Federated Learning with ResNet-50 on CIFAR-10

Alessandro Maria Feri'
Politecnico di Torino
Student ID: s305949
s305949@studenti.polito.it

Riccardo Giuseppe Ricevuto
Politecnico di Torino
Student ID: s292663
s292663@studenti.polito.it

***Abstract*—With this study we want to deepen the scenario of Federated Learning, implementing the best-known algorithms and investigating the problems concerning the statistical heterogeneity, systems heterogeneity and privacy. As CNN model we used the ResNet50 on the CIFAR-10 dataset. After a standard Federated scenario, we implement different algorithms: FedAVG, FedGKT, FedProx. Each one of the algorithms was tested with Batch and Group Normalization and compare the results considering the IID and balanced version of CIFAR-10 and also the non-IID and unbalanced version.**
https://github.com/ricevutoriccardo/Federated-Learning-with-ResNet-50-on-CIFAR-10

I. Introduction

Federated Learning is a cutting-edge approach to machine learning that enables organizations to train models on large amounts of decentralized data while maintaining the privacy of individual users. This has become increasingly relevant in recent years, as the volume of data generated by Internet of Things (IoT) devices and other edge computing nodes continues to grow. It was developed by Google in 2016 [1] as a solution to the challenge of processing huge amounts of data while keeping it safe and secure.
In Federated Learning, the model is sent to multiple devices where it trains on local data and sends the updated model back to a central server. The server aggregates the models from all devices to create a global model, which is repeated until the model is accurate enough. This approach enables organizations to train models on sensitive data without compromising privacy and allows them to train models on a larger and more diverse datasets than would be possible with a single device or server.
In this paper, we explore the performance of Federated Learning using ResNet50 [2] as convolutionals neural network architecture and CIFAR-10 [9] dataset, with a focus on comparing the impact of *IID* and *non-IID balanced* and *unbalanced* data distributions [8], as well as the effect of different normalization techniques such as Batch Normalization and Group Normalization [8]. For our study we implement a standard Federated architecture, then we implemented FedAVG [1], the most used and studied model in the federated scenario. After we analyzed the problem of system heterogeneity with the FedGKT [3] model, and the privacy with the use of the gradient inversion attack.

II. Related Works

A. *FedAvg*

FedAVG, that stands for Federated Averaging, is the most used Federated Learning algorithm introduce for the first time in 2017 by McMahan et al. [1]. In this implementation a model is sent to a different clients, each client train the model and after send back the model updated to a central server. The server use the models received by all the clients to update the global model taking the average of all the parameters received.

B. *FedGKT*

In the federated context, a large model size impedes training on resource-constrained edge device. The FedGKT (Federated Gaussian Knowledge Transfer [3]) aims to address the problems created by systems heterogeneity. In the algorithm the clients train a smaller model and periodically send their knowledge to a larger server-side CNN. In this way, the clients with less computing power to participate in the training process and ensures that the global model converges with more convergence speed and with an higher accuracy. For our test we use a *ResNet-8* as client model and a *ResNet-49* as server model. Compare with FedAVG, FedGKT improved the performance especially in scenarios with systems heterogeneity.

C. *FedProx*

FedProx [3] aims to enhance FedAVG by applying simple modifications to address the problem of statistical heterogeneity. Unlike FedAVG, FedProx allows for the use of information from stragglers. Stragglers are the devices that are slow or inactive, so this increases the statistical heterogeneity and slows down the rate of convergence. To mitigate this problem, FedProx includes a "proximal term" in the local objective function. This term helps to limit the effect of inconsistent local updates, because the update from the stragglers has less impact on the global model. A new hyperparameter, μ, is introduced to adjust the strength of the regularization. Compare with FedAVG, FedProx is more robust and improve the performance especially in scenarios with a large number of strugglers.

D. *Gradient Inversion Attack*

Gradient aggregation plays a vital role in federated or collaborative learning [5]. The gradient aggregation may suffer from some attacks, such as gradient inversion, where the private training data can be recovered from the shared gradients.

