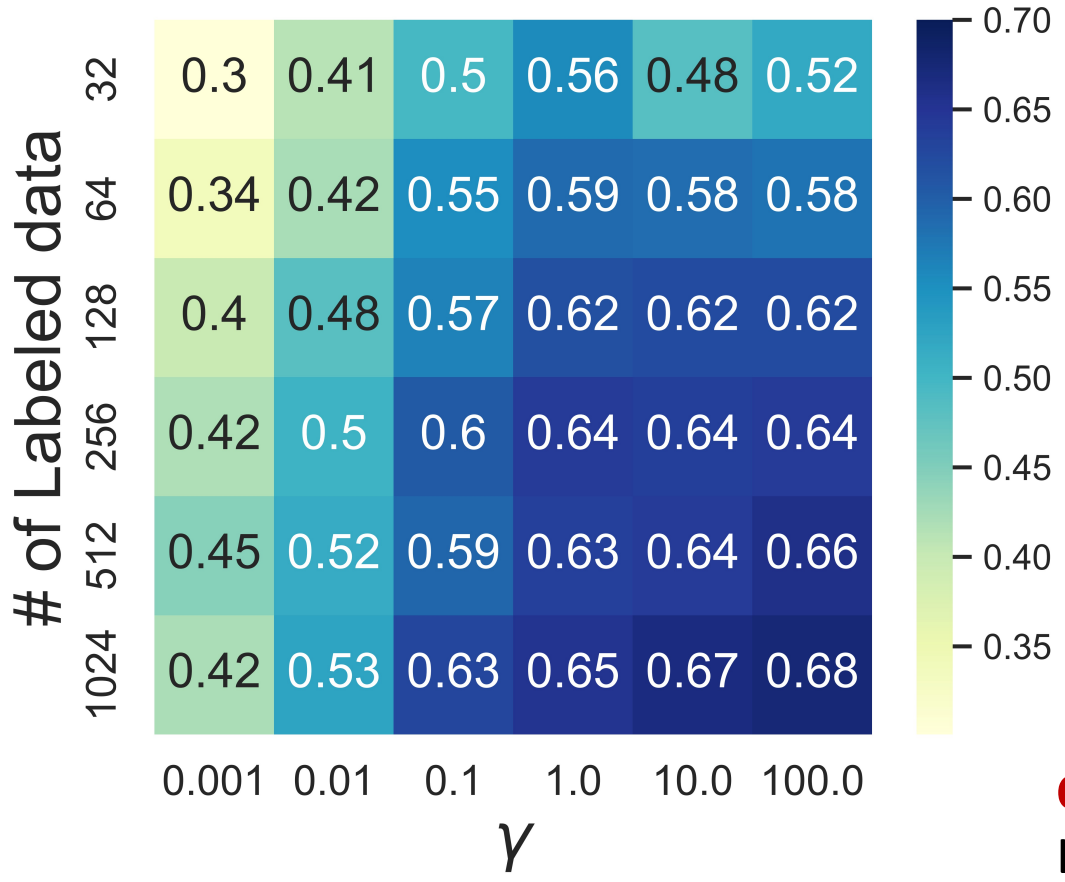


Labeled data Distribution



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

FedFSL Performance degradation



- More labeled data (n = 1024) will be 19% better than lack of labels (n = 32).
- Uniform distribution (γ = 100) will be 26% better than skewed distribution (γ = 0.001).

