The Resource Problem of Using Linear Layer Leakage Attack in Federated Learning

Joshua C. Zhao[1], Ahmed Roushdy Elkordy[2], Atul Sharma[1], Yahya H. Ezzeldin[2] Salman Avestimehr[2], Saurabh Bagchi[1]

[1]Purdue University, [2]University of Southern California

Linear layer leakage attacks

Linear layer leakage is a class of data reconstruction attacks that inserts fully-connected (FC) layers into a benign model to leak user data. The strengths of the attack include:

Scalability: The inserted layer size can be increased to work with larger batch sizes or aggregation while maintaining high leakage rate.

Single round attack: Only a single training round is required

Perfect reconstruction: Images recovered by the server are near perfect reconstructions of the client data.

Datatype domain agnostic: The attack is not limited to images and can work regardless of the datatype domain.

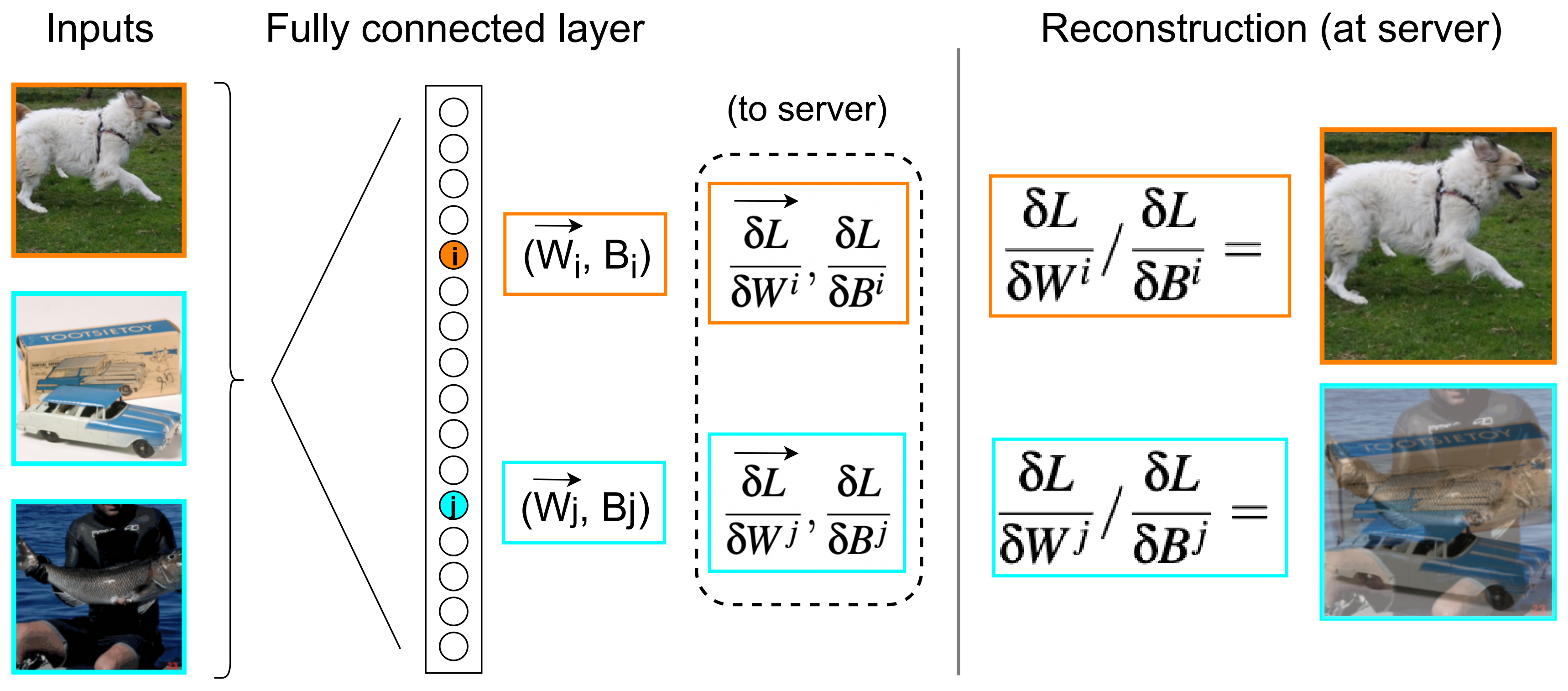

Inputs
Fully connected layer
Reconstruction (at server)
i
j
$(\overrightarrow{W_i}, B_i)$
$(\overrightarrow{W_j}, B_j)$
(to server)
$\frac{\delta L}{\delta \overrightarrow{W^i}}, \frac{\delta L}{\delta B^i}$
$\frac{\delta L}{\delta \overrightarrow{W^j}}, \frac{\delta L}{\delta B^j}$
$\frac{\delta L}{\delta W^i} / \frac{\delta L}{\delta B^i} =$
$\frac{\delta L}{\delta W^j} / \frac{\delta L}{\delta B^j} =$

Figure 1. Basic linear layer leakage through an inserted FC layer.

Scalability problems

When secure aggregation is used in FL, linear layer leakage privacy attacks such as Robbing the Fed [2] can still maintain high leakage rate by linearly increasing the FC layer size with the number of clients. However, this leads to several problems:

Model size: The FC layer size increase directly leads to a multiplicatively larger number of parameters in the model.

Communication cost: Clients also incur a significant communication cost increase when receiving models and sending updates due to model size.

Detectability: With 100 clients, the size of the FC layer can easily scale to over 10,000 units. This abnormally large layer is much more detectable.

Application of sparsity on attacks

This large increase in model overhead from prior work comes from an incorrect perspective on attacking aggregated updates. Since linear layer leakage requires enough parameters to store the image pixel information, this requires a model large enough for *all* images across *all* clients. Attacking the aggregate update as a large super-batch results in individual clients incurring the entire overhead.

However, even for an aggregate attack, client models only need enough parameters to store their individual batch of images. All other parameters can be zero. This creates very sparse attack layers that allows for sparse tensor storage and operations to decrease resource overhead.

Our Mandrake attack

The Mandrake [3] attack utilizes sparsity to improve aggregated leakage:

Identity mapping sets: Client leakage is separated by sending customized convolutional kernels to each client such that only one set of connections is non-zero. These kernels push the input images through different channels for each client.

Leakage: The FC layer following the convolutional layer leaks the images.

