EXPLORING THE IMPACT OF IID-NESS ON VARIOUS FL APPROACHES AT DIFFERENT STAGES OF CONVERGENCE

Anonymous authors
Paper under double-blind review

1 INTRODUCTION

Federated learning, characterized by decentralized model training across various distributed data sources, relies on aggregating insights from diverse datasets. In the realm of federated learning, a significant aspect requiring scholarly attention is comprehending data heterogeneity. Understanding this heterogeneity is pivotal to the success of the paradigm. The phrase "data heterogeneity" encompasses the diverse and disparate nature of datasets that originate from various sources and display differences in format, scale, distribution, and inherent features. Data heterogeneity is an acknowledged cause of the performance degradation of the global mode. Adapting to a global model may not be the optimal solution for the client's task. In the field of Federated Learning, different algorithms and strategies have addressed the issue of data heterogeneity. Nevertheless, there is a lack of comparison between the performance of various algorithms in varying degrees of data heterogeneity. In this study, we examine and compare the effects of varying degrees of data heterogeneity on model accuracy during diverse training phases and utilizing different algorithms. These empirical findings will aid in the advancement of effective scheduling techniques that can incorporate diverse federated learning algorithms at different stages of training within a federated learning framework. Studying the impact of adversarial attacks is crucial as it involves injecting malicious and diverse data into a computer network. It aids in devising effective defense mechanisms after a thorough examination of the severity of these attacks. We conducted several experiments by running different amounts of heterogeneity in the training data under different settings, despite the limitations in our work, we uncovered immense potential in this analysis.

2 RELATED WORKS

In federated learning, distributed clients engage with a central server to cooperatively train a solitary model without disclosing their data directly. McMahan et al. (2017). Despite the great success in homogeneous federated learning, where it heavily relies on the assumption that all the participants share the same network structure and possess similar data distributions Li et al. (2020). However, in practical large-scale situations, there may be considerable differences between data distributions, model structures, communication networks, and system edge devices, which make it challenging to realize federated collaboration. In our work, our primary interests are focused on statistical heterogeneity. Several studies Li et al. (2020); Karimireddy et al. (2020); Li et al. (2022) suggest that the local optimization goals of clients do not align with the global optimization goal because of differences in the local data distribution. Statistical heterogeneity can result in local models converging in different directions, reaching local optima instead of global optima, and ultimately reducing the effectiveness of federated learning.

Prior research has explored statistical heterogeneity in various manners. Many algorithms have been created in federated learning to handle diverse data, yet there lacks a systemic viewpoint and quantifiable methodology to tackle degraded performance. FedSGD and FedAvg McMahan et al. (2017) are two well-recognized algorithms that provide a solid comparison of how to balance convergence rate and handling heterogeneous data. Many algorithms have been developed for federated learning to manage heterogeneity in data. One noticeable work named FedChain Hou et al. (2021), proposed to select whether to adapt to a more local updated model in mid round, which provides higher convergence rate depending on how we adapt the model. Nevertheless, a systemic approach and quantifiable method to address performance degradation are lacking. FedSGD and FedAvg, two

widely recognized algorithms, present a solid comparison of how to balance convergence rate and handling data heterogeneity. In theory Hou et al. (2021), the FedAvg algorithm proposes that clients should take multiple optimization steps over their local data before communicating with the server. This allows for the local update to exploit the similarity between the clients' data, but results in a slower convergence rate. On the other hand, FedSGD has clients simply return a gradient vector in each round, resulting in faster coverage but failing to exploit the similarity between clients, even when they are homogeneous. In advance, the development of FedAdaGrad, FedAdam, and FedYogi Reddi et al. (2021) extended the convergence rate and communication efficiency, clients perform multiple epochs of training using a client optimizer to minimize loss on their local data, and server updates its global model by applying a gradient-based server optimizer to the average of the clients' model updates.

