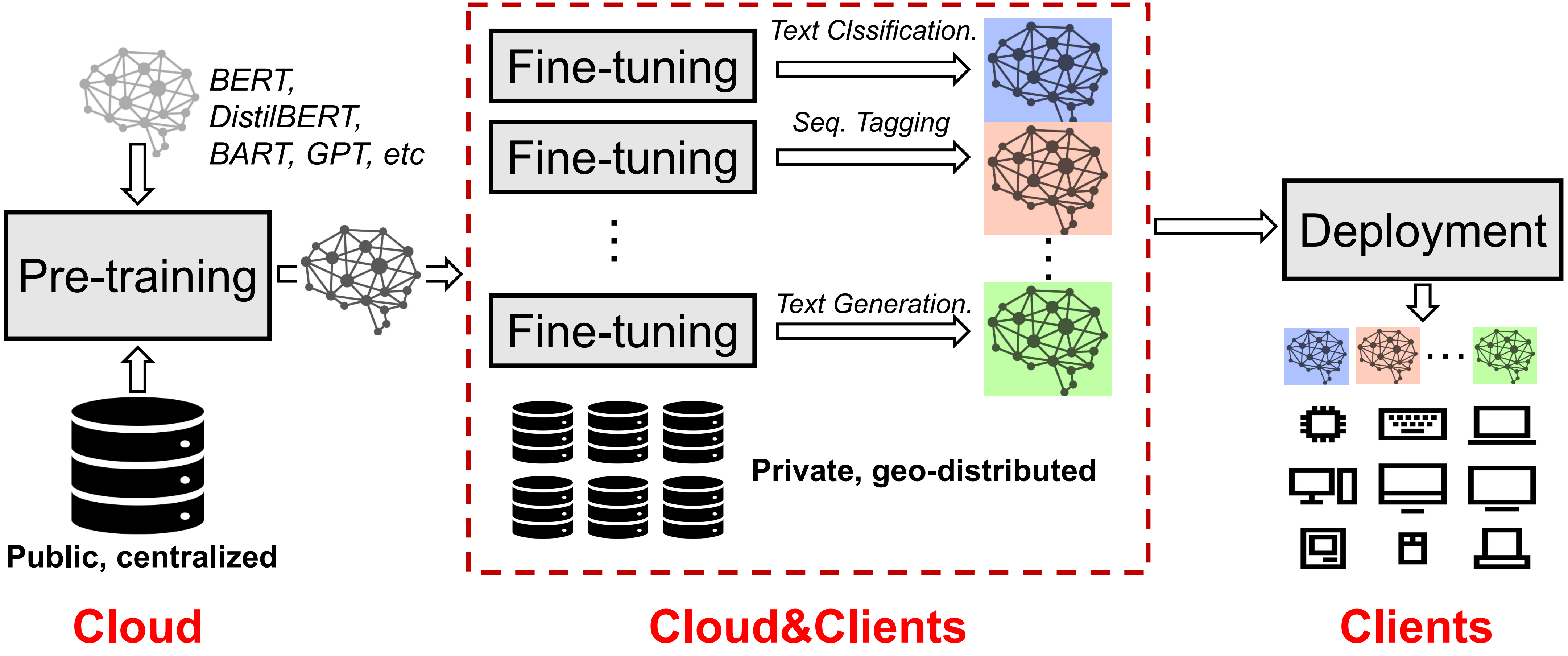
The 29th Annual International Conference
On Mobile Computing And Networking

Federated Few-shot Learning for Mobile NLP

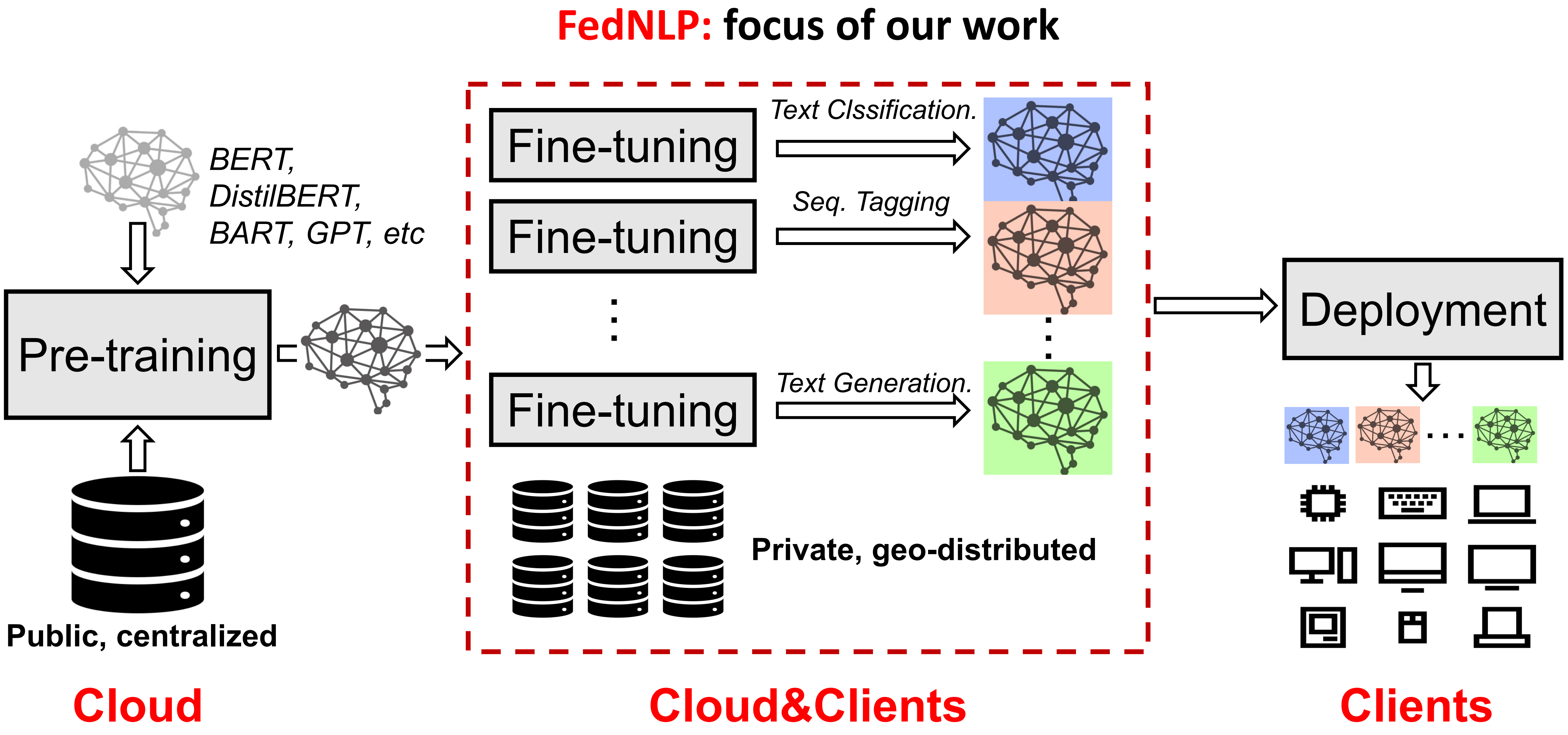
Dongqi Cai[1], Shangguang Wang[1], Yaozong Wu[1], Felix Xiaozhu Lin[2], Mengwei Xu[1]

1 Beiyou Shenzhen Institute
2 University of Virginia

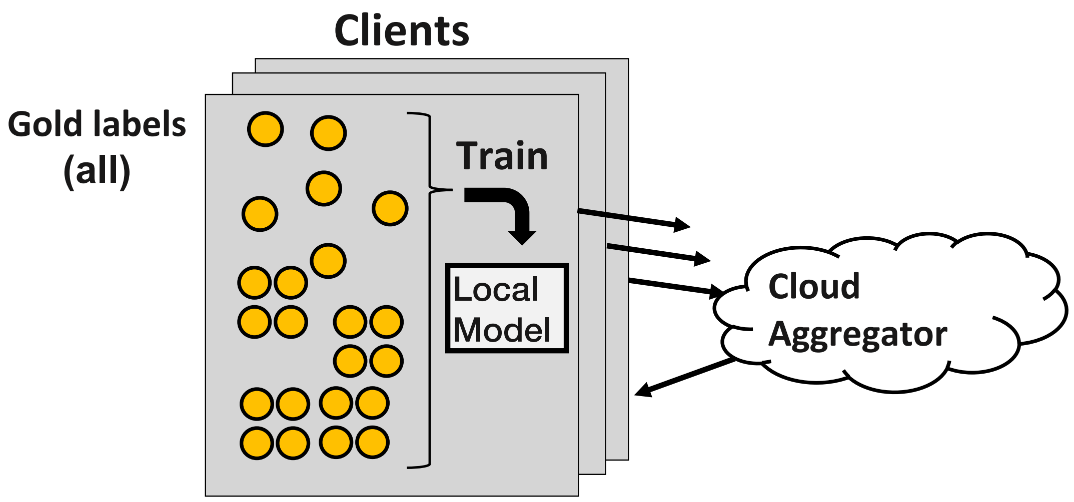
FedNLP: focus of our work



Where is the training data coming from?

