ICML
International Conference
On Machine Learning
2025

Federated Learning for Feature Generalization with Convex Constraints

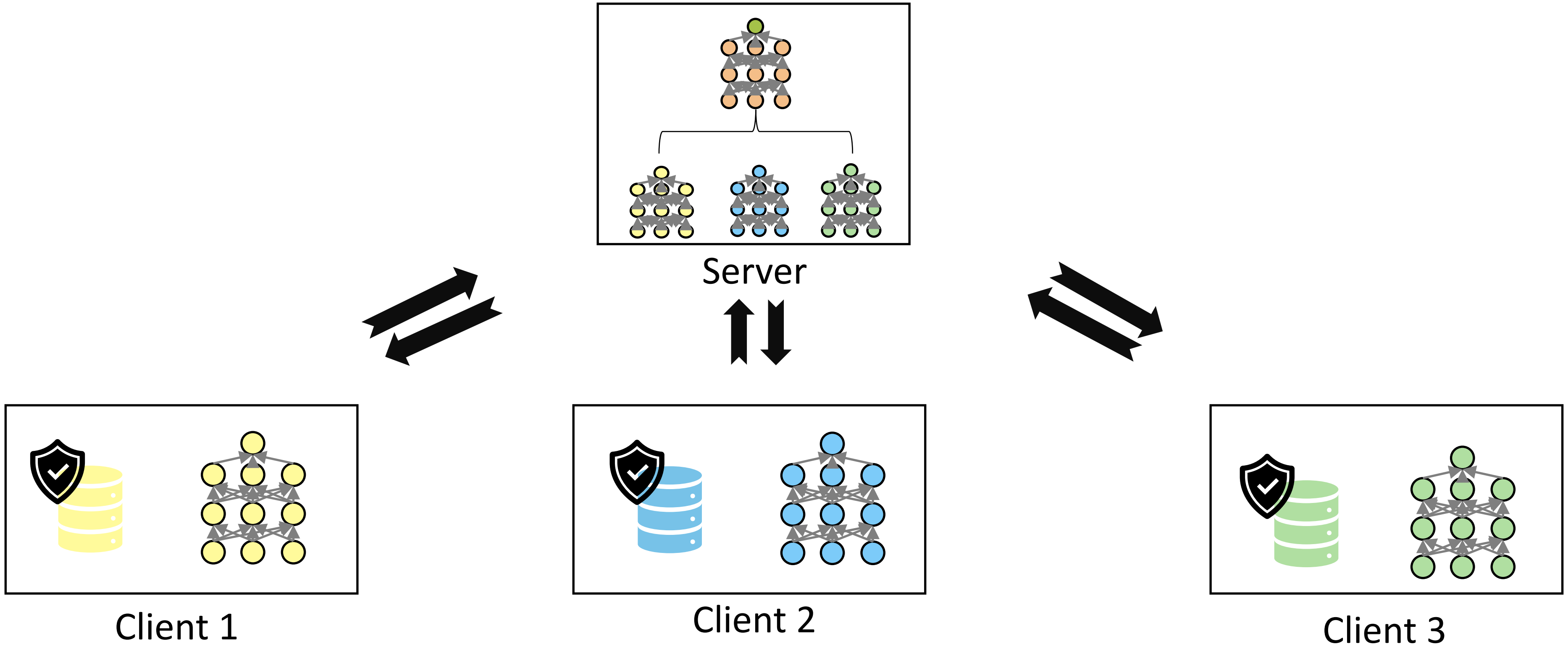
Dongwon Kim
Sungkyunkwan University

Donghee Kim
Sungkyunkwan University

Sung Kuk Shyn
Korea Advanced Institute of Science and Technology

Kwangsu Kim
Sungkyunkwan University

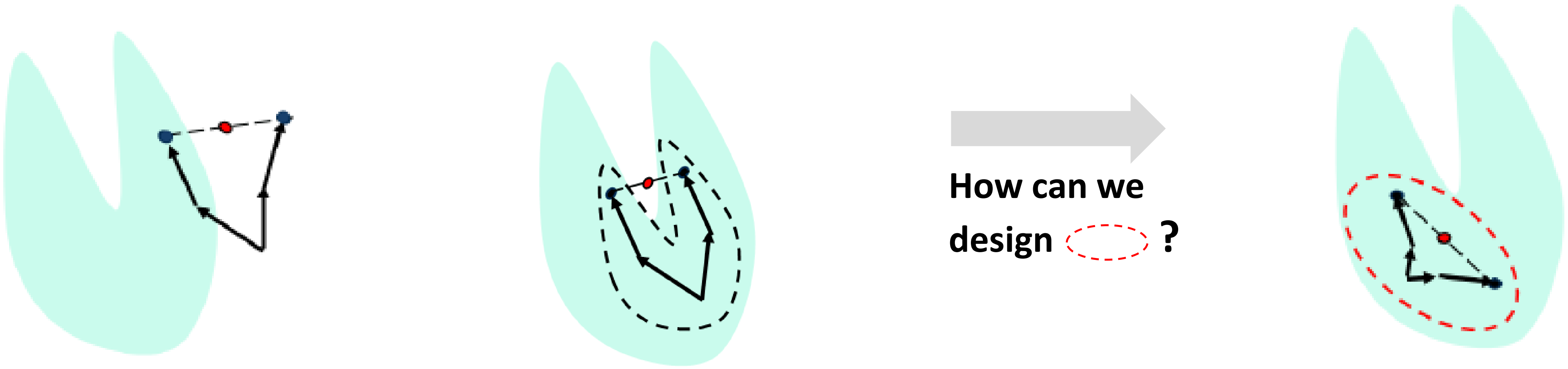
Motivation : Federated Learning



- Federated Learning is a decentralized training paradigm that preserves data privacy by keeping local data on-device.
- The main challenge lies in avoiding overfitting to non-i.i.d. client data while maintaining generalization to the overall data distribution.

Motivation : Previous Research

(1) Without Constraint (2) Non-convex Constraint



➢ What kind of constraint can promote generalization during local training, while preserving it after aggregation?

Generalization Area

Condition 1. The constraints boost weak features and preserve already strong features for generalization

Consistent Aggregation

Condition 2. The constraints should be conserved after aggregation to preserve generalization ability

Local Training Objective

$$\min_{W} \mathcal{L}_m(W)$$

$$\text{s.t. } {G_c^l}^\top (w_c^l - G_c^l) = 0, \quad \mathbf{1}^\top w_c^l = 0, \quad \forall c, l$$