3 METHODS

Our report is mainly an empirical analysis on the behavior of FL. We analyze the impacts of IID-ness in different environments and compare the behavior of FL approaches under certain IID circumstances. The analyses we explore include:

1. Performance of **FedAvg** McMahan et al. (2017) starting at models pre-trained to different accuracies with data organized with different IID-ness.
2. Comparison between performances **FedAvg**, **FedSGD** McMahan et al. (2017), and adaptive FL approaches including **FedAdaGrad**, **FedYogi**, and **FedAdam** Reddi et al. (2021) under different data circumstances.
3. Comparison of performances under other heterogeneous situations (clients have different number of data points, IID clients mixed with non-IID clients, etc.).

Federated Learning Simulation Used local desktop to simulate the federated learning environment. "Server" is run in the main Python process and each "client" is simulated as a function and "locally" trained sequentially. The trained weight is returned by the "Client" function and aggregated in the main process.
At each round, select 10 clients out of 100 clients in total. Each client is trained with its local data for 5 epochs, client learning rate is 0.1 (0.001 for the first experiment in Figure 2) and batch size is 64 (except for FedSGD where local epoch is 1 and batch size is ∞). The local optimizer is always SGD with momentum, with momentum = 0.9, and weight decay = 0.0001. The number of rounds is 50. For adaptive FL approaches, server learning rate is 0.1 when $\beta < 1$ else 0.01, and $\beta_1 = 0.9, \beta_2 = 0.99$. The reason why I chose server learning rate this way is through empirical tuning.

Model Setup To obtain models at different global accuracies, I pretrained the models and saved them at accuracies (0.0906, 0.2227, 0.2916, 0.3827, 0.4923, 0.5908, 0.6873, 0.797, 0.8917). As for the **Neural Network**, we used an MLP with two hidden layers. The hidden sizes are 512 and 128. Each is followed by a ReLU activation function and Batch Normalization.

Label Heterogeneity Gaining hints from Kim et al. (2023), we use **Dirichlet** distribution $Dir(\beta_1, \ldots, \beta_C)$, where C is the number of classes, to the dataset such that we organize data with a controlled IID-ness. Here, we are using the same β for each class, so it is actually $Dir(\beta)$. The smaller the β, the more non-IID the dataset is on each client. β is of 10^k with k from -3 (typically one label occupies the whole client) to 4 (even numbers of classes on each client). For each client, we compute the ratio of each label on this client by $Dir(\beta)$ and then organize data according to these ratios.

Device Heterogeneity We explored the device heterogeneity roughly by making half of the clients having only one tenth of the number of data points on the second half of the clients. This roughly simulates the situation where each client device has different number of data points (images).

Dataset We are using **FashionMNIST** Xiao et al. (2017) to do experiments. It consists of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 gray-scale image and is associated with a label from 10 classes.

4 EMPIRICAL ANALYSIS

Performance under Varying IID-ness At the beginning, we used FedAvg to learn FashionMNIST on model pretrained to different accuracies. The Figure 2 have complete graphs for the training results. We can observe that no matter what accuracy the model was pretrained to, the model goes to a similar range of accuracy at the end under different IID-ness. One meaningful takeaway of this is that even if the model is trained to a high accuracy, very non-IID circumstance eventually drags its accuracy down, making it forget what it learned before.
When $\beta = 0.001, 0.01$, the data is too heterogeneous on clients and thus show noticeably worse performance than other circumstances. When $\beta \geq 1$, the behaviors are nearly indistinguishable. $\beta = 0.1$ is a special condition (an example distribution is is shown in Figure 1a). We can see that this is still very non-iid, but the performance is noticebly better than $\beta = 0.001, 0.01$, and its improvement in the first two rounds when the pretrained accuracy is very low coincides with $\beta \geq 1$. When the model is pretrained to a high accuracy, training on $\beta = 0.1$ won't degrade the performance much. Thus, we could deduce that when label heterogeneity is of $Dir(0.1)$ level, it is possible for user to use FedAvg in first several rounds to gain a fast improvement at the beginning and then switch to better FL approaches.