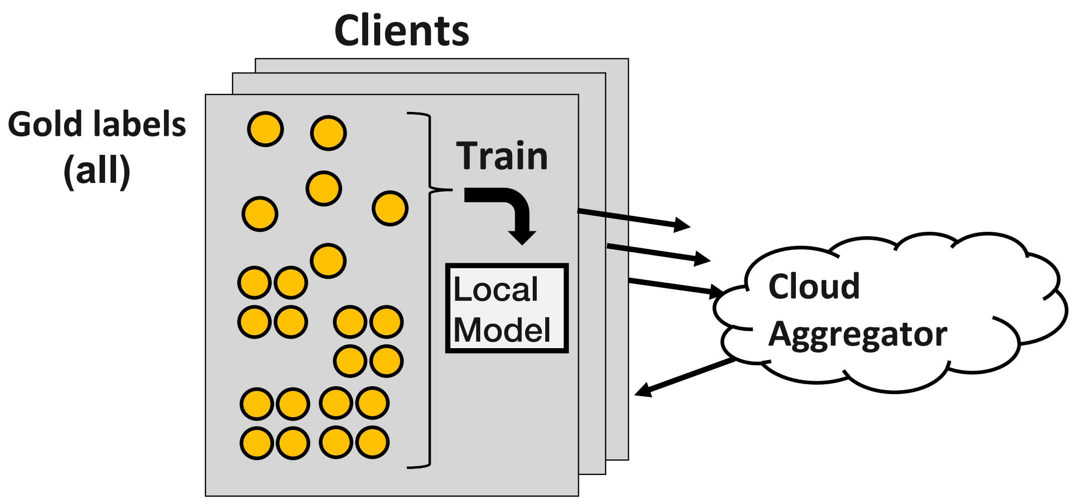


Background: Federated Few-shot Learning (FedFSL)



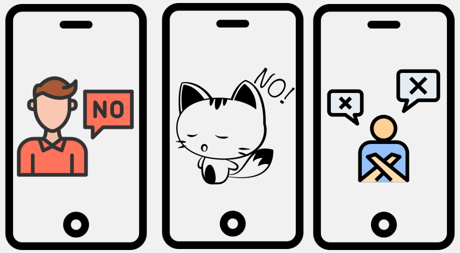
(a) Classic FL: rely on abundant labels

Background: Federated Few-shot Learning (FedFSL)

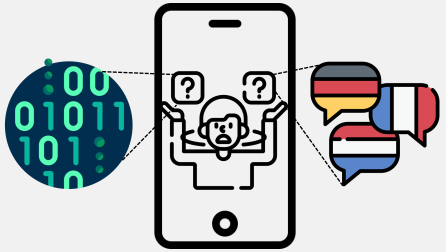


(a) Classic FL: rely on abundant labels

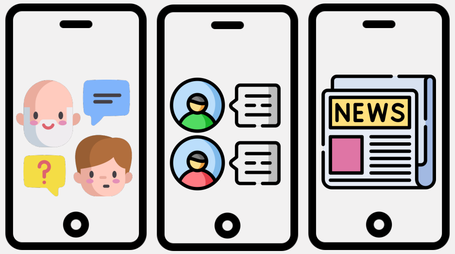
Well-curated labeled data is scarce on mobile devices



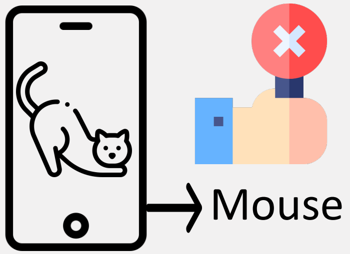
Users lack willingness.



Users lack expertise.

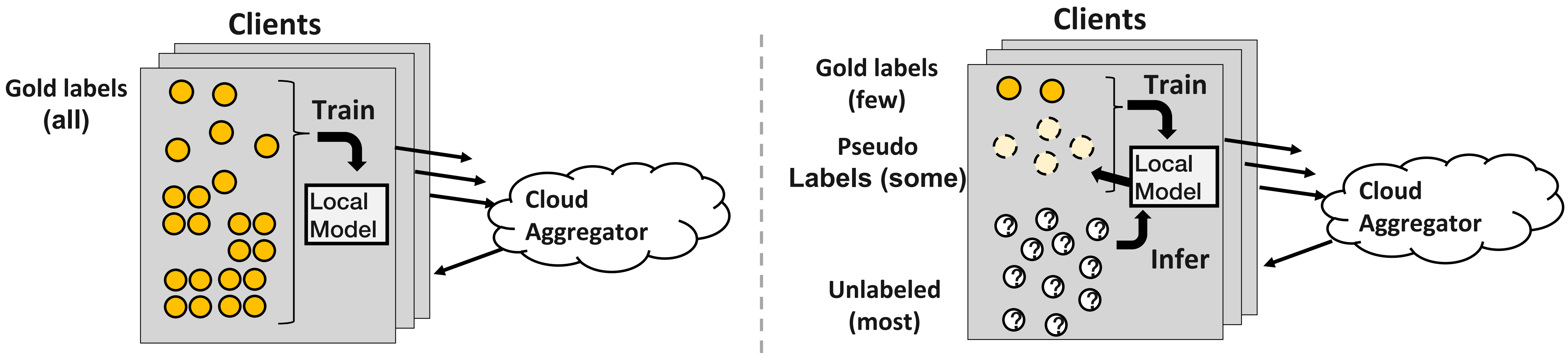


Diverse NLP tasks.



Mislabels are common.

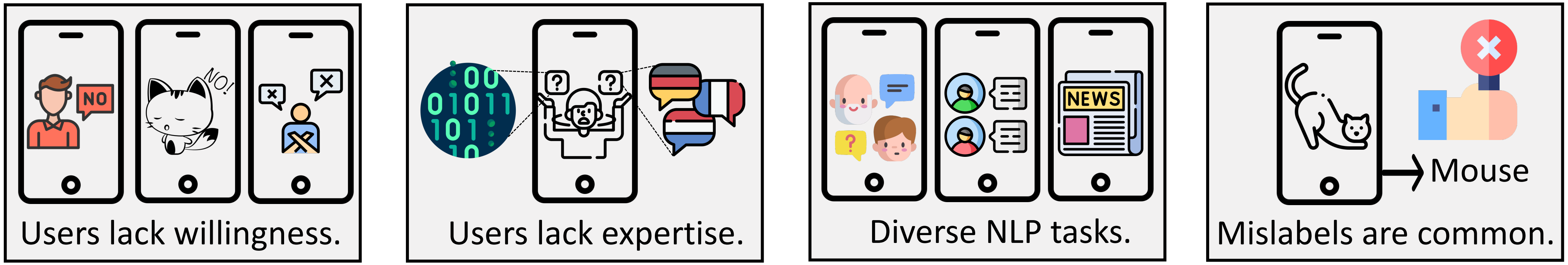
Background: Federated Few-shot Learning (FedFSL)



(a) Classic FL: rely on abundant labels

(b) Our FedFSL Scenario

Well-curated labeled data is scarce on mobile devices



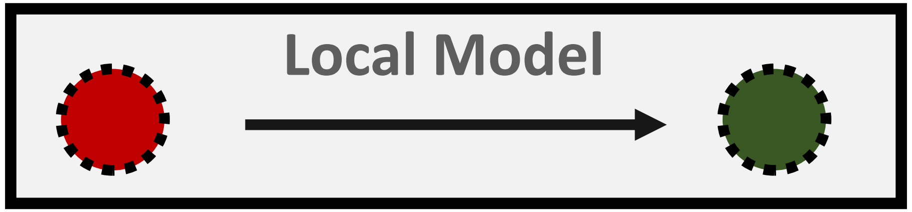
Background: Pseudo labeling

The rational behind pseudo labeling:

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

Data without labels

Data with **pseudo labels**

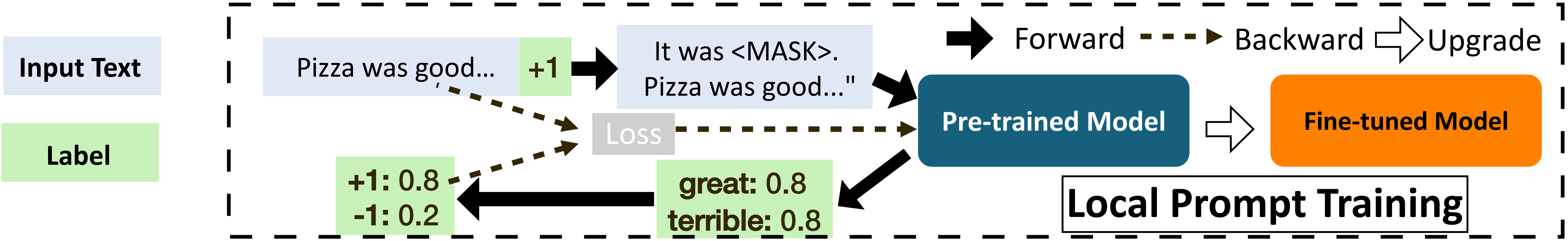


For example,
"great":0.9, "bad":0.1 rather than "great":0.6, "bad":0.4
Low class overlap ➡ Low entropy

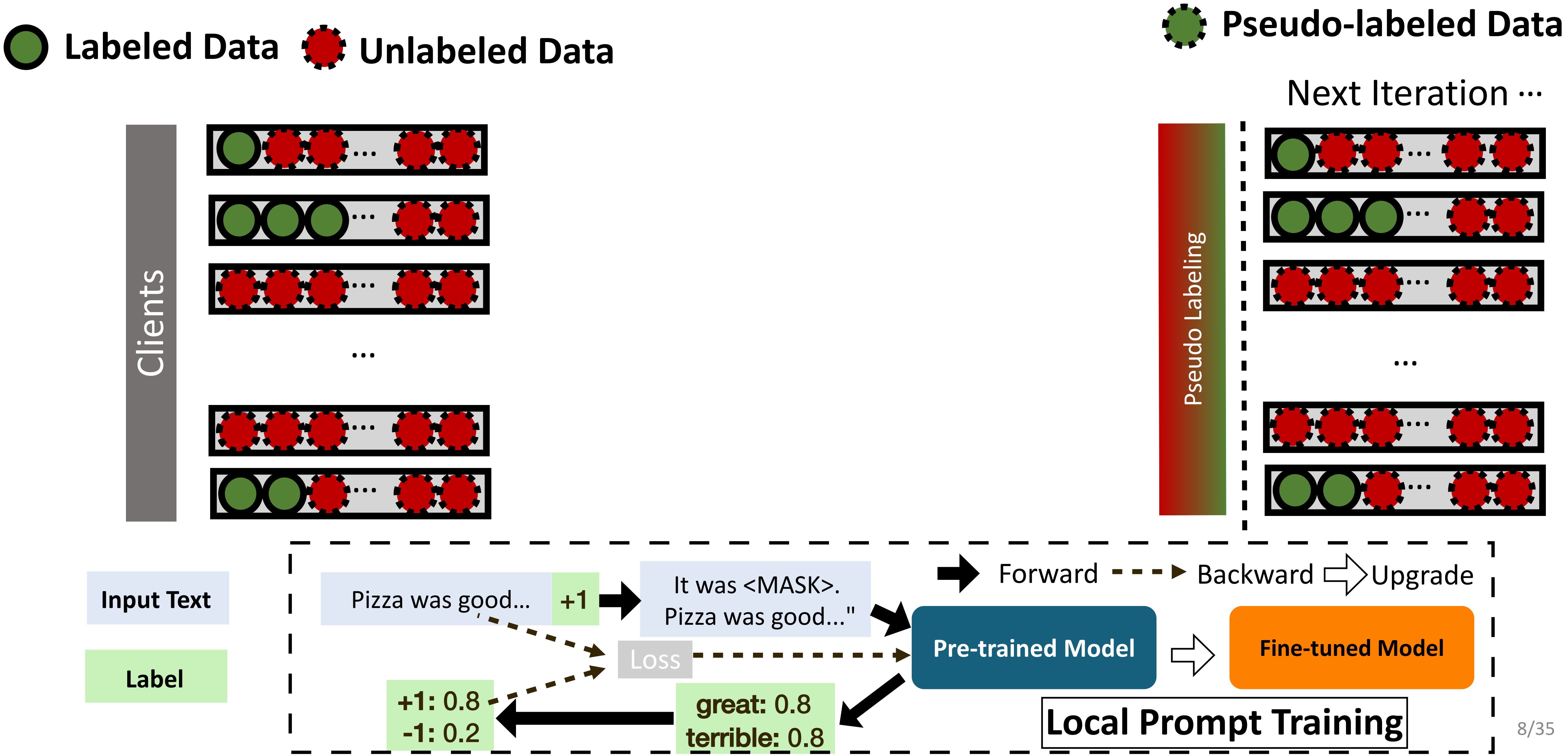
Background: Prompt learning

- 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.