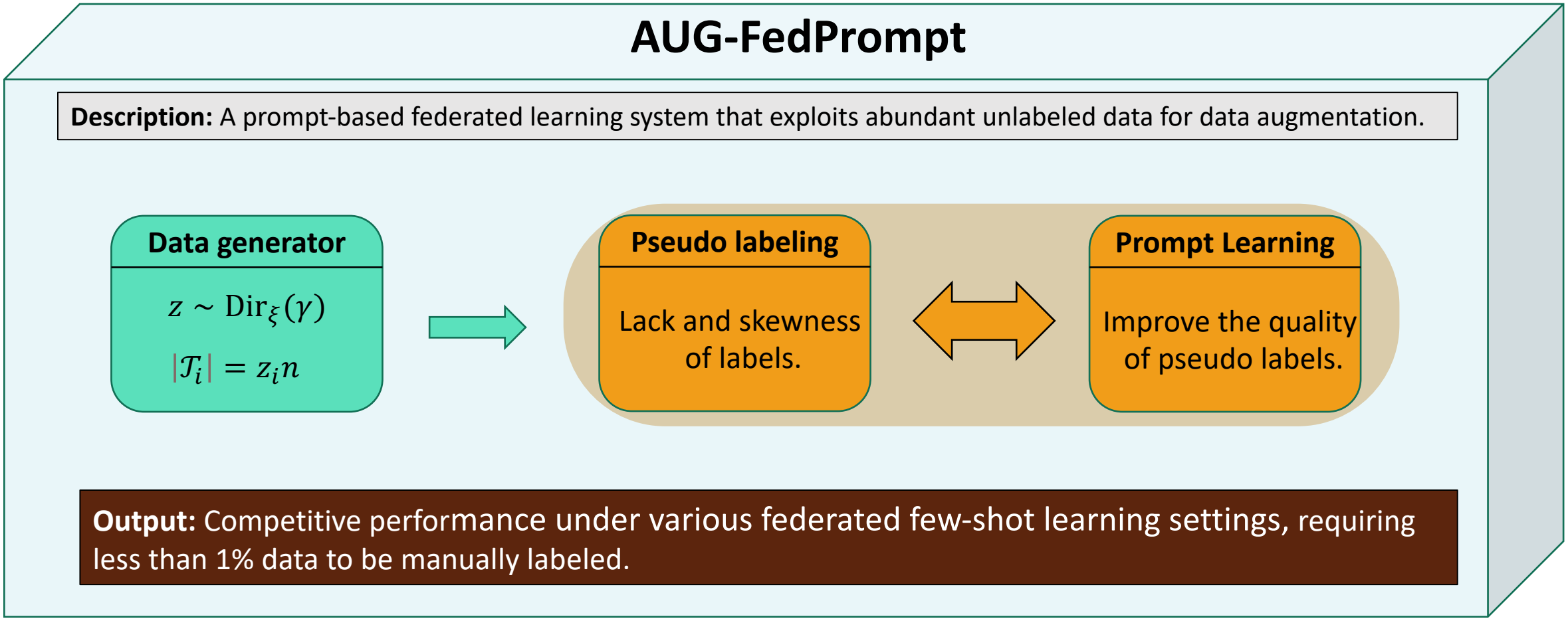
Conclusion:

Lack and skewness of labels will degrade performance.

Our System: AUG-FedPrompt

Two key building blocks: Pseudo labeling and Prompt Learning

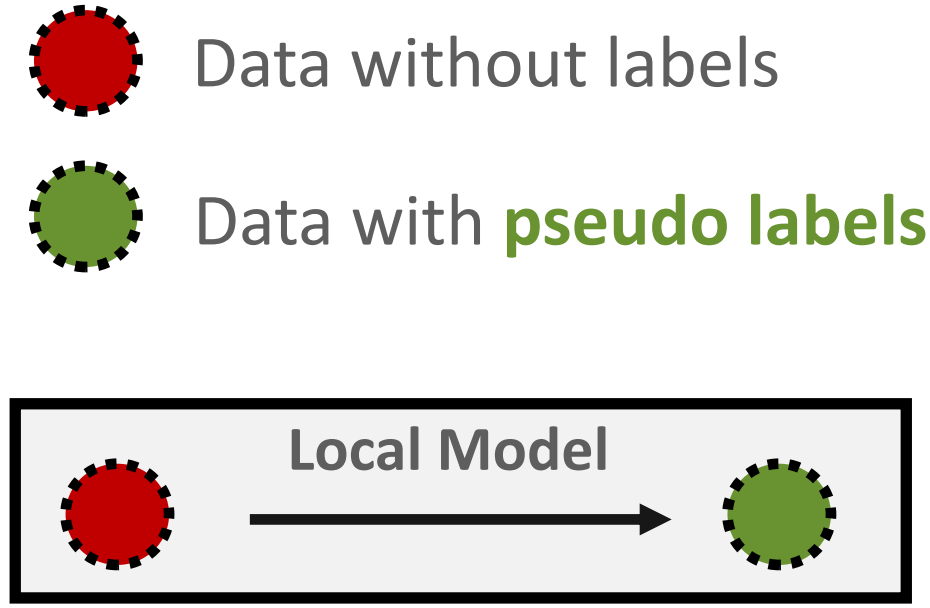
System Design



Pseudo labeling

The rational behind pseudo labeling:

