Research Track | Systems and Infrastructure for Web, Mobile, and WoT

Jiayun Zhang, Shuheng Li, Haiyu Huang, Zihan Wang, Xiaohan Fu, Dezhi Hong, Rajesh K. Gupta, Jingbo Shang

University of California, San Diego

How Few Davids Improve One Goliath: Federated Learning in Resource-Skewed Edge Computing Environments

Motivation

- **System Heterogeneity:** The deployment of federated learning requires orchestrating clients with widely varied compute resources.
- **Resource Skew:** We often see *skewed* computing environments where a few strong devices hold substantial data resources, alongside many weak devices.
- **Challenge:** Existing approaches identify shared patterns (i.e., neurons or layers) in local models and aggregate the common parts. The unshared portion of the large model rarely receives updates or gains benefits from weak collaborators.

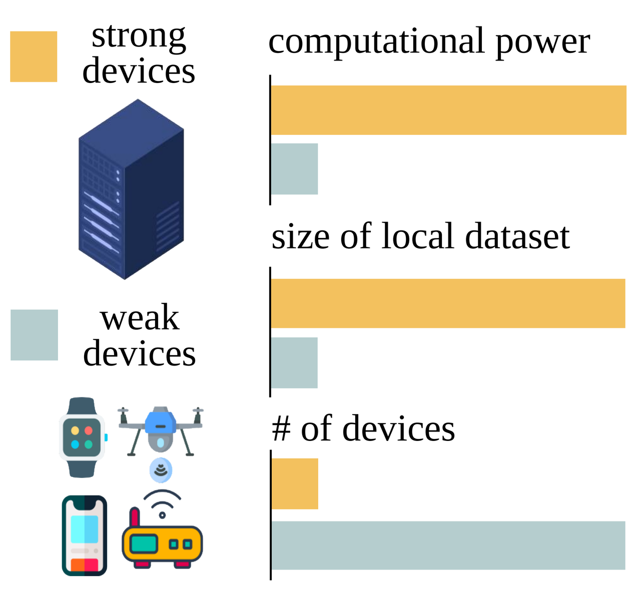
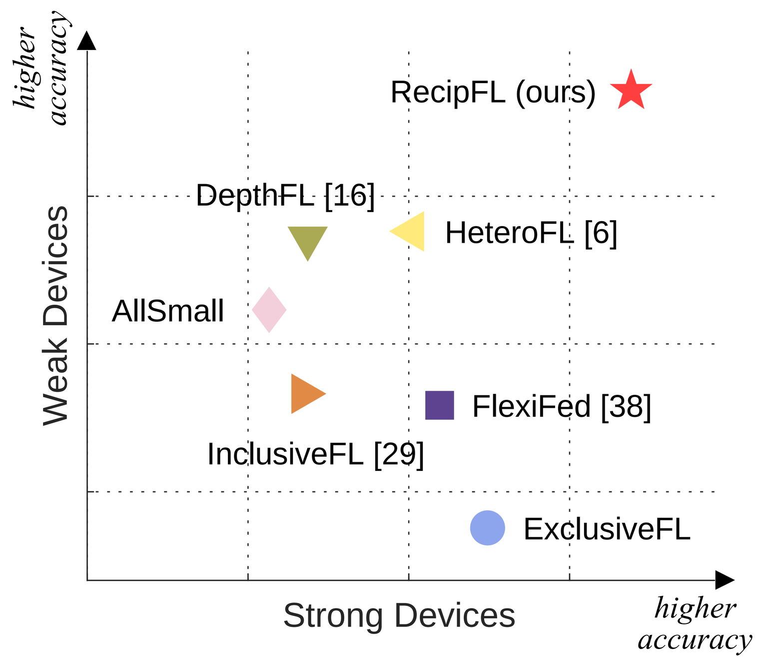