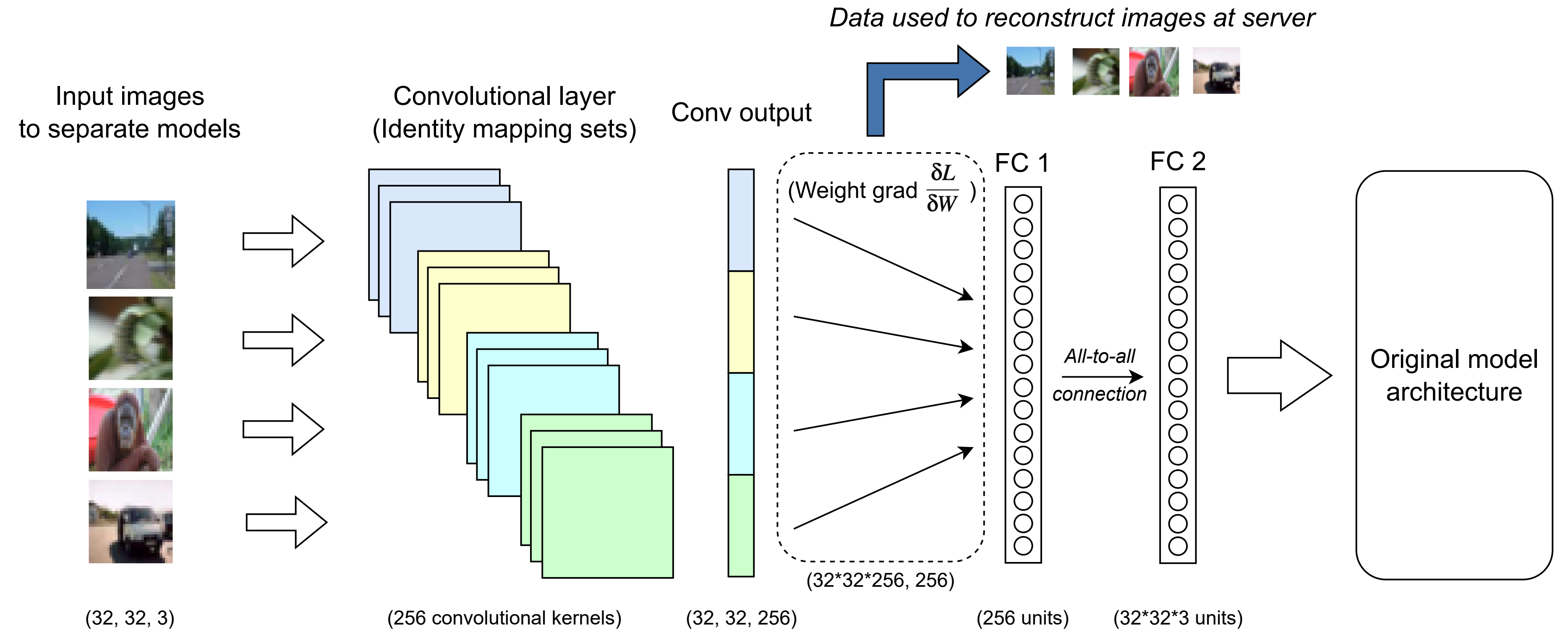
