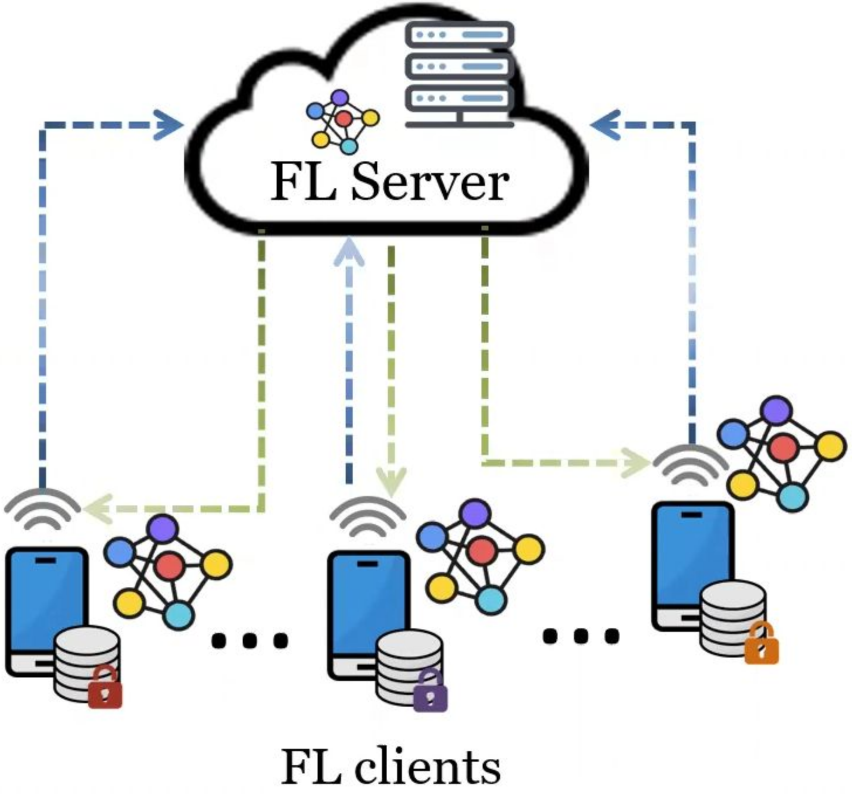
Politecnico
di Torino

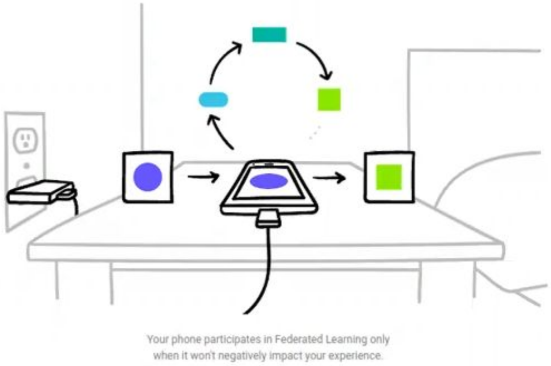
Investigating Statistical Heterogeneity and Privacy Issues In Federated Learning

S277711 GIUSEPPE DESIDERIO
S301905 VIDA AHMADI
S303073 SHYAM NANDAN RAI

MLDL
A.A 2021/2022

Introduction: Federated Learning





- Data never leaves local devices
- Learn on fresh real-world data

Content

Our project investigates two major problems in federated learning:

Content

Our project investigates two major problems in federated learning:

1. **Statistical Heterogeneity**
   a. We benchmarked the performance of the federated learning methods: FedAvg and FedGKT in IID and non-IID setting.
   b. Comparing Group Normalization vs Batch Normalization the former shows better performance in the presence of heterogeneity.
   c. We propose MOON-Prox that addresses the problem of data-heterogeneity by introducing a novel loss function.

2. **Privacy Concerns**
   a. We show that it is possible to reconstruct client data using gradient inversion attack.
   b. Extensive experiments on FedSGD and FedAvg are conducted showing client data reconstruction.

FedAvg

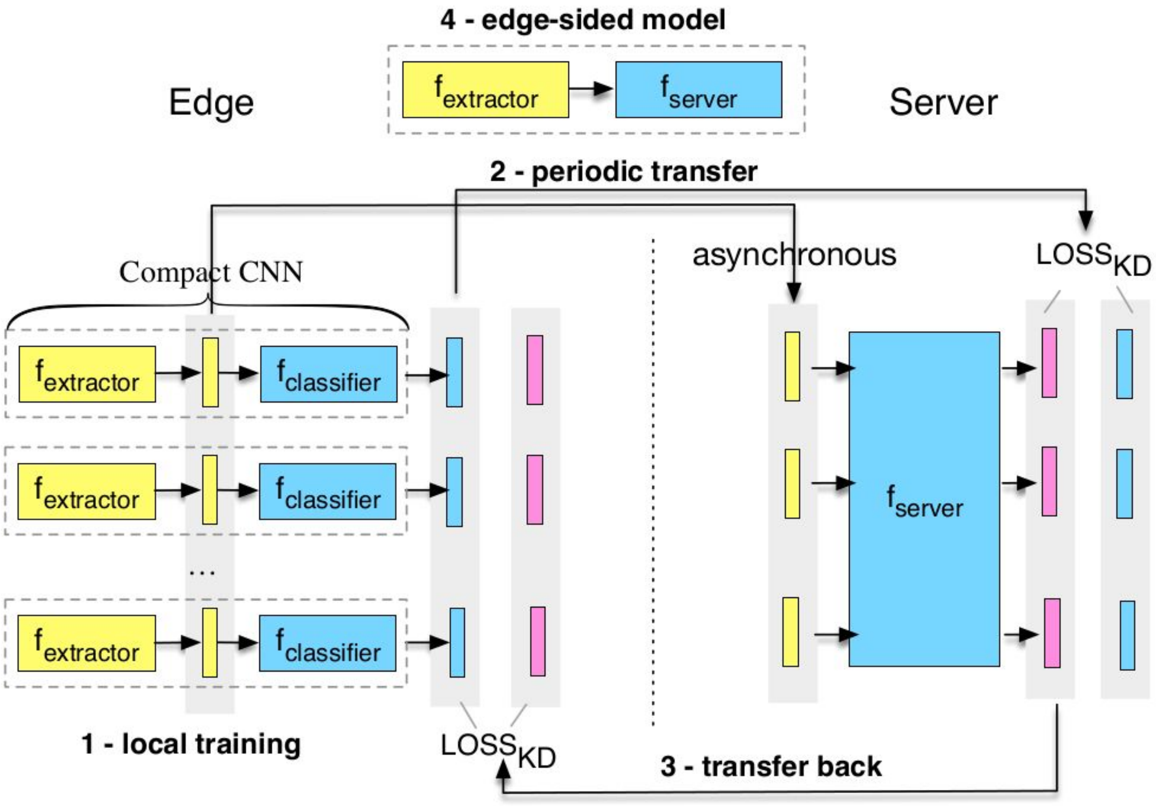
- **Client**
  receives the global model from server.
  train the model for E local epochs.
  sends the updated model to the server.

- **Server**
  it takes a weighted average of the resulting models.

- **the amount of computation is controlled by three key parameters :**
  C, the fraction of clients that perform computation on each round.
  E, the number of training passes each client makes over its local dataset on each round.
  B, the local minibatch size used for the client updates.

FedGKT

- FedGKT tackles the resource-constrained problem on edge devices.

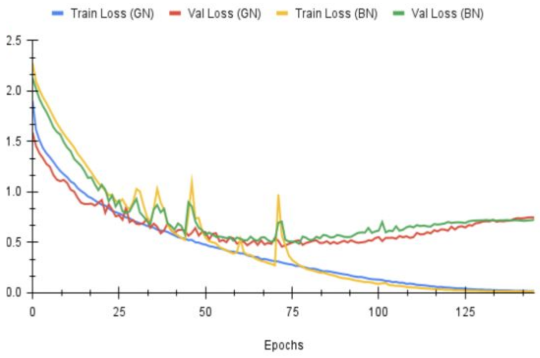


Experimental Setting

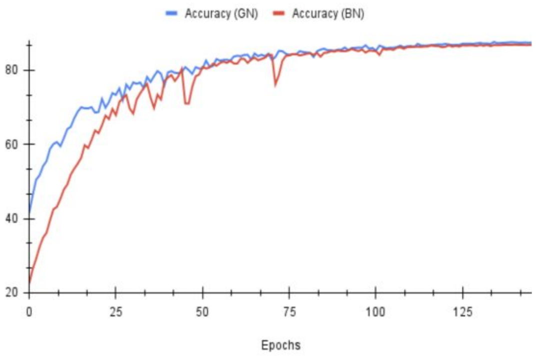
- **Architectures**:
  - FedAvg,Centralized Baseline: ResNet50
  - FedGKT: Resnet-8 (Client), Resnet-49(Server)
- **Normalization Layers:** Group Norm, Batch Norm
- **Dataset**: CIFAR-10
- **Baselines** : FedAvg, FedGKT, Centralized Baseline
- **Non-IID distribution**: Dirichlet [β=0.5].

