# Supplementary Material for FedCorr: Multi-Stage Federated Learning for Label Noise Correction

Jingyi Xu[1*] Zihan Chen[1,2*] Tony Q.S. Quek[1] Kai Fong Ernest Chong[1†]
[1]Singapore University of Technology and Design [2]National University of Singapore
{jinyi_xu,zihan_chen}@mymail.sutd.edu.sg {tonyquek,ernest_chong}@sutd.edu.sg

## 1. Outline

As part of supplementary material for our paper titled "FedCorr: Multi-Stage Federated Learning for Label Noise Correction", we provide further details, organized into the following sections:

- Sec. 2 introduces the implementation details for our method and baselines.
- Sec. 3 provides further details on our experiments.
  - Sec. 3.1 gives additional experiment results on CIFAR-100 with a non-IID data partition.
  - Sec. 3.2 shows that FedCorr is model-agnostic, via a comparison of the test accuracies and the distributions of cumulative LID scores, using different model architectures.
  - Sec. 3.3 gives a comparison of the communication efficiency of different methods.
  - Sec. 3.4 explains why cumulative LID scores are preferred over LID scores for identifying noisy clients.
  - Sec. 3.5 demonstrates the effectiveness of both the label noise identification and the label correction process in FedCorr.
  - Sec. 3.6 gives further details on the ablation study results for FedCorr.
  - Sec. 3.7 provides further intuition on the non-IID data settings used in our experiments, via explicit illustrations of the corresponding non-IID data partitions on CIFAR-10, over 100 clients.
- Sec. 4 discusses the potential negative societal impact of FedCorr.

## 2. Implementation details

All experiments were implemented using Pytorch. Among the baselines, we reimplemented FedAvg [8], FedProx [7], JointOpt [10], DivideMix [6] and PoC [2], and we used the official implementations of FedDyn [1] and ARFL [3]. For RoFL[1] and Median[2], we used their unofficial implementations. For all methods, we use an SGD local optimizer with a momentum of 0.5 and no weight decay, with a batch size of 10 for CIFAR-10/100 and 16 for Clothing1M. Note that at each noise level, we used the same training hyperparameters for both IID and non-IID data partitions.

| Hyperparameters | CIFAR-10 | CIFAR-100 | Clothing1M |
|---|---|---|---|
| # of iterations in stage 1, $T_1$ | 5 | 10 | 2 |
| # of rounds in stage 2, $T_2$ | 500 | 450 | 50 |
| # of rounds in stage 3, $T_3$ | 450 | 450 | 50 |
| Confidence threshold, $\theta$ | 0.5 | 0.5 | 0.9 |
| Relabel ratio, $\pi$ | 0.5 | 0.5 | 0.8 |
| Learning rate | 0.03 | 0.01 | 0.001 |

Table 1. Hyperparameters of FedCorr on different datasets.

For the implementation of each federated learning (FL) method, we define its *total communication cost* to be the cumulative number of clients that participate in training. For example, if a client participates in 10 communication rounds, then that client would contribute 10 to the total communication cost. For every method except JointOpt and DivideMix, we always reimplement the method using 5 local epochs per communication round, and the same total communication cost for each dataset, which corresponds to 1000 rounds of FedAvg for CIFAR-10/100 with fraction 0.1, and corresponds to 200 rounds of FedAvg for Clothing1M with fraction 0.02. Settings for JointOpt and DivideMix are discussed below.

In the rest of this section, we give full details on all remaining hyperparameters used for each method. For baseline methods, we also provide brief descriptions of their main underlying ideas.

- FedCorr. We fixed $k = 20$ for LID estimation, $\alpha = 1$ for mixup, and $\beta = 5$ for the proximal regularization term in all reported experiments. All re-

*Equal contributions. † Corresponding author.

[1]https://github.com/jangsoohyuk/Robust-Federated-Learning-with-Noisy-Labels

[2]https://github.com/fushuhao6/Attack-Resistant-Federated-Learning

maining hyperparameters can be found in Tab. 1. Note that the total communication cost for FedCorr is the same as for other baselines. Take CIFAR-10 for example: In each iteration of the pre-processing stage of FedCorr, every client participates exactly once. In contrast, in each communication round of our other baselines, only a fraction of 0.1 of the clients would participate. Hence, one iteration in the pre-processing stage of FedCorr has 10 times the total communication cost of one communication round of the other baselines. For the latter two stages of FedCorr, we used the usual 0.1 as our fraction. Hence the total communication cost of the entire implementation of FedCorr equals $100T_1 + 10T_2 + 10T_3 = 10000$; this is the same total communication cost for implementing FedAvg over 1000 communication rounds with fraction 0.1.