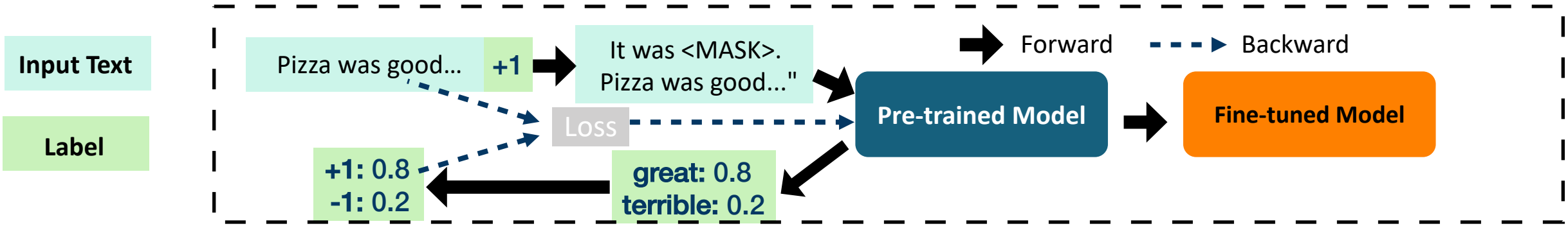
"Training with pseudo labels encourages the model to learn a decision boundary that lies in a region where the example density is lower. "