Gradient inversion attack is a type of attack on machine learning models, in particular on deep neural networks. Gradient inversion attack can be used to reconstruct an image from a trained neural network. This method can be used to obtain sensitive information that are present in the original image. To protect against this type of attack, it is important to develop robust defense mechanisms like use techniques of adversarial training, regularization, input. With these techniques we can make the model more robust to small changes in the input and to limit the model's reliance on adversarial examples.

E. CIFAR-10

The CIFAR-10 dataset [9] (Canadian Institute For Advanced Research) is a collection of images that are commonly used to train machine learning and computer vision algorithms. It is one of the most widely used datasets for machine learning research. The CIFAR-10 dataset contains 60,000 32x32 color images in 10 different classes. The 10 different classes represent airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. There are 6,000 images of each class. Real-life data (referring to objects, values, attributes, and other aspects) is essentially non-independent and identically distributed (non-IID). In contrast, most of the existing analytical or machine learning methods are based on IID data (like CIFAR-10). So, there needs to be a proper approach to handle such type of real-world dataset. Non-IID distribution is one of the main challenge in Federated scenario, causing unstable performances of deep learning models.

In our experiments, we use 3 different data distribution to evaluate our model:

1) IID: simple case. Data is equal distributed, each client has the same amount of images for each of the 10 classes.
2) non-IID balanced: In this case the same amount of images is assigned to each client and everyone of them has access to either one or two classes (Figure 4.a).
3) non-IID unbalanced: in this case dataset is divided based on the classes per client parameter. (Figure 4.b).

F. Normalization

Batch Normalization (BN) [8] has been an important component of many state-of-the-art deep learning models, especially in computer vision. It normalizes the layer inputs by the mean and variance computed within a batch. There are situations that we have to settle for a small batch size, for example when each data sample is highly memory-consuming or when we train a very large neural network, which leaves little GPU memory for processing data. Therefore we need alternatives to BN which work well with small batch size.

Group Normalization (GN) [7] is one of the latest normalization methods that avoids exploiting the batch dimension, thus is independent of batch size. GN normalizing the features across groups instead of across the entire batch. This makes the computation of group normalization independent of batch size and reduces the error that occurs with small batch sizes in batch normalization. When the batch size is small, GN consistently outperforms BN. However, when the batch size is significantly large, GN does not scale as well as BN and might not be able to match the performance of BN.

III. METHODS

TABLE I: Notation

Notation	Description
η	Learning rate
B	Batch size
E	Number of epochs
Ec	Number of epochs for the clients
K	Number of clients
T	Communication Round
C	Fraction of Client
μ	Proximal term
w_d	Weight decay

A. Centralized training

The performance of any model trained in a federated fashion is upper bounded by the results obtained in the centralized setting in Table II. We evaluated the performances, concerning the accuracy of *ResNet-50* on *CIFAR-10*. We compared two types of Normalization Layers, *Batch Normalization* and *Group normalization* (number of groups: 2). The hyperparameters tuning includes: learning rate, weight decay, the optimizer (sgd or adam) and the batch size. (Fig 5)

TABLE II: Centralized Training

Normalization layer	*Accuracy*	*# of model parameters*	*Hyperparameters*
BN	0.97	23,520,842	E = 40 B = 64 η = 0.0001 w_d = 1e-05 optimizer: adam
GN	0.93	23,520,842	E = 40 B = 16 η=0.001 w_d = 1e-05 optimizer: sgd

B. FedAvg

In the federated scenario, the training has been computed with the same model as the centralized baseline. The distribution of data among clients plays a primary role in performance. Specifically, we run the experiment in different clients distributions: **IID**, **non-IID balanced** and **non-IID unbalanced** which could be a simulation of a real world scenario.