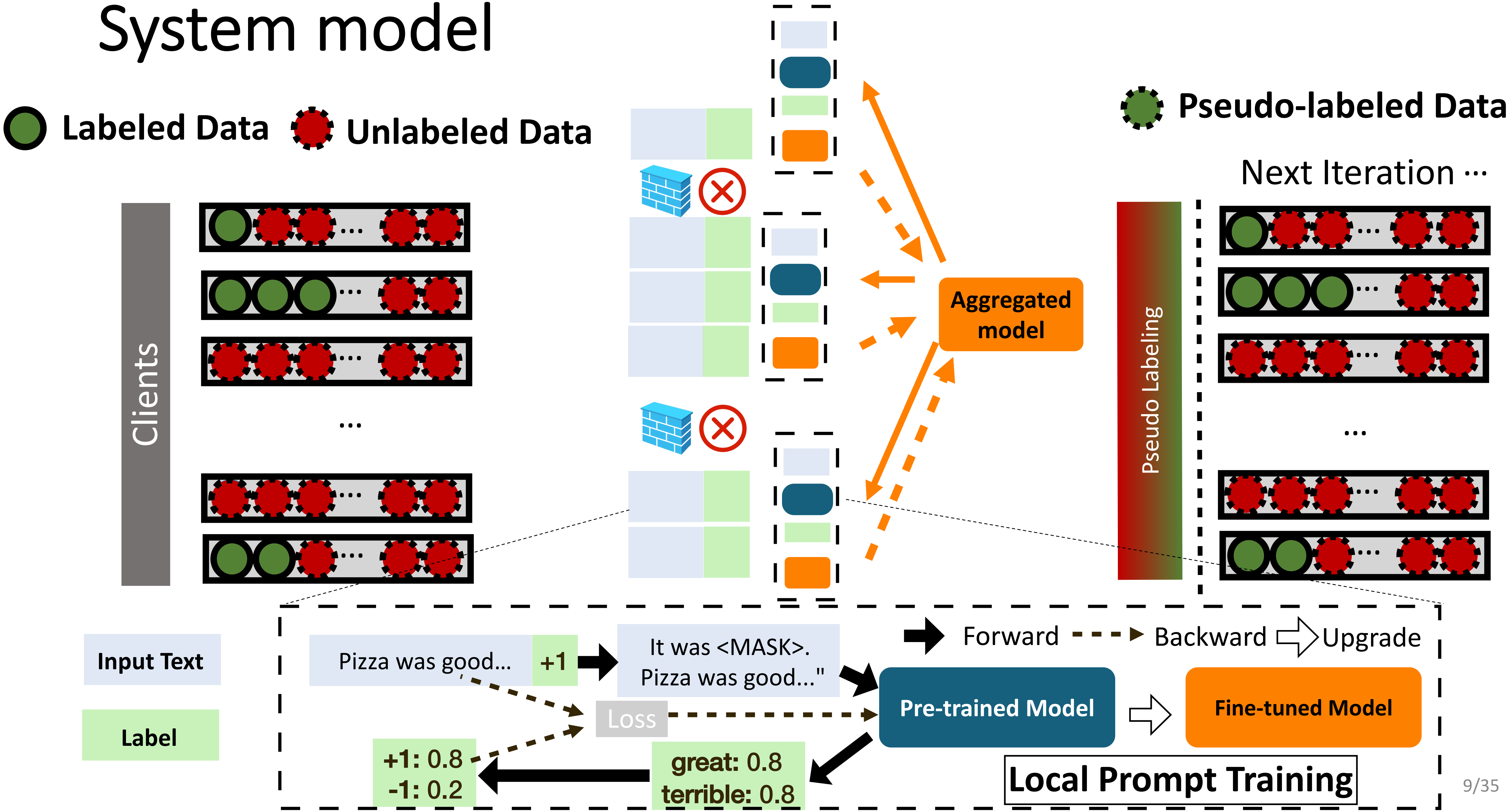
Prompt "It was <MASK>. Pizza was good..."



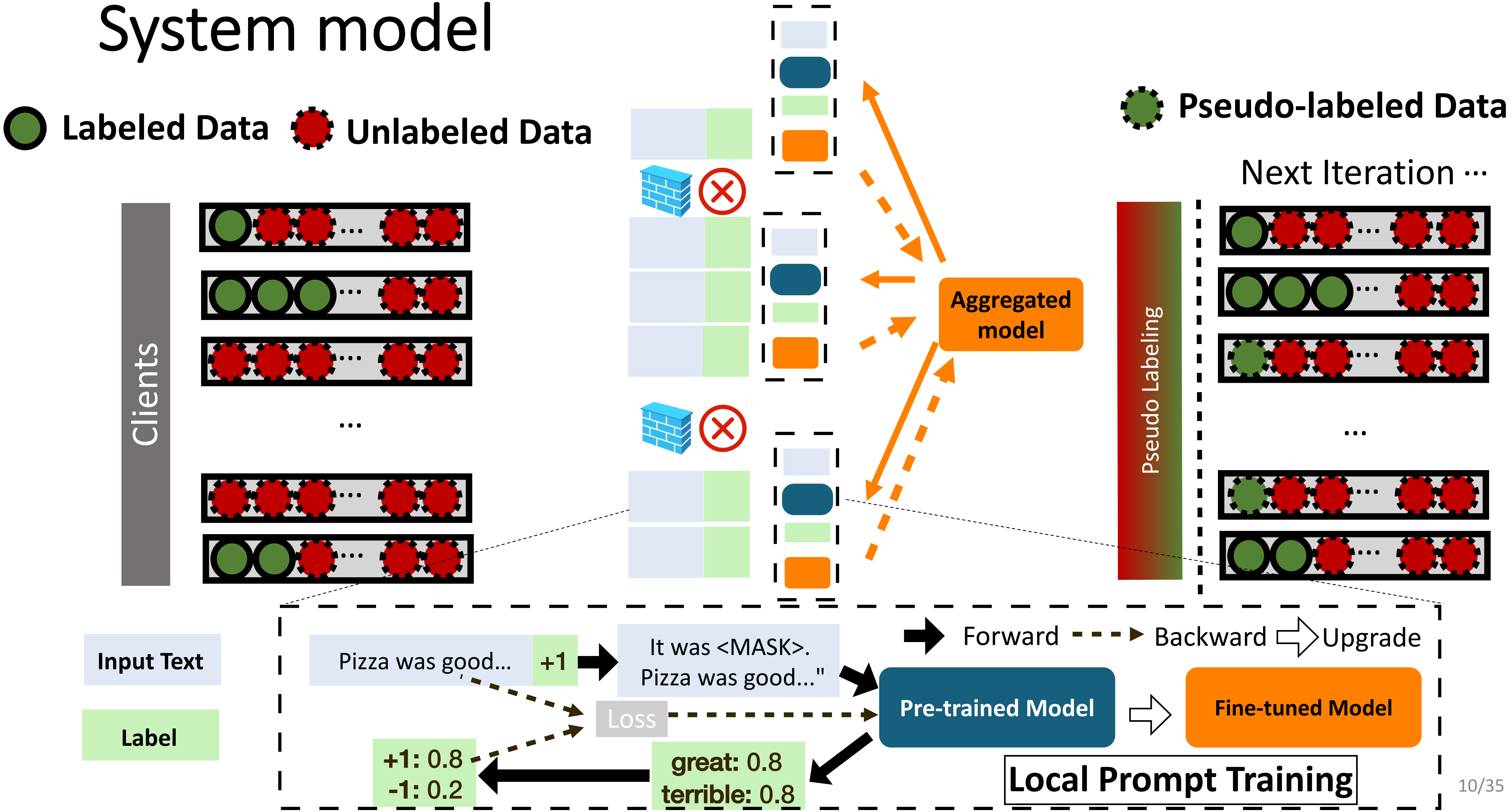
System model



System model

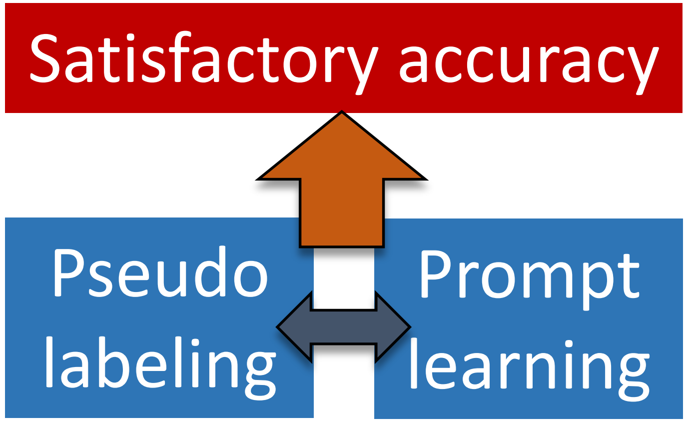


System model



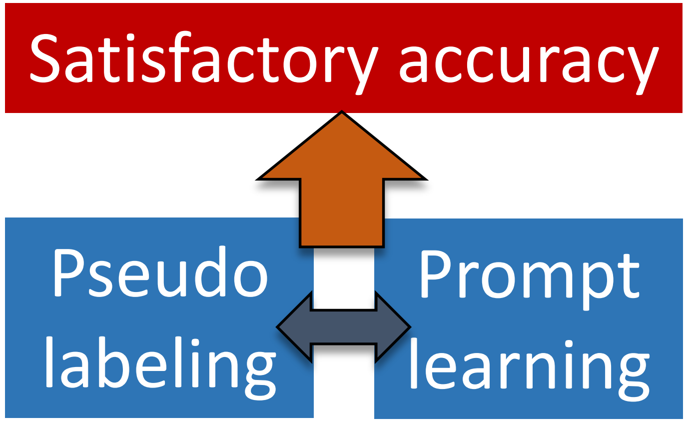
Preliminary: FedFSL performance

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	**77.4**±1.2
YAHOO (skewed)	78.0	24.4±10.3	41.8±4.3	31.0±2.0	**66.9**±1.1
YELP-F (skewed)	70.0	38.3±8.8	51.2±1.8	45.7±4.4	**58.2**±2.4
YELP-F (uniform)	70.0	54.0±0.1	58.1±1.5	57.0±2.2	**61.9**±0.7