Method : Method Component

Algorithm 1 Training procedure of FedCONST

Input: Batch size B, communication rounds K, number of clients M, local steps T, dataset $D = \bigcup_{m\in[M]} D_m$

Output: Global model parameters w^K

Server executes:

Initialize w^0 with He Initialization

for $k = 0, \ldots, K-1$ **do**

  for $m = 1, \ldots, M$ ***in parallel*** **do**

    Send w^k to client m

    $w_m^{k+1} \leftarrow$ **FedCONST: Client executes**(m, w^k)

  end

  $w^{k+1} \leftarrow \sum_{m\in[M]} \frac{|D_m|}{|D|} w_m^{k+1}$

end

return w^K

FedCONST: Client executes(m, w^k):

Assign global model to the local model $w_m^k \leftarrow w^k$

for *each local epoch* $t = 1, \ldots, T$ **do**

  for *batch* $(x_{m,1:B}, y_{m,1:B}) \in D_m$ **do**

    Per layer l and channel/feature c,

      Center gradient: $g_{m,t}^k \leftarrow \mathrm{C}(g_{m,t}^k)$ ①

      Project gradient: $g_{m,t}^k \leftarrow P_{w^k}(g_{m,t}^k)$ ②

      Apply update: $w_m^k \leftarrow w_m^k - \eta g_{m,t}^k$

  end

end

return w_m^{k+1} *to server*

Training Objective on Client

$$\min_W \mathcal{L}_m(W)$$

$$\text{s.t. } {G_c^l}^\top (w_c^l - G_c^l) = 0, \quad \mathbf{1}^\top w_c^l = 0, \quad \forall c, l$$