TABLE III: FedAVG Accuracy

Data Distribution	Normalization	Accuracy
IID	BN	0.69
	GN	0.45
non-IID balanced	BN	0.18
	GN	0.21
non-IID unbalanced	BN	0.20
	GN	0.15

K=100, C=0.1 T=100, B=10, η:0.001, w_d:1e-05, Ec=1

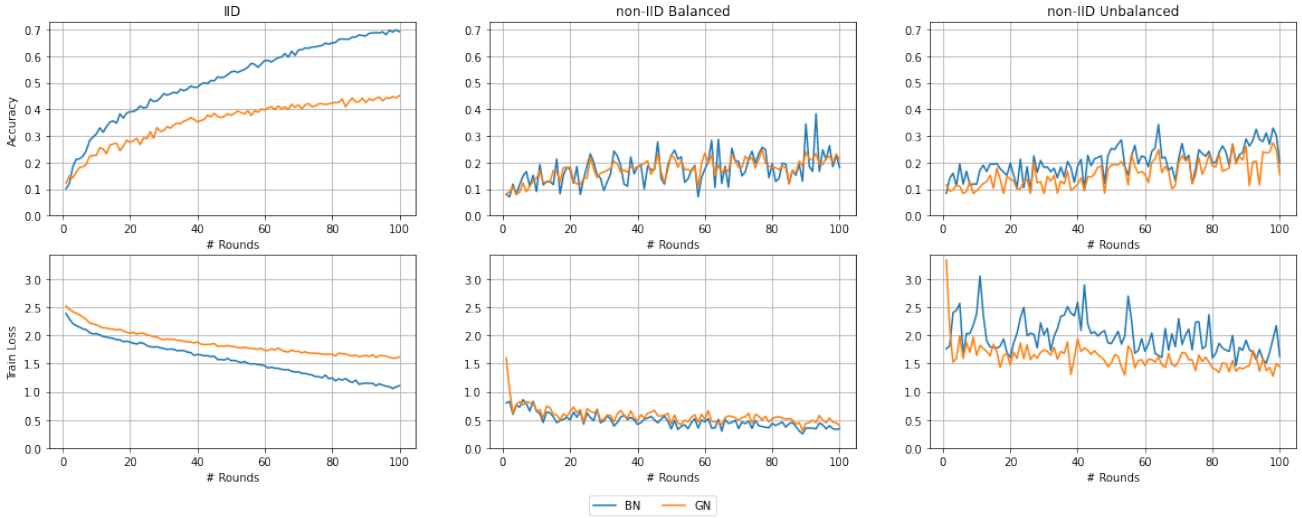


Fig. 1: FedAVG Training with Batch Normalization and Group Normalization

Table III shows the final accuracy of FedAVG with the previously mentioned distribution. (Plot on Figure 1)

C. *FedGKT*

FedGKT tackles the resource-constrained problem on edge devices algorithm. Due to computational limitation, we perform 10 epochs on the server , while for the clients 1 local epoch was set. We run this algorithm with Batch and Group Normalization for each of different data distributions.

TABLE IV: FedGKT

Data distribution	*Normalization*	*Accuracy*
IID	BN	0.74
	GN	0.68
non-IID unbalanced	BN	0.50
	GN	0.34
non-IID balanced	BN	0.32
	GN	0.22

K=100, C=0.1, T=10, η=0.001, Ec=10

Figure 8 show the performances of the algorithm with Batch Normalization and Group Normalization for each of the data settings. In figure 7 we can compare better the accuracy distributions of the different cases.

D. *FedProx*

We set $\mu > 0$ for the base implementation of the FedPROX algorithm. Specifically we set $\mu = 0.01$ and compared the results with FedAVG as we can see in Fig.2. The performance of the FedProx are showed in Figure 6

TABLE V: FedProx

Data distribution	*Normalization*	*Accuracy*
non-IID unbalanced	BN	0.20
	GN	0.19
non-IID balanced	BN	0.20
	GN	0.14

K=100, T=100, B=10, η=0.001, w_d=1e-05, Ec=5

Table V shows the final accuracy of FedPROX, that has similar results of FedAVG

E. *Gradient Inversion Attack*

To test privacy on our model, we simulated an attack, using the ResNet50 model on CIFAR10 dataset with two strong assumption. **Assumption 1**: Knowing BatchNorm statistics. Knowing BatchNorm statistics would enable the attacker to apply the same batch normalization used by the private batch on his recovered batch, to achieve a better reconstruction **Assumption 2**: Knowing or able to infer private labels. Private labels are not intended to be shared in Federated learning, but knowing them would improve the attack. [5] As it's shown in Figure 3, we test the attack by visualize 4 images of different classes (Deer, Frog, Automobile, Bird) and comparing the original (Figure 3.a) and the reconstructed images (Figure 3.b), we can notice some pattern of similarity but they are not recognizable.

