FuseFL: One-Shot Federated Learning through the Lens of Causality with Progressive Model Fusion

Zhenheng Tang[†*] **Yonggang Zhang**[†] **Peijie Dong**[♯] **Yiu-ming Cheung**[†]
Amelie Chi Zhou[†] **Bo Han**[†] **Xiaowen Chu**[♯,§]
[†] Department of Computer Science, Hong Kong Baptist University
[♯] DSA Thrust, The Hong Kong University of Science and Technology (Guangzhou)
{zhtang, ygzhang, ymc, amelieczhou, bhanml}@comp.hkbu.edu.hk
{pdong212, xwchu}@connect.hkust-gz.edu.cn

Abstract

One-shot Federated Learning (OFL) significantly reduces communication costs in FL by aggregating trained models only once. However, the performance of advanced OFL methods is far behind the normal FL. In this work, we provide a causal view to find that this performance drop of OFL methods comes from the isolation problem, which means that locally isolatedly trained models in OFL may easily fit to spurious correlations due to data heterogeneity. From the causal perspective, we observe that the spurious fitting can be alleviated by augmenting intermediate features from other clients. Built upon our observation, we propose a novel learning approach to endow OFL with superb performance and low communication and storage costs, termed as `FuseFL`. Specifically, `FuseFL` decomposes neural networks into several blocks and progressively trains and fuses each block following a bottom-up manner for feature augmentation, introducing no additional communication costs. Comprehensive experiments demonstrate that `FuseFL` outperforms existing OFL and ensemble FL by a significant margin. We conduct comprehensive experiments to show that `FuseFL` supports high scalability of clients, heterogeneous model training, and low memory costs. Our work is the first attempt using causality to analyze and alleviate data heterogeneity of OFL[2].

1 Introduction

Federated learning (FL) [96; 67] has become a popular paradigm that enables collaborative model training without sharing private datasets from clients. Two typical characteristics of FL limit its performance: (1) FL normally has non-IID (Independently and Identically Distributed) data, also called *data heterogeneity* [67], which causes unstable slow convergence [68; 146; 124; 135] and poor model performance [157; 94; 134; 162; 150]; (2) The extremely low bandwidth, e.g. $1 \sim 10$ MB/s of FL in Internet environments [67; 133; 129; 128; 132; 129], leads to high communication time of a large neural network. For example, communicating once ResNet-50 [51] with 25.56M parameters (102.24MB) or GPT-3 [14] with 175B parameters (700GB) will consume around 102.24 seconds or 194 hours, respectively. Current FL methods alleviate this problem by skipping the gradient synchronization of traditional distributed training to save communication costs [96; 67]. But the required hundreds or thousands of communication rounds still make the communication time unacceptable.

[*]This work is partially done during the visiting in The Hong Kong University of Science and Technology (Guangzhou).

[§] Correspondence to Xiaowen Chu (xwchu@hkust-gz.edu.cn).

[2]The code is publicly available: `https://github.com/wizard1203/FuseFL`

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

To reduce the communication costs at extreme, one-shot FL (OFL) [159; 38; 81; 164; 25; 23] only communicates the local trained model once. Thus, the communication cost is the model size S for each client, less than FedAvg-style algorithms for hundreds or thousands of times. However, averaging for only once cannot guarantee the convergence of FedAvg. Thus, the direct idea is to aggregate client models on the server and conduct inference as ensemble learning does. Some advanced works also consider better model averaging [63; 112; 90; 6], neurons matching [5; 141], selective ensemble learning [26; 52; 143], model distillation [81; 164; 25; 23]. These methods may be impractical due to the requirements of additional datasets with privacy concerns, and the extra large storage or computation costs. Most importantly, there still exists a large performance gap between OFL and the normal FL or the ensemble learning. This motivates the following question:

> *How to improve FL performance under* ***extremely low communication costs*** *with almost no extra computational and storage costs?*

