Robust Heterogeneous Federated Learning under Data Corruption

Xiuwen Fang, Mang Ye[*], Xiyuan Yang
National Engineering Research Center for Multimedia Software, Institute of Artificial Intelligence,
Hubei Key Laboratory of Multimedia and Network Communication Engineering,
School of Computer Science, Hubei Luojia Laboratory, Wuhan University, Wuhan, China
https://github.com/FangXiuwen/AugHFL

Abstract

Model heterogeneous federated learning is a realistic and challenging problem. However, due to the limitations of data collection, storage, and transmission conditions, as well as the existence of free-rider participants, the clients may suffer from data corruption. This paper starts the first attempt to investigate the problem of data corruption in the model heterogeneous federated learning framework. We design a novel method named Augmented Heterogeneous Federated Learning (AugHFL), which consists of two stages: 1) In the local update stage, a corruption-robust data augmentation strategy is adopted to minimize the adverse effects of local corruption while enabling the models to learn rich local knowledge. 2) In the collaborative update stage, we design a robust re-weighted communication approach, which implements communication between heterogeneous models while mitigating corrupted knowledge transfer from others. Extensive experiments demonstrate the effectiveness of our method in coping with various corruption patterns in the model heterogeneous federated learning setting.

1. Introduction

Modern society contains a large number of edge devices, such as smartphones, mobile networks, IoT devices, *etc*., which can be regarded as local clients with limited private data. The direct collection of private data from clients seriously compromises their privacy and security. Federated learning [48, 37, 7] is a distributed machine learning framework with secure encryption techniques. It is intended to enable multiple institutions to collaboratively train machine learning models while following the data-stay-local policy. In addition to the attention of academia [63, 68, 36, 45, 32, 59, 59, 56], federated learning has also been extensively explored in industrial fields such as healthcare [10, 64, 27, 15], finance [4], and data security [61], *etc*. Despite its remarkable success, the most prevalent federated learning algorithms [16, 51, 49] rely heavily on the assumption that all clients share an identical local model structure. For example, FedAvg [48], FedProx [35], and Per-FedAvg [13] enable communication between clients by taking the weighted average of the local model parameters.

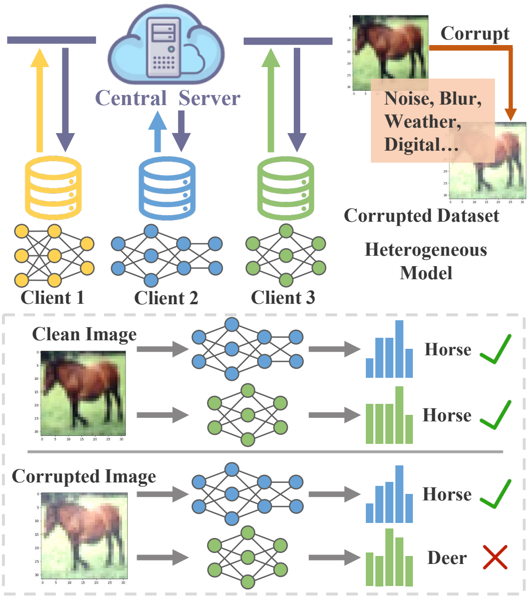

Figure 1. Illustration of heterogeneous federated learning with data corruption, where the clients may possess different model structures and corrupted private datasets.

In practical application scenarios, clients expect to independently design their own models to meet different tasks and specifications, which inevitably leads to the challenge of model heterogeneity [69, 46, 66, 72] in federated learning, as shown in Fig. 1. However, clients are often reluctant to share the details of their local model designs in consideration of privacy protection. Additionally, hardware, software, and communication capabilities [34] often differ between clients, which can further exacerbate model heterogeneity. Consequently, designing heterogeneous federated learning strategies to achieve collaborative communi-

[*] Corresponding Author: Mang Ye (yemang@whu.edu.cn)

cation between heterogeneous models has become the focus of significant research. FedMD [31] performs communication between heterogeneous clients based on the average class score output by the local models. FedDF [41] leverages unlabeled data or artificially generated examples to extract knowledge from the clients. RHFL [14] learns the knowledge distribution from other clients by aligning the feedback outputs of client models on public irrelevant data. However, the aforementioned methods all assume that each client has a private dataset with clean images, which is difficult to satisfy in practical applications.

The widely used heterogeneous federated learning algorithms show exceptional performance under ideal circumstances where the image samples are clean. However, clean image samples require substantial costs in terms of data collection, storage, and transmission. In real-world scenarios, the quality of the collected data can be influenced by several factors such as weather conditions, collection means, storage methods, *etc*. Consequently, the algorithms inevitably encounter corrupted data. Furthermore, there may be free-riding participants [40] in the federated learning system, who pretend to participate in the federated learning process, benefiting from the information contributed by other clients without actually providing any data. This phenomenon results in honest clients being reluctant to share their real information, raising concerns about data privacy and user fairness. As a means of safeguarding their privacy, honest clients may intentionally offer corrupted data. This data corruption strategy maintains the fairness of the federated learning system at the expense of model performance.

Data corruption is inevitable in data collection, storage, and transmission. Worse still, free-rider concerns exacerbate the potential for data corruption. As a consequence of data corruption, clients iteratively learn and share wrong knowledge, causing local models to update in incorrect directions. Ultimately, the performance of the federated learning system suffers a severe degradation, hindering the practical deployment of the models. In single-model machine learning, existing methods to mitigate the negative impact of data corruption during model training can be classified into three categories: data augmentation [2, 44, 53], noise injection [38, 42, 43], and pre-training [23, 50, 3]. The multi-model scenario of federated learning is more complex, thereby rendering overcoming data corruption more challenging. We expect to fully learn local knowledge and resist multiple corruptions under the federated learning framework. Thus, *how to mitigate model performance degradation caused by data corruption inside the client during the local update phase* is an important challenge.

In addition, data corruption in model heterogeneous scenarios encounters a new problem, *i.e*., how to avoid learning the corrupted knowledge from others while realizing heterogeneous models communication. In heterogeneous federated learning scenarios, the local models of clients may have different model architectures, thereby presenting inconsistent decision boundaries. It is noteworthy that corrupted samples are generally harder to classify correctly than clean samples, as illustrated in Fig. 2. For the same corrupted image, different model predictions have significant variance, as depicted in Fig. 1. Thus, during the collaborative update phase, the clients may learn erroneous predictions from other models. However, existing centralized methods for corruption can only handle internal corruption, but they cannot prevent external corruption from low-robustness models in the collaborative learning phase. Moreover, existing homogeneous FL methods for corruption are not suitable for model heterogeneous scenarios, as they rely on the local model training loss to measure the model reliability. The model training loss can be affected by many factors, and it is not fair or reliable to compare the training loss of heterogeneous models directly. Therefore, *how to minimize corrupted feedback from unreliable clients while implementing heterogeneous model communication* is a crucial challenge.