Convex Constraints

(1) Center Constraint

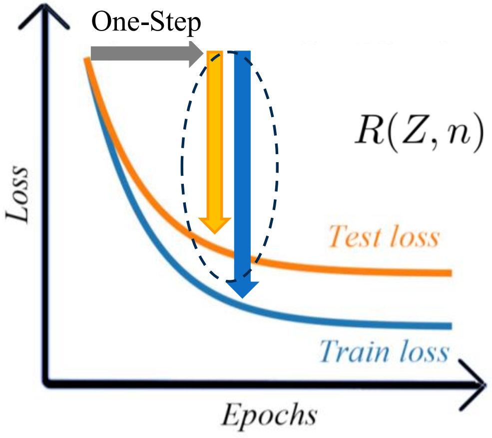
$$C(w) = w - \frac{1}{n}1^\top w$$

(2) Orthogonal Constraint

$$P_{w^k}(w) = (I - pp^\top)w$$

Condition 1 : Generalization Area

One Step Generalization Ratio (OSGR)



$$R(Z,n) = \frac{\mathbb{E}_{D,D' \sim \mathcal{Z}^n} \Delta L_{D'}}{\mathbb{E}_{D \sim \mathcal{Z}^n} \Delta L_D}$$

$$R(Z,n) = 1 - \frac{1}{n} \sum_j \frac{\mathbb{E}_{D \sim \mathcal{Z}^n}[g_j^2]}{\sum_{j'} \mathbb{E}_{D \sim \mathcal{Z}^n}[g_{j'}^2]} \cdot \frac{1}{r_j + \frac{1}{n}}$$

Generalization Ability

$$|w_j| \propto \frac{g_j^2}{\rho_j^2} = r_j$$

Theorem 1. *If we impose center constraint and orthogonal constraint, and if $W^l_{c,i} \leq W^l_{c,j}$, then*

$$\Pr(|\Delta W^l_{c,i}| \geq |\Delta W^l_{c,j}|) \geq \Pr(|\Delta W^l_{c,i}| \leq |\Delta W^l_{c,j}|).$$

Preserving Generalization

Z: Data distribution
D': Test Data
D: Training Data
g: Gradient
j: j-th parameter
r: Gradient Signal to Noise Ratio
n : #Sample

[1] Understanding Why Neural Networks Generalize Well Through GSNR of Parameters (ICLR2020)