- JointOpt [10] is one of the state-of-the-art centralized methods for tackling label noise, which alternately updates network parameters and corrects labels using the model prediction vectors. It introduced $\alpha_{Jo}$ and $\beta_{Jo}$ as two additional hyperparameters. In the centralized setting, we used the hyperparameters given in Tab. 2. In particular, we considered a total of seven noise settings, which we have divided into two groups: low noise levels (first four settings) and high noise levels (last three settings). Within each group, we used the same hyperparameters. Note that the hyperparameters are not exactly the same as those given in [10], as we used different architectures and different frameworks to generate synthetic label noise. In the federated setting, we used $\alpha_{Jo} = 1.2$, $\beta_{Jo} = 0.8$ and a learning rate of 0.01 for CIFAR-10/100. To boost performance, we used a warm-up process for CIFAR-10/100: We first trained using FedAvg over 20 communication rounds with 5 local epochs per communication round, after which we started using JointOpt for local training over 80 communication rounds with 20 local epochs per communication round. For Clothing1M, we used $\alpha_{Jo} = 1.2$, $\beta_{Jo} = 0.8$, and a learning rate of 0.001. As we used a ResNet-50 that is already pretrained on ImageNet, no warm-up process was used for our Clothing1M experiments. We trained using JointOpt over 40 communication rounds with 10 local epochs per round.
- DivideMix [6] is another state-of-the-art centralized method, which dynamically divides the training data into labeled (clean) and unlabeled (noisy) data, and trains the model in a semi-supervised manner. For CIFAR-10/100, we used the same two groups of noise settings, as described in the above configuration for JointOpt. The only hyperparameter we tuned is $\lambda_{Div}$, which is a hyperparameter specific to DivideMix. For low noise levels, we used $\lambda_{Div} = 0$ (resp. $\lambda_{Div} = 25$) for CIFAR-10 (resp. CIFAR-100). For high noise levels, we used $\lambda_{Div} = 25$ (resp. $\lambda_{Div} = 150$) for CIFAR-10 (resp. CIFAR-100). For all other hyperparameters for CIFAR-10/100, we used the values given in [6]. For Clothing1M, we use $\lambda_{Div} = 25$ and a learning rate of 0.01; for all other hyperparameters, we used the values given in [6]. We used the same warm-up process for CIFAR-10/100, and we used the same number of communication rounds and number of local epochs for all datasets, as described above in our configuration for JointOpt.
- FedAvg [8] is the first algorithm that introduced the idea of federated learning. We used a learning rate of 0.03, 0.01 and 0.003 for CIFAR-10, CIFAR-100 and Clothing1M, respectively.
- FedProx [7] was proposed to tackle data heterogeneity among clients by adding a fixed proximal term with coefficient $\mu_{prox}$ to every local loss function. We used $\mu_{prox} = 1$ for all experiments, and a learning rate of 0.01 and 0.003 for CIFAR-10/100 and Clothing1M, respectively.
- RoFL [11] is, to the best of our knowledge, the only method that is designed for label correction in FL. It is based on the idea of exchanging feature centroids between the server and clients, and it introduced $T_{pl}$ as an additional hyperparameter to control label correction. We set $T_{pl}$ to 100, 400 and 10 for CIFAR-10, CIFAR-100 and Clothing1M, respectively. All other hyperparameters are set to the same values as given in [11].
- ARFL [3] is a robust aggregation algorithm that resists abnormal attacks via residual-based reweighting, using two hyperparameters $\lambda_{ar}$ and threshold $\delta_{ar}$. We used $\lambda_{ar} = 2$ and $\delta_{ar} = 0.1$ for all experiments. We used a learning rate of 0.01 and 0.003 for CIFAR-10/100 and Clothing1M, respectively.
- FedDyn [1] proposed a dynamic regularizer, with coefficient $\alpha_{Dyn}$, for local optimization in each communication round, so as to tackle the inconsistency between the local and global empirical loss. We used $\alpha_{Dyn} = 0.01$, a learning rate of 0.1 with a decay of 0.998 for all the experiments.

| $(\rho, \tau)$ | $\alpha_{Jo}$ | $\beta_{Jo}$ | Learning rate |
|---|---|---|---|
| Low noise levels<br>$(0.0, 0.0), (0.4, 0.0), (0.4, 0.5), (0.6, 0.0)$ | 1.2 | 1.5 | 0.1 |
| High noise levels<br>$(0.6, 0.5), (0.8, 0.0), (0.8, 0.5)$ | 1.2 | 0.8 | 0.2 |

Table 2. Hyperparameters of JointOpt in the centralized setting on CIFAR-10/100.

- `Median` [12] is an aggregation method for robust distributed learning, whereby the notion of "average" in `FedAvg` is changed from "mean" to "median". For all experiments, we used a learning rate of 0.01; all other hyperparameters are the same as given in `FedAvg`.
- `Poc` [2] is a client selection algorithm that is biased towards clients with higher local losses within a given client pool. We used a learning rate of 0.01 and a client pool size of $d = 30$ for all experiments.

# 3. Details on experiment results

## 3.1. CIFAR-100 with non-IID data partition

In terms of robustness to the discrepancies in both local label quality and local data statistics, `FedCorr` significantly outperforms the baselines. In the main paper, we have reported the outperformance of `FedCorr` on CIFAR-100 with IID data partition. To further show the outperformance on non-IID data partitions, we also conducted experiments on CIFAR-100 with noise model $(\rho, \tau) = (0.4, 0.5)$ and non-IID hyperparameter $(p, \alpha_{Dir}) = (0.7, 10)$; here, we report our results in Tab. 3. We observe that `FedCorr` achieves an improvement in best test accuracy of at least 7% over our baselines.

| Method\$(p, \alpha_{Dir})$ | $(0.7, 10)$ |
|---|---|
| FedAvg | 64.75±1.75 |
| FedProx | 65.72±1.30 |
| RoFL | 59.31±4.14 |
| ARFL | 48.03±4.39 |
| JointOpt | 59.84±1.99 |
| DivideMix | 39.76±1.18 |
| Ours | **72.73±1.02** |

Table 3. Average (5 trials) and standard deviation of the best test accuracies of different methods on CIFAR-100 with non-IID data partition. The noise setting used is $(\rho, \tau) = (0.4, 0)$.