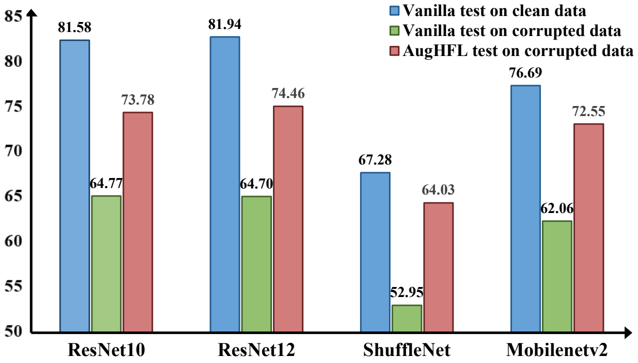

Figure 2. Evaluation of each client model on the clean and corrupted test data. We observe that clean data are more susceptible to correct prediction than corrupted data. Compared to vanilla models, AugHFL achieves significantly higher accuracy on corrupted data and maintains performance on clean data.

In this paper, we propose AugHFL to address the data corruption problem in heterogeneous federated learning, which consists of two stages: 1) For the negative impact of local data corruption, we perform multiple random data augmentation operations and then mix the augmented images. Simultaneously, a consistency constraint for diverse augmentations of the same image is imposed on the classifiers. This technique improves the classification robustness of the local models against corrupted data in the local update phase. 2) For corrupted knowledge learned from others, we design an adaptive re-weighted communication strategy called Public data Augmentation (PubAug). The main idea is to dynamically adjust the contributions of clients according to their reliability in the collaborative update phase. We measure the predicted distribution consistency of the local model on the corrupted public datasets and the original public dataset. Then, the robustness of models to uncertain corruption patterns is quantified by the computed distribution consistency. The local models that can effectively against multiple corruptions are considered reliable and thus their weights are increased, while the weights of unreliable models decrease synchronously. The main contributions of this work are as follows:

- We introduce AugHFL for a new and challenging model heterogeneous federated learning problem with data corruption. It simultaneously handles intra-client and inter-client corruption, enhancing the robustness against diverse corruptions.
- We propose PubAug, an adaptive re-weighted communication method. It adaptively adjusts the contribution of clients based on their ability to cope with data corruption while enabling communication between heterogeneous models.
- Extensive experiments demonstrate that AugHFL outperforms State-Of-The-Art (SOTA) methods in terms of both performance and robustness, especially on datasets with severe data corruption.

2. Related Work

Federated Learning. Federated learning, proposed by McMahan *et al*. [48] in 2017, is a distributed learning paradigm that enables multiple clients to jointly train local models without compromising their privacy. In the classical FedAvg algorithm, clients train local models on their private datasets, the server averages model parameters across all clients, and iterates the process for several rounds. To address the problem of high parameter-transmission costs, Wu *et al*. [65] proposed the SmartIdx algorithm, which includes a kernel-based parameter selection strategy and a parameter compression algorithm that reduces communication overhead. Additionally, the FedProx algorithm proposed by Li *et al*. [35] improves the stability and convergence of the training process by adjusting client-training epochs.

Unlike traditional homogeneous federated learning, model heterogeneous federated learning involves several clients designing their own local models. Several current approaches [55, 6, 47, 20] leverage knowledge distillation techniques proposed by Hinton *et al*. [25] to communicate between clients. For example, FedMD proposed by Li *et al*. [31] enables heterogeneous models to communicate by learning the average soft label across all clients. Many methods [6, 72] utilize public data to extract, aggregate, and transfer knowledge. Furthermore, other approaches [41, 26, 14] leverage unlabeled data to implement knowledge distillation and transfer among clients.

However, these SOTA methods assume that training images are flawless and do not account for the data corruption problem of clients. Consequently, these methods may face difficulties in model convergence or suffer from poor performance in the presence of data corruption.

Data Corruption Learning. Data corruption is a prevalent type of disturbance [52], which can occur during data collection, storage, and transmission in real-world scenarios, and can also be caused by free-riders in federated learning systems. Dodge *et al*. [12] point out that models under training are very sensitive to continue training on corrupted datasets which means training models under data corruption severely degrades model performance. The SOTA methods mainly employ the following techniques to mitigate performance degradation caused by data corruption.

1) *Data augmentation* [58, 71, 62] aims to enhance the robustness and reduce generalization error by augmenting corrupted data. MixUp [71] and Manifold Mixup [62] are widely used augmentation algorithms that linearly combine training data to augment the dataset. Teach Augment [58] improves the generalizability of models by using an adversarial model to prevent feature loss caused by excessive data augmentation. 2) *Noise injection* [1, 18, 5, 39, 29] prevents the model from overfitting to the corrupted data by adding noise to one or more parts of training process. Early noise injection methods [1] added noise to training data and subsequent works extended this approach to activation functions [18] and fully-connected layers of MLP or CNN [5, 39], which improves model robustness against data corruption. 3) *Pre-training* [23, 67] on diverse datasets with large domain gaps also contributes to model robustness. Comprehensive studies [23] have shown that pre-trained models have stronger robustness and generalization. Noisy Student Training [67] with the pre-training process has also shown good robustness against various data corruption. Currently, there are also some works investigating the problem of corruption in federated learning [33, 57], both of which mainly use empirical risk or to set the aggregation weight of each round. However, their proposals are only applicable to the model homogeneous setting and cannot be applied to scenarios with model heterogeneity.

Previous methods for data corruption mainly focus on centralized training and model homogeneous federated learning, without addressing the negative impact of data corruption in model heterogeneous federated learning.