Preliminary: FedFSL performance

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	**77.4**±1.2
YAHOO (skewed)	78.0	24.4±10.3	41.8±4.3	31.0±2.0	**66.9**±1.1
YELP-F (skewed)	70.0	38.3±8.8	51.2±1.8	45.7±4.4	**58.2**±2.4
YELP-F (uniform)	70.0	54.0±0.1	58.1±1.5	57.0±2.2	**61.9**±0.7



How about the system cost?

Challenge: FedFSL system cost

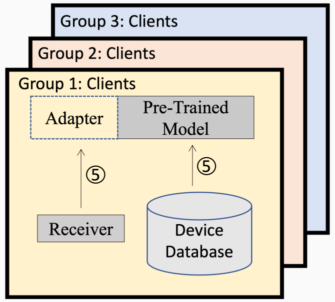
Training efficiency

Adapter

LoRA

BitFit

...



AdaFL: Efficient FL

Challenge: FedFSL system cost

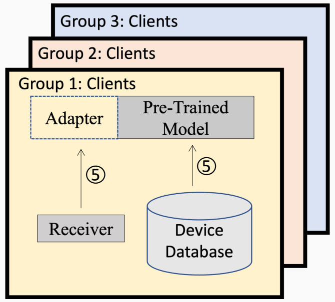
Training efficiency

Adapter

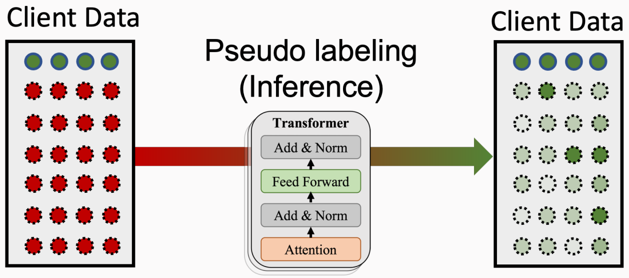
LoRA

BitFit

...

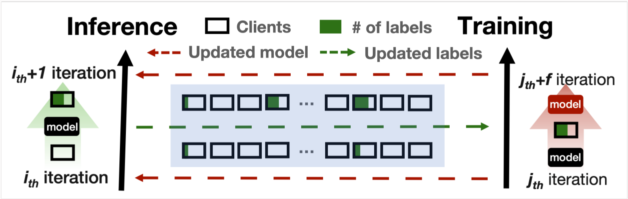


Labeling redundancy



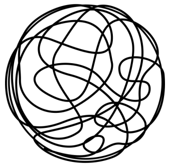
>87%

Labeling-training pace



Number?

...



Time?

...

Frequency?