Illustration of Resource Skew



Performance Summary

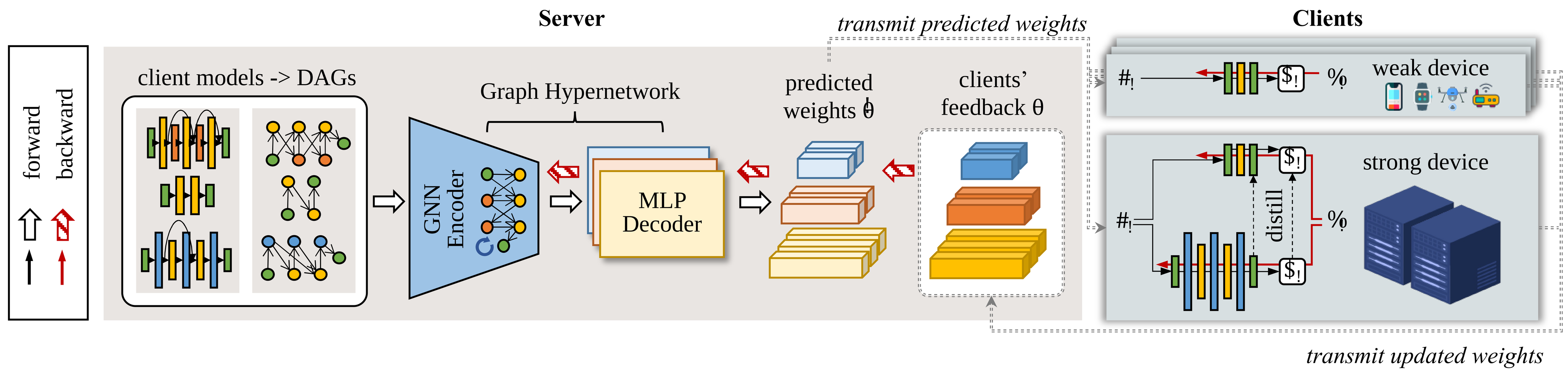
Goal: We aim to facilitate reciprocal benefits between strong and weak devices in resource-skewed federated learning environments, incentivizing both devices to actively engage.

Key Contributions

- Address a new research question in federated learning: *Can strong devices benefit from weak devices in resource-skewed environments?*
- Propose a novel framework to effectively generate weights for heterogeneous client models based on graph hypernetwork, compatible with arbitrary model scaling strategies.
- Establish the generalization bound of RecipFL through theoretical analysis and validate its performance through extensive experiments.

Code available on GitHub: https://github.com/jiayunz/RecipFL

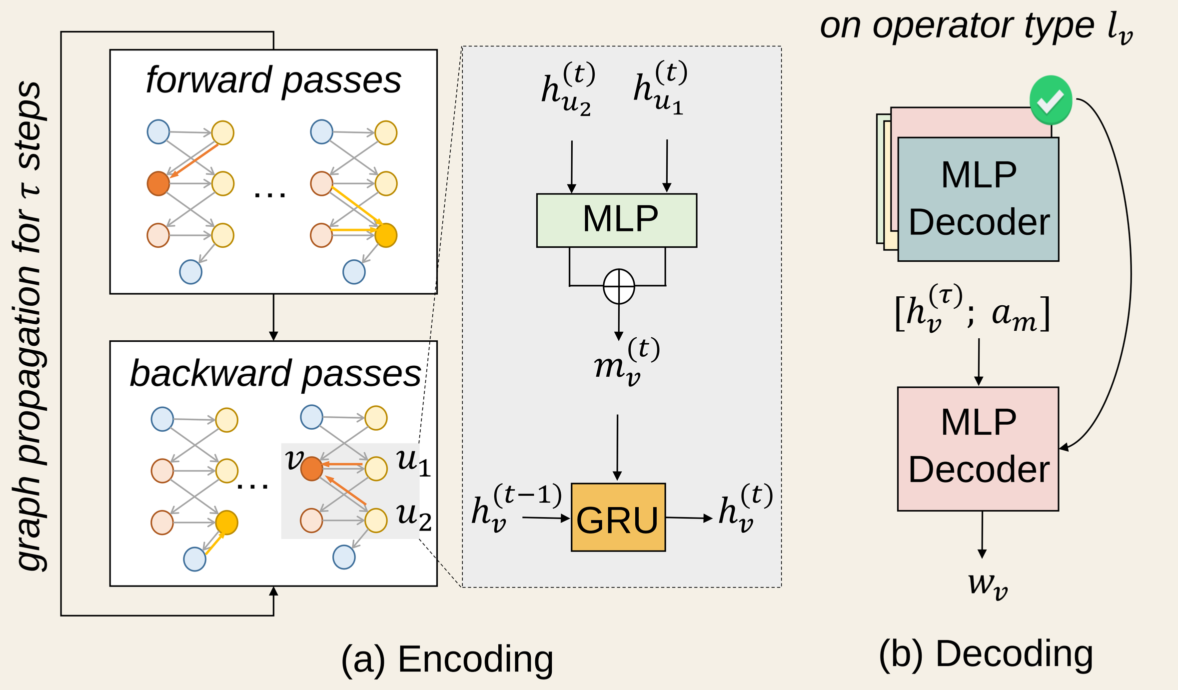
Methodology



RecipFL Overview. The server transforms client models into directed acyclic graphs (DAGs) to represent the computation flow among operations and trains a central graph hypernetwork to generate weights for customized client models.