3. Proposed Method

3.1. Preliminaries

In this paper, we consider the C-class image classification task and assume a federated learning system with K clients and one server. The k-th client c_k has a private dataset $D_k = \{(x_i^k, y_i^k)\}_{i=1}^{N_k}$ with $|x^k| = N_k$ and $y_i^k \in \{0,1\}^C$. To protect basic privacy, the client c_k never shares the private dataset D_k with the server and other clients $c_{k' \neq k}$. In the context of model heterogeneity, each client c_k has an independently designed local model $f(\theta_k)$ with a unique neural structure, and θ_k represents the model pa-

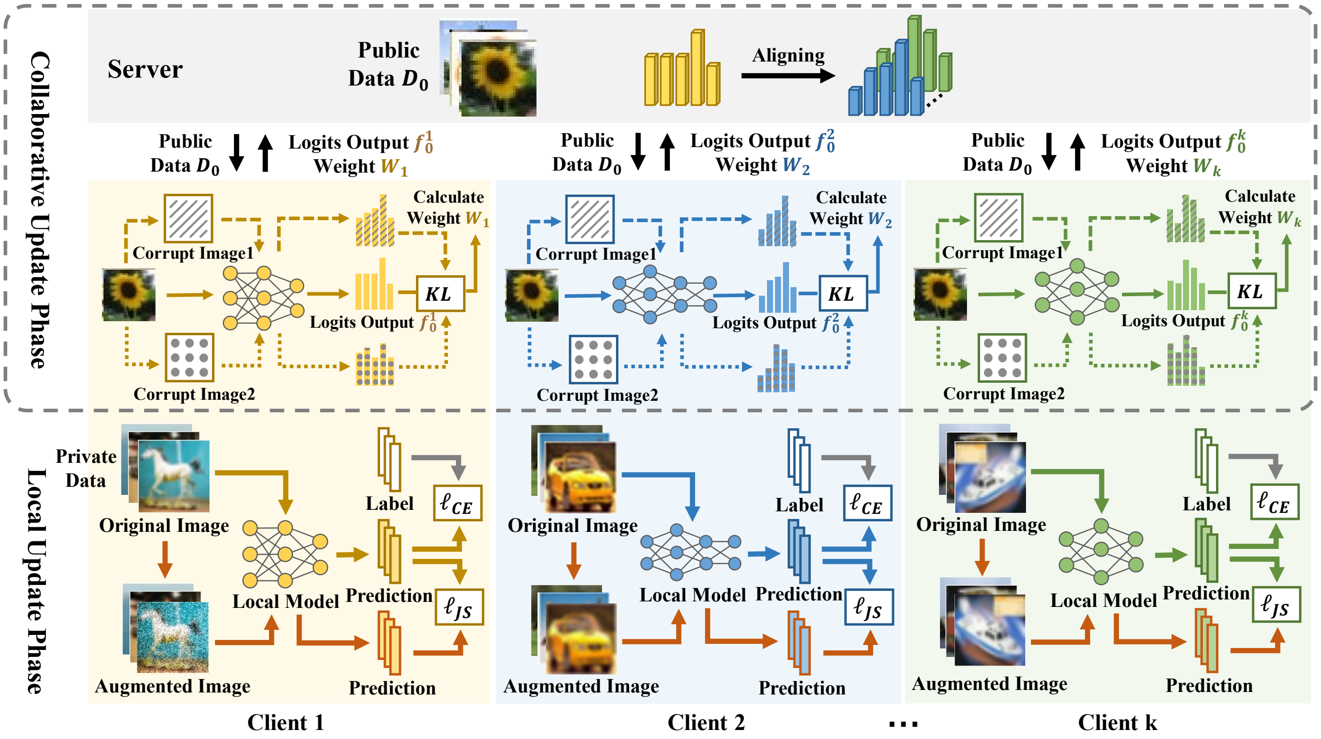

Figure 3. Illustration of AugHFL, which enables knowledge communication between heterogeneous models by aligning the output distributions of local models on public data. A robust data augmentation strategy is employed in the local update phase to handle local corrupted knowledge (Sec. 3.2). An adaptive re-weighted communication method, PubAug, is designed in collaborative update phase to mitigate corrupted knowledge learned from others (Sec. 3.3).

rameters. To explore the data corruption problem in heterogeneous federated learning, we consider that each client c_k possesses a corrupted private dataset $\tilde{D}_k = \{(\tilde{x}_i^k, y_i^k)\}_{i=1}^{N_k}$, where $\tilde{x}_i^k$ is an image that can be either clean or corrupted. For communication purposes, the server has an unlabeled public dataset $D_0 = \{\tilde{x}_i^0\}_{i=1}^{N_0}$ with $|\tilde{x}^0| = N_0$, which can be directly accessed by all clients. And the samples in this public dataset are potentially corrupted.

Federated learning generally consists of the local update phase and the collaborative update phase. We denote the rounds of local update and collaborative update by T_l and T_c, respectively. It is necessary to enhance the stability and robustness of classifiers against local corruption when local updates. Besides, due to the difficulty of correctly classifying corrupted data, heterogeneous models may have inconsistent prediction outputs for the same corrupted images, which can be expressed as $f(\tilde{x}, \theta_{k_1}) \neq f(\tilde{x}, \theta_{k_2})$. Therefore, we need to prevent the client from learning corrupted predictions from others during the collaborative update phase. Each client c_k is allocated an adaptive weight W_k to minimize the exchange of corrupted feedback.

In this context, the objective is to obtain an optimal set of model parameters $\{\theta_1, \theta_2, ..., \theta_K\}$ that minimizes the empirical risk $\mathbb{E}(f)$, as follows:

$$\arg\min_{\theta} \mathcal{L}(\theta) = \sum\nolimits_{k=1}^{K} W_k \cdot \mathcal{L}(\theta_k), \tag{1}$$

where $\mathcal{L}(\theta_k) = \mathbb{E}_{(\tilde{x},y)\sim D_k}[\ell(f(\tilde{x}, \theta_k), y)]$ is the empirical loss of c_k, and $\ell(\cdot)$ is the loss function.

3.2. Local Learning with Data Corruption

In the local update phase, we strive to minimize the adverse effects of data corruption while enabling the local model to learn sufficient local knowledge. In machine learning, data augmentation [11, 70, 71, 60, 19, 8, 9] is a prevalent technique to enhance the model generalization performance. The principle is to improve the model performance on unseen data by generating richer and more diverse training samples. Additionally, data augmentation methods have been demonstrated to make models more robust against data corruption [17]. To enhance the robustness of the local model against data corruption, we learn the solution strategy from Hendrycks *et al.* [24].