## 3.2. Comparison of different architectures

To demonstrate that our proposed `FedCorr` is model-agnostic, especially with respect to the noisy client identification scheme via cumulative LID scores, we conducted experiments on CIFAR-10 with IID data partition using different architectures: ResNet-18 [4], VGG-11 [9] and LeNet-5 [5]. Tab. 4 shows the best test accuracies of each model trained on CIFAR-10 with various levels of synthetic noise. For experiments on VGG-11, we used hyperparameters with the same values as used in the experiments on ResNet-18. For LeNet-5, we only tuned the learning rate and fixed it at 0.003 in all experiments. Fig. 1 shows a further comparison between different architectures in terms of the distribution of the cumulative LID scores and the corresponding separations of the clients via Gaussian Mixture Models.

## 3.3. Comparison of communication efficiency

In this subsection, we discuss the communication efficiency of different methods. Here, given any implementation of an FL method, and any desired target accuracy $\zeta$, we define its *targeted communication cost for $\zeta$ test accuracy* to be the lowest total communication cost required (in the experiments) to reach the target $\zeta$ test accuracy. Informally, the lower the targeted communication cost, the higher the communication efficiency.

Tab. 5 and Tab. 6 show the comparison of the communication efficiency on CIFAR-10, in terms of the targeted communication cost at test accuracies $\zeta = 80\%$ and $\zeta = 65\%$, respectively. Tab. 7 shows the comparison on CIFAR-100, in terms of the targeted communication cost at test accuracy $\zeta = 50\%$. As our results show, `FedCorr` achieves improvements in communication efficiency, by a factor of at least 1.9 on CIFAR-10, and at least 1.3 on CIFAR-100.

## 3.4. Distribution of cumulative LID scores

Fig. 2 shows the comparison between the distribution of the LID scores and the distribution of the cumulative LID scores, after each iteration in the pre-processing stage. The LID scores of clean clients and noisy clients can be well-separated after the second iteration and the third iteration. This is also true for the cumulative LID scores. However, after the fourth iteration, the LID scores of noisy clients and clean clients start overlapping, while in contrast, the cumulative LID scores of noisy clients and clean clients remain well-separated. As already discussed in the main paper, cumulative LID scores have a stronger linear relation with local noise levels, as compared to LID scores. Hence, the cumulative LID score is a more robust metric for identifying noisy clients.

## 3.5. Evaluation of label noise identification and label correction

Fig. 3 demonstrates the effectiveness of the label noise identification and correction process in the pre-processing stage on CIFAR-10. Note that in Fig. 3, we used the noise setting $(\rho, \tau) = (0.6, 0.5)$, which means on average 60% of the clients are randomly selected for the addition of synthetic noise to their local datasets before training, whereby the local noise level for each selected client is at least 0.5. The top plot in Fig. 3 shows the estimated noise levels, in comparison with the ground-truth noise levels (before training and after stage 1), across all 100 clients. In particular, the huge gap between the ground-truth noise levels before training (blue dotted line) and after stage 1 (or-

| Method | Best Test Accuracy (%) ± Standard Deviation (%) | | | | | | |
|---|---|---|---|---|---|---|---|
| | $\rho = 0.0$ | $\rho = 0.4$ | | $\rho = 0.6$ | | $\rho = 0.8$ | |
| | $\tau = 0.0$ | $\tau = 0.0$ | $\tau = 0.5$ | $\tau = 0.0$ | $\tau = 0.5$ | $\tau = 0.0$ | $\tau = 0.5$ |
| ResNet-18 | 93.82±0.41 | 94.01±0.22 | 94.15±0.18 | 92.93±0.25 | 92.50±0.28 | 91.52±0.50 | 90.59±0.70 |
| VGG-11 | 88.96±0.84 | 87.93±0.41 | 87.53±0.40 | 84.78±1.68 | 84.82±0.79 | 83.34±0.42 | 80.82±2.62 |
| LeNet-5 | 72.03±0.35 | 70.47±0.86 | 70.02±1.39 | 69.09±0.16 | 67.48±0.54 | 67.49±0.74 | 65.16±0.53 |

Table 4. Comparison of the average (5 trials) and standard deviation of best test accuracies, when trained on CIFAR-10 with IID data partition using different architectures.