Results: Centralized Baseline

Loss Plots



Accuracy Plots

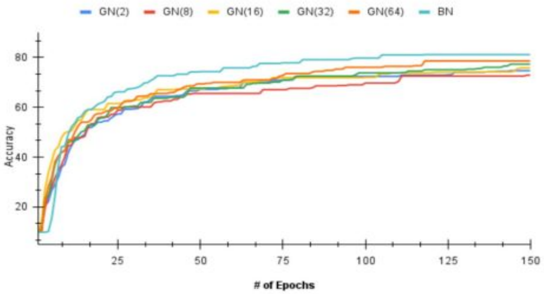


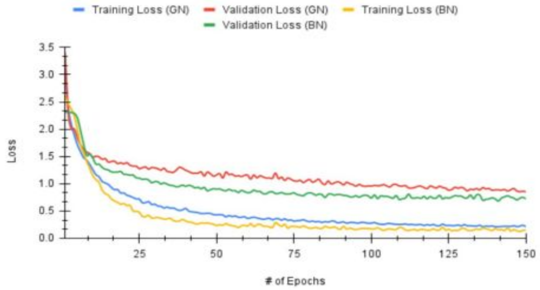
Method	Model	Norm.	Accuracy
Centralized	ResNet-50	BN	86.91
Centralized	ResNet-50	GN	87.48
FedAVG[IID]	ResNet-50	BN	81.14
FedAVG[IID]	ResNet-50	GN	78.60
FedAVG[nonIID]	ResNet-50	BN	60.00
FedAVG[nonIID]	ResNet-50	GN	60.51
FedGKT[IID]	ResNet-49	BN	55.28
FedGKT[IID]	ResNet-49	GN	53.27
FedGKT[nonIID]	ResNet-49	BN	27.01
FedGKT[nonIID]	ResNet-49	GN	19.79

Results: FedAvg

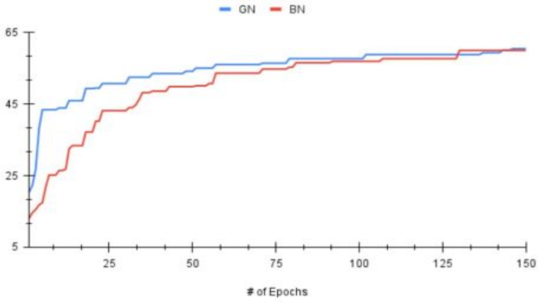
Accuracy Plots | **Loss Plots**

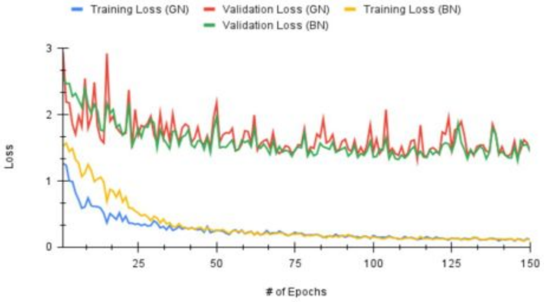
IID





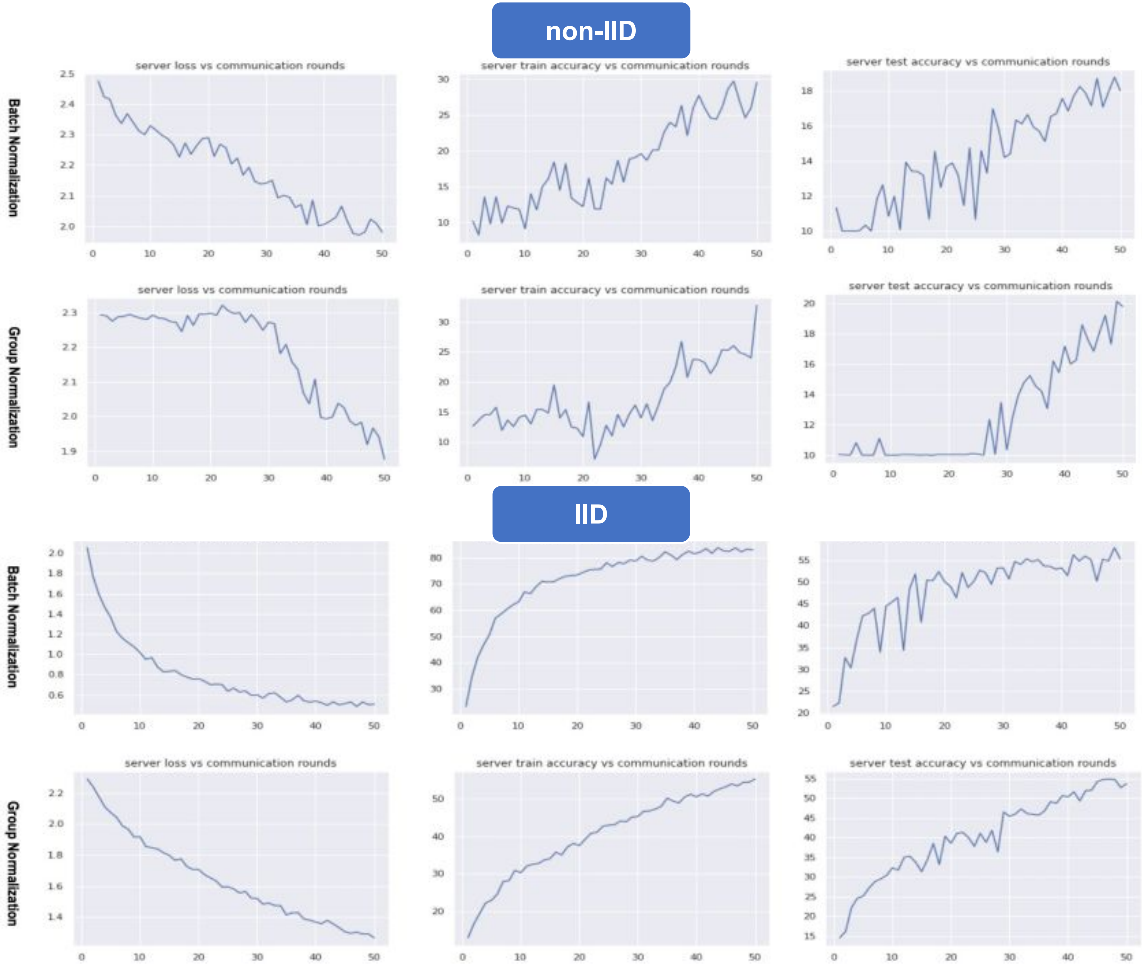
non-IID





Method	Model	Norm.	Accuracy
Centralized	ResNet-50	BN	86.91
Centralized	ResNet-50	GN	87.48
FedAVG[IID]	ResNet-50	BN	81.14
FedAVG[IID]	ResNet-50	GN	78.60
FedAVG[nonIID]	ResNet-50	BN	60.00
FedAVG[nonIID]	ResNet-50	GN	60.51
FedGKT[IID]	ResNet-49	BN	55.28
FedGKT[IID]	ResNet-49	GN	53.27
FedGKT[nonIID]	ResNet-49	BN	27.01
FedGKT[nonIID]	ResNet-49	GN	19.79

Results: FedGKT



Method	Model	Norm.	Accuracy
Centralized	ResNet-50	BN	86.91
Centralized	ResNet-50	GN	87.48
FedAVG[IID]	ResNet-50	BN	81.14
FedAVG[IID]	ResNet-50	GN	78.60
FedAVG[nonIID]	ResNet-50	BN	60.00
FedAVG[nonIID]	ResNet-50	GN	60.51
FedGKT[IID]	ResNet-49	BN	55.28
FedGKT[IID]	ResNet-49	GN	53.27
FedGKT[nonIID]	ResNet-49	BN	27.01
FedGKT[nonIID]	ResNet-49	GN	19.79

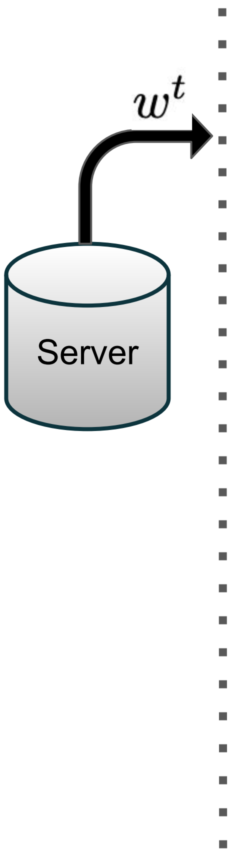
MOON: Motivation

Intuition: Model trained on the whole dataset **[Global Model]** extracts better feature representation than model trained on skewed subset **[Local Model]**.