Condition 2: Consistent Aggregation

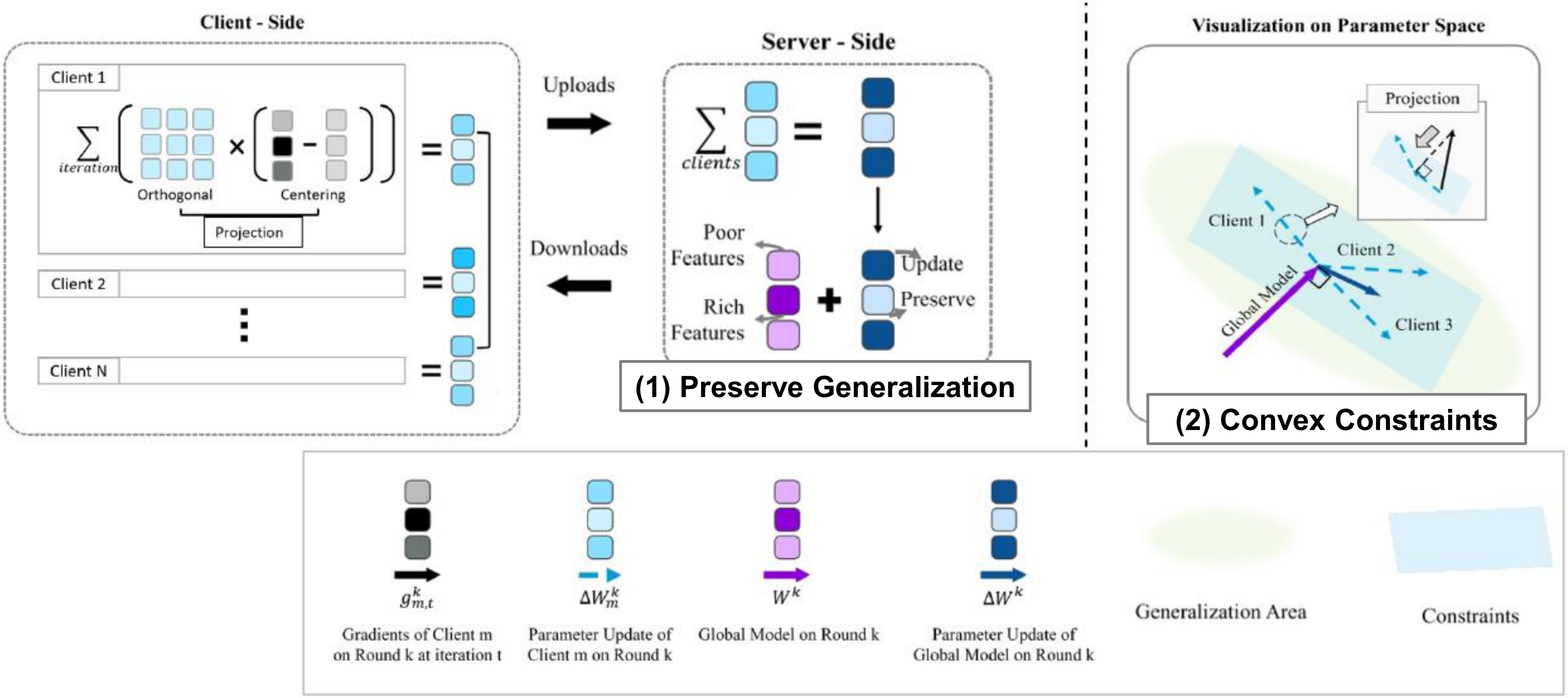
Generalization Area

Constraint

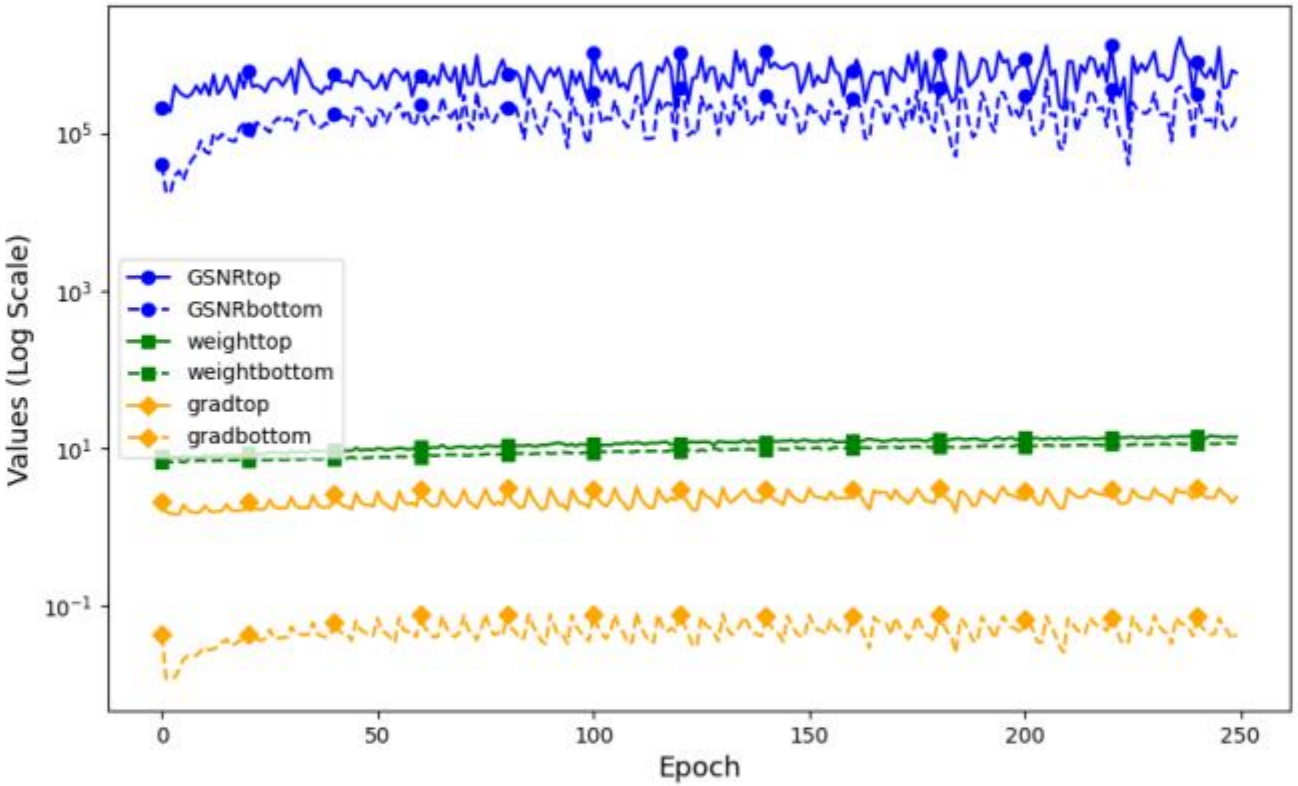
	Local Training	Aggregation
Center Constraint:	$1^\top g_{m,t}^k = 0$	$\frac{1}{M}\sum_{m\in M} 1^\top \Delta w_m^k = 1^\top \Delta w^k = 0$
Orthogonal Constraint :	$(w^k)^\top g_{m,\,t}^k = 0$	$\frac{1}{M}\sum_{m\in M} (w^k)^\top \Delta w_m^k = (w^k)^\top \Delta w^k = 0$