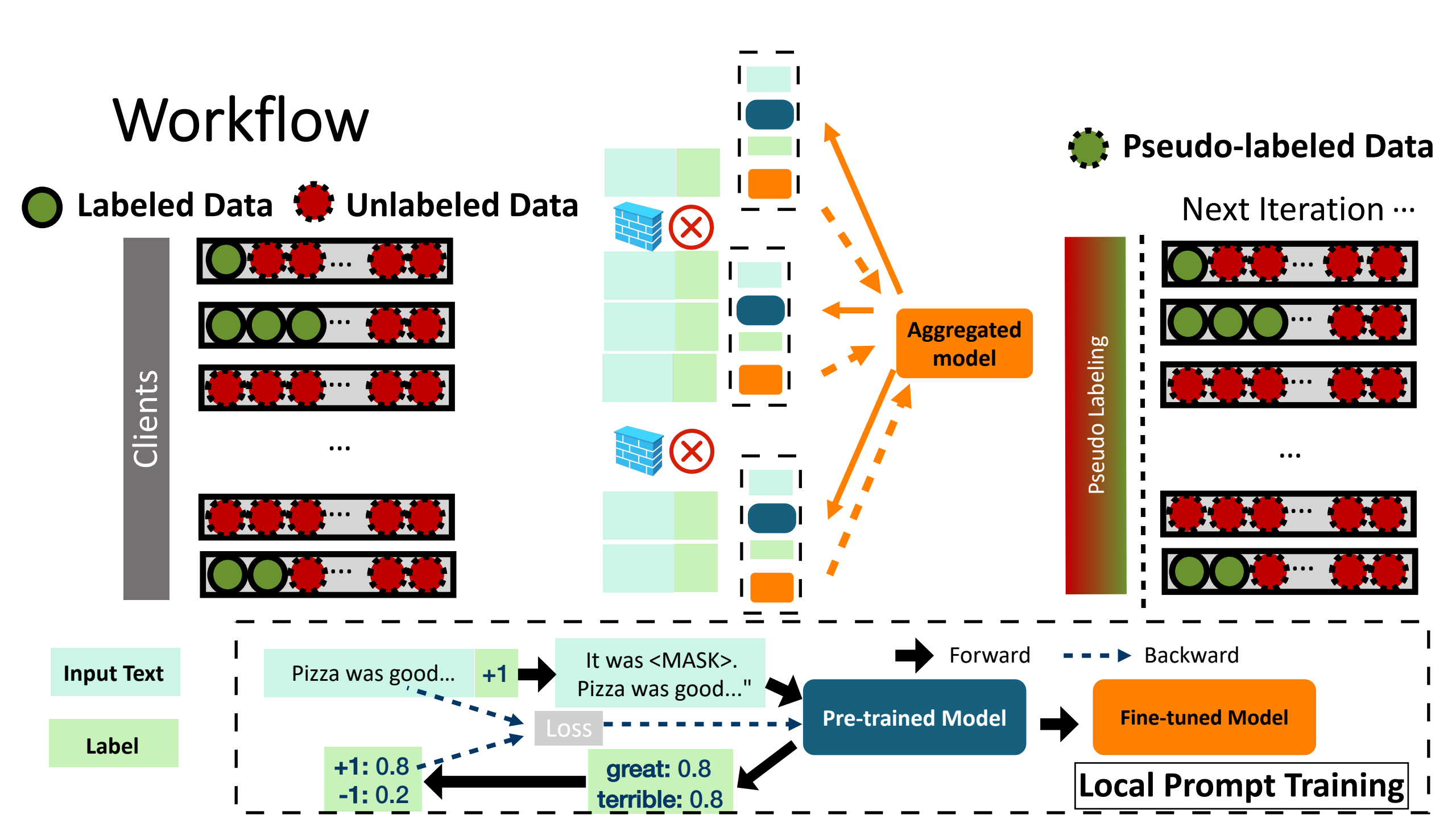


For example,
"great":0.9, "bad":0.1 rather than "great":0.6, "bad":0.4
Low class overlap ➡ Low entropy

Prompt learning

- T1 (label = +1): “Most delicious pizza I’ve ever had.”
- T2 (label = -1): “You can get better sushi for half the price.”
- T3 (label = ?): Pizza was good. Not worth the price.




Workflow
Labeled Data
Unlabeled Data
Pseudo-labeled Data
Clients
Aggregated model
Pseudo Labeling
Next Iteration ···
Input Text
Label
Pizza was good...
+1
It was <MASK>. Pizza was good..."
Loss
Forward
Backward
Pre-trained Model
Fine-tuned Model
+1: 0.8
-1: 0.2
great: 0.8
terrible: 0.8
Local Prompt Training

Experiments

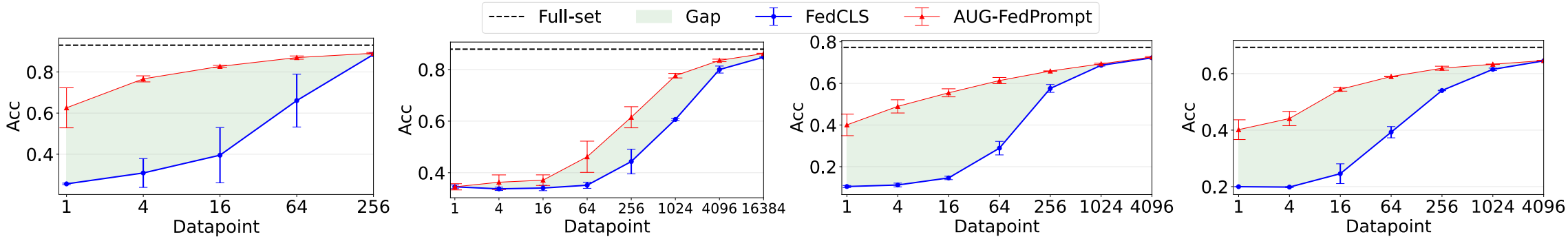
Convergence performance and system cost

Experiment setup

Dataset	Prompt	Train	Test
AGNEWS [20]	a (____) b	120,000	7.600
MNLI [24]	“a” ? ‖ ____, “b”	392,702	9,815
YAHOO [20]	[Category:] a ____ b	1,400,000	60,000
YELP-F [20]	It was ____. a	650,000	50,000

Server	Edge device
a GPU server with 8x NVIDIA A40.	Jetson TX2, 256-core NVIDIA Pascal™ GPU.

Performance across data scales



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

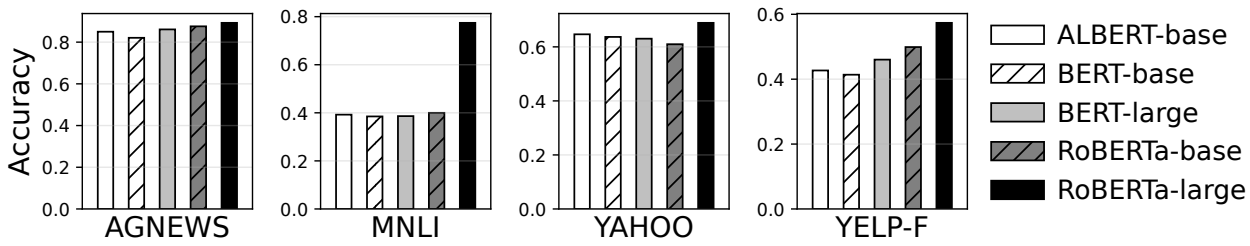
- AUG-FedPrompt enjoys up to 50%, 25%, 55%, 38% **accuracy** improvement separately for 4 datasets.
- For a usable accuracy, AUG-FedPrompt saves up to 99% training **data** compared to full-set federated finetuning.