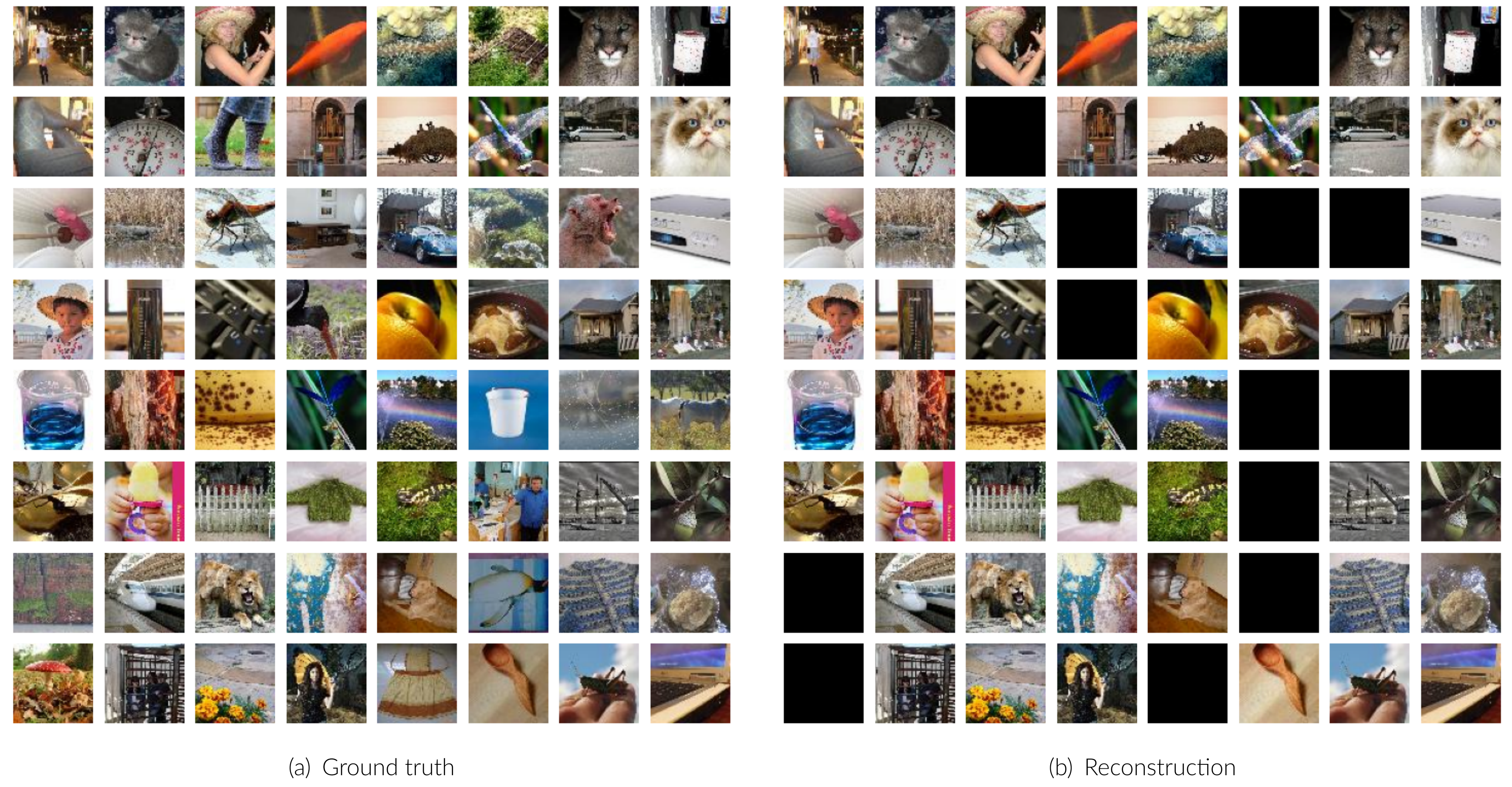