Consistent Aggregation

FedCONST Framework



Experiments : Weight, GSNR, Gradient

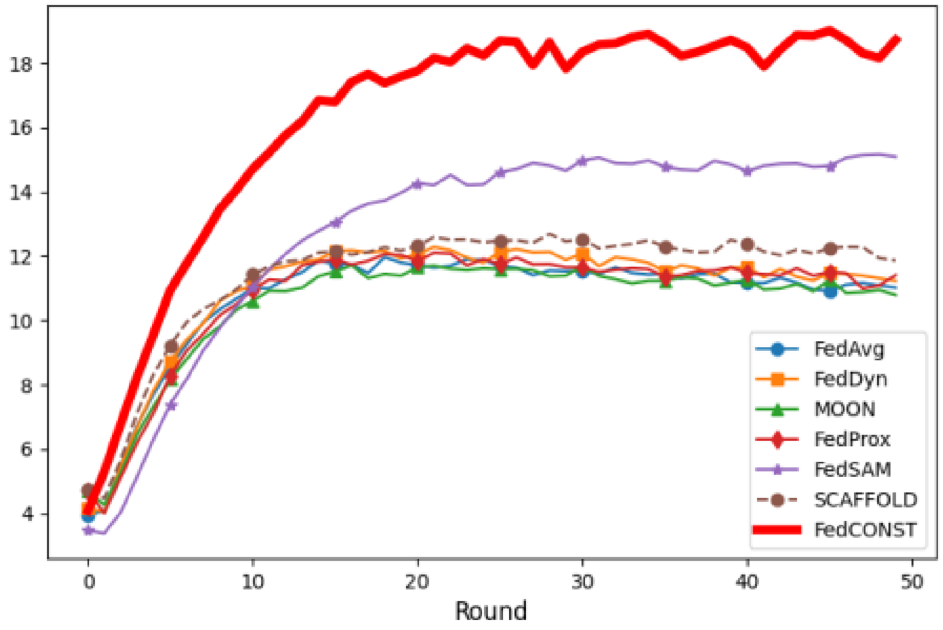


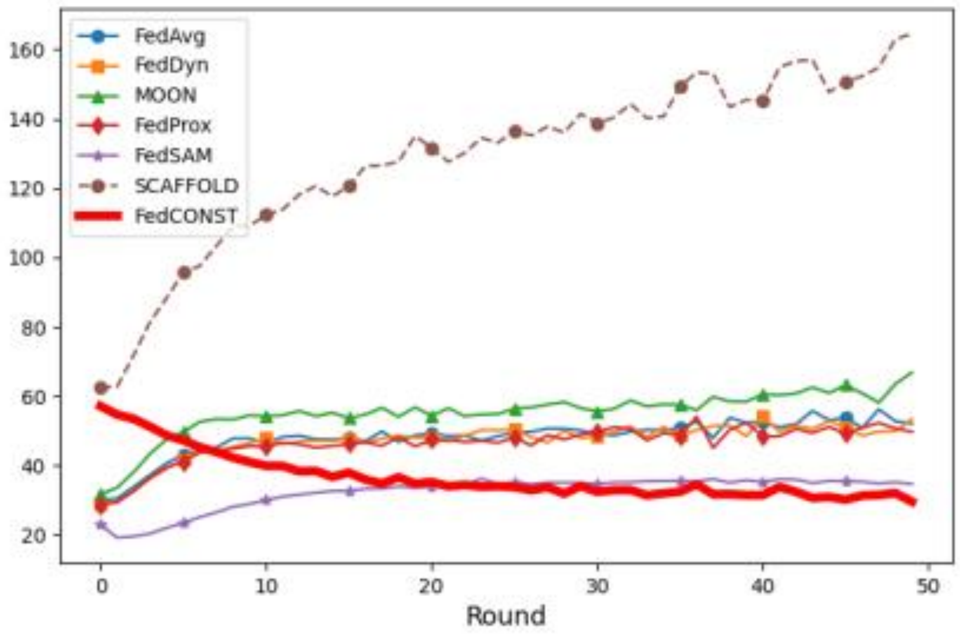
- Weight magnitude suggest feature strength, showing positive correlation with GSNR in FL framework.
- Parameters of global model can be directly utilized for preserving generalization ability of the model.