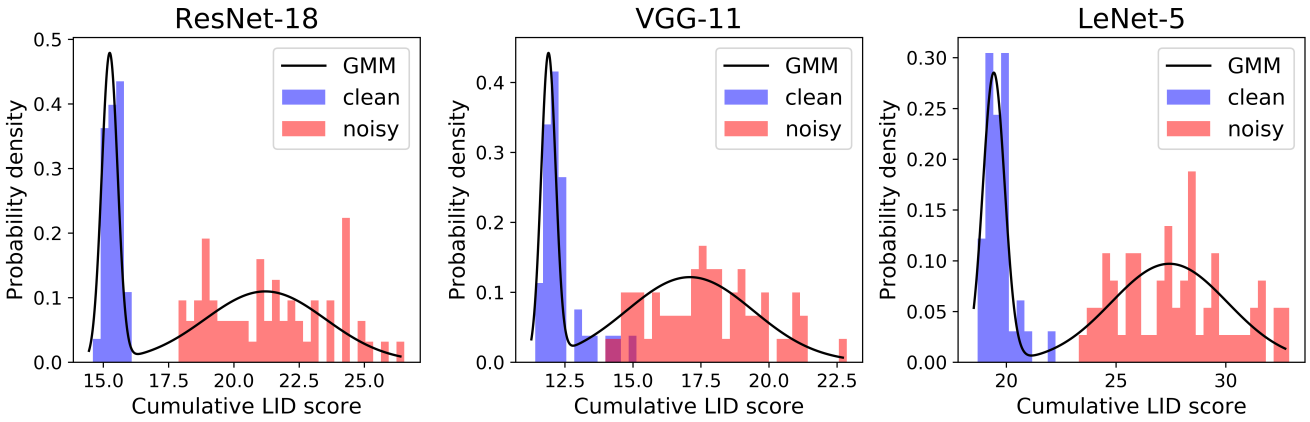


Figure 1. Comparison of cumulative LID score distribution after 5 iterations in pre-processing stage among different architectures. The experiments were conducted on CIFAR-10 with IID data partition and noise setting $(\rho, \tau) = (0.6, 0.5)$.

| Method | $\rho = 0.0$ | $\rho = 0.4$ | | $\rho = 0.6$ | | $\rho = 0.8$ | |
|---|---|---|---|---|---|---|---|
| | $\tau = 0.0$ | $\tau = 0.0$ | $\tau = 0.5$ | $\tau = 0.0$ | $\tau = 0.5$ | $\tau = 0.0$ | $\tau = 0.5$ |
| Ours | 150 | 210 | 230 | 230 | 330 | 360 | 510 |
| FedAvg | 370(2.6×) | 450(2.1×) | 470(2.0×) | 550(2.4×) | 930(2.8×) | 810(2.3×) | - |
| FedProx | 690(4.9×) | 1050(5.0×) | 1190(5.2×) | 1230(5.3×) | 1600(4.8×) | 1730(4.8×) | 4640(9.1×) |
| RoFL | 990(7.1×) | 1390(6.6×) | 1580(6.9×) | 1900(8.3×) | 4200(12.7×) | 2080(5.8×) | - |
| ARFL | 290(2.1×) | 740(3.5×) | 1180(5.1×) | - | - | - | - |
| JointOpt | 330(2.4×) | 420(2.0×) | 760(3.3×) | 550(2.4×) | - | - | - |
| DivideMix | - | - | - | - | - | - | - |

Table 5. A comparison of communication efficiency for different methods on CIFAR-10 with IID data partition, in terms of the targeted communication cost at $\zeta = 80\%$ test accuracy. Values in brackets represent the ratios of the targeted communication costs as compared to our method `FedCorr`. Note that the test accuracies are evaluated after each communication round. In the case of methods and noise settings for which the target test accuracy $\zeta$ is not reached, we indicate '-'.

ange line) represents the effectiveness of our label correction process, while the small gap between the estimated noise levels (green line) and the ground-truth noise levels after stage 1 (orange line) reflects the effectiveness of our local noise level estimation. Note that for clean clients (with zero ground-truth noise levels before training), `FedCorr` is able to estimate their noise levels to be exactly zero in most cases. Consequently, no additional label noise is introduced to these identified clean clients in our label correction process.

The bottom plot in Fig. 3 shows the separation results between noisy and clean samples (via a Gaussian Mixture Model) for each identified noisy client, in terms of true/false positives/negatives.In particular, the small numbers of false

| Method | $\rho = 0.0$ | $\rho = 0.4$ | | $\rho = 0.6$ | | $\rho = 0.8$ | |
|---|---|---|---|---|---|---|---|
| | $\tau = 0.0$ | $\tau = 0.0$ | $\tau = 0.5$ | $\tau = 0.0$ | $\tau = 0.5$ | $\tau = 0.0$ | $\tau = 0.5$ |
| Ours | 50 | 60 | 90 | 70 | 110 | 90 | 190 |
| FedAvg | 160(3.2×) | 200(3.3×) | 210(2.3×) | 230(3.3×) | 300(2.7×) | 270(3.0×) | 470(2.5×) |
| FedProx | 300(6.0×) | 430(7.2×) | 500(5.6×) | 480(6.9×) | 690(6.3×) | 670(7.4×) | 1840(9.7×) |
| RoFL | 350(7.0×) | 420(7.0×) | 470(5.2×) | 440(6.3×) | 670(6.1×) | 490(5.4×) | 1710(9.0×) |
| ARFL | 120(2.4×) | 230(3.8×) | 170(1.9×) | 240(3.4×) | 390(3.5×) | 270(3.0×) | - |
| JointOpt | 160(3.2×) | 200(3.3×) | 220(2.4×) | 220(3.1×) | 250(2.3×) | 250(2.8×) | 860(4.5×) |
| DivideMix | 480(9.6×) | 560(9.3×) | 580(6.4×) | 590(8.4×) | 690(6.3×) | 930(10.3×) | 970(5.1×) |