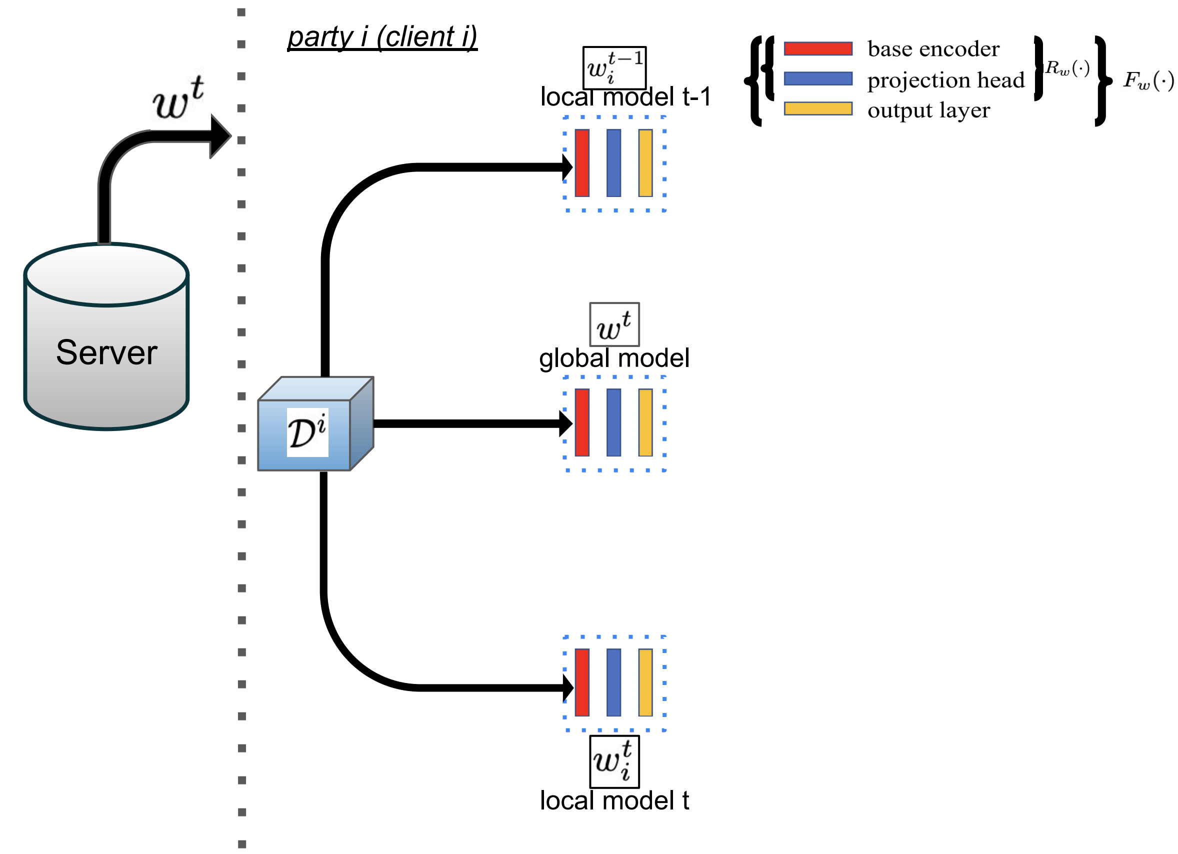
- Bridge the gap between the representation of local model and global model. Minimize the drift caused in the local model updates.
- Contrastive Learning: ↓ global and local model distance & ↑ local model (t) and local model (t-1).