Performance Comparison between Different Aggregations We measure the performance of different aggregation methods on model pretrained to different accuracies under different data heterogeneities. The graphs are in Figure 3. We observe that: (1) FedSGD is never worse than FedAvg and always better than FedAvg when data heterogeneity is high. (2) Adaptive FL methods are smooth, and degrade/improve less rapidly than FedSGD or FedAvg, and thus does better when data is very non-iid and the model has been pretrained to higher accuracies. (3) The trend of changes in performances are similar for all aggregations, difference mainly lies in rapidness. (4) When data heterogeneity is low, those adaptive methods might not be a good option as they typically work worse than FedSGD and does not gain advantages over FedAvg.

Performance under Varying Device Heterogeneity We also measure the performance of FedAvg under varying device heterogeneities (roughly, though). Graphs shown in Figure 4. In this experiment, the performance of both cases are similar, while the case when all clients have full data is always very slightly better than the case where half of the clients get only one tenth of the data. We could deduce that the influence of device heterogeneity is less significant than label heterogeneity.

Performance when IID Mixed with Non-IID Given the poor performance of FedAvg under extremely non-iid situations and the smoother curves of adaptive FL approaches, we further try the situation where in each round the certain amount of clients have IID data ($\beta = 10,000$) and the rest of the selected clients have completely non-IID data ($\beta = 0.001$). Graphs are in Figure 5.
We can observe that this mixation makes the FedAvg oscillates even more rapidly while FedAdam stays smooth. The ratios of IID clients we tried are (0.2, 0.4, 0.6, 0.8). When the ratio is 0.2, though performance of FedAvg is generally going down, the performance of FedAdam is going up and converging to about 80% accuracy. When the ratio is 0.4, though the FedAvg is still very unstable, it is generally improving. When ratio is 0.6, FedAvg starts to outperform FedAdam. When ratio is 0.8, FedAvg is much more stable and converging to 80% accuracy. This observation is very useful as it provides a condition of very non-IID circumstance in which adaptive FL approaches like FedAdam could still give reasonable performances.

5 LIMITATIONS/FUTURE WORK

Environment Setup Issue In this paper, we have solely examined the datasets of FMNIST and Cifar10, along with the federated learning algorithms of FedAvg, FedSGD, FedAdaGrad,

FedAdam, and FedYogi. The original goal was to explore the algorithm that was targeted to solve data heterogeneity in order to form a solid comparison. However, due to the lack of time and computational power, we can only explore this many strategies. In the future, our goal should be to explore various datasets and algorithms that aim to address the data heterogeneity issue.
Furthermore, our setup may yield inaccurate results due to partitioning the data and sampling client data from the same partition. To produce non-iidness, we may have oversampled a particular data point in the partitioned dataset, creating a significant issue during model training. Given more time, we aim to reconstruct synthetic non-iid data to achieve fairness in experiment set up.

Defining Data Heterogeneity While the Dirichlet distribution seems like a reasonable way to partition a global dataset into non-iid data. There are still different methods that redefine the notion of data heterogeneity and empirical way to split data into non-id data. Also, our definition of **data heterogeneity** is limited. Heterogeneity can also be caused by features in images. One example is when device owners took pictures in their homes. The background would become one source of data heterogeneity. In the future, exploring impact of such data heterogeneity at different stages of convergence can be a future direction. Additionally, we should also look more into methods that synthesize non-iid data and resemble the characteristics of real-life dataset in federated environment.

Framework Working under Extremely Non-IID circumstance According to my experiment observations for "**Performance when IID Mixed with Non-IID**", a framework robust to extremely non-IID circumstances is also possible. During the process of training, the server could utilize some techniques (or simply look at the global accuracy) to check if it is learning in an extremely non-IID circumstance. If so, the developer could deploy some stablizers (fake clients of their choice and have very IID datas) and apply adaptive FL methods like FedAdam Reddi et al. (2021) to obtain reasonable performances. If possible, developers could even add more "stablizers" to allow FedAvg to give reasonable performances (note that FedAvg starts to improve at ratio = 0.4) for its smaller computation costs at aggregation.