Table 6. A comparison of communication efficiency for different methods on CIFAR-10 with IID data partition, in terms of the targeted communication cost at $\zeta = 65\%$ test accuracy. Values in brackets represent the ratios of the targeted communication costs as compared to our method `FedCorr`. Note that the test accuracies are evaluated after each communication round. Note that the test accuracies are evaluated after each communication round. In the case of methods and noise settings for which the target test accuracy $\zeta$ is not reached, we indicate '-'.

| Method | $\rho = 0.0$ $\tau = 0.0$ | $\rho = 0.4$ $\tau = 0.5$ | $\rho = 0.6$ $\tau = 0.5$ | $\rho = 0.8$ $\tau = 0.5$ |
|---|---|---|---|---|
| Ours | 95 | 140 | 295 | 505 |
| FedAvg | 135(1.4×) | 210(1.5×) | 420(1.4×) | - |
| FedProx | 465(4.9×) | 705(5.0×) | 1110(3.8×) | 1885(3.7×) |
| RoFL | 380(4.0×) | 2350(16.8×) | 4740(16.1×) | - |
| ARFL | 150(1.6×) | 265(1.9×) | - | - |
| JointOpt | 125(1.3×) | 210(1.5×) | - | - |
| DivideMix | - | - | - | - |

Table 7. A comparison of communication efficiency for different methods on CIFAR-100 with IID data partition, in terms of the targeted communication cost at $\zeta = 50\%$ test accuracy. Values in brackets represent the ratios of the targeted communication costs as compared to our method `FedCorr`. Note that the test accuracies are evaluated after each communication round. Note that the test accuracies are evaluated after each communication round. In the case of methods and noise settings for which the target test accuracy $\zeta$ is not reached, we indicate '-'.

positives across all identified noisy clients imply the effectiveness of `FedCorr` in identifying noisy samples.

To further illustrate the effectiveness of the label correction process, we compared the confusion matrices of the given labels before training, the corrected labels after the pre-processing stage, and the corrected labels after the finetuning stage. Fig. 4 depicts the confusion matrices for the first 5 clients, in the experiments conducted on CIFAR-10 with IID data partition and noise setting $(\rho, \tau) = (0.6, 0.5)$. For all five selected clients, the ground-truth noise levels after label correction are close to 0. Notice also that for client 2, whose dataset initially has no noisy labels, only a minimal amount of label noise is introduced during the label correction process.

### 3.6. Additional ablation study results

In Fig. 5, we show the effects of the components of `FedCorr` on test accuracies during training.In particular, note that without the finetuning stage, the total communication cost would be 5000. Hence in Fig. 5, the curve plotted for `FedCorr` without finetuning ends at the 5000 communication cost mark, which is to the left of the second red dotted line (5500 communication cost). As we mentioned in the main paper, fraction scheduling plays the most significant role in `FedCorr`. In addition, the label correction process would significantly improve training stability, especially in the usual training stage.

### 3.7. Illustration of non-IID data partitions on CIFAR-10

As reported in the main paper, we used 3 different non-IID local data settings $((p, \alpha_{Dir}) = (0.7, 10), (0.7, 1), (0.3, 10))$ for our experiment involving non-IID data partitions. In Fig. 6, we illustrate the detailed local class distributions and local dataset sizes for these three non-IID data settings on CIFAR-10, over 100 clients.

## 4. Potential negative impact: the issue of freeloaders

In real-world FL implementations, there is the implicit assumption that clients are collaborative and jointly collaborate to train a global model. Although `FedCorr` allows for a robust training of a global model even when some clients have label noise, this also includes the case when a client is a "freeloader", where the client's local dataset has completely random label noise (e.g. randomly assigning labels to an unlabeled dataset, without any actual non-trivial annotation effort). By participating in the `FedCorr` FL framework, such a "freeloader" would effectively use