In this work, we provide a *causal view* [109; 110; 3; 119] to analyze the performance drop of OFL. We firstly construct a causal graph to model the data generation process in FL, where the spurious features build up the data heterogeneity between clients, and invariant features of the same class remain constant in each client (domain) [3; 119; 22; 151; 161]. Then, we show the performance drop comes from the *isolation problem*, which means that locally isolatedly trained models in OFL may easily fit to spurious correlations like adversarial shortcuts [40; 35; 53], instead of learning invariant features [3; 119], causing a performance drop of OFL on the test dataset. Consider a real-world example, Alice takes photos of birds in the forests, while Bob near the sea. Now, the isolated models will mistakenly identify birds according to the forests or the sea [53; 35]. Based on the causal graph, we intuitively and empirically show that such spurious fitting can be alleviated by augmenting intermediate features from other clients (Section 3).

Built upon this observation, we propose a simple yet effective learning approach to realizing OFL with superb performance and extremely low communication and storage costs, termed as `FuseFL`, which builds up the global model through bottom-up training and fusion to improve OFL performance (Section 4). Specifically, we split the whole model into multiple blocks [3] (The "block" means a single or some continuous layers in a DNN.). For each block, clients first train and share their local blocks with others; then, these trained local blocks are assembled together, and the features outputted from these blocks are fused together and fed into the next local blocks. This process is repeated until the whole model is trained. Through this bottom-up training-and-fusion method, local models can learn better feature extractors that learn more invariant features from other clients to avoid the isolation problem. To avoid the large storage cost, given the number of clients M, we assign each local client with a small model with reduced hidden dimension with ratio $\sqrt{M}$, to ensure the final learned global model has the same size S as the original model.

Our main contributions can be summarized as follows:

- We provide a causal view to understand the gap between multi-round FL and OFL, showing that augmenting intermediate features from other clients contributes helps improve OFL. As far as we know, this is the first work using causality to analyze the data heterogeneity of OFL.
- To leverage causality to improve OFL, we design `FuseFL`, which decomposes models into several modules and transmits one module for feature augmentation at each communication round.
- We conduct comprehensive experiments to show how `FuseFL` significantly promotes the performance of OFL without *no* additional communication and computation cost.

2 Preliminary

2.1 Federated Learning

In FL, a set of clients $\mathcal{M} = \{m | m \in 1, 2, ..., M\}$ have their own dataset $\mathcal{D}_m$. Given C classes indexed by $[C]$, a sample in $\mathcal{D}_m$ is denoted by $(x, y) \in \mathcal{X} \times [C]$, where x is the input in the space $\mathcal{X}$ and y is its corresponding label. These clients cooperatively learn a model $F(\theta, x) : \mathcal{X} \to \mathbb{R}^C$ that is

[3] Note that this method is general to any neural network, and the fusion is different from averaging.

parameterized as $\theta \in \mathbb{R}^d$. Formally, the global optimization problem can be formulated as [96]:

$$\min_{\theta} L(\theta) \triangleq \sum_{m=1}^{M} p_m L_m(\theta) = \sum_{m=1}^{M} p_m \mathbb{E}_{(x,y)\sim\mathcal{D}_m} \ell(F; x, y),$$

where the local objective function of mth-client $\ell(F; x, y) \triangleq CE(\hat{y}, y)$ with $\hat{y} \triangleq F(\theta; x)$, CE denotes the cross-entropy loss, $p_m > 0$ and $\sum_{m=1}^{M} p_m = 1$. Usually, $p_m \triangleq n_m/N$, where n_m denotes the number of samples on client m ($n_m = |\mathcal{D}_m|$) and $N = \sum_{m=1}^{M} n_m$.

The classic FL algorithm is FedAvg [96]. In each communication round t, the central server randomly samples a part of clients $\mathcal{S}^t \subseteq \mathcal{M}$ and broadcasts the model θ^t to all selected clients, and then each m-th client performs multiple local updates. After local training, all selected clients send the optimized $\theta^t_{m,E}$ to the server, and the server aggregates and averages local models to obtain a global model. Such a multi-round communication introduces large communication costs [67].