(a) global model (b) local model

MOON

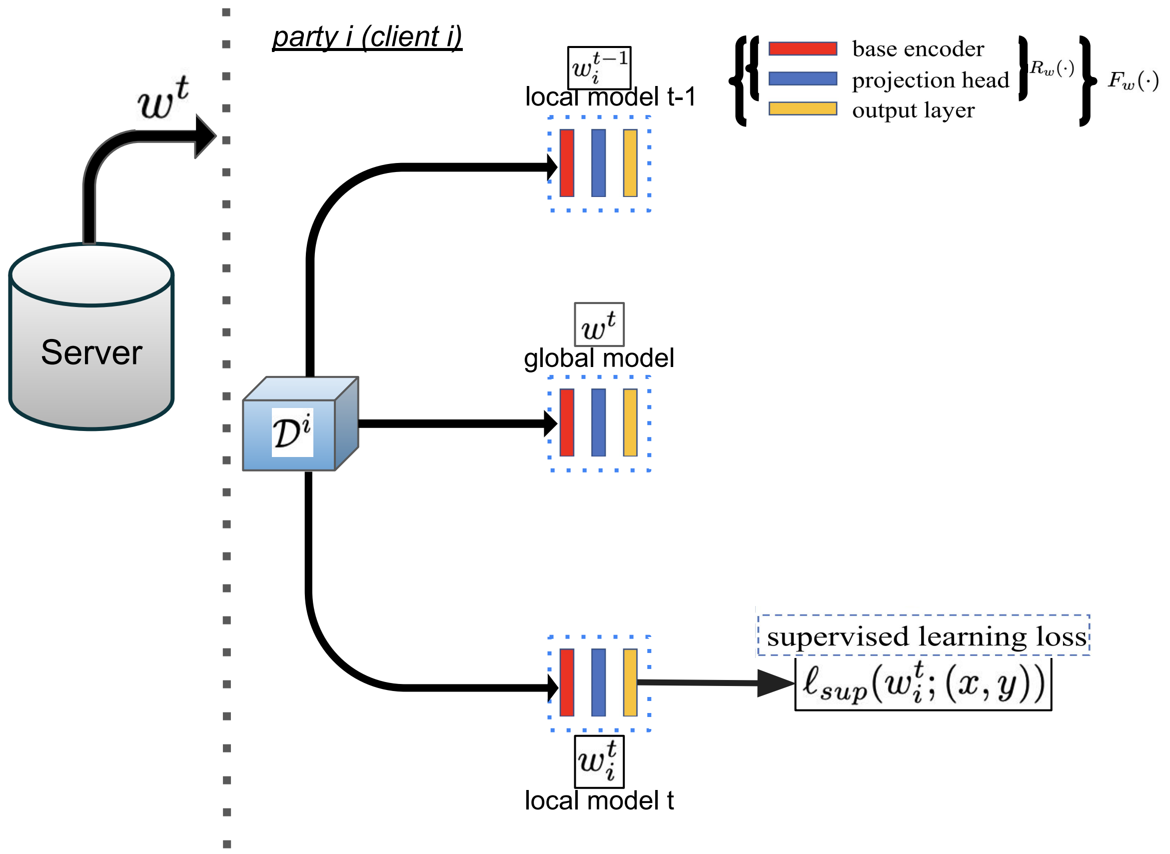


MOON



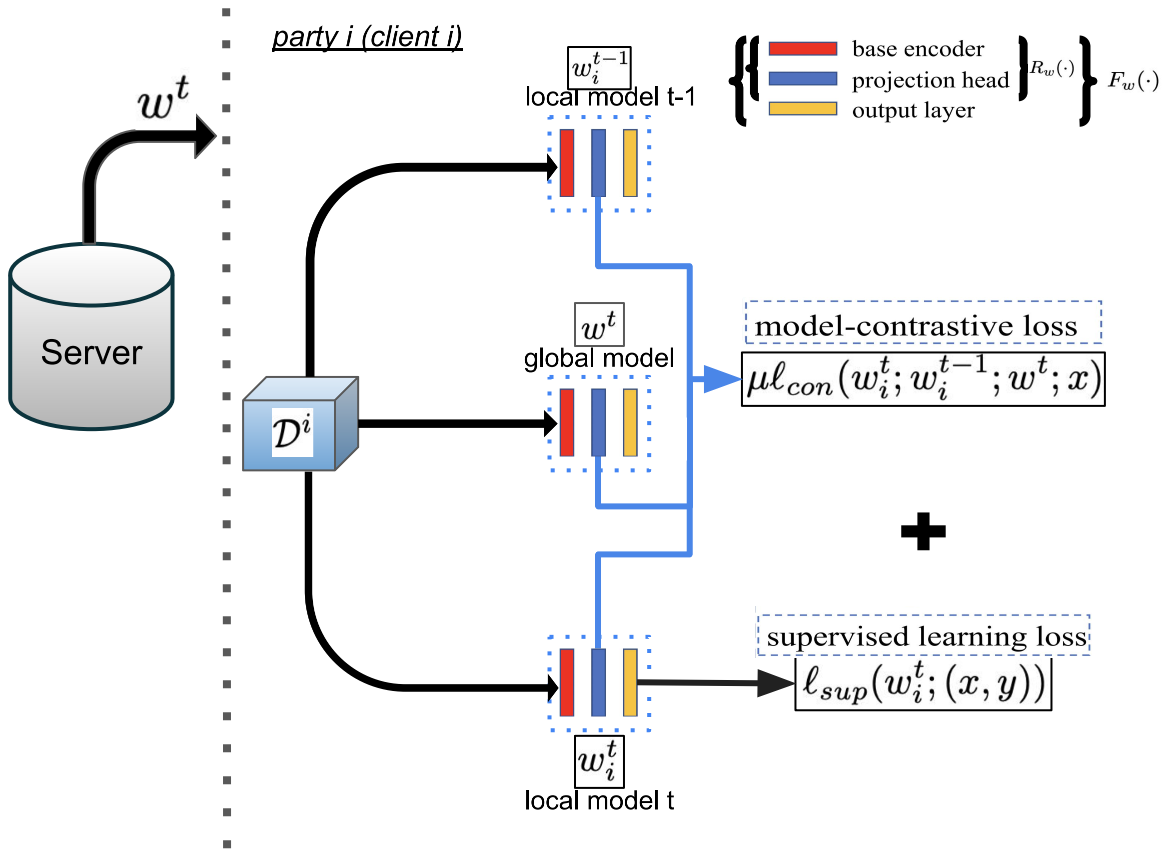
w^t
Server
party i (client i)
w_i^{t-1}
local model t-1
w^t
global model
$\mathcal{D}^i$
w_i^t
local model t
base encoder
projection head
output layer
$R_w(\cdot)$
$F_w(\cdot)$

MOON



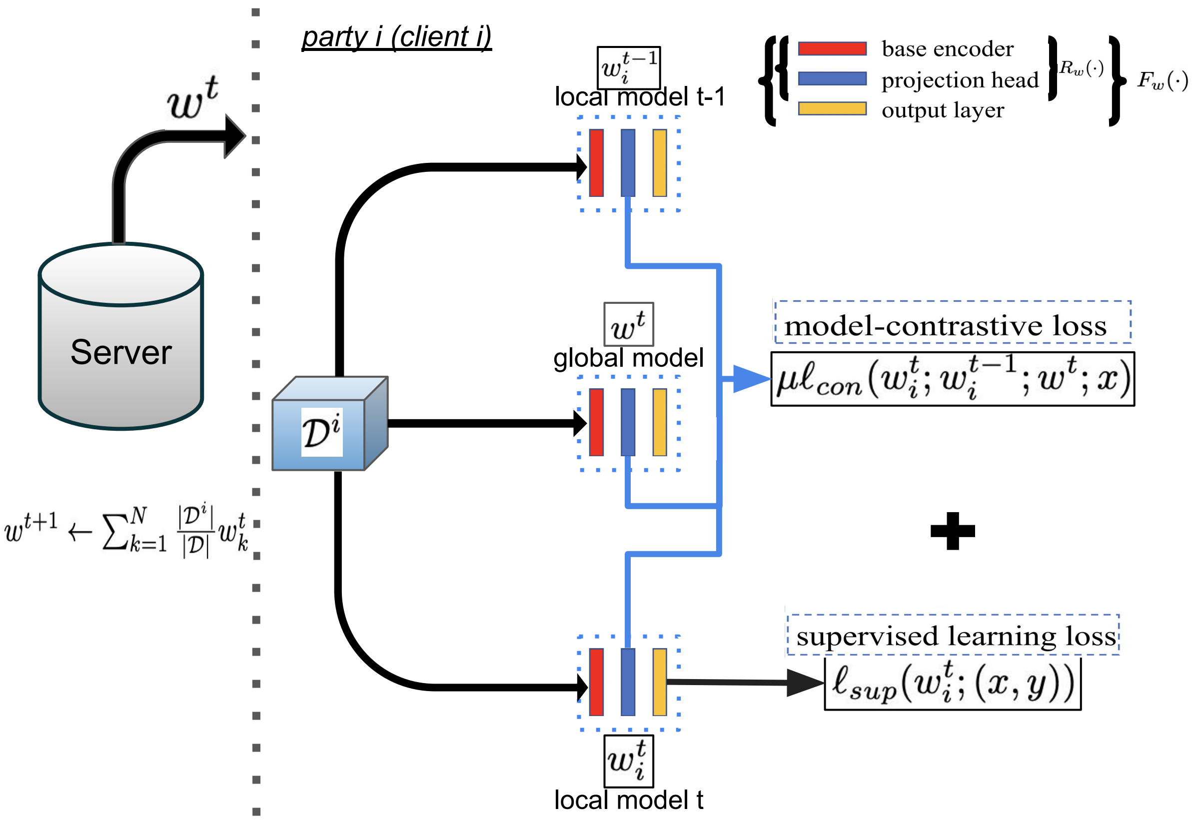
w^t
Server
party i (client i)
w_i^{t-1}
local model t-1
base encoder
projection head
output layer
$R_w(\cdot)$
$F_w(\cdot)$
w^t
global model
$\mathcal{D}^i$
supervised learning loss
$\ell_{sup}(w_i^t;(x,y))$
w_i^t
local model t

MOON



party i (client i)
w^t
Server
$\mathcal{D}^i$
w_i^{t-1}
local model t-1
base encoder
projection head
output layer
$R_w(\cdot)$
$F_w(\cdot)$
w^t
global model
model-contrastive loss
$\mu\ell_{con}(w_i^t; w_i^{t-1}; w^t; x)$
supervised learning loss
$\ell_{sup}(w_i^t; (x, y))$
w_i^t
local model t

MOON



party i (client i)
w^t
Server
$w^{t+1} \leftarrow \sum_{k=1}^{N} \frac{|\mathcal{D}^i|}{|\mathcal{D}|} w_k^t$
$\mathcal{D}^i$
w_i^{t-1}
local model t-1
w^t
global model
w_i^t
local model t
base encoder
projection head
output layer
$R_w(\cdot)$
$F_w(\cdot)$
model-contrastive loss
$\mu \ell_{con}(w_i^t; w_i^{t-1}; w^t; x)$
supervised learning loss
$\ell_{sup}(w_i^t; (x, y))$

MOON-Results

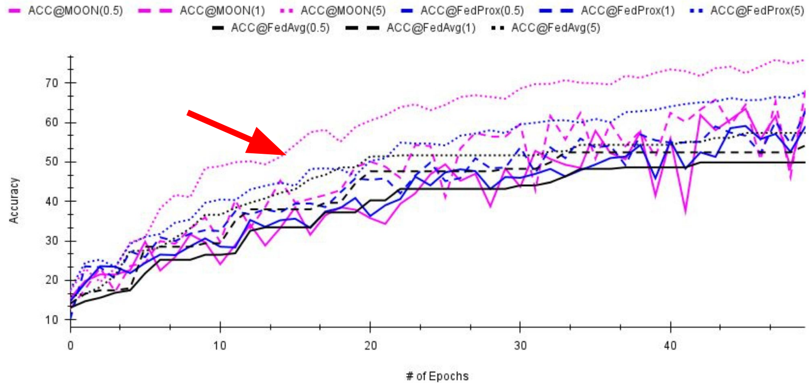
Dataset: CIFAR-10

Network: ResNet-18

Accuracy Comparison

Method	Acc.@β=0.5	Acc.@β=1	Acc.@β=5
FedAvg	49.87	54.17	57.34
FedProx	59.10	62.88	67.61
MOON	63.62	68.23	75.93