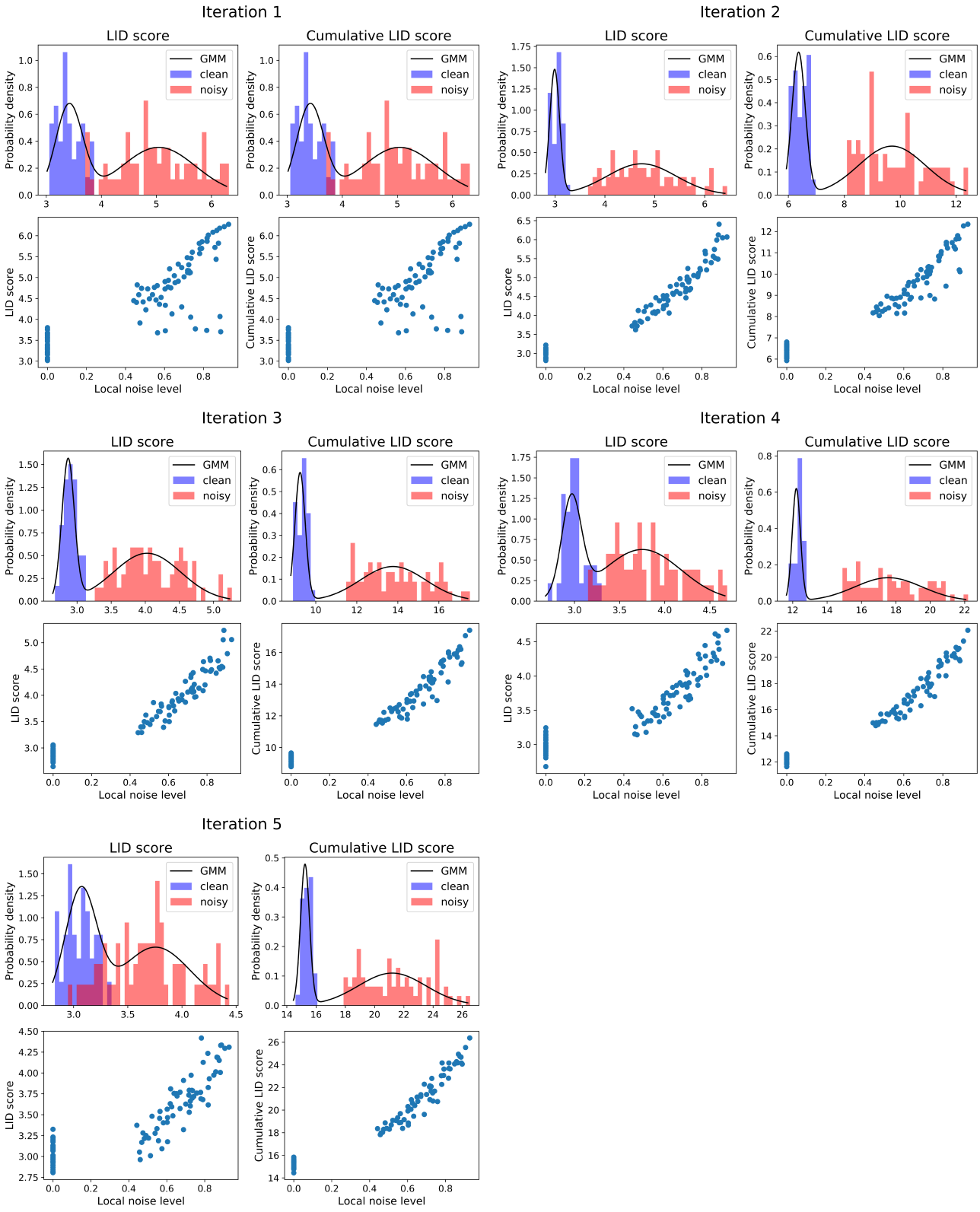


Figure 2. Distributions of the LID/cumulative LID scores during all 5 iterations of the pre-processing stage of `FedCorr`, evaluated on CIFAR-10 with IID data partition and noise setting $(\rho, \tau) = (0.6, 0.5)$, over 100 clients.