Impact of Data Augmentation

Dataset		AGNEWS	MNLI	YAHOO	YELP-F
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
Skewed	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

- AUG-FedPrompt shows competitive performance under various federated few-shot learning settings, regardless of uniform or **skewed** label distribution.
- Up to 95% of unlabeled data is pseudo-labeled correctly at the convergence round.

System cost analysis



AUG-FedPrompt prefers large language models.

Model	ALBERT-base [29]	BERT-base [1]	BERT-large [1]	RoBERTa-base [25]	RoBERTa-large [25]
Memory (GB)	3.7	5.4	OOM (9.8)	5.8	OOM (10.4)
Latency (s)	1.4	1.9	~7.8	2.1	~8.1
Param. (M)	11.7	109.5	334.9	124.6	355.3

Finetuning these ‘behemoths’ can be extremely resource-intensive.

Future work

Challenges	Possible Solutions
Huge training latency	Model structure optimization [28, 29].
Large memory requirement	Rematerialization [30, 31], paging [32].
Excessive inference for pseudo labeling	Pacing [23, 33], early-exit [34, 35].
High communication cost	Quantization [36, 37], sparsity [38, 39].

Takeaway

- **Target:** Data labels can be scarce and skewed in federated learning.
- **Contribution:**

1. Data generator for federated few-shot learning task.
2. The lack and skewness of labeled data can significantly degrade federated learning convergence performance.
3. We propose AUG-FedPrompt, a novel federated few-shot learning system that orchestrates prompt learning and pseudo labeling.
4. AUG-FedPrompt shows competitive performance under various federated few-shot learning settings, requiring less than 1% data to be manually labeled.

- **Future work:** Improve resource efficiency.

Thank you for listening!

cdq@bupt.edu.cn

Appendix

Ablation study of key components

- Neither pseudo labeling nor prompt learning alone is enough to exhibit a usable accuracy

Dataset	Full-set (oracle)	Vanilla-FedFSL	Prompt-Only	Pseudo-Only	Both (Ours)
AGNEWS (skewed)	93.0	64.8±3.1	68.4±2.4	67.5±1.3	**90.2**±0.5
MNLI (skewed)	85.0	37.7±5.6	42.4±5.8	42.7±6.3	**75.7**±1.2
YAHOO (skewed)	78.0	24.4±10.3	41.8±4.3	31.0±2.0	**66.9**±1.1
YELP (skewed)	70.0	38.3±8.8	51.2±1.8	45.7±4.4	**58.2**±2.4
YELP (uniform)	70.0	54.0±0.1	58.1±1.5	57.0±2.2	**61.9**±0.7

Table 1: Convergence accuracy with 64 gold labels. "Full-Set" assumes every data is labeled (an oracle case). "skewed" means the gold labels are located on few clients instead of uniformly distributed across clients.

How to select prompts

- 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 [1].

[1]. Timo Schick and Hinrich Sch¨ utze, "Exploiting cloze questions for few shot text classification and natural language inference," arXiv preprint arXiv:2001.07676, 2020.

Why AUG-FedPrompt instead of FedPrompt?

- FedPrompt is used in previous work [1-3].
- However, we are different from them due to we are “hard” prompt, targeting few-shot learning instead of parameter-efficient.