AdaFL: Efficient FL

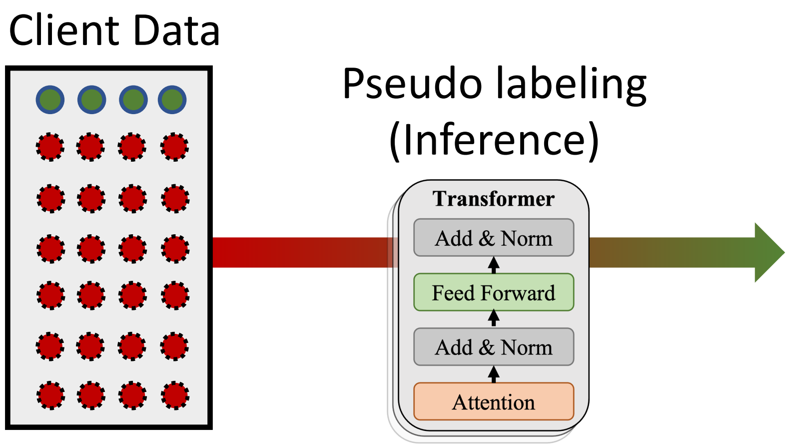
FeS: Train without labels

2021.12

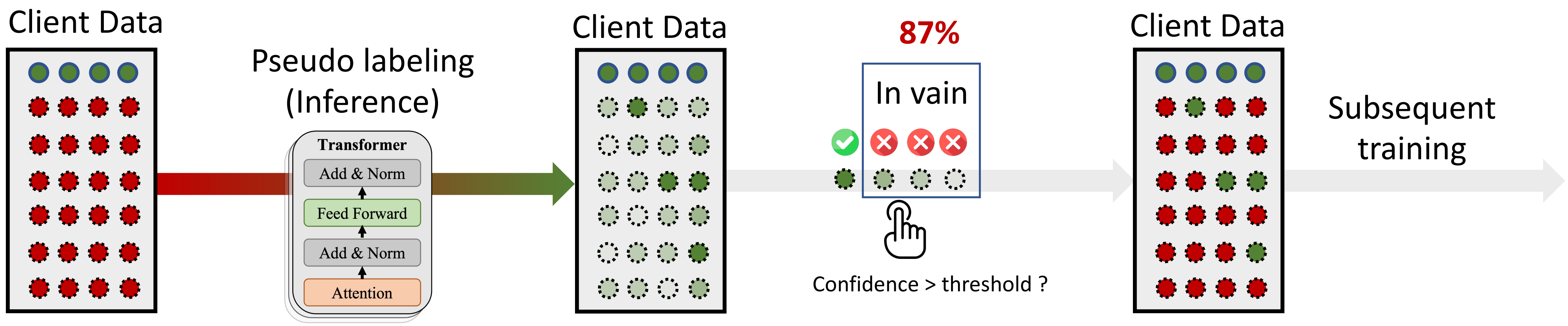
2022.08

2023.03

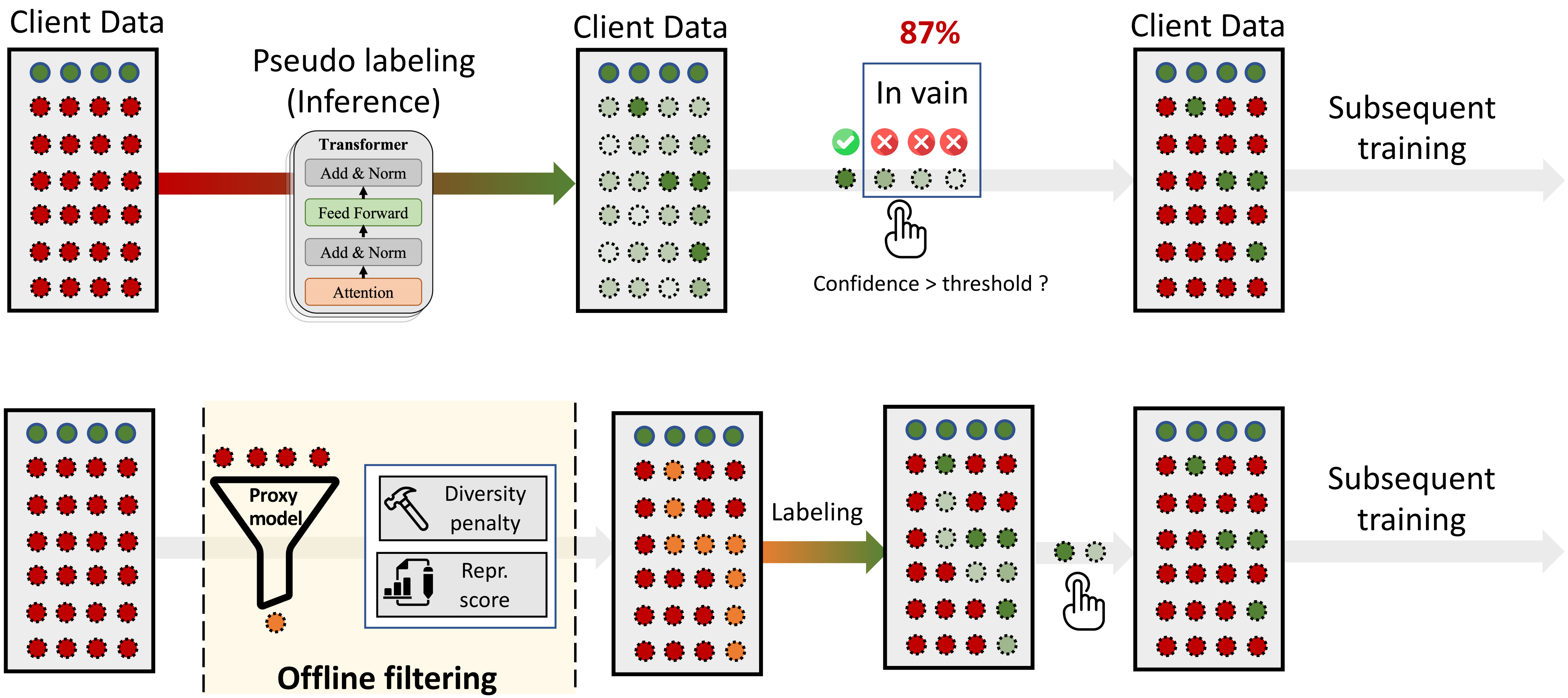
Design 1: Representational Filtering



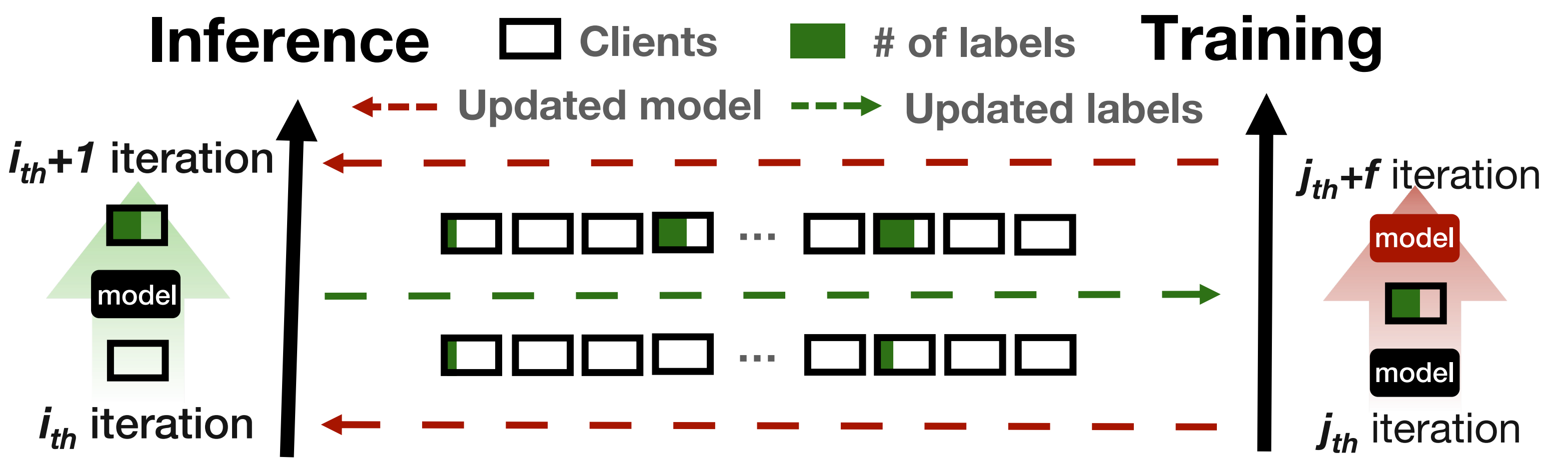
Design: Representational Filtering



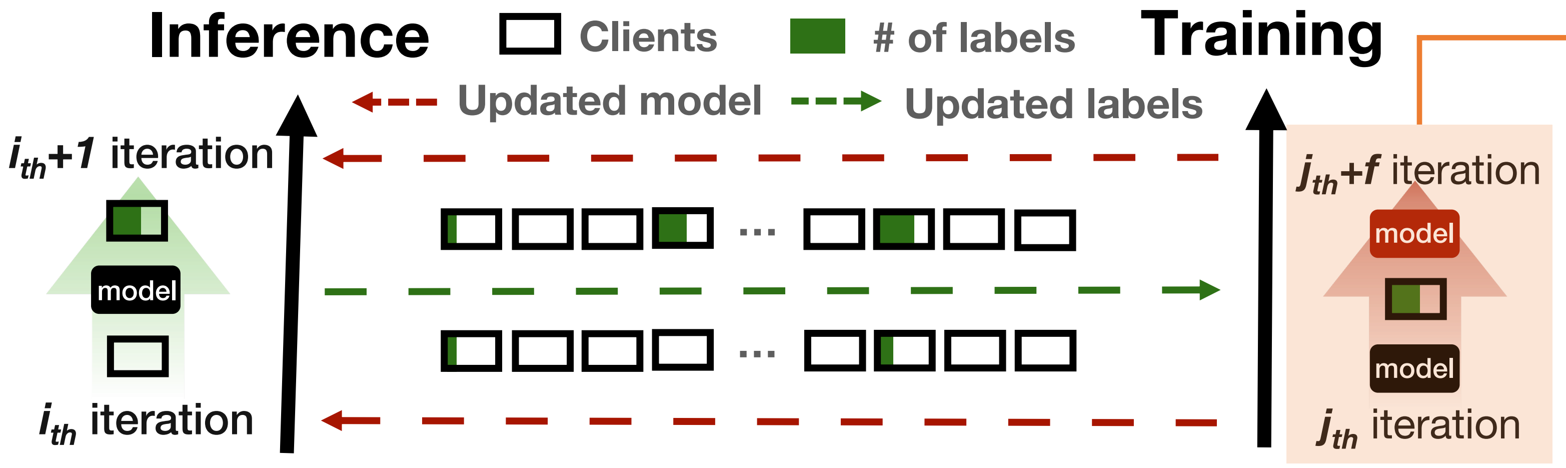
Design: Representational Filtering



Design: Curriculum Pacing



Design: Curriculum Pacing



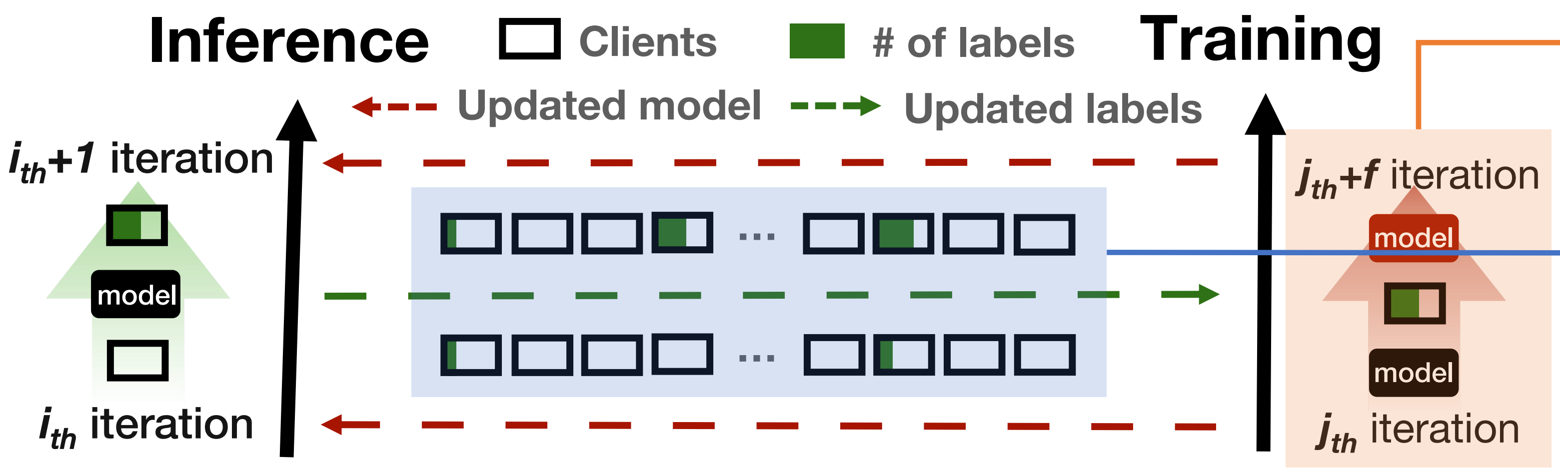
f: frequency of updating pseudo labels

n: number of clients selected to perform pseudo labeling

k: ratio of selected pseudo labels for the subsequent training

Pacing configuration <f,n,k>

Design: Curriculum Pacing



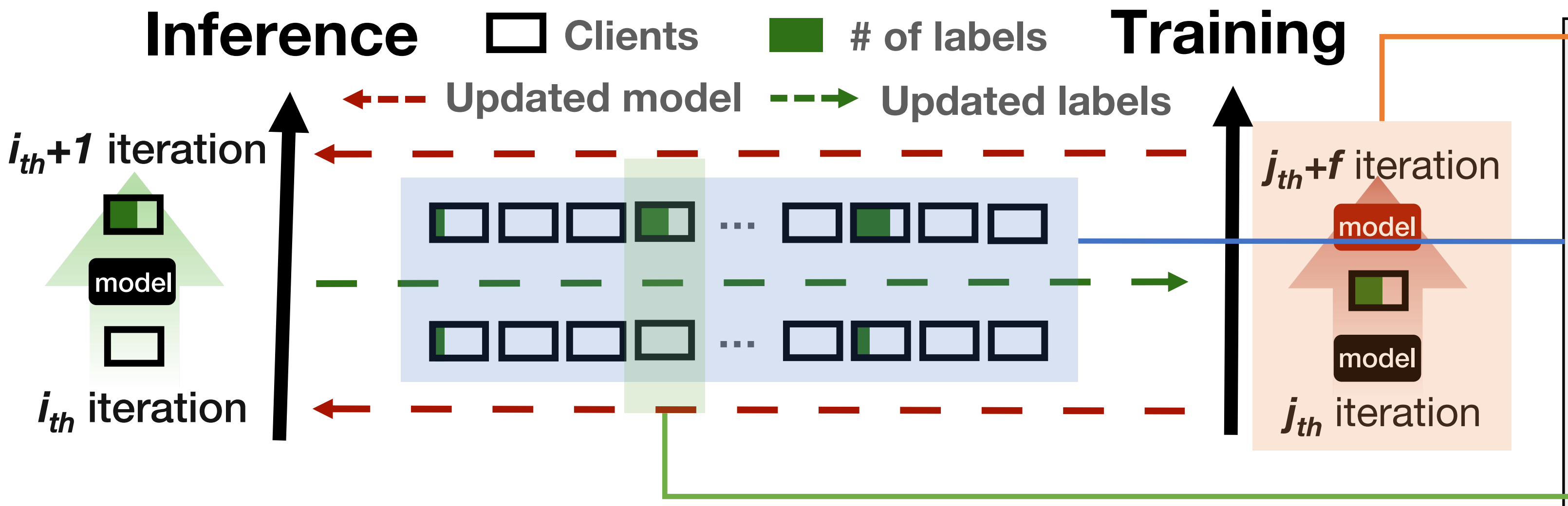
f: frequency of updating pseudo labels

n: number of clients selected to perform pseudo labeling

k: ratio of selected pseudo labels for the subsequent training

Pacing configuration <f,n,k>

Design: Curriculum Pacing



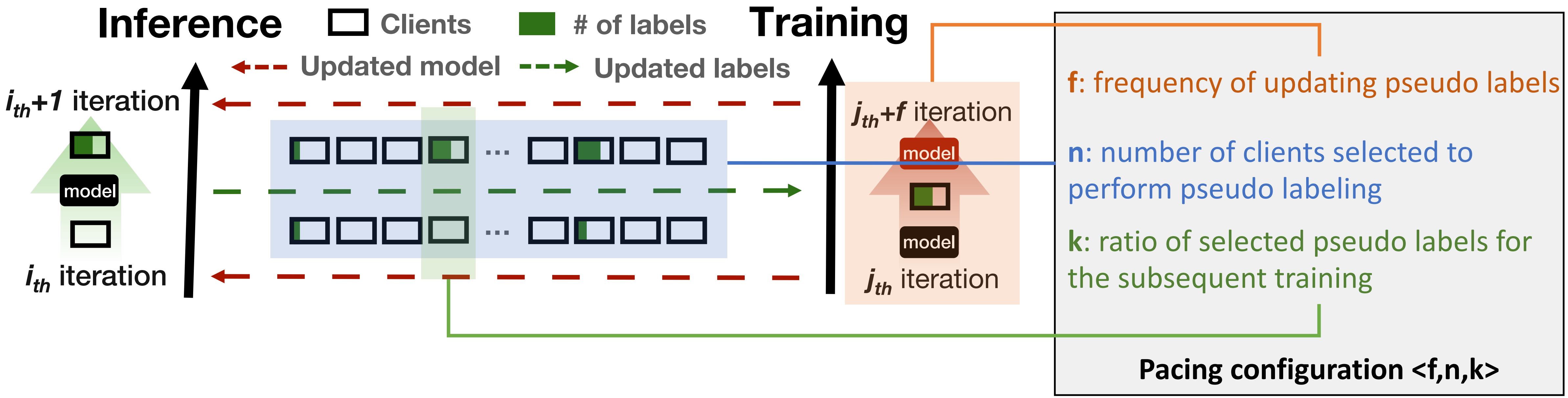
f: frequency of updating pseudo labels

n: number of clients selected to perform pseudo labeling

k: ratio of selected pseudo labels for the subsequent training

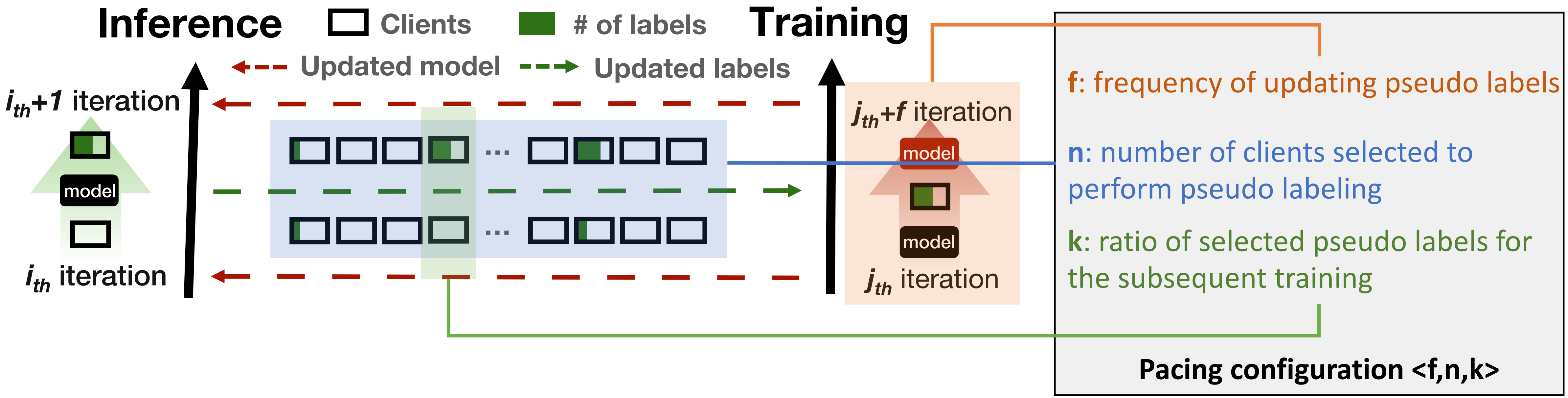
Pacing configuration <f,n,k>

Design: Curriculum Pacing