Experiments : Gradient Update Analysis

Drift Diversity: $\frac{\sum_{m \in M} \frac{|D_m|}{|D|} \left\| \boldsymbol{\Delta w}_m^k \right\|_2^2}{\left\| \boldsymbol{\Delta w}^{k+1} \right\|_2^2}$

Consistency: $\sum_{m \in M} \frac{|D_m|}{|D|} \; |\boldsymbol{\Delta} w_m^k \;|_2^2$





- Lower consistency value indicate client updates remain aligned.
- Higher drift diversity indicate each client fully reflect its own information.

Experiments : Loss Landscape

Algorithm	w/o Constraints		Constraints	
	C_{convex}	H_{trace}	C_{convex}	H_{trace}
FedAvg	2.466	-4951	31.7	10102
FedProx	2.739	-3873	23.09	12294
MOON	3.015	-3416	16.09	9640
SCAFFOLD	2.914	-3245	21.81	9145
FedDyn	2.16	-4590	12.82	9121
FedCONST	**31.7**	**10102**	-	-

$C_{convex} = |\lambda_{max} / \lambda_{min}|$
H_{trace} : Hessian trace

- Our constraints governs convex loss landscape of the global model.

Experiments: Performance Comparison

Model	Algorithm	Cross-Device CIFAR-10 $\alpha = 0.5$	Cross-Silo CIFAR-10 $\alpha = 0.2$	Cross-Silo CIFAR-10 $\alpha = 0.5$	Cross-Silo CIFAR-100 $\alpha = 0.5$
LeNet-5	FedAvg	46.12	46.42	53.12	17.46
	FedAvg + CONST	**54.28** (+8.16)	54.79 (+8.37)	59.66 (+6.54)	26.86 (+9.40)
	FedProx	45.58	45.27	55.15	18.42
	FedProx + CONST	53.09 (+7.51)	56.18 (+10.91)	60.70 (+5.55)	26.78 (+8.36)
	MOON	43.89	46.66	55.79	18.72
	MOON + CONST	48.66 (+4.77)	52.88 (+6.22)	59.86 (+4.07)	26.76 (+8.04)
	SCAFFOLD	45.66	45.67	52.74	17.66
	SCAFFOLD + CONST	53.82 (+8.16)	**56.62** (+10.95)	**63.03** (+10.29)	26.74 (+9.08)
	FedDyn	44.93	48.05	51.05	16.79
	FedDyn + CONST	54.07 (+9.14)	55.67 (+7.62)	59.76 (+8.71)	**27.14** (+10.35)
ResNet-18	FedAvg	54.07	57.04	64.25	33.51
	FedAvg + CONST	66.51 (+12.44)	68.41 (+11.37)	72.44 (+8.19)	36.82 (+3.31)
	FedProx	56.79	53.92	64.51	34.11
	FedProx + CONST	63.51 (+6.72)	68.07 (+14.15)	71.96 (+7.45)	36.56 (+2.45)
	MOON	57.84	51.51	68.45	35.19
	MOON + CONST	**66.94** (+9.10)	62.52 (+11.01)	71.84 (+3.39)	36.80 (+1.61)
	SCAFFOLD	56.47	59.30	64.50	37.18
	SCAFFOLD + CONST	63.49 (+7.02)	68.63 (+9.33)	**75.09** (+10.59)	38.93 (+1.75)
	FedDyn	52.64	55.09	65.50	35.07
	FedDyn + CONST	64.29 (+11.65)	66.00 (+10.91)	71.76 (+6.26)	37.22 (+2.15)
	FedSAM	62.52	61.35	69.45	38.43
	FedSAM + CONST	63.45 (+0.93)	**68.87** (+7.52)	72.64 (+3.19)	**39.61** (+1.18)

- Consistently boosts existing FL Algorithms.

Conclusion

We present **FedCONST**, a simple yet effective FL algorithm that leverages convex constraints based on weight magnitude to preserve strong features and reinforce weak ones.

- Improves generalization by feature-aware local learning
- Ensures stability, convexity, and consistency
- Compatible with many FL algorithms without extra cost
- Simple and scalable

Bridging local learning and generalization of global model in FL

Thank you