Adversarial Attack by Client-Level Poisoning According to my experiment observations for "**Performance when IID Mixed with Non-IID**", adversarial attacks by inserting extremely non-IID clients into client pool is also possible. Though according to Figure 5, inserting two clients with extremely non-IID data at each round (which is already strong assumption) does not deteriorate FedAvg too much, research could be done on this to amplify the effects.

6 ACKNOWLEDGEMENT

Thanks to Jong-Ik Park's code for Discussion 1. We built our work on top of his codes, which gave us a basic sight on how to simulate FL on our desktop.

REFERENCES

Charlie Hou et al. Fedchain: Chained algorithms for near-optimal communication cost in federated learning. 2021.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5132–5143. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/karimireddy20a.html.

Yeachan Kim, Junho Kim, Wing-Lam Mok, Jun-Hyung Park, and SangKeun Lee. Client-customized adaptation for parameter-efficient federated learning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 1159–1172, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.75. URL https://aclanthology.org/2023.findings-acl.75.

Qinbin Li et al. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2022.

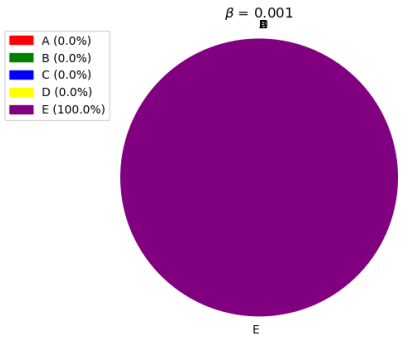
T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated learning: Challenges, methods, and future directions. volume 37, pp. 50–60, May 2020. doi: 10.1109/MSP.2020.2975749.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 20–22 Apr 2017. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

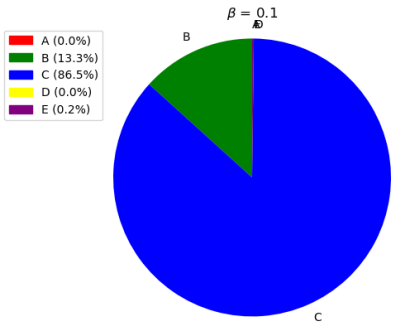
Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=LkFG3lB13U5.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

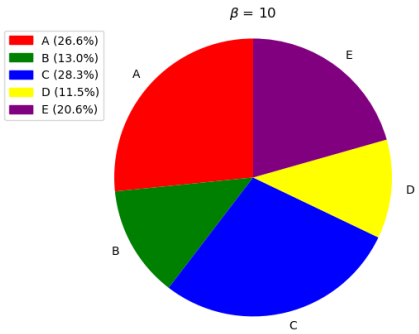
A Appendix



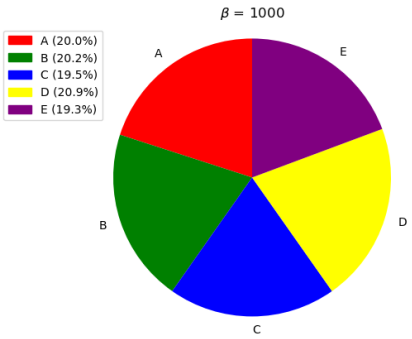
(a) $\beta = 0.001$



(b) $\beta = 0.1$

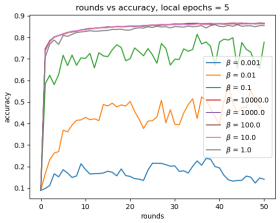


(c) $\beta = 10$

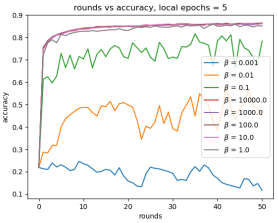


(d) $\beta = 10,000$

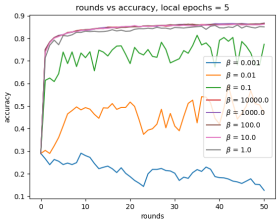
Figure 1: A plot interpretation of Dirichlet Distribution with different values of β