We select a set of traditional data augmentation operations $\mathcal{A}$, including autocontrast, equalize, posterize, rotate, solarize, shear_x, shear_y, translate_x, and translate_y, to ensure inconsistency with the original corruption patterns in the private datasets. In addition, the data augmentation operations use varying augmentation magnitudes. Then, one to three augmentation operations $a \sim \mathcal{A}$ are randomly selected, and they are stacked to construct several augmentation operation sequences Seq with different depths. This process can be formulated as:

$$\begin{aligned} &Seq \sim \{a_1, a_{12}, a_{123}\}, \\ a_{12} =&a_1 \oplus a_2,\ a_{123} = a_1 \oplus a_2 \oplus a_3, \end{aligned} \tag{2}$$

where $a_1, a_2, a_3 \sim \mathcal{A}$, and a_{12}, a_{123} denote the operation combination sequences with depths of 2 and 3, respectively. However, the resulting image $Seq(x)$ obtained by cascading multiple data augmentation operations may be distorted and drift away from the original image x. To alleviate this image degradation, a set of weights $(w_1, ..., w_{\mathcal{S}})$ is randomly sampled from the $Dirichlet(\alpha, ..., \alpha)$ distribution. $\mathcal{S}$ represents the number of random augmentation operation sequences. The resulting images $Seq(x)$ from different augmentation sequences are mixed according to the corresponding weights w, which can be expressed as:

$$x_{seq} = \sum\nolimits_{i=1}^{\mathcal{S}} w_i \cdot Seq_i(x). \tag{3}$$

Random weighted mixing of $\mathcal{S}$ augmented images can create more diversity than simple augmentation. To prevent the mixed image from losing the semantic information of the original image, we combine the mixed image with the original image, thus obtaining the image after Aug:

$$x_{aug} = \eta \cdot x + (1 - \eta) \cdot x_{seq}, \tag{4}$$

where η refers to the weight randomly sampled from the $Beta(\alpha, \alpha)$ distribution. In this way, various augmentation operations, augmentation magnitudes, augmentation sequences, and multiple random mixing operations guarantee the diversity of the augmented images.

To further improve model stability, an additional Jensen-Shannon (JS) divergence consistency loss is utilized to constrain model updates. The main idea is to constrain the output consistency of the classifier on different augmentations of the same image. For each original image x, two different augmented samples x_{aug_1}, x_{aug_2} are generated by the above method. Subsequently, the loss function of the local update phase can be specifically expressed as:

$$\ell_{local} = \ell_{CE}(f, y) + \mu \cdot \ell_{JS}(f, f', f''). \tag{5}$$

Here, μ controls the strength of JS consistency constraint and ℓ_{CE} denotes the cross-entropy loss. f, f', f'' denote the output distributions of the original image x and two augmented images x_{aug_1}, x_{aug_2} on the local model θ respectively. JS divergence is a symmetric and smooth version based on Kullback–Leibler (KL) divergence, which ensures output stability for samples from the same original image.

In the local update phase, the clients update the local models with their private datasets, so that the local models can sufficiently learn the local information and prevent from forgetting local knowledge after several rounds of communication. However, the client may repeatedly learn corrupted knowledge and optimize in the wrong direction, which will eventually lead to non-convergence of the local model. Thus, we utilize ℓ_{local} in Eq. (5) as the loss function of the local update, and the local update process of client c_k can be formulated as:

$$\theta_k^{t_l} \leftarrow \theta_k^{t_l-1} - \lambda \nabla_\theta \ell_{local}(f(\tilde{x}^k; \theta_k^{t_l-1}), y^k), \tag{6}$$

where λ denotes the local learning rate and $t_l \in [0, T_l]$ represents the t_l-th local update epoch. The data augmentation strategy enables the models to successfully cope with unseen corruptions during the local update phase.

3.3. Robust Corrupted Clients Communication

In the collaborative update phase, we follow HFL [14] to implement knowledge communication between heterogeneous models in a model-agnostic manner. Concurrently, the propagation of corrupted knowledge in the mutual learning process is minimized by our proposed PubAug.

The private datasets carry abundant exclusive information which is safeguarded and not viable for exchange. The unlabeled public dataset is relatively easy to obtain, which is compiled from diverse sources and can be corrupted. The output distributions of public data on local models reflect the discriminative capabilities of the models. Consequently, we leverage the output distributions of the public dataset to execute communication among heterogeneous clients. Specifically, the client c_k expresses local knowledge information through the output class distribution $f(D_0, \theta_k)$ computed by the local model θ_k on public dataset D_0. We implement collaborative learning with KL divergence, which is a metric used to measure the distance between two probability distributions. The discrepancy between the output class distribution of client c_p and client c_q is expressed as:

$$KL(p||q) = \sum\nolimits_{i=1}^{N_0} p(\tilde{x}_i^0) \log \frac{p(\tilde{x}_i^0)}{q(\tilde{x}_i^0)}, \tag{7}$$

where $p(\tilde{x}_i^0) = f(\tilde{x}_i^0, \theta_p)$ and $q(\tilde{x}_i^0) = f(\tilde{x}_i^0, \theta_q)$ indicate the output class distributions of public data $\tilde{x}_i^0$ on client c_p and client c_q, respectively. Furthermore, the communication of local knowledge information between clients can be realized by narrowing the gap between their output class distributions. Thus, we leverage KL divergence as the loss function in the collaborative update phase, where minimizing KL divergence can be interpreted as a process of acquiring knowledge from other clients. The loss of the client c_k in the t_c-th collaborative round is

$$\ell_{col}^{k,t_c} = \sum\nolimits_{i=1, i \neq k}^{K} KL(f_0^{k,t_c} || f_0^{i,t_c}), \tag{8}$$

where $f_0^{k,t_c} = f(D_0, \theta_k^{t_c})$ is the prediction distribution of the local model θ_k on public dataset, and f_0^{i,t_c} is similar.

In this way, each client c_k learns from others by fitting the output class distributions of other clients. Besides, clients with more distinct differences in output class distributions can acquire richer knowledge from each other.

To minimize corrupted knowledge learned from others, we design PubAug to dynamically adjust the contribution of clients to the federated learning system. Our weight calculation strategy takes inspiration from semi-supervised learning. Ideally, the output prediction distributions of the corrupted sample and the original sample should be similar. Therefore, for each client c_k, we obtain two different corrupted public datasets D_{0,k_1}, D_{0,k_2} by randomly corrupting the original public dataset D_0. Among them, the corruption patterns and severity level of each sample are randomly selected to provide a comprehensive measure of client reliability. The corrupted public datasets D_{0,k_1}, D_{0,k_2} and the original public dataset D_0 are simultaneously fed into the local model θ_k, resulting in three distinct prediction distributions. We exploit the consistency of these prediction distributions to quantify the robustness of local models against different corruption patterns. Consequently, the reliability R_k of the client c_k against corruption can be measured as:

$$R_k = \frac{1}{KL(f^k_{0,k_1}||f^k_0) + KL(f^k_{0,k_2}||f^k_0)}, \tag{9}$$

where f^k_{0,k_1}, f^k_{0,k_2} are the prediction distributions of corrupted public datasets D_{0,k_1} and D_{0,k_2} on the local model θ_k, respectively. The reliability of the local model is proportional to the degree of consistency in prediction distributions between the corrupted data and the original data. Thus, the local models with high reliability computed by Eq. (9) can be considered highly resistant to corruption.