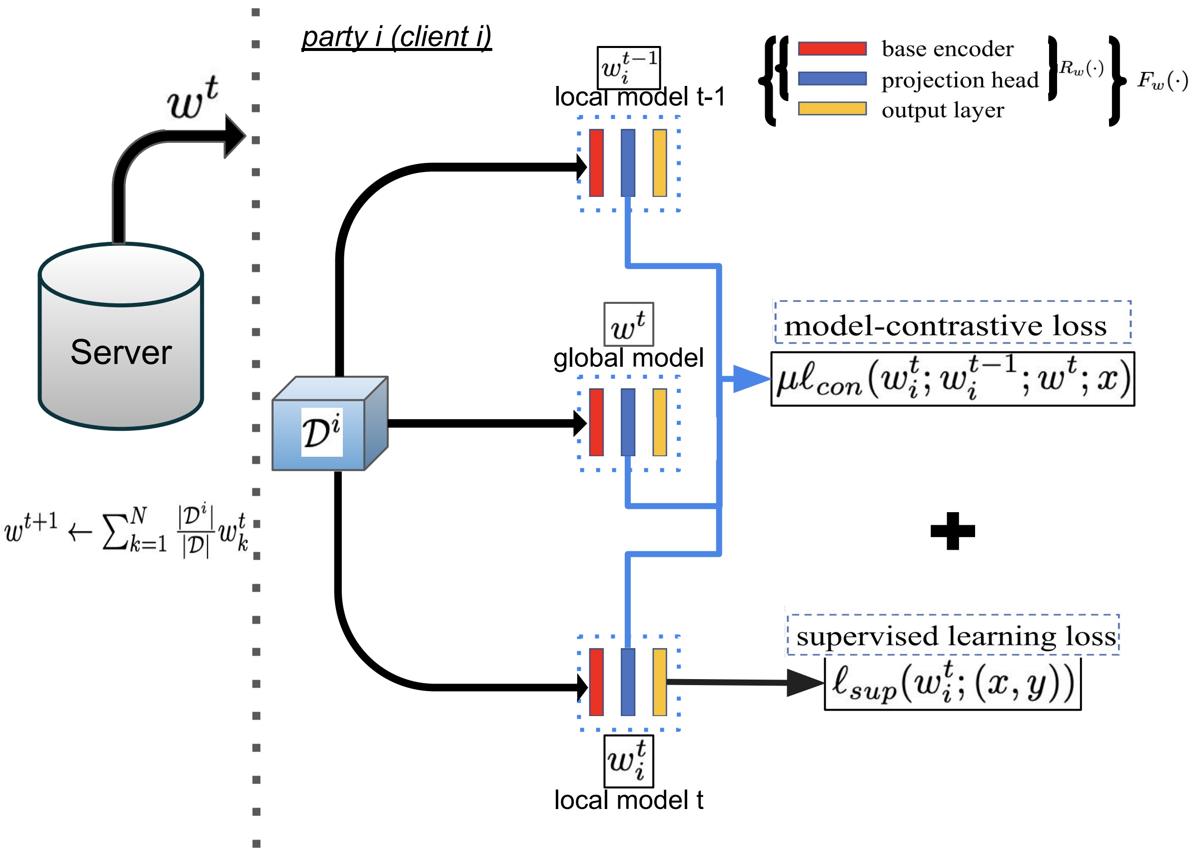
Accuracy Plots



Speed-up Comparison

Method	Rounds@$\beta = 0.5$	Speedup@$\beta = 0.5$	Rounds@$\beta = 1$	Speedup@$\beta = 1$	Rounds@$\beta = 5$	Speedup@$\beta = 5$
FedAvg	50	1x	50	1x	50	1x
FedProx	35	1.42x	34	1.47x	27	1.85x
MOON	25	2x	23	2.17x	16	3.12x