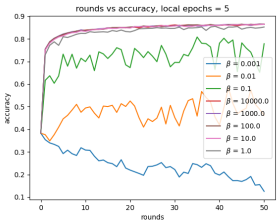
(a) pretrained accuracy = 9.06%



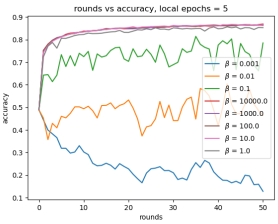
(b) pretrained accuracy = 22.27%



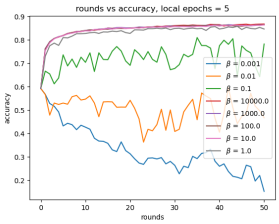
(c) pretrained accuracy = 29.16%



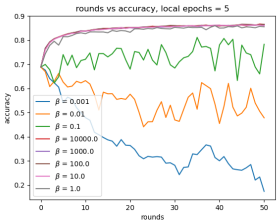
(d) pretrained accuracy = 38.27%



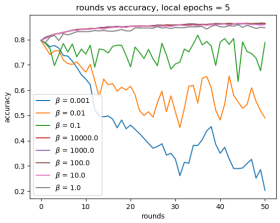
(e) pretrained accuracy = 49.23%



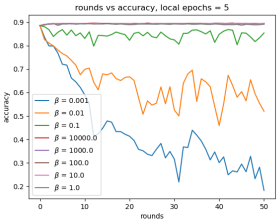
(f) pretrained accuracy = 59.08%



(g) pretrained accuracy = 68.73%

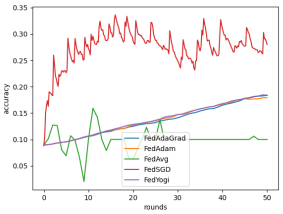


(h) pretrained accuracy = 79.7%

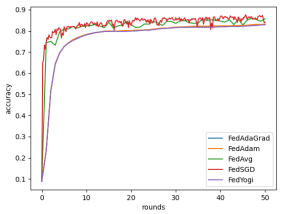


(i) pretrained accuracy = 89.17%

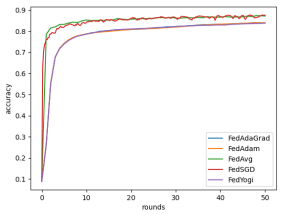
Figure 2: Accuracy of FedAvg from models pretrained to different accuracies under data of different IID-ness



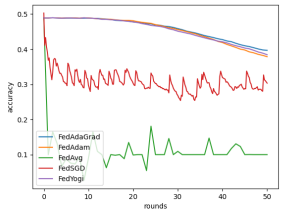
(a) pretrained accuracy = 9.06%, $\beta = 0.001$



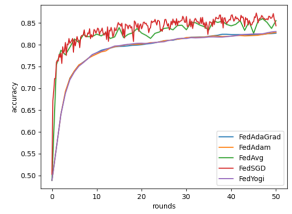
(b) pretrained accuracy = 9.06%, $\beta = 1$



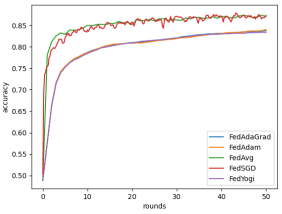
(c) pretrained accuracy = 9.06%, $\beta = 10,000$



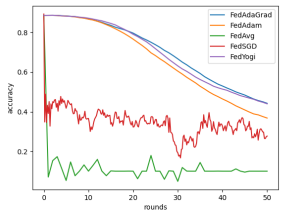
(d) pretrained accuracy = 49.23%, $\beta = 0.001$



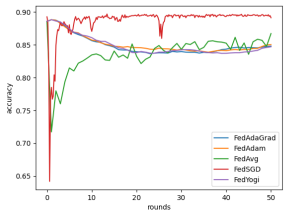
(e) pretrained accuracy = 49.23%, $\beta = 1$