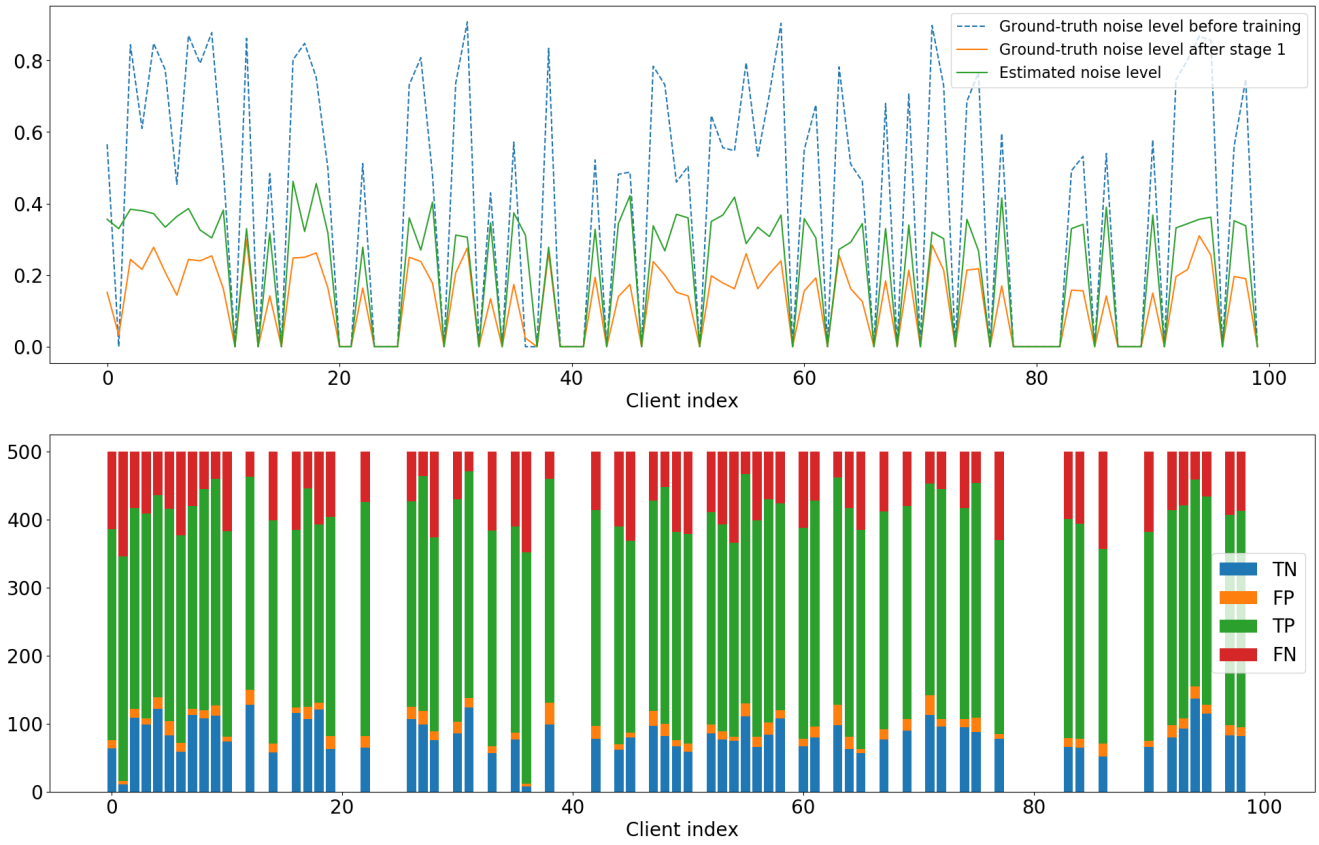


Figure 3. An evaluation of the label noise identification and label correction process after 5 iterations in the pre-processing stage, conducted on CIFAR-10 with IID data partition and noise setting $(\rho, \tau) = (0.6, 0.5)$. Top: Evaluation of noise level estimation and label correction process in the pre-processing stage. Bottom: Evaluation of label noise identification.

`FedCorr` as the actual annotation process, whereby identified noisy labels are corrected. Hence, this would be unfair to clients that have performed annotation on their local datasets prior to participating in `FedCorr`.

## References


[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2020. 1, 2

[2] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *arXiv preprint arXiv:2010.01243*, 2020. 1, 3

[3] Shuhao Fu, Chulin Xie, Bo Li, and Qifeng Chen. Attack-resistant federated learning with residual-based reweighting. In *AAAI Workshop Towards Robust, Secure and Efficient Machine Learning*, 2021. 1, 2

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[5] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 3

[6] Junnan Li, Richard Socher, and Steven C.H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2020. 1, 2

[7] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, volume 2, pages 429–450, 2020. 1, 2

[8] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 1, 2

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International*

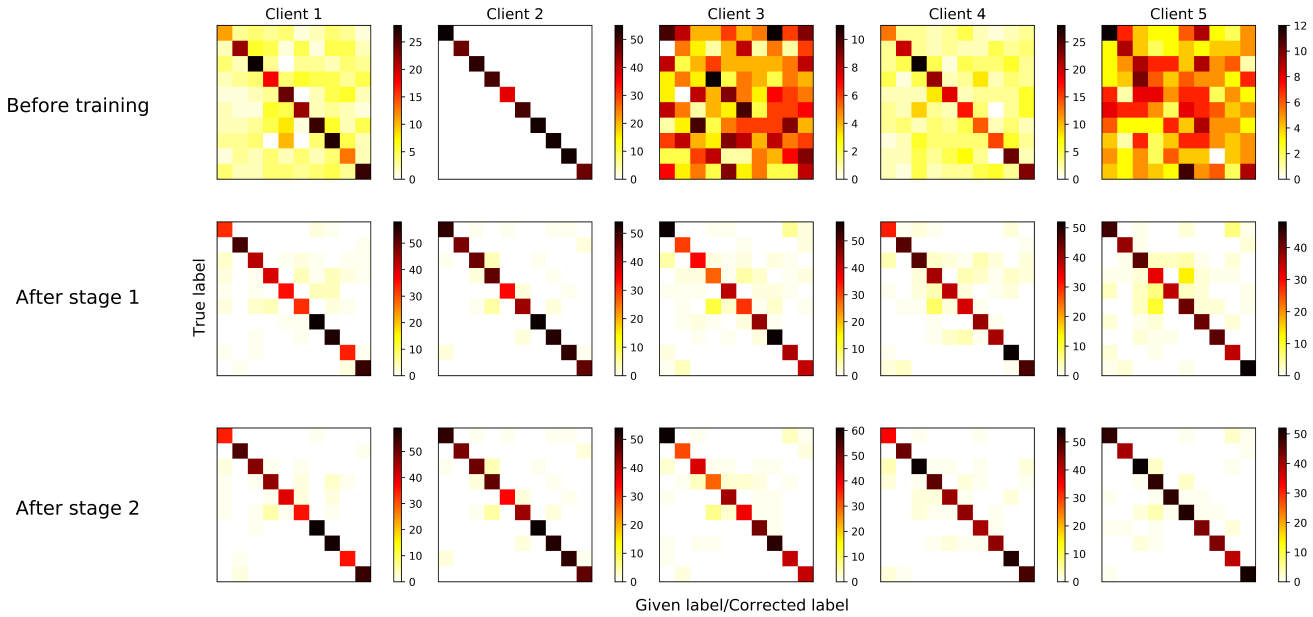


Figure 4. An evaluation of the label correction process on the first five clients, conducted on CIFAR-10 with IID data partition and noise setting $(\rho, \tau) = (0.6, 0.5)$. For each client, we give the heat maps of three confusion matrices, associated to the given labels before training, the corrected labels after the pre-processing stage (stage 1), and the corrected labels after the finetuning stage (stage 2).