MOON

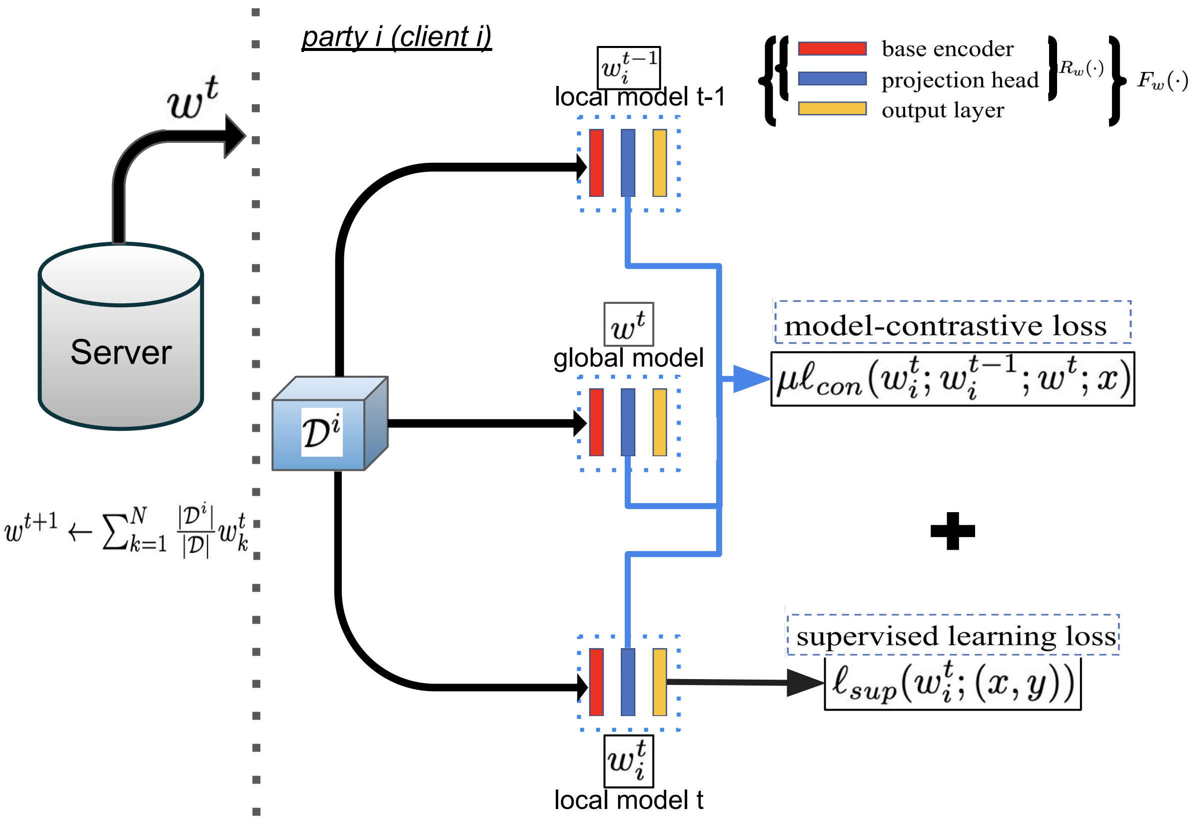


Maximize

Feature Alignment

Weight Alignment

Problem



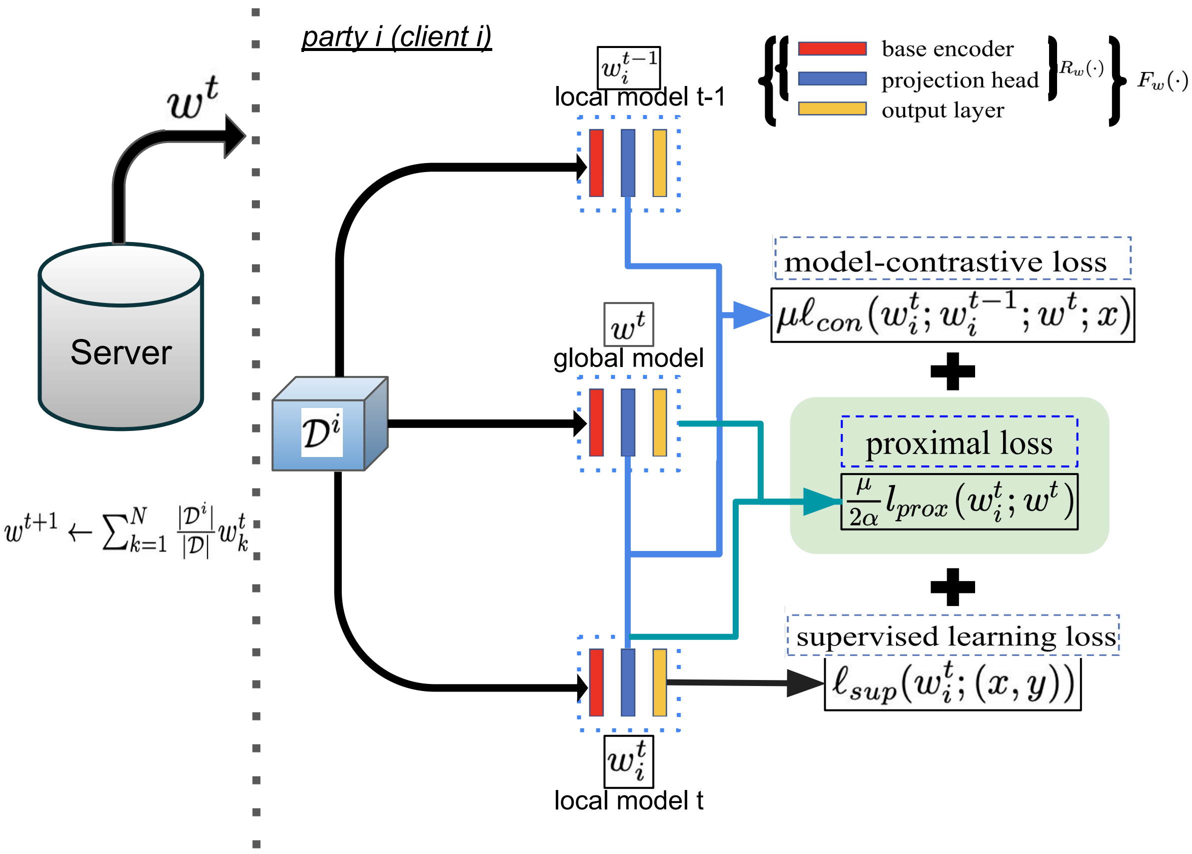
Maximize

Feature Alignment

Weight Alignment

- Large weight updates
- Unstable Training

MOON-Prox

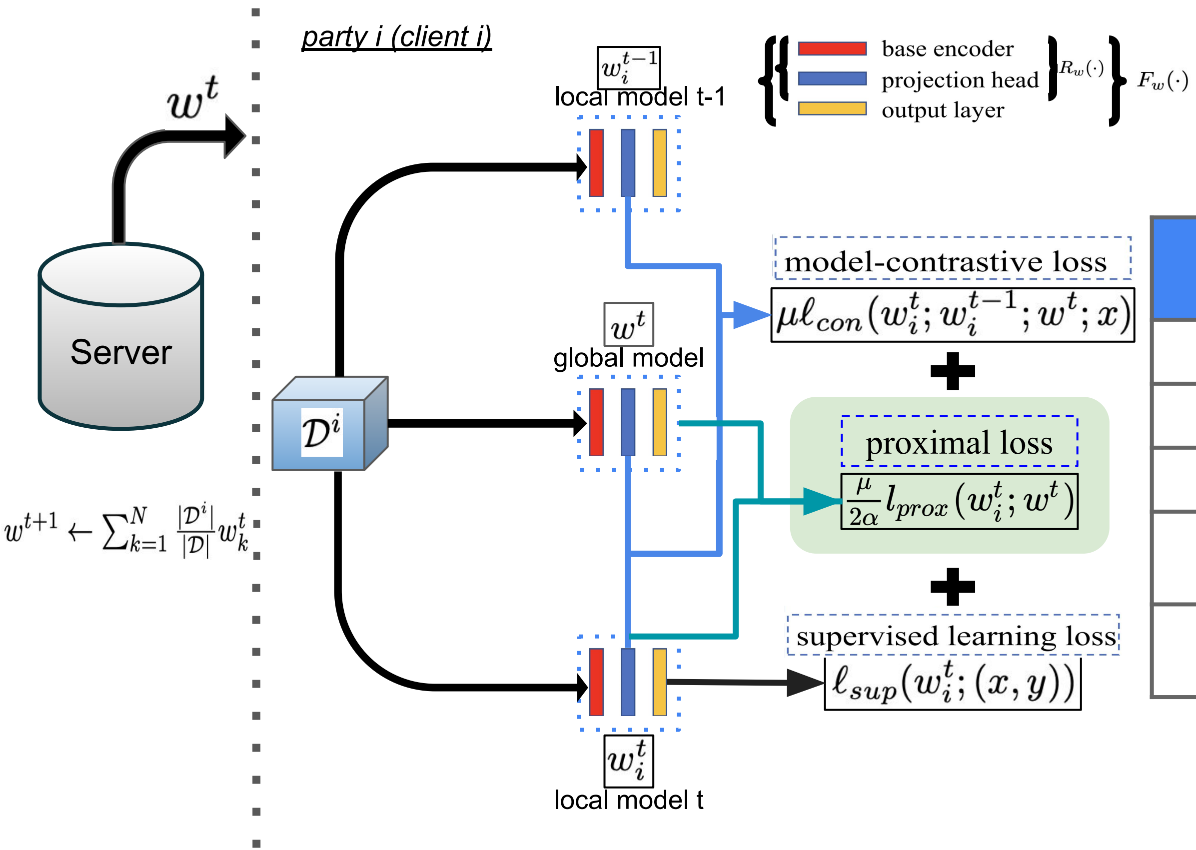


Maximize

Feature Alignment

Weight Alignment

MOON-Prox



Method	Acc@ CR=50	Acc@CR=100
FedAvg	49.87	56.96
FedProx	59.10	65.21
MOON	63.62	77.03
Ours *(alpha =100)*	66.50 **(+2.88)**	77.25
Ours *(alpha=500)*	63.16	77.59 **(+0.56)**

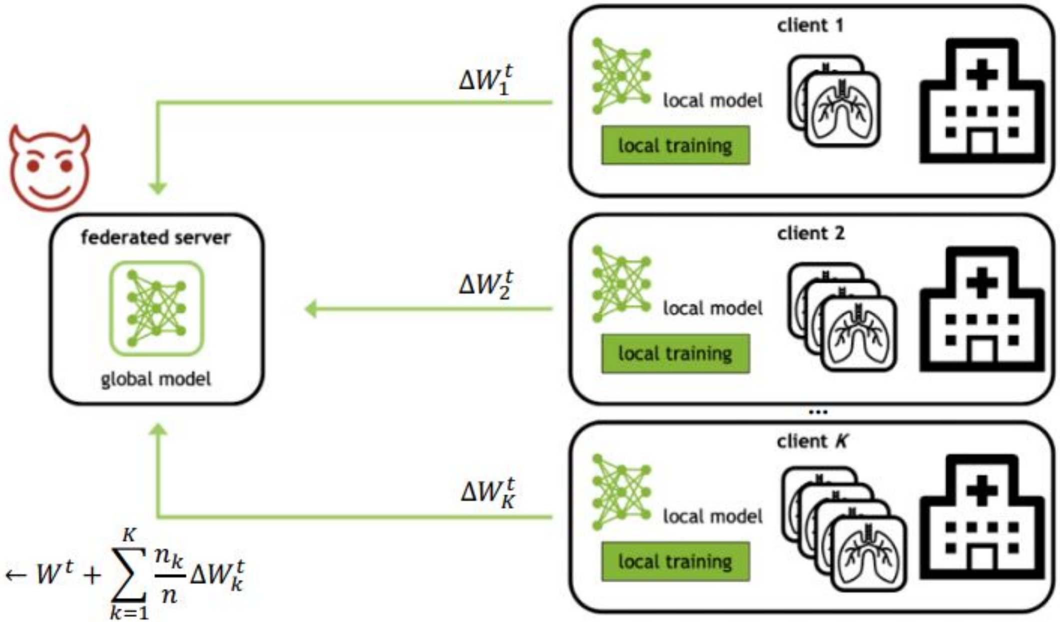
Privacy Concerns: Gradient Inversion Attack

- Might federated learning give users a false sense of privacy?