In the collaborative update phase, we devise an adaptive re-weighting scheme based on client reliability that effectively mitigates the detrimental impact of corrupted clients on the federated learning system. reduces the contribution of corrupted clients to the federated learning system, while learning more from reliable clients. By prioritizing reliable clients, our approach optimizes the distribution of learning effort in a manner that enhances the overall performance of the system. Specifically, the collaborative update process of client c_k in the t_c-th iteration can be formulated as:

$$\theta^{t_c}_k \leftarrow \theta^{t_c-1}_k - \lambda \nabla_\theta (W_k \cdot \ell^{k,t_c-1}_{col}), W_k = \frac{R_k}{\sum^K_{i=1} R_i}, \tag{10}$$

where $t_c \in [0, T_c]$ indicates the t_c-th collaborative update iteration. The weight W_k assigned to the client c_k is dynamically adjusted according to its reliability R_k. Through this adaptive re-weighting strategy, we are able to emphasize the contribution of more reliable clients while reducing the noise introduced by corrupted clients during heterogeneous model communication. This approach optimizes the learning process and refines the communication performance of heterogeneous models under data corruption.

4. Experimental

4.1. Experimental Setup

Datasets and Models. According to latest works [14, 24], our experiments are based on two datasets, Cifar-10-C [22] and Cifar-100 [30], due to their significant influence in the research of data corruption learning and image classification tasks. Both Cifar-10 [30] and Cifar-100 contain 60,000 color images of size 32 × 32, including 50,000 training images and 10,000 testing images. Furthermore, Cifar-10-C is obtained by introducing common visual corruptions in Cifar-10. We randomly divide Cifar-10-C to clients as their private datasets and select a subset of Cifar-100 as the public dataset on the server. To meet the requirements of the heterogeneous model scenario, we assign four different local models, ResNet10 [21], ResNet12 [21], ShuffleNet [73] and Mobilenetv2 [54] to four clients respectively.

Corruption Patterns. To better simulate actual data corruption, our approach to constructing corruption patterns for federated learning follows Hendrycks *et al*. [22]. Cifar-10-C consists of 15 corruption types (from four major categories: Noise, Blur, Weather, and Digital) and each corruption pattern has five severity levels. Corrupted images are generated by randomly sampling the corruption types and severity levels from a uniform distribution. We randomly set different corruption rates, corruption types, and severity levels for private datasets of different clients. Moreover, there may be different corruption patterns between different samples within a private dataset.

Baselines. We compare AugHFL with advanced methods to prove its effectiveness in the heterogeneous model scenario. Specifically, AugHFL is compared with FedMD [31], FedDF [41], RHFL [14] and FCCL [26] under the same setting. FedMD communicates based on the average class scores output by client models on the public dataset. FedDF is a distillation framework for robust federated model fusion utilizing unlabeled or artificially generated data. RHFL simultaneously handles the label noise and heterogeneous model communication in a single framework. FCCL constructs a cross-correlation matrix for collaborative learning and prevents catastrophic forgetting in local updates. Due to the differences in experimental settings between different algorithms, we implement the main ideas of these algorithms in our framework for a fair comparison.

Implementation Details. Initially, the local model on each client will be pre-trained for 40 epochs on its private dataset. The size of private datasets and the public dataset is specified as $N_k = 10,000$ and $N_0 = 5,000$ respectively. To ensure the effectiveness of federated learning, all clients perform $T_c = 40$ rounds of collaborative updates. The num-

Table 1. Ablation experiment with corruption rate $\xi = 0$ on private dataset, θ_k represents the local model of the client c_k.

Components				Test on clean dataset					Test on random corrupted dataset				
HFL	PubAug	Aug	JSD	θ_0	θ_1	θ_2	θ_3	Avg	θ_0	θ_1	θ_2	θ_3	Avg
				82.63	**82.57**	68.58	78.14	77.98	68.03	66.21	56.04	63.23	63.38
✓				81.18	81.29	**71.46**	**80.50**	78.61	65.75	65.78	58.08	64.20	63.45
✓	✓			82.06	82.09	70.86	80.15	**78.79**	66.51	67.91	57.25	67.84	64.88
✓	✓	✓		75.01	77.47	61.59	74.33	72.10	62.60	66.48	46.97	62.98	59.76
✓	✓	✓	✓	79.86	81.45	70.67	79.47	77.86	**73.78**	**74.46**	**64.03**	**72.55**	**71.21**

Table 2. Ablation experiment with corruption rate $\xi = 0.5$ on private dataset, θ_k represents the local model of the client c_k.

Components				Test on clean dataset					Test on random corrupted dataset				
HFL	PubAug	Aug	JSD	θ_0	θ_1	θ_2	θ_3	Avg	θ_0	θ_1	θ_2	θ_3	Avg
				68.75	68.46	57.14	57.41	62.94	64.03	64.75	50.86	52.40	58.01
✓				61.37	63.71	56.08	59.34	60.13	57.19	60.07	50.79	56.25	56.08
✓	✓			62.67	62.37	57.25	59.58	60.47	58.20	59.18	53.94	55.96	56.82
✓	✓	✓		70.27	69.08	57.65	67.74	66.19	62.17	59.22	49.09	59.29	57.44
✓	✓	✓	✓	**76.22**	**76.50**	**66.66**	**73.31**	**73.17**	**71.96**	**71.26**	**61.28**	**69.58**	**68.52**

Table 3. Ablation experiment with corruption rate $\xi = 1$ on private dataset, θ_k represents the local model of the client c_k.