[1] [Reduce Communication Costs and Preserve Privacy: Prompt Tuning Method in Federated Learning](https://arxiv.org/abs/2208.12268)
[2] [PromptFL: Let Federated Participants Cooperatively Learn Prompts Instead of Models — Federated Learning in Age of Foundation Model](https://arxiv.org/abs/2208.11625)
[3] [Privacy-preserving, Efficient, and Effective Machine Learning](https://www.researchsquare.com/article/rs-1682972/v1)

Difference between hard and soft prompt learning

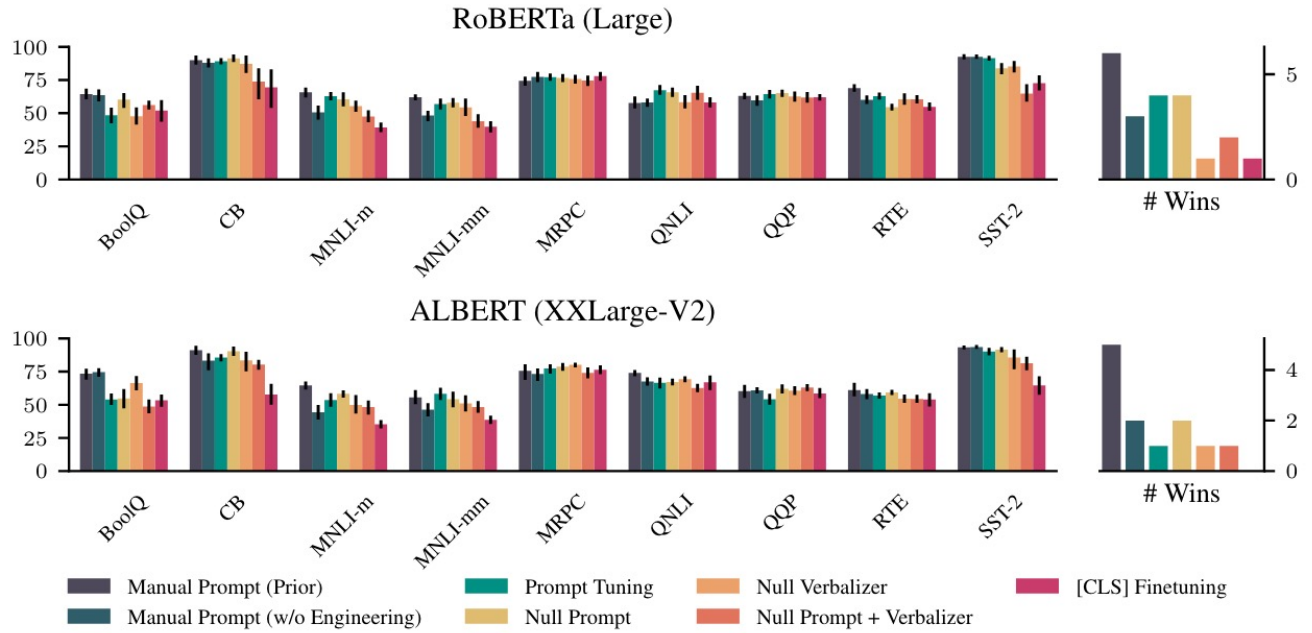


Figure 3: **Simplifying the Selection of Prompts**. We apply prompt-based finetuning in conjunction with six different types of prompts. We report accuracy or F_1 on each dataset. Manually-designed prompts from prior work achieve the best accuracy but require manual tuning on validation sets. On the other hand, null prompts and prompt tuning both perform competitively without requiring any tuning of the pattern.

Please see recent surveys on prompt learning [1-2] for more details.

Copied from [3]

[1]. Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258, 2021.
[2]. Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. arXiv preprint arXiv:2107.13586, 2021.
[3]. Logan IV, et al. Cutting Down on Prompts and Parameters: Simple Few-Shot Learning with Language Models.

Appendix

Figures from paper

Tips for preparing high-quality talks

Now you can present your paper just as you would in a typical international conference. It may feel strange talking to yourself, but rest assured, your presentation will have a live audience during the online conference.

Here are some tips for giving a high-quality talk.

Keep it simple. Do not try to squeeze a lot of material into each presentation slide. You have a time limit of 18 minutes, but you do not have limits on the total number of slides you have. Keep each slide as simple as possible, and explain one idea at a time. Use large font sizes. If you feel that you must cover a lot of material in the same slide, use **builds** to show your material a step at a time.

Avoid having too much technical detail. Your talk should be designed to get the audience interested in your paper, rather than replacing the paper. Do not try to present all the technical detail you have in your paper; instead, try to present only a few highlights of your original contributions in your paper, and emphasize high-level ideas on why your contributions are original in the context of related work.

Start strongly. The beginning of your talk is the most important as you need to grab the attention from your audience. Start from a compelling introduction of the background of your work, and motivate your ideas with convincing arguments.