Privacy Concerns: Gradient Inversion Attack

- Might federated learning give users a false sense of privacy?

Threat model: Honest-but-Curious Server



$$W^{t+1} \leftarrow W^t + \sum_{k=1}^{K} \frac{n_k}{n} \Delta W_k^t$$

✔ Allow to separately store and process updates transmitted by individual users.

✘ Not interfere with the collaborative learning algorithm.

✘ Not modify the model architecture to better suit their attack.

✘ Cannot send malicious global parameters that do not represent the actually learned global model.

Privacy Concerns: Gradient Inversion Attack

- Optimization problem:

$$\arg\min_{x} \mathcal{L}_{grad}(x;\theta, \nabla_\theta \mathcal{L}_\theta(x*, y*)) + \alpha \mathcal{R}_{aux}(x)$$

- θ = neural network parameters
- $\nabla_\theta \mathcal{L}_\theta(x*, y*)$ = the gradient computed with a private data batch $(x^*, y^*) \in \mathcal{R}^{bxd} \times \mathcal{R}^{b}$ (b, d being the batch size, image size)

Privacy Concerns: Gradient Inversion Attack

- Optimization problem:

$$\arg\min_x \boxed{\mathcal{L}_{grad}(x;\theta,\nabla_\theta\mathcal{L}_\theta(x*,y*))} + \alpha\mathcal{R}_{aux}(x)$$

- θ = neural network parameters
- $\nabla_\theta\mathcal{L}_\theta(x*,y*)$ = the gradient computed with a private data batch $(x^*,y^*) \in \mathcal{R}^{bxd} \times \mathcal{R}^b$ (b, d being the batch size, image size)

  - $\mathcal{L}_{grad}(x;\theta,\nabla_\theta\mathcal{L}_\theta(x*,y*))$ enforces to produce a gradient that matches the observed model updates.

Privacy Concerns: Gradient Inversion Attack

- Optimization problem:

$$\arg\min_{x} \mathcal{L}_{grad}(x;\theta,\nabla_\theta\mathcal{L}_\theta(x*,y*)) + \boxed{\alpha\mathcal{R}_{aux}(x)}$$

- θ = neural network parameters
- $\nabla_\theta\mathcal{L}_\theta(x*,y*)$ = the gradient computed with a private data batch $(x^*, y^*) \in \mathcal{R}^{bxd} \times \mathcal{R}^b$ (b, d being the batch size, image size)

- $\mathcal{L}_{grad}(x;\theta,\nabla_\theta\mathcal{L}_\theta(x*,y*))$ enforces to produce a gradient that matches the observed model updates.
- $\mathcal{R}_{aux}(x)$ regularizes the recovered image based on image prior(s).

Privacy Concerns: Gradient Inversion Attack

- Optimization problem:

$$\arg\min_{x} \mathcal{L}_{grad}(x;\theta,\nabla_\theta\mathcal{L}_\theta(x*,y*)) + \alpha\mathcal{R}_{aux}(x)$$

- $\theta =$ neural network parameters
- $\nabla_\theta\mathcal{L}_\theta(x*,y*)$ = the gradient computed with a private data batch $(x^*,y^*) \in \mathcal{R}^{bxd} \times \mathcal{R}^{b}$ (b, d being the batch size, image size)

  - $\mathcal{L}_{grad}(x;\theta,\nabla_\theta\mathcal{L}_\theta(x*,y*))$ enforces to produce a gradient that matches the observed model updates.
  - $\mathcal{R}_{aux}(x)$ regularizes the recovered image based on image prior(s).

- Jonas Geiping implementation from [1]:

$$\arg\min_{x} 1 - \frac{\langle\ \nabla_\theta\mathcal{L}_\theta(x,y)), \nabla_\theta\mathcal{L}_\theta(x*,y*))\ \rangle}{\|\nabla_\theta\mathcal{L}_\theta(x,y))\|\|\nabla_\theta\mathcal{L}_\theta(x*,y*))\|} + \alpha_{TV}\mathcal{R}_{TV}(x)$$

  - $\mathcal{L}_{grad}(x;\theta,\nabla_\theta\mathcal{L}_\theta(x*,y*))$ = cosine similarity loss.
  - $\mathcal{R}_{TV}(x)$ = total variation of images.

[1]: Jonas Geiping, Hartmut Bauermeister, Hannah Dr ̈oge, and Michael Moeller. Inverting gradients–how easy is it to break privacy in federated learning? In NeurIPS, 2020

Privacy Concerns: Gradient Inversion Attack

- Optimization problem:

$$\arg\min_x \ \mathcal{L}_{grad}(x; \theta, \nabla_\theta \mathcal{L}_\theta(x*, y*)) + \alpha \mathcal{R}_{aux}(x)$$

- θ = neural network parameters
- $\nabla_\theta \mathcal{L}_\theta(x*, y*)$ = the gradient computed with a private data batch $(x^*, y^*) \in \mathcal{R}^{bxd} \times \mathcal{R}^b$ (b, d being the batch size, image size)

- $\mathcal{L}_{grad}(x; \theta, \nabla_\theta \mathcal{L}_\theta(x*, y*))$ enforces to produce a gradient that matches the observed model updates.
- $\mathcal{R}_{aux}(x)$ regularizes the recovered image based on image prior(s).

- Jonas Geiping implementation from [1]:

$$\arg\min_x 1 - \frac{\langle\ \nabla_\theta \mathcal{L}_\theta(x, y)), \nabla_\theta \mathcal{L}_\theta(x*, y*))\ \rangle}{\|\nabla_\theta \mathcal{L}_\theta(x, y))\| \|\nabla_\theta \mathcal{L}_\theta(x*, y*))\|} + \alpha_{TV} \mathcal{R}_{TV}(x)$$

- $\mathcal{L}_{grad}(x; \theta, \nabla_\theta \mathcal{L}_\theta(x*, y*))$ = cosine similarity loss.
- $\mathcal{R}_{TV}(x)$ = total variation of images.

Assumptions:
- Batch normalization statistics ✔
- Private labels ✔

[1]: Jonas Geiping, Hartmut Bauermeister, Hannah Dr¨oge, and Michael Moeller. Inverting gradients–how easy is it to break privacy in federated learning? In NeurIPS, 2020

Experimental Setup

- **Architectures**: ResNet-18, ResNet-50.
- **Normalization Layers:** Group Norm, Batch Norm.
- **Algorithms**: FedAvg, FedSGD.
- **Dataset**: CIFAR-10.
- **Metrics**: LPIPS, PSNR.
- **Repository:** Breaching at [2]