Components				Test on clean dataset					Test on random corrupted dataset				
HFL	PubAug	Aug	JSD	θ_0	θ_1	θ_2	θ_3	Avg	θ_0	θ_1	θ_2	θ_3	Avg
				65.58	67.01	56.00	57.11	61.43	62.47	63.88	52.57	53.49	58.10
✓				62.02	62.24	58.51	59.84	60.65	59.22	59.88	55.00	56.74	57.71
✓	✓			60.41	62.40	57.67	59.46	59.99	56.76	59.10	53.96	55.96	56.45
✓	✓	✓		68.90	67.65	55.99	66.59	64.78	62.55	61.26	48.82	58.46	57.77
✓	✓	✓	✓	**76.98**	**77.82**	**65.79**	**74.03**	**73.66**	**73.65**	**74.04**	**61.50**	**70.92**	**70.03**

ber of local learning epochs is adaptively set to $T_l = \frac{N_0}{N_k}$ to balance local and shared knowledge. Furthermore, we use the Adam [28] optimizer with an initial learning rate of $\lambda = 0.001$ and the batch size of 256. The number of augmentation operation sequences $\mathcal{S}$ is set as 3 and the hyperparameter μ is 12. Since this paper focuses on the data corruption problem in federated learning, we consider three cases with corruption rates ξ of 0, 0.5, and 1, meaning the private dataset is clean, half corrupted, and fully corrupted. The client c_k randomly selects N_K samples from Cifar-10-C generated by random corruption. The effectiveness of our method is tested on clean and randomly corrupted datasets.

4.2. Ablation Study

We train on private datasets with three corruption rates $\xi = 0, 0.5, 1$, and evaluate the components on the completely clean and the completely random corrupted datasets.

Effectiveness of HFL. According to Tab. 1, we observe that adding HFL improves the average accuracy of four local models from 77.98% to 78.61% when both the training and testing datasets are clean. This shows that federated communication enables clients to learn from each other. However, in data corruption scenarios, the effect of adding HFL decreased instead. We attribute this phenomenon to the fact that, in the presence of data corruption, federated learning causes clients to repeatedly exchange incorrect knowledge, which degrades the model performance. Therefore, data corruption severely hinders heterogeneous federated learning, which illustrates the necessity of exploring this issue.

Effectiveness of PubAug. We add the PubAug module to reduce the negative impact of corrupted clients during the collaborative update phase. The addition of PubAug brings better performance improvements in scenarios where private datasets are either clean or partially corrupt, as demonstrated in Tables 1 and 2. However, when the dataset is fully corrupted, we observe limited impact from PubAug. This is due to the fact that when all clients have fully corrupted data, their reliability levels are similar, making it difficult for them to help each other mitigate data corruption. Consequently, in addition to external data corruption, internal data corruption also needs to be addressed.

Effectiveness of Aug. As shown in the Tabs. 2 and 3, we can observe that Aug can effectively handle datasets with data corruption rather than clean datasets. In the local update phase, Aug alleviates the learning of corrupted knowledge and revises the optimization directions of the models.

Effectiveness of JSD. Adding JSD on the basis of Aug has significantly improved the model performance. Especially when the corruption rate of the private dataset is 0.5, the performance on the clean dataset and the random corrupted dataset is improved by 8.4% and 11.51%, respectively. JSD component imposes a consistency constraint on the output distribution of the images obtained by Aug. This regularization strategy enhances model stability and contributes directly to improved performance.

Table 4. Compare with the SOTA methods with corruption rate $\xi = 0$ on private dataset, θ_k represents the local model of the client c_k.

Model	Test on clean dataset					Test on random corrupted dataset				
	θ_0	θ_1	θ_2	θ_3	Avg	θ_0	θ_1	θ_2	θ_3	Avg
Baseline	82.63	82.57	68.58	78.14	77.98	68.03	66.21	56.04	63.23	63.38
FedMD[31]	82.86	83.06	73.02	80.74	79.92	68.70	68.28	61.01	67.53	66.38
FedDF[41]	82.40	82.40	73.55	78.48	79.21	65.54	67.00	59.39	63.36	63.82
RHFL[14]	82.30	82.83	71.72	77.55	78.60	66.32	62.28	57.65	62.94	62.30
FCCL[26]	**83.26**	**83.07**	**73.69**	**81.53**	**80.39**	69.55	67.78	60.57	68.25	66.54
AugHFL	79.86	81.45	70.67	79.47	77.86	**73.78**	**74.46**	**64.03**	**72.55**	**71.21**

Table 5. Compare with the SOTA methods with corruption rate $\xi = 0.5$ on private dataset, θ_k represents the local model of the client c_k.

Model	Test on clean dataset					Test on random corrupted dataset				
	θ_0	θ_1	θ_2	θ_3	Avg	θ_0	θ_1	θ_2	θ_3	Avg
Baseline	68.75	68.46	57.14	57.41	62.94	64.03	64.75	50.86	52.40	58.01
FedMD[31]	64.25	65.30	55.97	58.58	61.03	59.66	61.28	51.66	54.56	56.79
FedDF[41]	61.72	63.51	57.29	57.89	60.10	58.09	59.57	53.15	54.35	56.29
RHFL[14]	58.05	59.47	51.13	55.07	55.93	62.42	63.46	56.78	59.81	60.62
FCCL[26]	63.75	62.62	58.43	59.78	61.15	59.91	59.03	53.71	56.24	57.22
AugHFL	**76.22**	**76.50**	**66.66**	**73.31**	**73.17**	**71.96**	**71.26**	**61.28**	**69.58**	**68.52**

Table 6. Compare with the SOTA methods with corruption rate $\xi = 1$ on private dataset, θ_k represents the local model of the client c_k.

Model	Test on clean dataset					Test on random corrupted dataset				
	θ_0	θ_1	θ_2	θ_3	Avg	θ_0	θ_1	θ_2	θ_3	Avg
Baseline	65.58	67.01	56.00	57.11	61.43	62.47	63.88	52.57	53.49	58.10
FedMD[31]	62.11	63.74	56.57	58.12	60.14	59.11	60.58	52.51	55.39	56.90
FedDF[41]	61.66	63.11	58.39	58.06	60.31	58.87	59.45	55.44	55.31	57.27
RHFL[14]	62.55	64.21	57.57	58.14	60.62	58.14	59.79	54.75	55.77	57.11
FCCL[26]	62.60	63.64	58.60	59.71	61.14	59.36	59.55	55.82	56.92	57.91
AugHFL	**76.98**	**77.82**	**65.79**	**74.03**	**73.66**	**73.65**	**74.04**	**61.50**	**70.92**	**70.03**

4.3. Comparison with SOTA Methods

We provide a comprehensive comparison of AugHFL with SOTA methods in several different scenarios, where the models are trained on datasets with different corruption rates, and their performance is tested on both clean and randomly corrupted datasets, as shown in Tabs. 4 to 6. The baseline refers to the method that clients train local models on their private datasets without a federated learning process. The experiments show that our method, AugHFL, significantly outperforms other existing methods under most settings, with many test accuracy even exceeding existing methods by more than 12%. Notably, our method performs similarly to SOTA methods in the ideal scenario of completely clean training and testing data, though this is unlikely in practical applications. The performance of all SOTA methods degrades severely as the corruption rates in training datasets increase from completely clean to completely corrupted. It can be observed that the average accuracy of the two classic algorithms FedMD[31] and FedDF[41] decreases by about 19% on clean datasets and 8% on corrupted datasets. The other two latest algorithms, RHFL [14] and FCCl [26]have an average accuracy drop of about 18% on clean datasets and 7% on corrupted datasets. Meanwhile, as the corruption rate increases from 0 to 1, AugHFL drops 4.2% accuracy on the clean dataset and 1.18% on the corrupted dataset, still achieving the highest accuracy among these algorithms. The above demonstrates the strong robustness of our proposed AugHFL against data corruption and the validity of the method in the real world.