IV. Discussion & Results

The results of our study shows how the *ResNet-50* model work using the most significant Federated Learning's algorithms.

1) *Performances*:

- FedAvg: Batch Normalization get better results for each data distribution. As we expected, the non-IID case get lower accuracy respect to the IID.
- FedGKT: As the previous case, Batch Normalization performs better than Group Normalization, and the non-IID unbalanced case achieves better results than the balanced case.
- FedProx: Compared to FedAVG, it achieve a *comparable* accuracy level for the non-IID cases. This outcome can be attributed to the proximity of the local epochs (*Ec*) in both algorithms, leading to an inadequate regularization of the loss function.

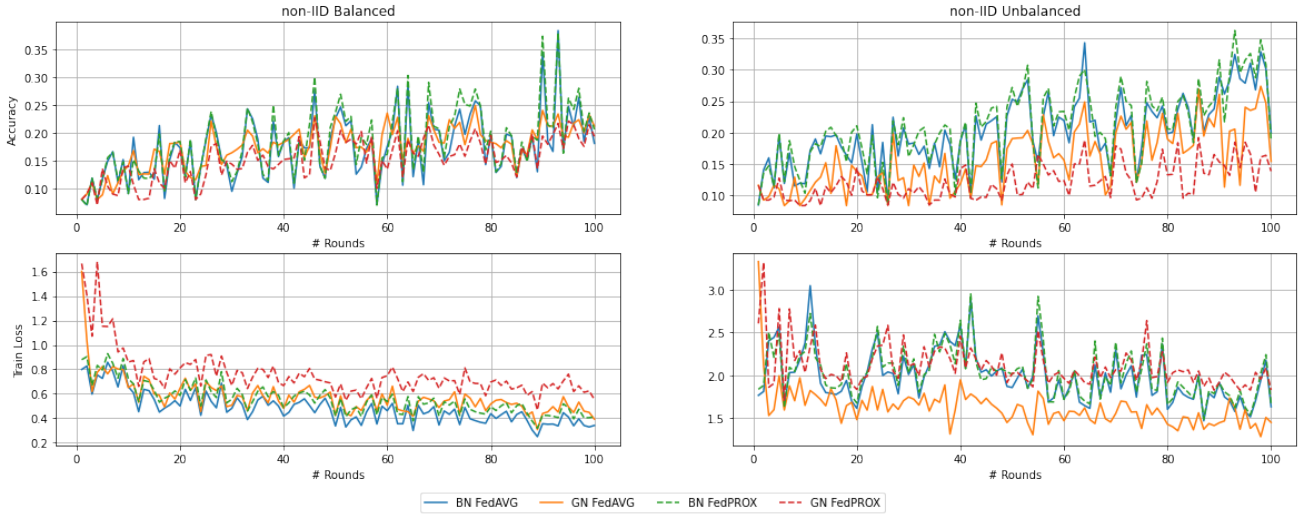


Fig. 2: Comparison between FedAVG and FedPROX

- Compared to other algorithms, FedGKT achieved a higher accuracy and reduced the overall computational cost thanks to the lighter model in the clients.

2) *Data distribution and Normalization Method*:

- In the IID case the Batch Normalization perform better for his dependence on mini-batch, mean and variance as reported in *Hsieh, Kevin, et al. "The non-IID data quagmire of decentralized machine learning."* [8].
- In the non-IID settings this dependence is a problem, indeed as we can notice from the results, the accuracy is going to be low.
- Our analysis of the model implementation results reveals a superiority of the unbalanced non-IID scenarios vs the balanced non-IID scenarios in terms of performance.