Figure 2. Mandrake attack architecture.

Only the number of kernels increases with more clients instead of the FC layer size.

Parameters: The absolute number of parameters is $\approx \frac{1}{2}$ compared to current SOTA Robbing the Fed. The number of non-zero parameters per client is only $\approx \frac{1}{N}$.

Leakage quality: Even with the use of sparsity, the reconstructed images are still near exact copies of client data.

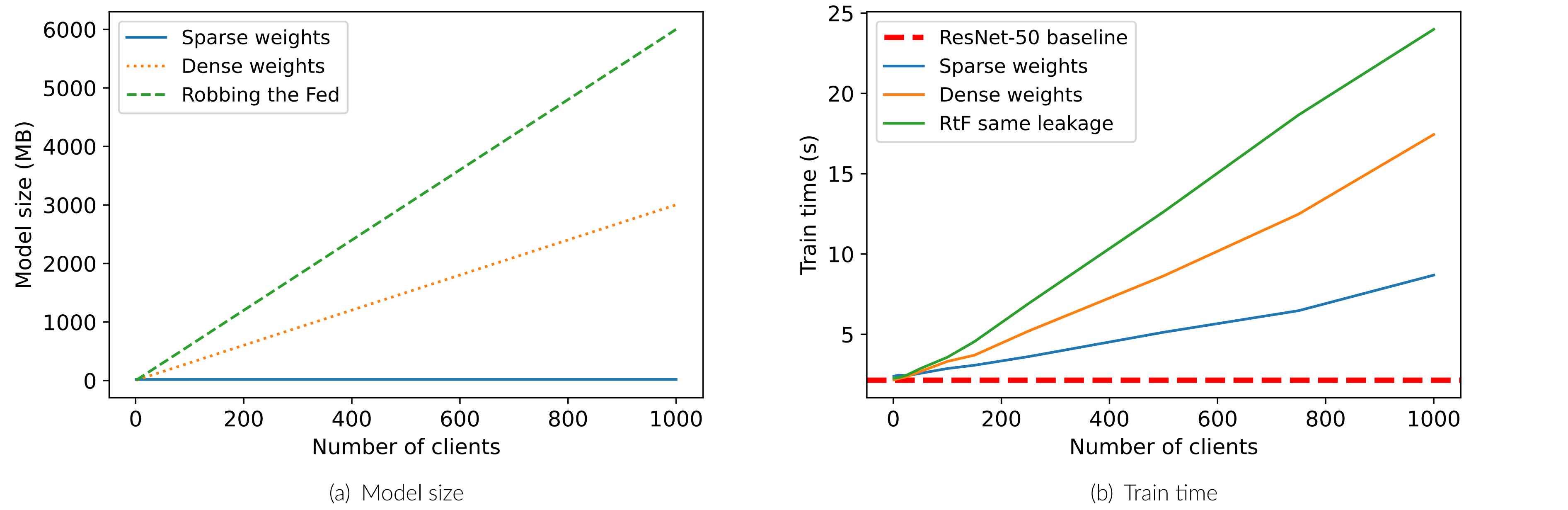

(a) Ground truth

(b) Reconstruction

Figure 3. Tiny ImageNet reconstructions from a client with a batch size of 64 using Mandrake.

Leakage rate, model size, training time

Using sparsity with the Mandrake attack, the additional model size added and computation time added by the attack is $327\times$ and $3.34\times$ smaller than Robbing the Fed for 1000 clients on Tiny ImageNet respectively.



(a) Model size

(b) Train time

Figure 4. (a) Client model size overhead and (b) training time for 1-1000 clients for Robbing the Fed compared to Mandrake using a sparse and dense tensor representation on Tiny Imagenet ($32 \times 32 \times 3$).

The added model size from the sparse attack barely changes regardless of the number of clients being attacked.

	Clients	Robbing the Fed	Dense weights	Sparse weights
MNIST (28x28x1)	100	153.2	77.3	4.6
	1000	1532.2	766.4	4.6
CIFAR-100 (32x32x3)	100	600.1	303.0	18.0
	1000	6001.0	3003.3	18.3
Tiny ImageNet (64x64x3)	100	2400.1	1212.1	72.1
	1000	24001.0	12012.4	72.4
ImageNet (256x256x3)	100	38400.9	19392.8	1152.8
	1000	384001.7	192193.1	1153.1

Table 1. Comparison of model size overhead (MB) using different datasets with batch size 64 and 100 and 1000 clients. At 1000 clients on ImageNet, the sparse representation adds a 1.1GB overhead while Robbing the Fed adds 375GB.

Despite having a much smaller added model size and computation time, the leakage rate of the sparse Mandrake attack maintains near equivalent leakage rate to Robbing the Fed.

	Sparse Mandrake	Robbing the Fed
CIFAR-100	77.5% (4957)	77.1% (4931)
MNIST	71.0% (4546)	75.1% (4803)
Tiny ImageNet	77.8% (4978)	77.7% (4970)

Table 2. Total leakage rate of sparse Mandrake and Robbing the Fed on various datasets. For all three datasets, 100 aggregated clients and batch size of 64 were used for 6400 total images.

References

[1] Joshua C. Zhao, Ahmed Roushdy Elkordy, Atul Sharma, Yahya H. Ezzeldin, Salman Avestimehr, and Saurabh Bagchi. The Resource Problem of Using Linear Layer Leakage Attack in Federated Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[2] Liam H Fowl, Jonas Geiping, Wojciech Czaja, Micah Goldblum, and Tom Goldstein. Robbing the fed: Directly obtaining private data in federated learning with modified models. In *International Conference on Learning Representations*, 2022.

[3] Joshua C. Zhao, Atul Sharma, Ahmed Roushdy Elkordy, Yahya H. Ezzeldin, Salman Avestimehr, and Saurabh Bagchi. Secure aggregation in federated learning is not private: Leaking user data at large scale through model modification. *arXiv preprint arXiv:2303.12233*, 2023.

[4] Franziska Boenisch, Adam Dziedzic, Roei Schuster, Ali Shahin Shamsabadi, Ilia Shumailov, and Nicolas Papernot. When the curious abandon honesty: Federated learning is not private. In *IEEE European Symposium on Security and Privacy (IEEE Euro S&P)*, 2023.