- Progressively speed up the pseudo labeling speed, i.e., adding more pseudo labels at a higher frequency.

Design: Curriculum Pacing

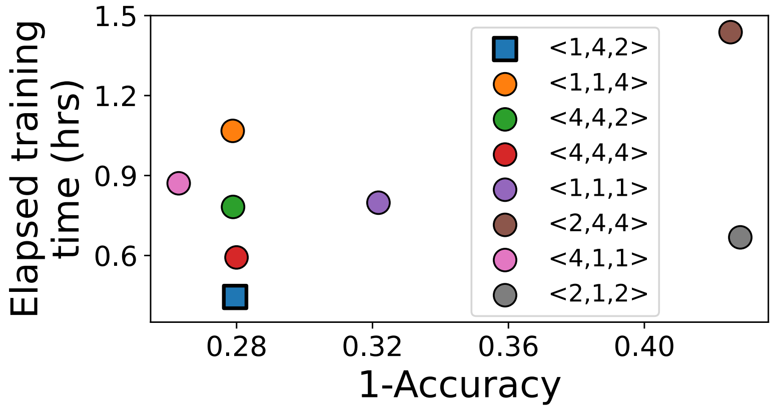


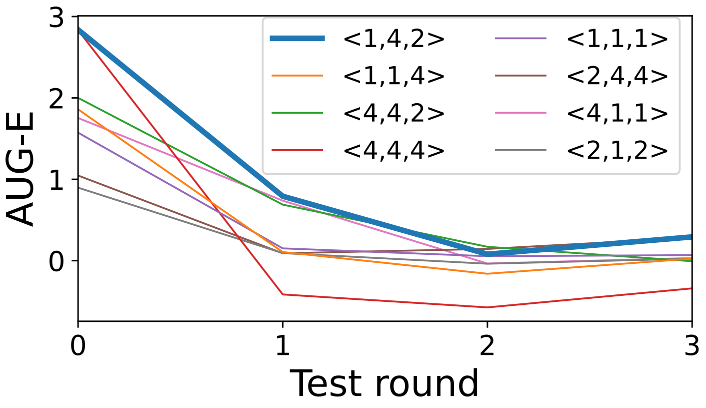
- Progressively speed up the pseudo labeling speed, i.e., adding more pseudo labels at a higher frequency.
- **Progressive upgrading is only a coarse-grained plan, how to control the pace more concisely?**

Design: Curriculum Pacing

Augment efficiency (AUG-E):
measure the gradient of the time-to-accuracy curve to search for an effective configuration with low cost

$$AUG-E(f,n,k) \leftarrow \frac{\eta\Delta(acc)}{C_{\text{infer}}(f,n)+\theta\cdot C_{\text{train}}(k)}$$



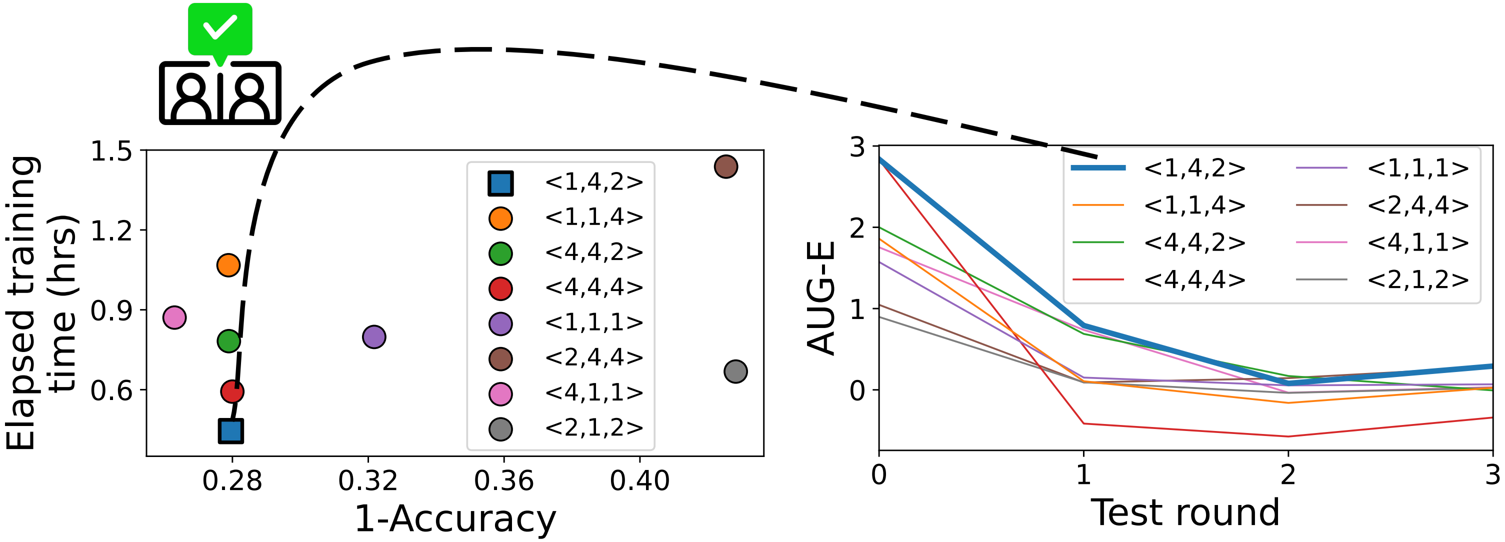


Our system selects a configuration with **best AUG-E** from a candidate list (hand-picked through extensive offline experiments) for future pseudo labeling.

Design: Curriculum Pacing

Augment efficiency (AUG-E):
measure the gradient of the time-to-accuracy curve to search for an effective configuration with low cost

$$\boxed{AUG-E(f,n,k) \leftarrow \frac{\eta\Delta(acc)}{C_{\text{infer}}\,(f,n)+\theta\cdot C_{\text{train}}\,(k)}}$$



Our system selects a configuration with **best AUG-E** from a candidate list (hand-picked through extensive offline experiments) for future pseudo labeling.