Weight Generation with Graph Hypernetwork

- Graph hypernetwork: A neural network that generates model weights for another network.
- A client model is represented as a directed acyclic graph $\mathcal{G}(\mathcal{V},\mathcal{E})$. Nodes $\mathcal{V}$ are operators (e.g., linear, conv, pool) and edges $\mathcal{E}$ describe computation flow.



(a) Encoding (b) Decoding

Federated Training

- In each round, the server randomly selects a subset of clients S^t, uses the hypernetwork to generate model weights $\{\tilde{\theta}_m | m \in S^t\}$ and sends them to clients.
- The client trains its local model with the initial weight value $\theta_m = \tilde{\theta}_m$. After local training, it sends the updated client model weights back to the server.
- The server then calculates the client weight change $\Delta\theta_m$ and update the hypernetwork:

$$\phi \leftarrow \phi - \eta_s \sum_{m \in S_t} (\nabla_\phi \theta_m)^T \Delta\theta_m$$

- The learning objective of federated learning is:

$$\operatorname{argmin}_\phi \frac{1}{M} \sum_{m=1}^{M} \mathcal{L}(\mathrm{GHN}(\mathcal{G}_m, a_m; \phi))$$

Strong-to-Weak Device Knowledge Transfer

- Generate and train both small and large models on strong devices
- Knowledge distillation from large to small model

$$\mathcal{L}_m^S(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left[CE\left(f_m^S(x_i;\theta_m^S), y_i\right) + CE\left(f_m^S(x_i;\theta_m^S), f_m^L(x_i;\theta_m^L)\right) + D_{KL}(p_i^L \| p_i^S)\right]$$

Experiment Results

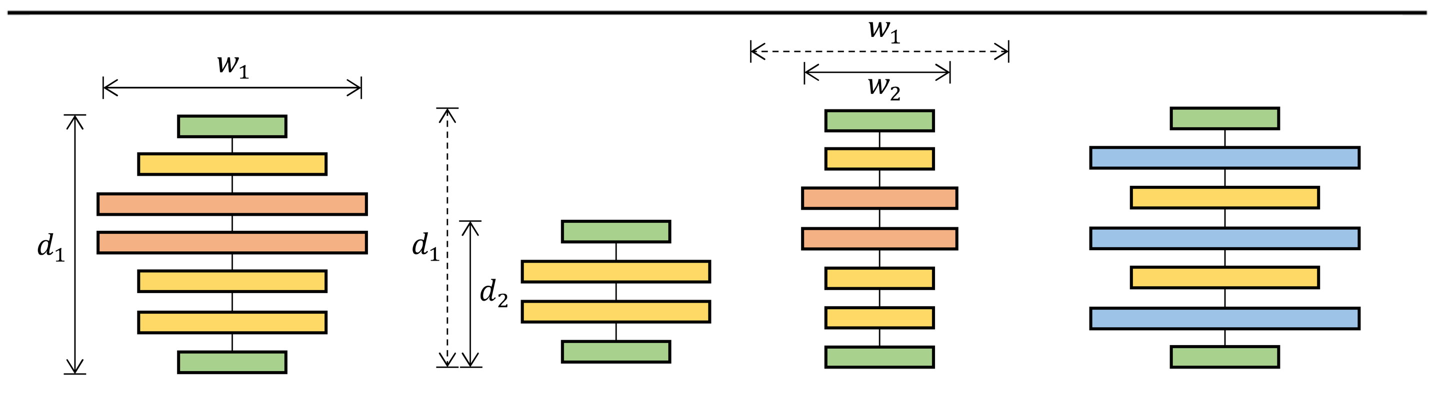
Main Experiment results. RecipFL consistently outperforms the compared methods across all datasets and model scaling strategies, benefiting both strong and weak devices.