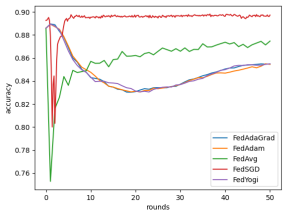
(f) pretrained accuracy = 49.23%, $\beta = 10,000$



(g) pretrained accuracy = 89.17%, $\beta = 0.001$

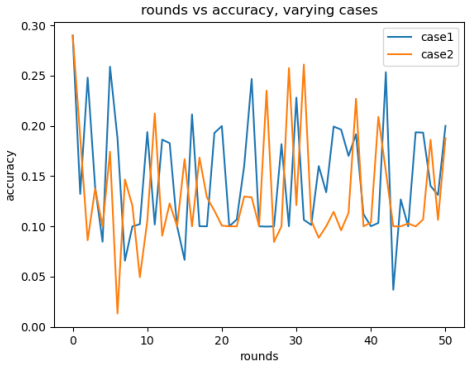


(h) pretrained accuracy = 89.17%, $\beta = 1$

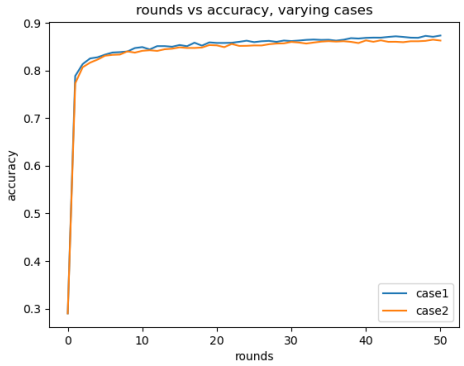


(i) pretrained accuracy = 89.17%, $\beta = 10,000$

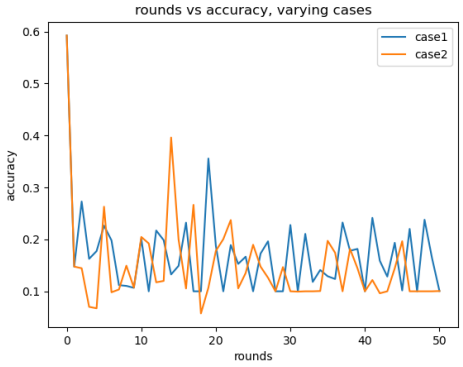
Figure 3: Accuracy comparison between aggregation methods from models pretrained to different accuracies under data of different IID-ness



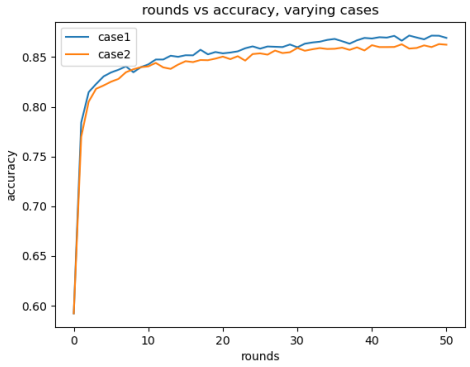
(a) pretrained accuracy = 29.16%, $\beta = 0.01$



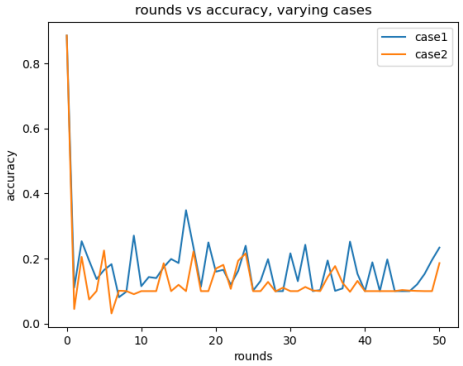
(b) pretrained accuracy = 29.16%, $\beta = 10,000$