2.2 Ensembled FL

The FedAvg requires multiple communication rounds T for convergence [146; 83], which might be extremely large [67]. Given the model size S, FedAvg-style FL methods introduce communication costs as $T \times S$. As shown in Table 1, the current lowest communication cost of FL is reduced as S in OFL, making FL possibly deployable in low communication bandwidth scenarios [133; 129]. Thus, we analyze what causes the performance drop of OFL and how to improve it. As the performance of average-based and model distillation OFL methods is upper bounded by ensemble learning [159; 38; 81; 164; 23], we mainly focus on analyzing ensemble learning and differentiating `FuseFL` from it. The output of the ensemble learning can be formalized as: $F_{\text{ens}}(x) \triangleq \frac{1}{M}\sum_{m\in\mathcal{M}} F_m^{\text{loc}}(\theta_m; x)$, in which the local model F_m^{loc} parameterized with θ_m is isolatedly trained with minimizing empirical risk minimization (ERM) objective [152; 3; 22] function $\ell(F(\theta, x), y), (x, y) \sim \mathcal{D}_m$ by SGD.

3 Federated Learning: A Causal View

3.1 The Sequential Structure of Neural Networks

A neural network can be decomposed into the sequential module-wise structure as shown in Figure 1. Formally, it can be defined as:

$$F = \Lambda \circ H^K \circ H^{K-1} \circ \cdots \circ H^1, \text{for } 1 \le k \le K, \tag{1}$$

where Λ is the final classifier, and H_k is the module that may consist of single or multiple blocks. The Λ and each H_k are parameterized by $\theta^\Lambda \in \mathbb{R}^{d_\Lambda}$ and $\theta^k \in \mathbb{R}^{d_k}$. The $H^i \circ H^j(\cdot)$ means $H^i(H^j(\cdot))$. Thus, the parameter θ of F are concatenated by the θ^Λ and $\{\theta^k | k \in 1, 2, \cdots K\}$, and $d_\Lambda + \sum_{k=1}^{K} d_k = d$.

As Figure 1 illustrates, each module H^k receives the output of module H^{k-1}, and the final classifier receives the output of the final hidden module H^K and makes predictions $\hat{y} = f(x)$ on the input x. We call the output from each module, $h^k = H^k(h^{k-1})$ and $h^1 = H^1(x)$, as the feature for simplicity.

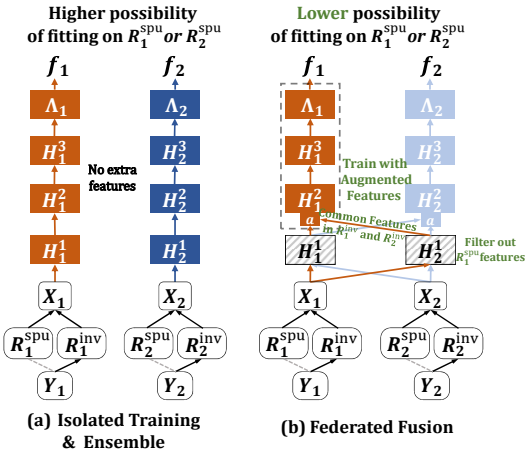


Figure 1: Structure Equation Model [109] of FL.

3.2 Structure Equation Model of FL

Inspired from the analysis of out-of-distribution (OOD) generalization [3] through the lens of mutual information [122; 2] and structure equation

model (SEM) in causality [110; 3; 119; 161], we define the data generation SEM of FL as shown in Figure 1. For local training dataset $\mathcal{D}_m$ at client m, the SEM is $Y_m \rightarrow R_m^{\text{inv}} \rightarrow X_m \leftarrow R_m^{\text{spu}}$, where R_m^{inv} and R_m^{spu} are invariant and spurious features, Y_m and X_m are label and input data respectively. Here, the dataset $\mathcal{D}_m$ is a subset of the whole dataset $\mathcal{D}$. The R_m^{spu} is actually the nuisance at a global level (respect to Y), being independent of Y, but dependent on X_m.