V. Conclusion

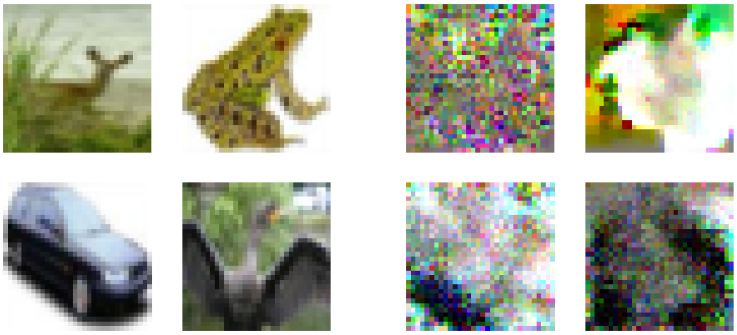
In this project, using ResNet-50, we explore the state of the art of Federated Learning algorithms, deal the main challenges of these scenario: statistical heterogeneity, system heterogeneity and privacy concerns. From the literature [8] we know that the Group Normalization are usually more effective than Batch Normalization ones in federated scenarios. So we compared this Normalization techniques, showing how the Batch Normalization achieve better performance in the IID cases and similar results in the others distributions settings. In the case of Federated Averaging, it was observed that there was a high consumption of computational resources. As a result, FedGKT performs better in real-world scenarios where lighter models are deployed at the edges. To evaluate the privacy concerns, we try to attack our model using the gradient inversion attack, and we observed that the model response well, preserving the privacy of the client.

References

[1] McMahan, Brendan et al. "Communication-Efficient Learning of Deep Networks from Decentralized Data" Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, PMLR 54:1273-1282, (2017).
[2] He, Kaiming, et al. "Deep residual learning for image recognition." arXiv preprint arXiv:1512.03385 (2015).
[3] Tian Li, et al. "Federated optimization in heterogeneous networks." In I. Dhillon, D. Papailiopoulos, and V. Sze, editors, Proceedings of Machine Learning and Systems, volume 2, pages 429–450, 2020.
[4] He, Chaoyang, Murali Annavaram, and Salman Avestimehr. "Group knowledge transfer: Federated learning of large CNNs at the edge." Advances in Neural Information Processing Systems 33 (2020): 14068-14080.
[5] Geiping, Jonas, et al. "Inverting gradients-how easy is it to break privacy in federated learning?." Advances in Neural Information Processing Systems 33 (2020): 16937-16947.
[6] Huang, Yangsibo, et al. "Evaluating gradient inversion attacks and defenses in federated learning." Advances in Neural Information Processing Systems 34 (2021).
[7] Wu, Yuxin, and Kaiming He. "Group normalization." Proceedings of the European conference on computer vision (ECCV). 2018.
[8] Hsieh, Kevin, et al. "The non-iid data quagmire of decentralized machine learning." International Conference on Machine Learning. PMLR, 2020.
[9] https://paperswithcode.com/dataset/cifar-10

APPENDIX

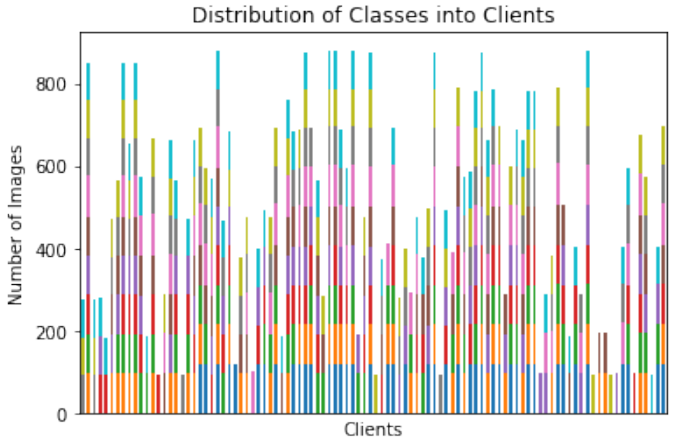
In this appendix, we provide additional results and figures to support the main text.