Use examples. Your talk will be more understandable if you use a few simple examples, and work through your algorithm or theoretical proof in the context of your examples. Examples are your best friend in a high-quality talk.

Keep a calm pace. Do not rush through your presentation slides with a breakneck pace. Deliver your talk at a leisurely pace.

Use a timer. When you are presenting your slides, it is easy to lose track of time. Use a timer on your side, and be keenly aware of the amount of time left.

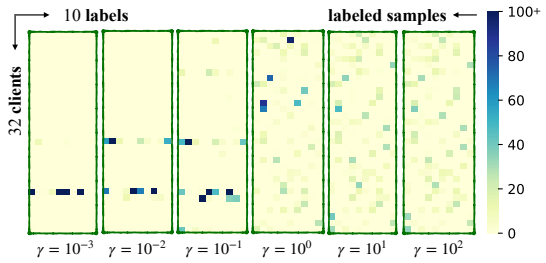


Figure 1: Visualizing the skewness of labeled data on YAHOO [20] with n=1024, ξ=32, γ being 10^n, n=-3,-2,..,2. Each sub-figure is a 32×10 matrix, where 32 is the number of clients and 10 is the number of labels. The intensity of each cell represents the number of labeled samples for a specific label in the client-side local data.

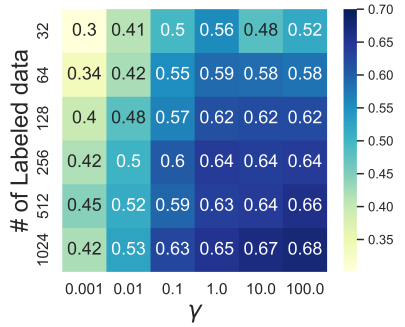
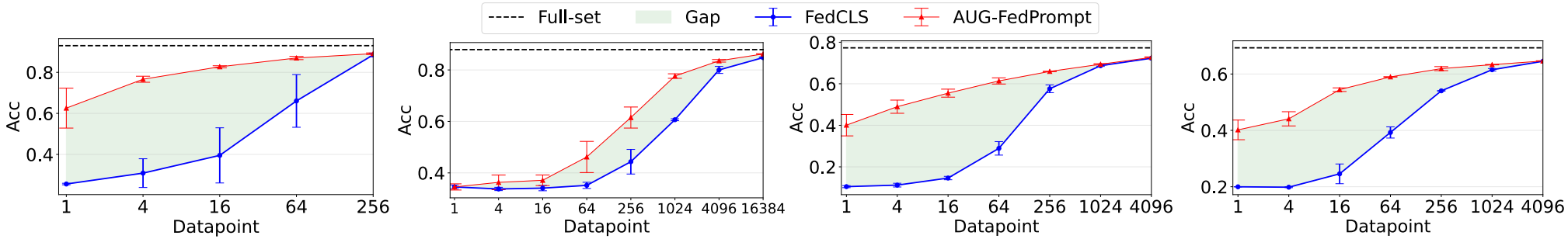


Figure 2: Average accuracy of federated few-shot learning under different data quantity and skewness. When skewness γ grows larger, labeled data will be more uniformly distributed, and vice versa. Dataset: YAHOO [20].



(a) AGNEWS **(b) MNLI** **(c) YAHOO** **(d) YELP-F**

Figure 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.