[2]: https://github.com/JonasGeiping/breaching.git

Experimental Results: BatchNorm Case

ResNet-18 Batch Norm | **ResNet-50 Batch Norm**

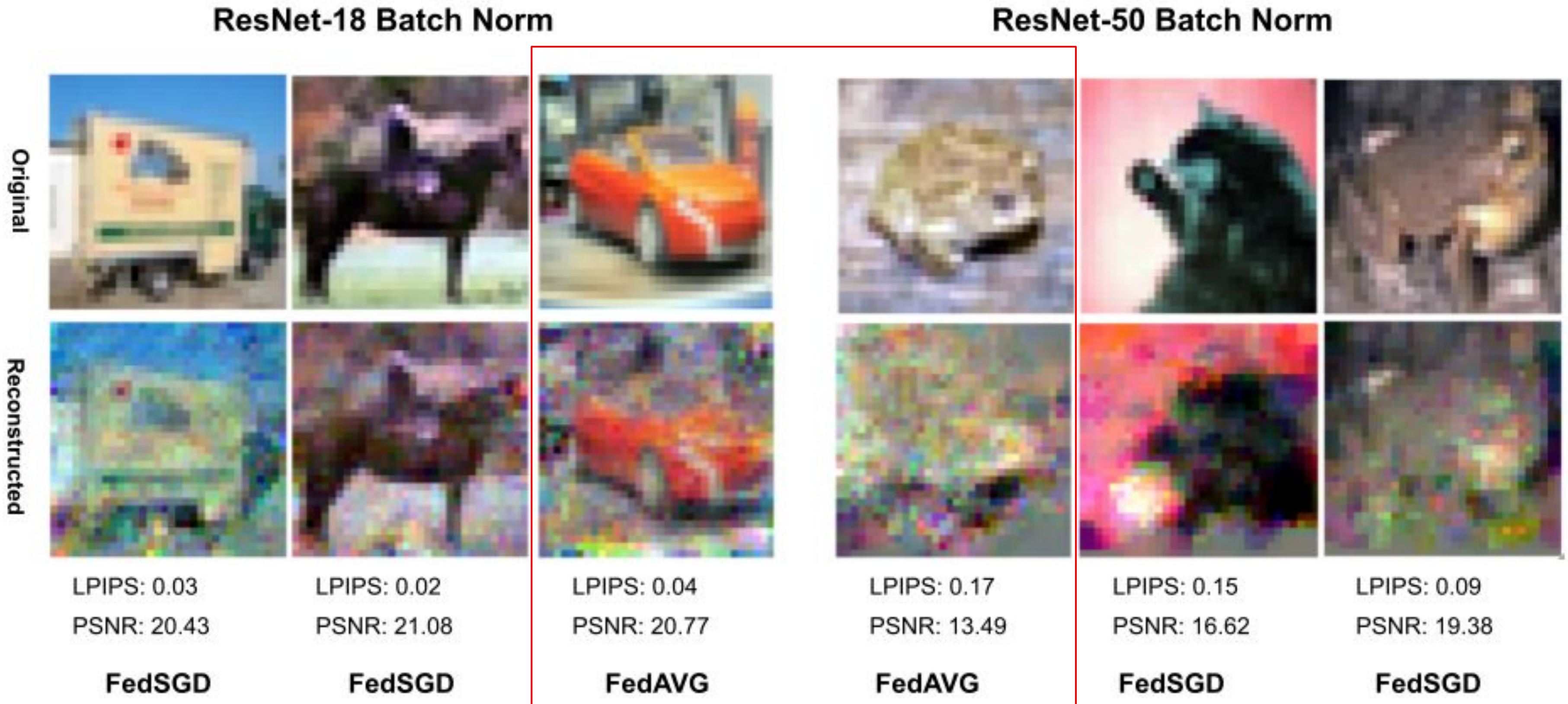
LPIPS: 0.03	LPIPS: 0.02	LPIPS: 0.04	LPIPS: 0.17	LPIPS: 0.15	LPIPS: 0.09
PSNR: 20.43	PSNR: 21.08	PSNR: 20.77	PSNR: 13.49	PSNR: 16.62	PSNR: 19.38
FedSGD	**FedSGD**	**FedAVG**	**FedAVG**	**FedSGD**	**FedSGD**

Experimental Results: BatchNorm Case

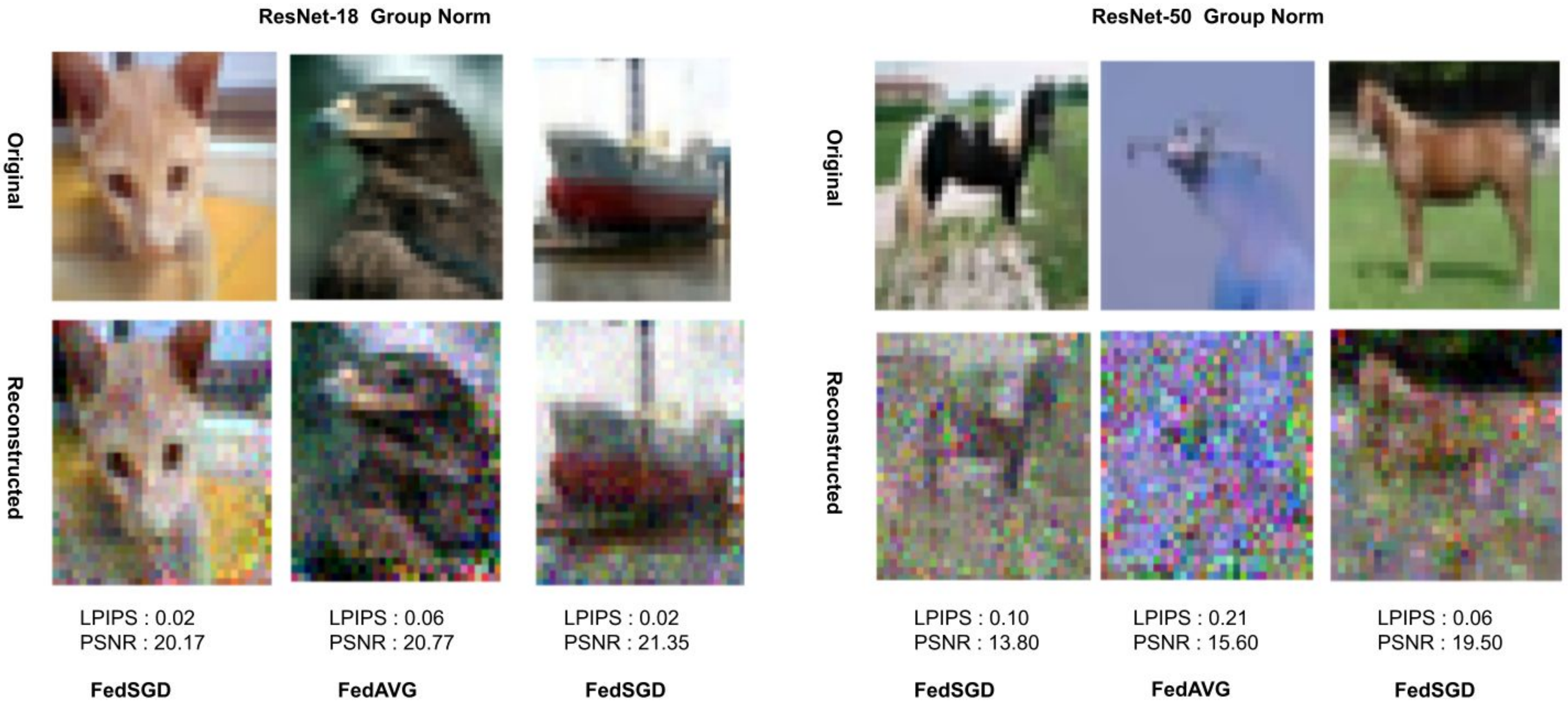
ResNet-18 Batch Norm | **ResNet-50 Batch Norm**

LPIPS: 0.03	LPIPS: 0.02	LPIPS: 0.04	LPIPS: 0.17	LPIPS: 0.15	LPIPS: 0.09
PSNR: 20.43	PSNR: 21.08	PSNR: 20.77	PSNR: 13.49	PSNR: 16.62	PSNR: 19.38
FedSGD	**FedSGD**	**FedAVG**	**FedAVG**	**FedSGD**	**FedSGD**

Experimental Results: BatchNorm Case



Experimental Results: GroupNorm Case



Conclusion

- We benchmarked recent federated learning algorithms: FedAvg, FedGKT and MOON and assessed their performance in IID and non-IID data distribution.

- We conclude that GN gives better performance than BN for non-IID data distribution.

- We proposed MOON-Prox that performs best among all the compared federated learning algorithms.

- We investigate security issues faced by federated learning methods via gradient attack and were able to reconstruct the private data of the client.

Thanks for your attention

Method	Model	Norm.	Accuracy
Centralized	ResNet-50	BN	86.91
Centralized	ResNet-50	GN	87.48
FedAVG[IID]	ResNet-50	BN	81.14
FedAVG[IID]	ResNet-50	GN	78.60
FedAVG[nonIID]	ResNet-50	BN	60.00
FedAVG[nonIID]	ResNet-50	GN	60.51
FedGKT[IID]	ResNet-49	BN	55.28
FedGKT[IID]	ResNet-49	GN	53.27
FedGKT[nonIID]	ResNet-49	BN	27.01
FedGKT[nonIID]	ResNet-49	GN	19.79
FedGKT*[IID]	ResNet-49	BN	74.55
FedGKT*[IID]	ResNet-49	GN	74.28
FedGKT*[nonIID]	ResNet-49	BN	47.53
FedGKT*[nonIID]	ResNet-49	GN	48.58