(a) Original images (b) Reconstructed images

Fig. 3: Gradient Inversion Attack

(a) non-IID Balanced



(b) non-IID Unbalanced

Fig. 4: Client data split

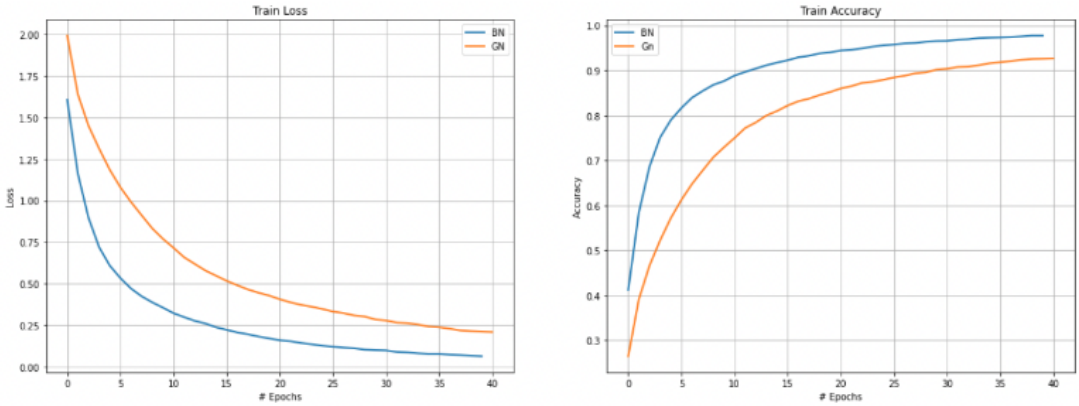


Fig. 5: Centralized Training after the hyperparameater tuning