5. Conclusion

This paper studies a novel and challenging robust heterogeneous federated learning problem with data corruption. We introduce a comprehensive framework AugHFL, to cope with this issue. First, we incorporate a robust data augmentation strategy to minimize the impact of intra-client data corruption. Second, we design PubAug, a mechanism that adaptively adjusts the client contributions based on their reliability, to address inter-client corruption during the collaborative update phase. Extensive experiments demonstrate the effectiveness of our method under different corrupted scenarios..

Acknowledgement. This work is partially supported by the Key Research and Development Program of Hubei Province (2021BAA187), National Natural Science Foundation of China under Grant (62176188), Zhejiang lab (NO.2022NF0AB01), the Special Fund of Hubei Luojia Laboratory (220100015) and CAAI-Huawei MindSpore Open Fund.

References

[1] Guozhong An. The effects of adding noise during backpropagation training on a generalization performance. *Neural computation*, pages 643–674, 1996. 3

[2] Rowel Atienza. Improving model generalization by agreement of learned representations from data augmentation. In *WACV*, pages 372–381, 2022. 2

[3] Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, et al. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *ICML*, pages 642–652, 2020. 2

[4] David Byrd and Antigoni Polychroniadou. Differentially private secure multi-party computation for federated learning in financial applications. In *ICAIF*, pages 16:1–16:9, 2020. 1

[5] Alexander Camuto, Matthew Willetts, Umut Simsekli, Stephen J Roberts, and Chris C Holmes. Explicit regularisation in gaussian noise injections. *NeurIPS*, 2020. 3

[6] Hongyan Chang, Virat Shejwalkar, Reza Shokri, and Amir Houmansadr. Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer. *NeurIPS*, 2019. 3

[7] Anda Cheng, Peisong Wang, Xi Sheryl Zhang, and Jian Cheng. Differentially private federated learning with local regularization and sparsification. In *CVPR*, pages 10122–10131, 2022. 1

[8] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. In *CVPR*, 2019. 4

[9] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR*, pages 3008–3017, 2020. 4

[10] Ittai Dayan, Holger R Roth, Aoxiao Zhong, Ahmed Harouni, Amilcare Gentili, Anas Z Abidin, Andrew Liu, Anthony Beardsworth Costa, Bradford J Wood, Chien-Sung Tsai, et al. Federated learning for predicting clinical outcomes in patients with covid-19. *Nature medicine*, 2021. 1

[11] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 4

[12] Samuel Dodge and Lina Karam. Understanding how image quality affects deep neural networks. In *QoMEX*, pages 1–6, 2016. 3

[13] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In *NeurIPS*, 2020. 1

[14] Xiuwen Fang and Mang Ye. Robust federated learning with noisy and heterogeneous clients. In *CVPR*, pages 10062–10071, 2022. 2, 3, 5, 6, 8

[15] David Froelicher, Juan R Troncoso-Pastoriza, Jean Louis Raisaro, Michel A Cuendet, Joao Sa Sousa, Hyunghoon Cho, Bonnie Berger, Jacques Fellay, and Jean-Pierre Hubaux. Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption. *Nature communications*, 2021. 1

[16] Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *CVPR*, pages 10112–10121, 2022. 1

[17] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2019. 4

[18] Caglar Gulcehre, Marcin Moczulski, Misha Denil, and Yoshua Bengio. Noisy activation functions. In *ICML*, pages 3059–3068, 2016. 3

[19] Hongyu Guo, Yongyi Mao, and Richong Zhang. Mixup as locally linear out-of-manifold regularization. In *AAAI*, pages 3714–3722, 2019. 4

[20] Chaoyang He, Murali Annavaram, and Salman Avestimehr. Group knowledge transfer: Federated learning of large cnns at the edge. *NeurIPS*, 2020. 3

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6

[22] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019. 6

[23] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *ICML*, pages 2712–2721, 2019. 2, 3

[24] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *ICLR*, 2020. 4, 6

[25] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *NeurIPS*, 2015. 3

[26] Wenke Huang, Mang Ye, and Bo Du. Learn from others and be yourself in heterogeneous federated learning. In *CVPR*, pages 10133–10143, 2022. 3, 6, 8

[27] Georgios Kaissis, Alexander Ziller, Jonathan Passerat-Palmbach, Théo Ryffel, Dmitrii Usynin, Andrew Trask, Ionésio Lima, Jason Mancuso, Friederike Jungmann, Marc-Matthias Steinborn, et al. End-to-end privacy preserving deep learning on multi-institutional medical imaging. *Nature Machine Intelligence*, pages 473–484, 2021. 1

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 7

[29] Klim Kireev, Maksym Andriushchenko, and Nicolas Flammarion. On the effectiveness of adversarial training against common corruptions. In *UAI*, pages 1012–1021, 2022. 3

[30] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009. 6

[31] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. In *NeurIPS Workshop*, 2019. 2, 3, 6, 8

[32] Jingtao Li, Adnan Siraj Rakin, Xing Chen, Zhezhi He, Deliang Fan, and Chaitali Chakrabarti. Ressfl: A resistance transfer framework for defending model inversion attack in split federated learning. In *CVPR*, pages 10194–10202, 2022. 1

[33] Shenghui Li, Edith Ngai, Fanghua Ye, and Thiemo Voigt. Auto-weighted robust federated learning with corrupted data sources. *ACM TIST*, pages 1–20, 2022. 3

[34] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, pages 50–60, 2020. 1

[35] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *MLSys*, 2020. 1, 3

[36] Xin-Chun Li, Yi-Chu Xu, Shaoming Song, Bingshuai Li, Yinchuan Li, Yunfeng Shao, and De-Chuan Zhan. Federated learning with position-aware neurons. In *CVPR*, pages 10082–10091, 2022. 1

[37] Zhuohang Li, Jiaxin Zhang, Luyang Liu, and Jian Liu. Auditing privacy defenses in federated learning via generative gradient leakage. In *CVPR*, pages 10132–10142, 2022. 1