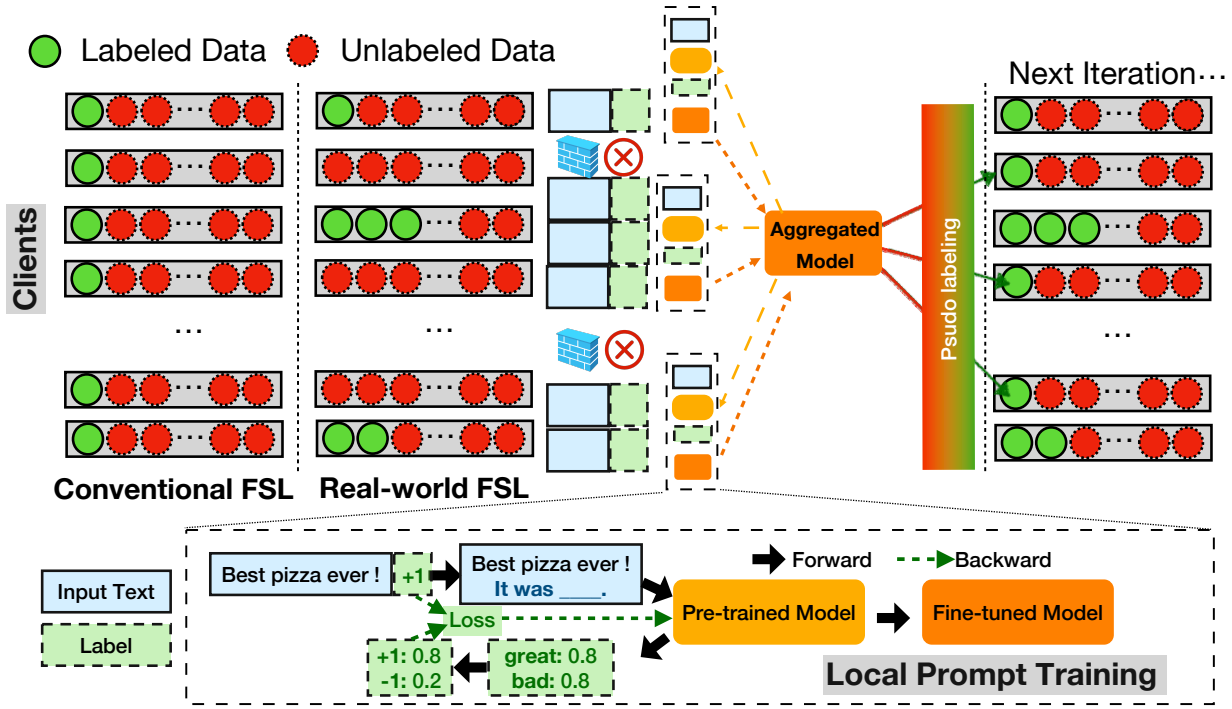


Figure 3: Workflow of AUG-FedPrompt.

Dataset	Prompt	Train	Test
AGNEWS [20]	a (____) b	120,000	7.600
MNLI [24]	"a" ? ‖ ____, "b"	392,702	9,815
YAHOO [20]	[Category:] a ____ b	1,400,000	60,000
YELP-F [20]	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 [11] where $\alpha = 1$.

Dataset		AGNEWS	MNLI	YAHOO	YELP-F
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
Skewed	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 YHAOO and YELP-F.

Challenges	Possible Solutions
Huge training latency	Model structure optimization [28, 29].
Large memory requirement	Rematerialization [30, 31], paging [32].
Excessive inference for pseudo labeling	Pacing [23, 33], early-exit [34, 35].
High communication cost	Quantization [36, 37], sparsity [38, 39].

Table 3: Challenges and possible solutions.

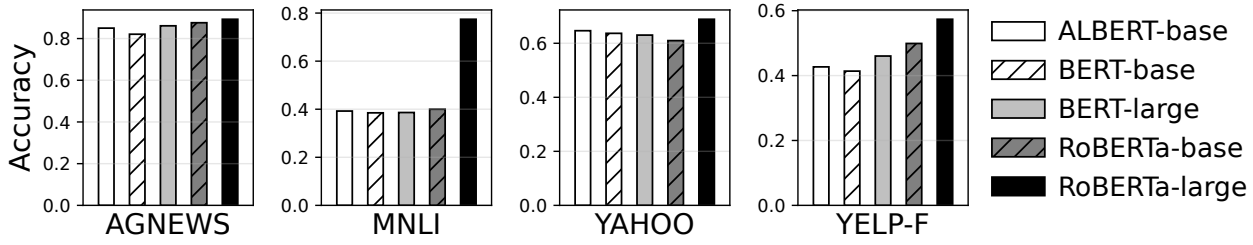


Figure 5: AUG-FedPrompt convergence performance with different models and datasets. 0.1% labeled data uniformly distributed in 32 clients.

Model	ALBERT-base [29]	BERT-base [1]	BERT-large [1]	RoBERTa-base [25]	RoBERTa-large [25]
Memory (GB)	3.7	5.4	OOM (9.8)	5.8	OOM (10.4)
Latency (s)	1.4	1.9	~7.8	2.1	~8.1
Param. (M)	11.7	109.5	334.9	124.6	355.3

Table 4: System cost of different NLP models. Tested on NVIDIA TX2. Batch size: 4.