Scaling	Method	CIFAR-10		CIFAR-100	
		Strong	Weak	Strong	Weak
Depth	AllSmall	64.34±2.14	68.50±3.42	17.86±2.56	25.56±3.11
	ExclusiveFL	<u>84.85±1.85</u>	59.11±4.22	<u>32.21±3.81</u>	19.22±2.07
	FlexiFed [38]	82.86±1.77	67.66±3.93	28.60±3.48	27.84±2.98
	InclusiveFL [29]	83.22±0.47	67.66±3.14	18.98±3.49	28.71±2.87
	DepthFL [16]	73.90±1.49	<u>78.16±1.48</u>	22.08±3.58	<u>36.83±2.87</u>
	RecipFL	**85.28±0.22**	**78.65±1.35**	**41.63±2.24**	**45.52±3.12**
Width	AllSmall	82.86±1.77	<u>78.90±2.87</u>	29.80±3.32	37.90±2.83
	ExclusiveFL	83.96±1.97	70.65±3.99	<u>32.22±6.66</u>	24.49±3.52
	HeteroFL [6]	<u>84.76±1.19</u>	77.93±2.92	26.51±2.70	<u>39.05±2.82</u>
	RecipFL	**85.06±0.13**	**82.88±1.29**	**43.64±2.84**	**42.00±3.88**

Method	MNIST		MNLI	
	Strong	Weak	Strong	Weak
AllSmall	91.73±3.94	77.05±7.47	73.47±0.52	82.13±2.89
ExclusiveFL	92.97±1.98	77.85±5.14	<u>80.20±0.20</u>	70.52±6.04
FlexiFed [38]	92.70±2.72	73.89±8.08	79.65±0.18	<u>82.31±6.15</u>
InclusiveFL [29]	84.77±3.12	75.72±6.27	79.87±0.30	81.17±4.31
DepthFL [16]	<u>94.33±1.95</u>	<u>78.39±7.52</u>	77.11±0.90	80.92±6.64
HeteroFL [6]	89.53±3.22	75.95±8.01	79.65±0.18	<u>82.31±6.15</u>
RecipFL	**97.07±1.87**	**86.36±6.60**	**82.78±0.57**	**83.37±4.72**

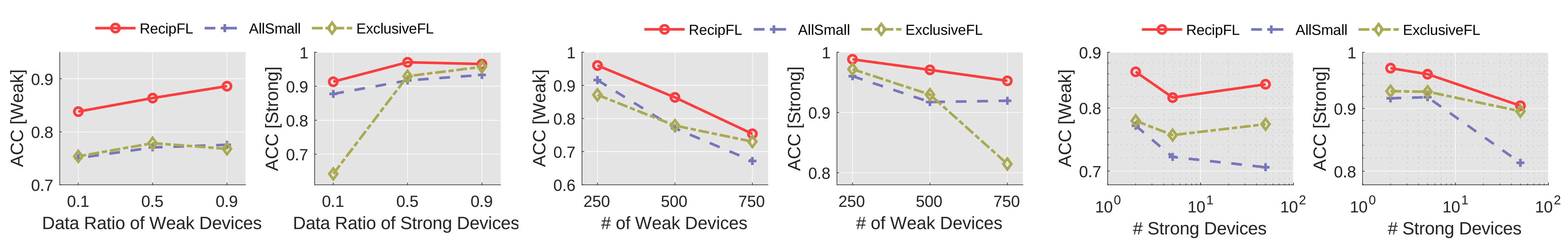
Compatibility with model scaling strategies. RecipFL is compatible with various ways of model scaling, showing more flexibility than existing solutions.

Scaling Strategy	HeteroFL	InclusiveFL	FlexiFed	DepthFL	RecipFL (ours)
depth-wise		✓	✓	✓	✓
width-wise	✓				✓
architecture-wise					✓



(a) original model (b) depth-wise (c) width-wise (d) architecture-wise

Exploratory studies. RecipFL exhibits superior scalability and robustness across a range of resource skew scenarios compared to the baselines, consistently enhancing the performance of both strong and weak devices.



(a) Impact of data ratio (b) Varying number of weak devices (c) Varying number of strong devices

More diverse device capacities. RecipFL is not limited to the setup of one large and one small model architecture and can work with diverse device capacities.

Method	Small (LeNet-5)	Medium (ResNet-101)	Large (VGG-16)
AllSmall	69.22±2.16	66.58±1.84	59.06±1.41
ExclusveFL	46.00±3.71	72.38±3.80	79.86±2.52
RecipFL	**70.12±2.33**	**86.37±1.72**	**81.23±0.41**