[38] Soon Hoe Lim, N Benjamin Erichson, Liam Hodgkinson, and Michael W Mahoney. Noisy recurrent neural networks. In *NeurIPS*, pages 5124–5137, 2021. 2

[39] Soon Hoe Lim, N Benjamin Erichson, Francisco Utrera, Winnie Xu, and Michael W Mahoney. Noisy feature mixup. *ICLR*, 2021. 3

[40] Jierui Lin, Min Du, and Jian Liu. Free-riders in federated learning: Attacks and defenses. *arXiv preprint arXiv:1911.12560*, 2019. 2

[41] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. 2020. 2, 3, 6, 8

[42] Xuanqing Liu, Tesi Xiao, Si Si, Qin Cao, Sanjiv Kumar, and Cho-Jui Hsieh. Neural sde: Stabilizing neural ode networks with stochastic noise. *arXiv preprint arXiv:1906.02355*, 2019. 2

[43] Xuanqing Liu, Tesi Xiao, Si Si, Qin Cao, Sanjiv Kumar, and Cho-Jui Hsieh. How does noise help robustness? explanation and exploration under the neural sde framework. In *CVPR*, pages 282–290, 2020. 2

[44] Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin D Cubuk. Improving robustness without sacrificing accuracy with patch gaussian augmentation. *arXiv preprint arXiv:1906.02611*, 2019. 2

[45] Xiaosong Ma, Jie Zhang, Song Guo, and Wenchao Xu. Layer-wised model aggregation for personalized federated learning. In *CVPR*, pages 10092–10101, 2022. 1

[46] Disha Makhija, Xing Han, Nhat Ho, and Joydeep Ghosh. Architecture agnostic federated learning for neural networks. In *ICML*, pages 14860–14870, 2022. 1

[47] Disha Makhija, Xing Han, Nhat Ho, and Joydeep Ghosh. Architecture agnostic federated learning for neural networks. *ICML*, pages 14860–14870, 2022. 3

[48] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, pages 1273–1282, 2017. 1, 3

[49] Matias Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, Zhengming Ding, and Chen Chen. Local learning matters: Rethinking data heterogeneity in federated learning. In *CVPR*, pages 8397–8406, 2022. 1

[50] Siyu Qi, Lahiru D Chamain, and Zhi Ding. Hierarchical training for distributed deep learning based on multimedia data over band-limited networks. In *ICIP*, pages 2871–2875, 2022. 2

[51] Liangqiong Qu, Yuyin Zhou, Paul Pu Liang, Yingda Xia, Feifei Wang, Ehsan Adeli, Li Fei-Fei, and Daniel Rubin. Rethinking architecture design for tackling data heterogeneity in federated learning. In *CVPR*, pages 10061–10071, 2022. 1

[52] Evgenia Rusak, Lukas Schott, Roland S Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. A simple way to make neural networks robust against diverse image corruptions. In *ECCV*, pages 53–69, 2020. 3

[53] Shahd Safarani, Arne Nix, Konstantin Willeke, Santiago Cadena, Kelli Restivo, George Denfield, Andreas Tolias, and Fabian Sinz. Towards robust vision by multi-task learning on monkey visual cortex. In *NeurIPS*, pages 739–751, 2021. 2

[54] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018. 6

[55] Felix Sattler, Arturo Marban, Roman Rischke, and Wojciech Samek. Communication-efficient federated distillation. *arXiv preprint arXiv:2012.00632*, 2020. 3

[56] Yiqing Shen, Yuyin Zhou, and Lequan Yu. Cd2-pfed: Cyclic distillation-guided channel decoupling for model personalization in federated learning. In *CVPR*, pages 10041–10050, 2022. 1

[57] Dimitris Stripelis, Marcin Abram, and Jose Luis Ambite. Performance weighting for robust federated learning against corrupted sources. *arXiv preprint arXiv:2205.01184*, 2022. 3

[58] Teppei Suzuki. Teachaugment: Data augmentation optimization using teacher knowledge. In *CVPR*, pages 10894–10904, 2022. 3

[59] Minxue Tang, Xuefei Ning, Yitu Wang, Jingwei Sun, Yu Wang, Hai Li, and Yiran Chen. Fedcor: Correlation-based active client selection strategy for heterogeneous federated learning. In *CVPR*, pages 10102–10111, 2022. 1

[60] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Between-class learning for image classification. In *CVPR*, pages 5486–5494, 2018. 4

[61] Dmitrii Usynin, Alexander Ziller, Marcus Makowski, Rickmer Braren, Daniel Rueckert, Ben Glocker, Georgios Kaissis, and Jonathan Passerat-Palmbach. Adversarial interference and its mitigations in privacy-preserving collaborative machine learning. *Nature Machine Intelligence*, pages 749–758, 2021. 1

[62] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *ICML*, pages 6438–6447, 2019. 3

[63] Chunnan Wang, Xiang Chen, Junzhe Wang, and Hongzhi Wang. Atpfl: Automatic trajectory prediction model design under federated learning framework. In *CVPR*, pages 6563–6572, 2022. 1

[64] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. Communication-efficient federated learning via knowledge distillation. *Nature communications*, 2022. 1

[65] Donglei Wu, Xiangyu Zou, Shuyu Zhang, Haoyu Jin, Wen Xia, and Binxing Fang. Smartidx: Reducing communication cost in federated learning by exploiting the cnns structures. In *AAAI*, pages 4254–4262, 2022. 3

[66] Qiong Wu, Kaiwen He, and Xu Chen. Personalized federated learning for intelligent iot applications: A cloud-edge based framework. *IEEE OJ-CS*, pages 35–44, 2020. 1

[67] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *CVPR*, pages 10684–10695, 2020. 3

[68] Jingyi Xu, Zihan Chen, Tony QS Quek, and Kai Fong Ernest Chong. Fedcorr: Multi-stage federated learning for label noise correction. In *CVPR*, pages 10184–10193, 2022. 1

[69] Mang Ye, Xiuwen Fang, Bo Du, Pong C Yuen, and Dacheng Tao. Heterogeneous federated learning: State-of-the-art and research challenges. *arXiv preprint arXiv:2307.10616*, 2023. 1

[70] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6022–6031, 2019. 4

[71] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 3, 4

[72] Jie Zhang, Song Guo, Xiaosong Ma, Haozhao Wang, Wenchao Xu, and Feijie Wu. Parameterized knowledge transfer for personalized federated learning. *NeurIPS*, pages 10092–10104, 2021. 1, 3

[73] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, pages 6848–6856, 2018. 6