Evaluation: Setup

- **Implementation**
  - FedNLP[1]
  - PET[2]
- **Setups**
  - 2 devices (TX2, RPI 4B)
  - 2 models (RoBERTa-base & large)
  - 4 datasets
- **Baselines**
  1. Vanilla Fine-Tuning (FedCLS)
  2. Vanilla Few-shot Tuning (FedFSL)
  3. Vanilla Few-shot Tuning + Bias-tuning (FedFSL-BIAS)

64 labels in total instead of per client

Dataset	AGNEWS [108]	MNLI [89]	YAHOO [108]	YELP-F [108]
# Training	120k	392.7k	1.4M	650k
# Test	7.6k	9.8k	60k	50k
# Clients	100	1000	1000	1000
# Labels	64	64	64	64
Distribution	Skewed	Uniform	Skewed	Skewed
Prompt	a ____ b	a ?____, b	Category: a ____ b	It was ____. a

Setup	Labeling		Training	
	Pacing	**Optimization**	**Method**	**Optimization**
FedCLS	/	/	Head-based	/
FedFSL	Static	/	Prompt-based	/
FedFSL-BIAS	Static	/	Prompt-based	Bias-only tuning
FeS (Ours)	Curriculum (§3.1)	Filtering (§3.2)	Prompt-based (§2.2)	Depth/Capacity Co-planning (§3.3)

[1] Yuchen Lin B, He C, Zeng Z, et al. FedNLP: Benchmarking Federated Learning Methods for Natural Language Processing Tasks[J]. Findings of NAACL, 2022.
[2] Schick T, Schütze H. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference[C]//Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. 2021: 255-269.

Evaluation: End-to-end Performance

- Our system significantly speeds up model convergence at high accuracy.

Dataset	AGNEWS					MNLI					YAHOO					YELP-F				
Perf.	Conv. Acc.	Time-to-acc (hr)				Conv. Acc.	Time-to-acc (hr)				Conv. Acc.	Time-to-acc (hr)				Conv. Acc.	Time-to-acc (hr)			
		TX2		RPI			TX2		RPI			TX2		RPI			TX2		RPI	
		acc1	acc2	acc1	acc2		acc1	acc2	acc1	acc2		acc1	acc2	acc1	acc2		acc1	acc2	acc1	acc2
FedCSL	27.9%	X	X	X	X	37.3%	X	X	X	X	34.6%	X	X	X	X	35.7%	X	X	X	X
FedFSL	92.5%	3.3	3.3	50.0	50.0	74.1%	9.2	X	137.5	X	84.3%	8.3	X	125.0	X	75.3%	2.1	X	31.3	X
FedFSL-BIAS	92.5%	1.7	1.7	25.0	25.0	88.1%	0.5	11.7	7.5	175.0	85.9%	3.3	5.3	50.0	80.0	79.4%	0.2	2.1	2.5	10.4
Ours	95.9%	0.4	0.4	5.5	5.5	92.2%	0.2	0.8	2.5	12.5	88.5%	0.3	0.7	5.0	10.0	86.8%	0.1	0.5	1.3	7.5

260× 68.0%

Table 1: The final convergence accuracy ("Conv. Acc.") and the elapsed training time ("Time-to-acc") to reach different relative accuracy. "acc1"/"acc2" are the final convergence accuracy of FedFSL/FedFSL-BIAS, respectively. "X" means the accuracy cannot be achieved.

Evaluation: Key deign

- Our key designs contribute to the results significantly.

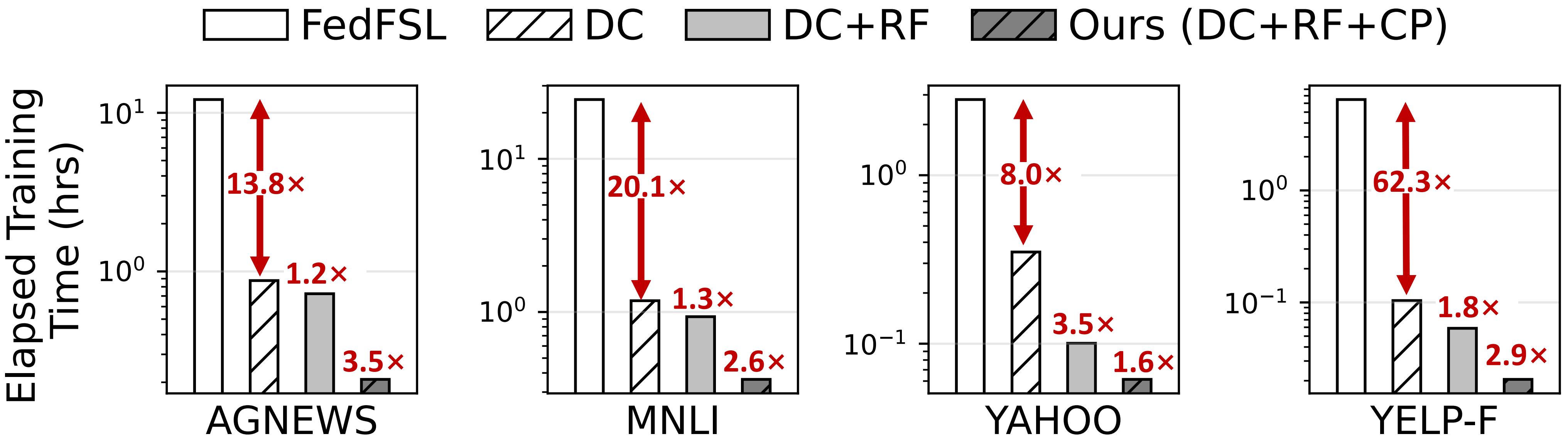


Fig. 1: Model convergence delays with and without Our system's key designs, showing their significance. **DC**: training depth/capacity co-planning; **RF**: representative filtering; **CP**: curriculum pacing.

Evaluation: System Cost

Our system is resource-efficient.

- It saves up to 3000.0× **network traffic**. (Fig. 1)
- It reduces up to 41.2× **energy consumption**. (Fig. 2)
- It reduces the **memory usage** by 4.5×. (Fig. 3)

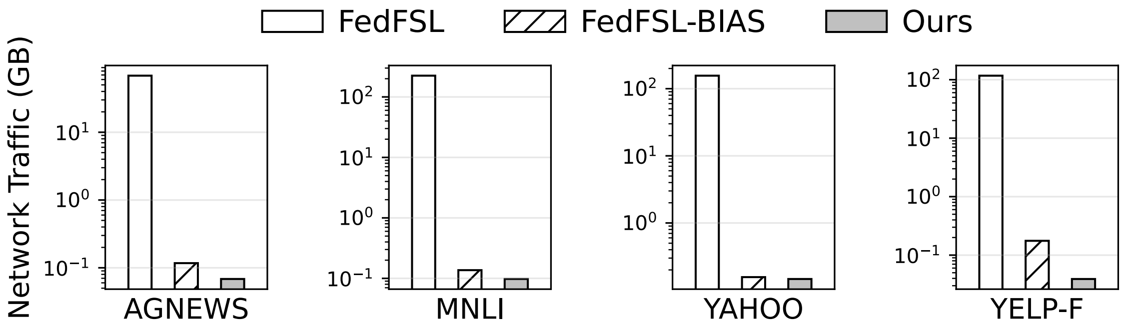


Fig. 1: The total network traffic of all clients.

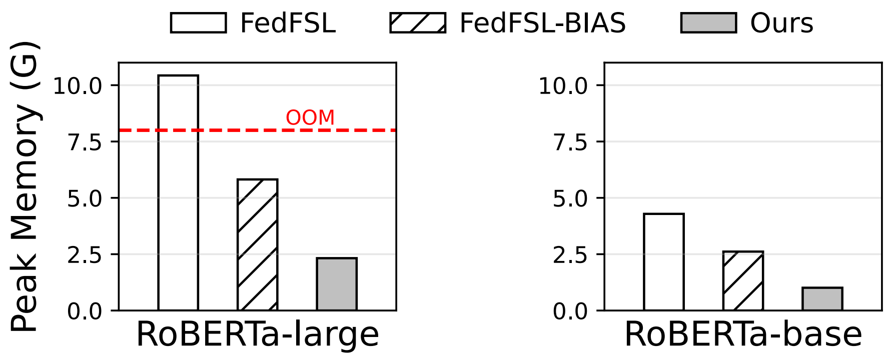


Fig. 3: Memory footprint of on-device training.

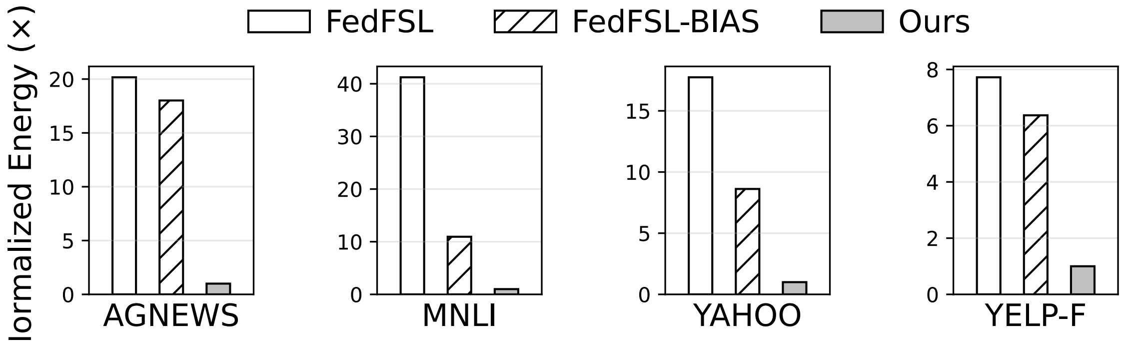


Fig. 2: The total energy consumption of all clients, normalized to that of ours

Dongqi Cai, Yaozong Wu, Shangguang Wang, Felix Xiaozhu Lin, Mengwei Xu

Contact: *cdq@bupt.edu.cn*

Conclusion

- Our system is a FedFSL framework that enables practical few-shot NLP fine-tuning on federated mobile devices.

Code: https://github.com/UbiquitousLearning/FeS

Conclusion

- Our system is a FedFSL framework that enables practical few-shot NLP fine-tuning on federated mobile devices.
- It incorporates pseudo labeling and prompt learning to achieve usable accuracy with only tens of data labels.