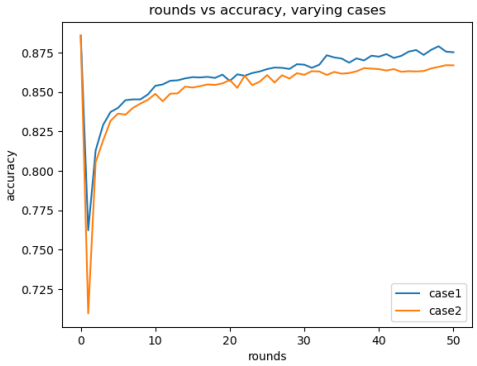
(c) pretrained accuracy = 59.08%, $\beta = 0.01$



(d) pretrained accuracy = 59.08%, $\beta = 10,000$

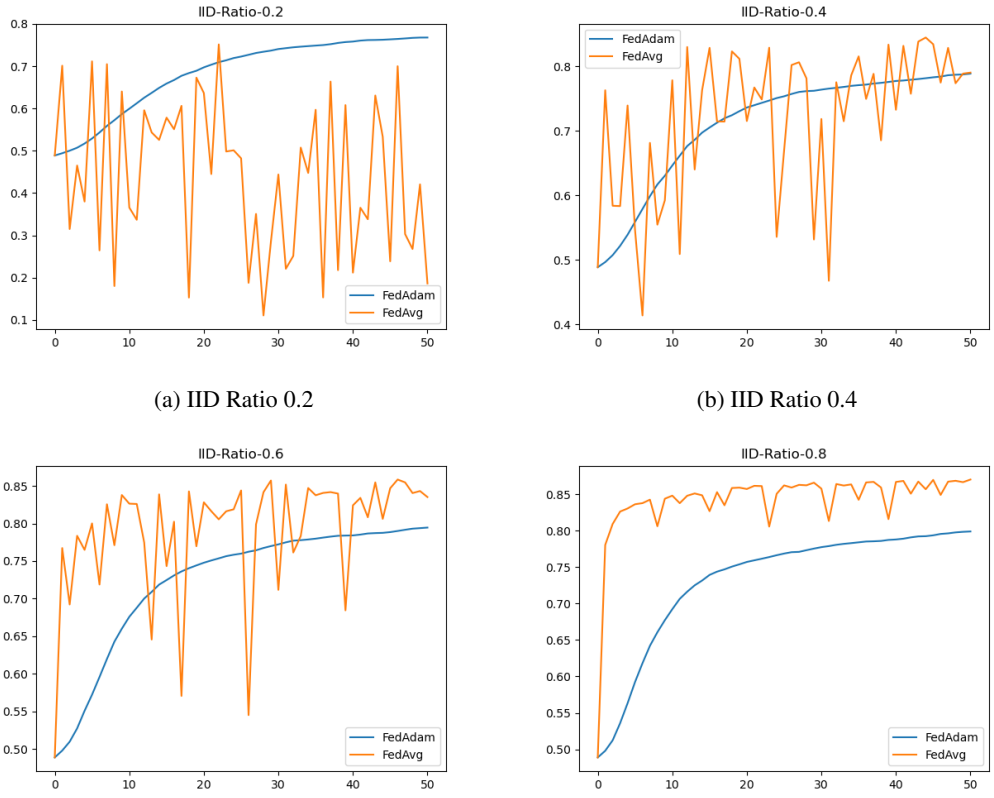


(e) pretrained accuracy = 89.17%, $\beta = 0.01$



(f) pretrained accuracy = 89.17%, $\beta = 10,000$

Figure 4: Comparison of two cases of learning from models pretrained to different accuracies under data of different IID-ness. Case 1 is when all clients have full data. Case 2 is when half of the clients have only one tenth of data. Use FedAvg as aggregation



(a) IID Ratio 0.2

(b) IID Ratio 0.4

(c) IID Ratio 0.6

(d) IID Ratio 0.8

Figure 5: Comparison of different ratios of IID ($\beta = 10,000$) clients among non-IID ($\beta = 0.001$) clients. Models were pre-trained to 49.23% accuracy.