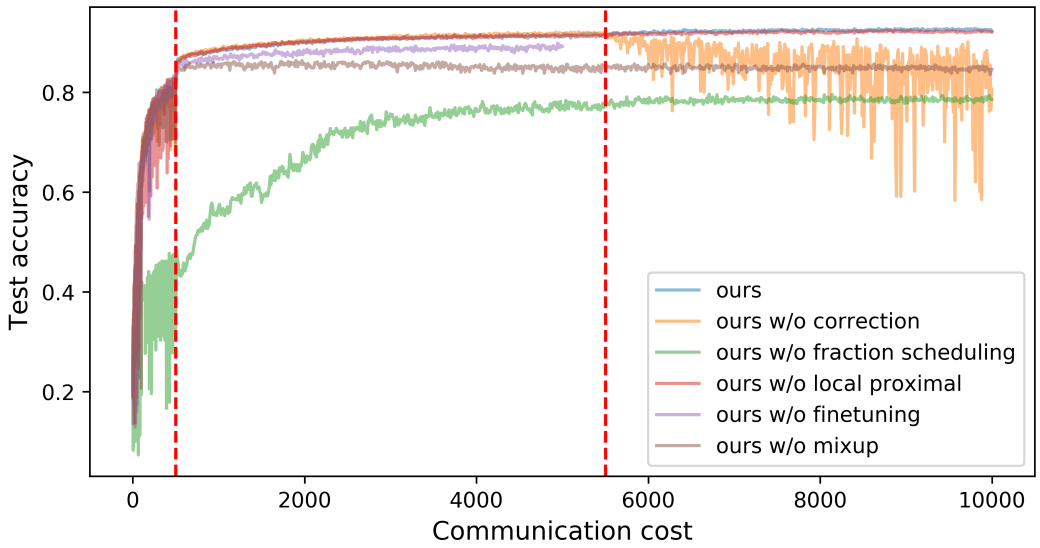


Figure 5. Ablation study results on the test accuracies of `FedCorr` during the training process, with each component removed. The experiments are evaluated on CIFAR-10 with IID data partition and noise setting $(\rho, \tau) = (0.6, 0.5)$. The dotted lines represent the separation of the training process into our three stages.

*Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 3

[10] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5552–5560, 2018. 1, 2

[11] Seunghan Yang, Hyoungseob Park, Junyoung Byun, and Changick Kim. Robust federated learning with noisy labels. *IEEE Intelligent Systems*, 2022. 2

[12] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter

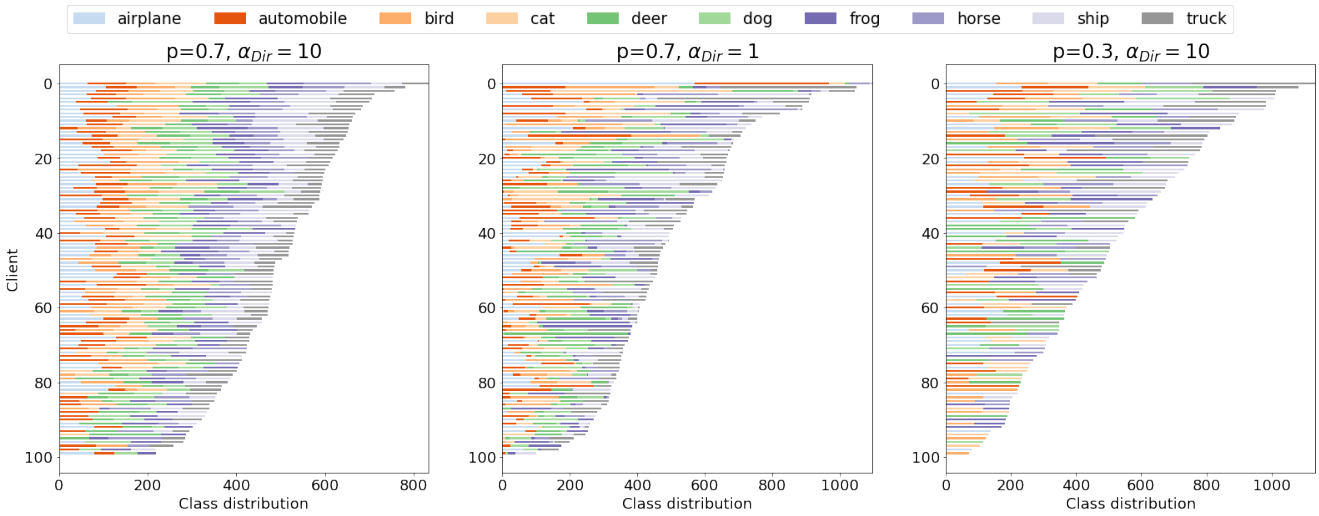


Figure 6. An illustration of three non-IID data partitions (via three different values for ($p$,$\alpha_{Dir}$)) on CIFAR-10, as used for our experiments reported in the main paper. For ease of viewing, we have sorted the clients according to their local dataset sizes in each of the three non-IID data partitions depicted. It should be noted that in our experiments, clients are *not* sorted; instead, clients are assigned data samples according to the data partition described in the main paper, which is a random process with no sorting.

Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5650–5659. PMLR, 10–15 Jul 2018. 3