Code: https://github.com/UbiquitousLearning/FeS

Dongqi Cai, Yaozong Wu, Shangguang Wang, Felix Xiaozhu Lin, Mengwei Xu

Contact: *cdq@bupt.edu.cn*

Conclusion

- Our system is a FedFSL framework that enables practical few-shot NLP fine-tuning on federated mobile devices.
- It incorporates pseudo labeling and prompt learning to achieve usable accuracy with only tens of data labels.
- At system aspect, it proposes three novel techniques, i.e., early filtering unlabeled data, reducing the tuning depth/capacity, and curriculum orchestrate them to address the unique challenge of huge resource cost raised by its algorithmic.

Code: https://github.com/UbiquitousLearning/FeS

Dongqi Cai, Yaozong Wu, Shangguang Wang, Felix Xiaozhu Lin, Mengwei Xu

Contact: cdq@bupt.edu.cn

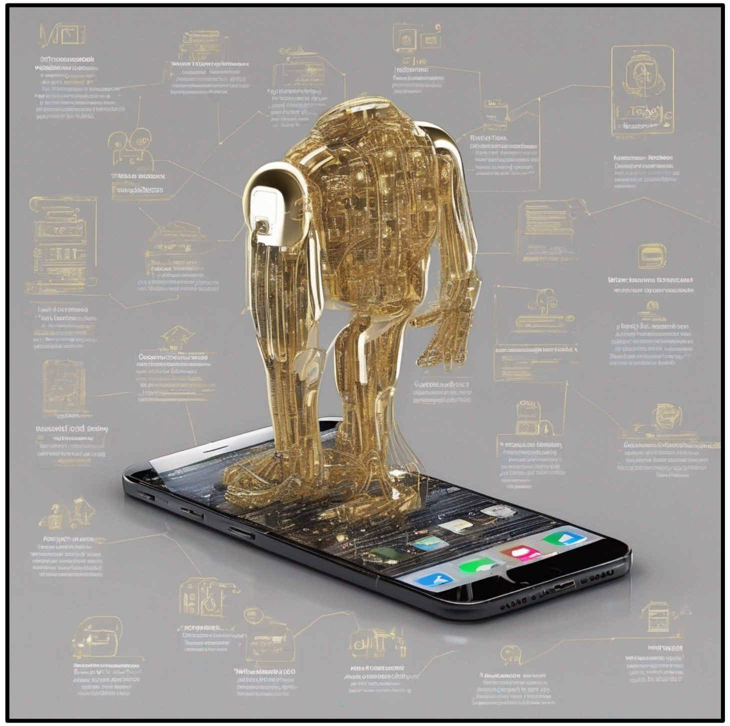
Conclusion

- Our system is a FedFSL framework that enables practical few-shot NLP fine-tuning on federated mobile devices.
- It incorporates pseudo labeling and prompt learning to achieve usable accuracy with only tens of data labels.
- At system aspect, it proposes three novel techniques, i.e., early filtering unlabeled data, reducing the tuning depth/capacity, and curriculum orchestrate them to address the unique challenge of huge resource cost raised by its algorithmic.
- Compared to vanilla FedFSL, Our system reduces the training delay, client energy, and network traffic by up to 46.0×, 41.2× and 3000.0×, respectively.

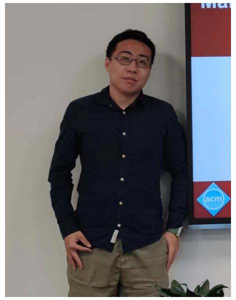
Code: https://github.com/UbiquitousLearning/FeS

Concluding Remarks by Mengwei

- The recent AI wave (large, foundational, multimodal models) is going to make another Golden Era for mobile computing.
  - Think of Smartphones/IoTs as humans-level assistants
- Two key research directions
  - Making LLMs run fast and learn rapidly on devices (hw-sw-algo. codesign)
  - Building killer apps atop LLMs (agents, searching, AIGC, etc)
- Open to collaboration and debate!
  - **Who are we**: a junior faculty plus a group of passionate graduate students who believe in LLM as a game changer to mobile research



Generated by Stable Diffusion XL







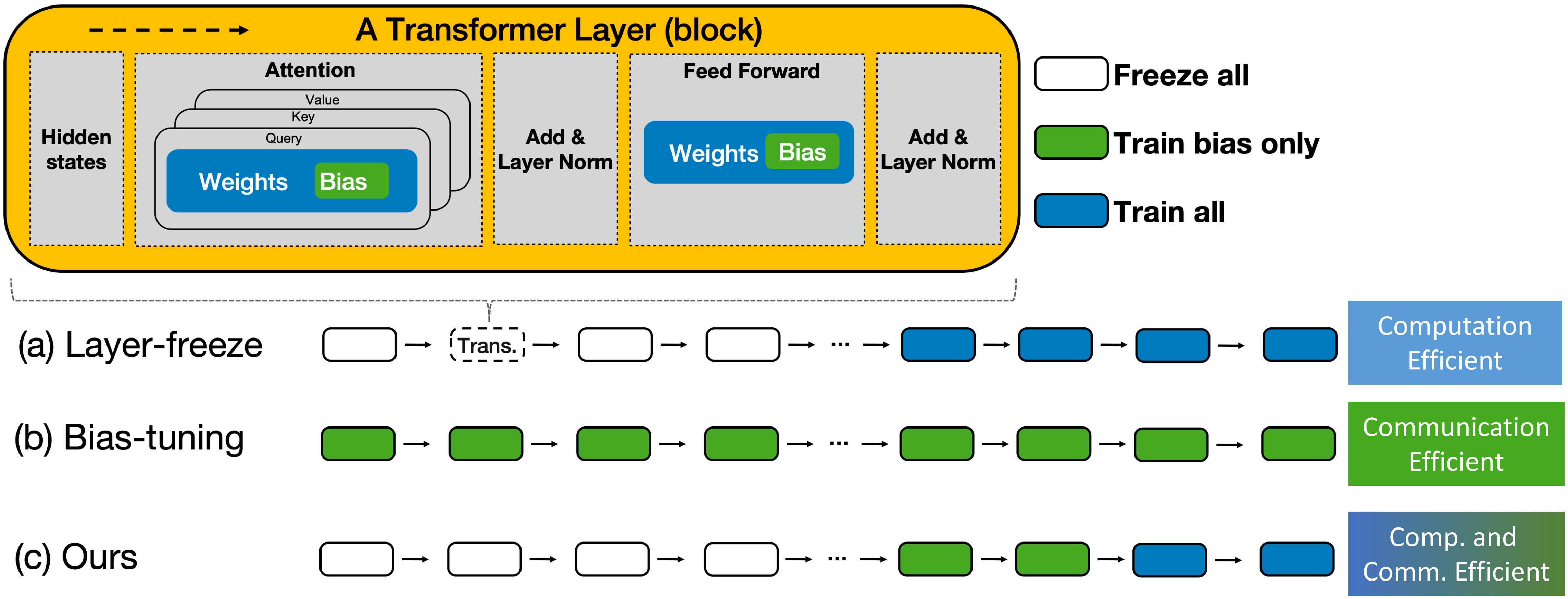
Appendix for Q&A

Different parameter-efficient methods

- Adapter is not only for ”adapters”.
- Parameter-efficient methods are unified (He, ICLR’22).
- Bias-tuning provides the best accuracy-efficiency tradeoff under few-shot learning scenarios (Logan, ACL’22).

He, Junxian, et al. "Towards a Unified View of Parameter-Efficient Transfer Learning.", ICLR 2022.
Logan R L, et al. “Cutting Down on Prompts and Parameters: Simple Few-Shot Learning with Language Models”, ACL 2022.

Design 2: Training Depth/Capacity Co-planning

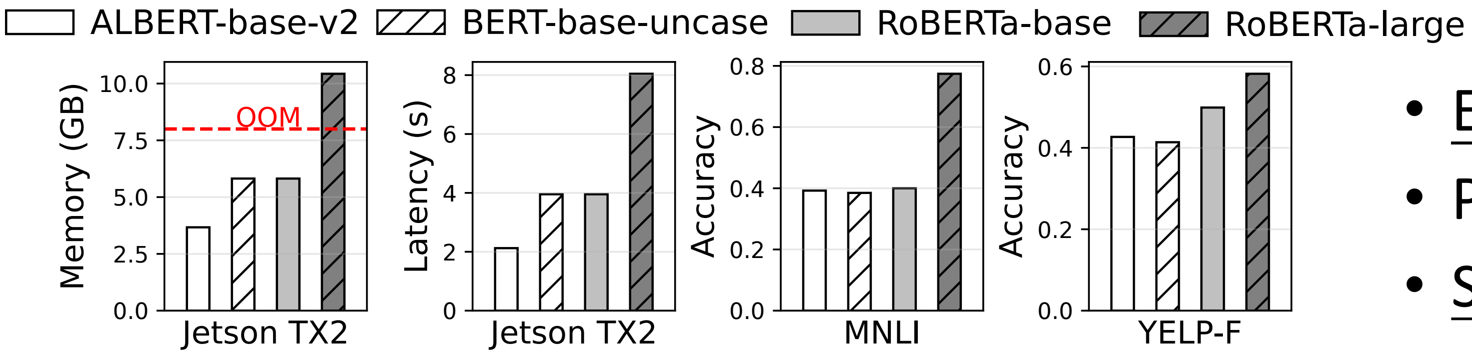


Preliminary: FedFSL performance and cost

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	**77.4**±1.2
YAHOO (skewed)	78.0	24.4±10.3	41.8±4.3	31.0±2.0	**66.9**±1.1
YELP-F (skewed)	70.0	38.3±8.8	51.2±1.8	45.7±4.4	**58.2**±2.4
YELP-F (uniform)	70.0	54.0±0.1	58.1±1.5	57.0±2.2	**61.9**±0.7

Satisfactory accuracy

Both pseudo labeling and prompt learning are indispensable.



Huge system cost

- Excessive on-device **inference**.
- Prompt **learning** needs large NLP model.
- Sophisticated **orchestration** workflow.

Paths towards practical federated learning

	AdaFL: Efficient FL	FeS: Few-shot FL
Solution	Activation cache Configurator	Curriculum pacing Tuning co-planner Representative filter
Key block	Adapters	Prompt learning Pseudo learning
Challenge	Network transmission	Data labels
	2021.12 – 2022.08	2022.08 – 2023.03