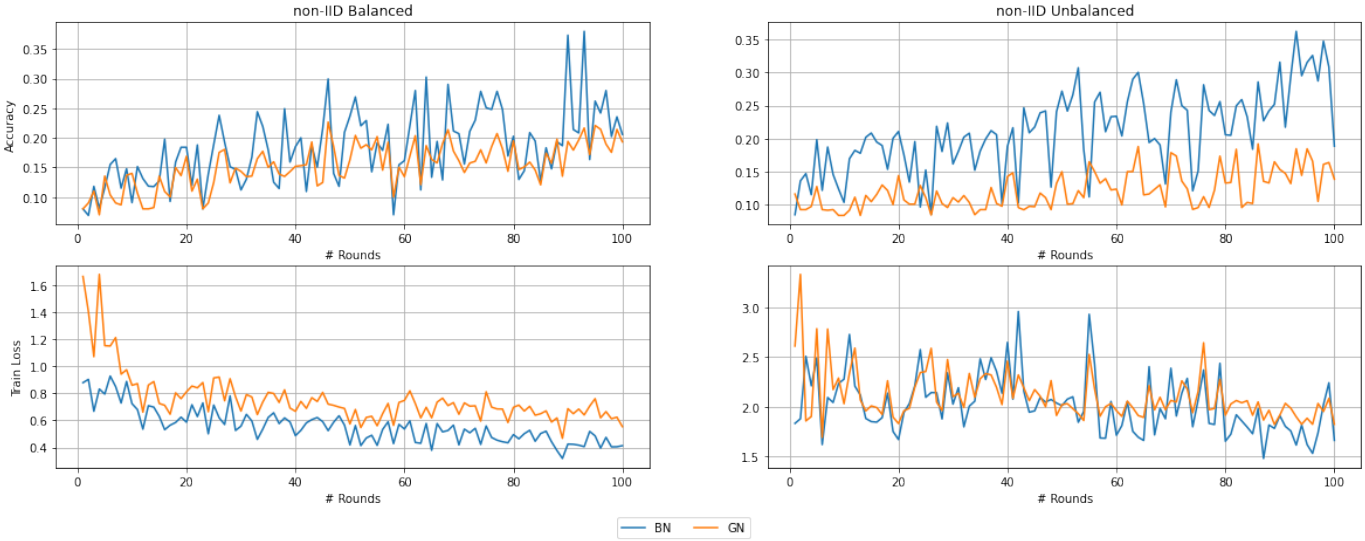


Fig. 6: FedPROX Training with Batch Normalization and Group Normalization

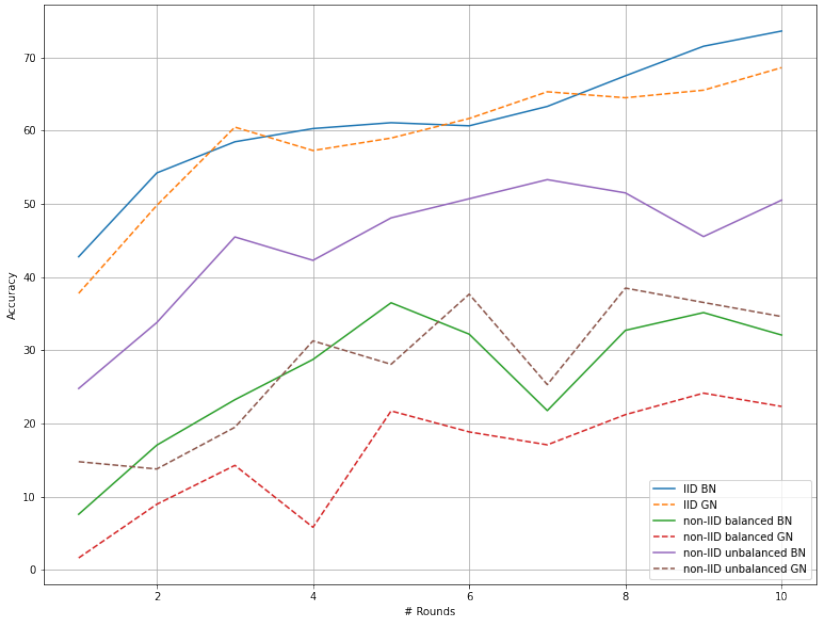


Fig. 7: FedGKT accuracy

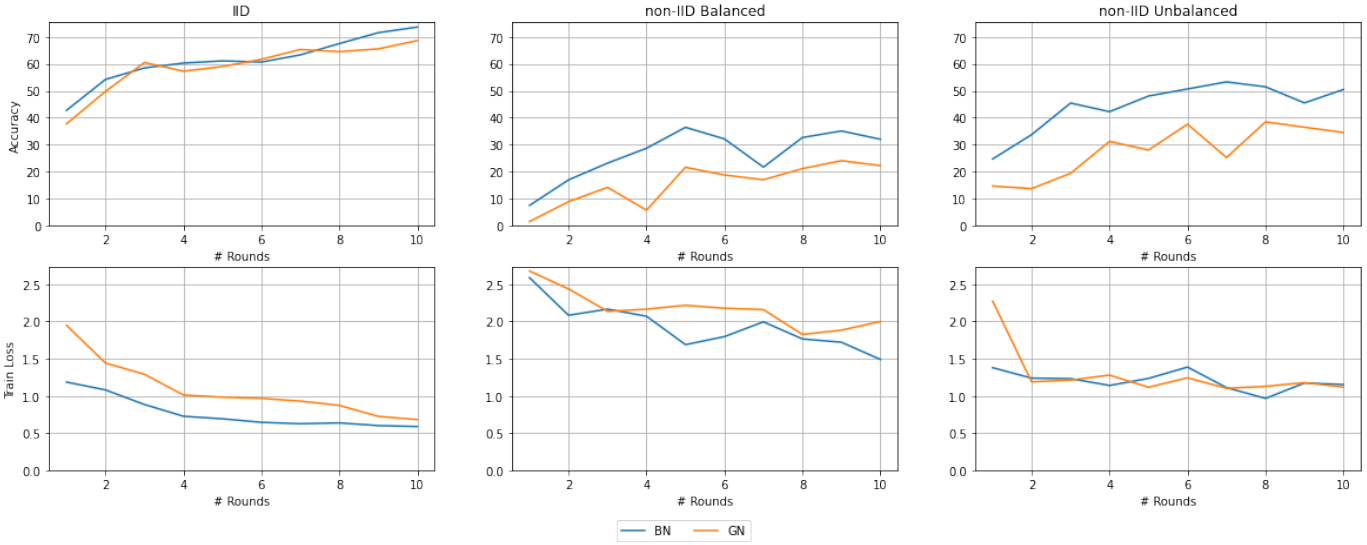


Fig. 8: FedGKT Training with Batch Normalization and Group Normalization