Non-IID data and causality. For a groundtruth label Y, its corresponding invariant features R_m^{inv} do not change across clients [3; 119; 22; 151]. However, the spurious features R_m^{spu} are other factors that occasionally exist in data and do not have a relationship to Y, which means that the heterogeneous features of data (non-iid) with the same class come from the spurious features R_m^{spu} (concept shift [67]). For example, in photos of birds in the forests or the sea, pixels of birds are R_m^{inv} while the forests and the sea are R_m^{spu}. Considering the test dataset includes all client data distribution and even OOD data, the Y_{test} is largely dependent on $R_{\text{test}}^{\text{inv}}$: $P(Y_{\text{test}}|X_{\text{test}}, R_{\text{test}}^{\text{inv}}) \gg P(Y_{\text{test}}|X_{\text{test}}, R_{\text{test}}^{\text{spu}})$[4].

Non-IID scenarios. This SEM model considers the label shift ($p_i(y) \neq p_j(y)$) and concept shift ($p_i(x|y) \neq p_j(x|y)$) scenarios [67; 80], or both of them appear simultaneously. When the support[5] $\mathcal{Y}_m$ of Y_m is different or partly overlapped between clients $m = 1, ..., M$, this would be the severe non-IID scenario [155]. And it is obvious that spurious features R_m^{spu} relate to the concept shift.

Spurious fitting. By conducting isolated local training on local dataset $\mathcal{D}_m$ at client m, the model F_m^{loc} is prone to learn to predict Y_m based on spurious features R_m^{spu}, i.e. low distance $d_{\text{loc},m}^{\text{spu}} = d(P(Y_m|X_m, R_m^{\text{spu}}), P(F_m^{\text{loc}}|X_m, R_m^{\text{spu}}))$ but high distance $d(P(Y_m|X_m, R_m^{\text{inv}}), P(F_m^{\text{loc}}|X_m, R_m^{\text{inv}}))$, in which the distance d could be CE loss or KL divergence. The reason for the spurious fitting by isolated training is that the invariant features $R_{i\neq m}^{\text{inv}}$ from other clients are not observed by client m, while the R_m^{spu} frequently appears in the local dataset $\mathcal{D}_m$ like the adversarial attacks or shortcuts [40; 35; 53]. This guarantees low error on the training dataset $\mathcal{D}_m$, because it has much less data than $\mathcal{D}_{\text{test}}$ and $\mathcal{D}_{1,...,M}$, thus introducing high probability $P(Y_m|X_m, R_m^{\text{spu}})$. However, on test dataset $\mathcal{D}_{\text{test}}$, the low $d_{\text{loc},m}^{\text{spu}}$ of model F_m^{loc} but high $d_{\text{loc},m}^{\text{inv}}$ of model F_m^{loc} leads to high test error. Different from isolated training, FedAvg alleviates this problem by multiple times of averaging models to find those common features, including more $R_{1,...,M}^{\text{inv}}$ and removing $R_{1,...,M}^{\text{spu}}$.

Feature augmentation. Through the above analysis, the key to improve OFL performance is to endow OFL with the ability of training to see invariant features across all clients. It has been found that training on noised datasets with SGD to optimize ERM can still result in some feature representations consisting of both spurious and invariant features; exploiting the invariant features is the key to helping improve OOD performance [158; 3; 8]. In light of this, we introduce augmenting features by fusing client models block by block. Concisely speaking, in `FuseFL`, each local model can conduct local training with the view from other clients $H_{i\neq m}(X_m)$, which helps filter out R_m^{spu} but retain R_m^{inv}, as other clients cannot see R_m^{spu} in their dataset $\mathcal{D}_{i\neq m}$. This method can be seen as a kind of invariant feature augmentation [22]. The details of `FuseFL` are shown in Section 4.

3.3 Mutual Information

The goal of FL is to obtain a model that performs well on all client datasets [67]. Thus, here we consider the random variable X, Y sampled from the global dataset $\mathcal{D}$. In this section, we also write H^k as the features that output from $H^k(H^{k-1} \dots (H^1(X)))$ for simplicity.

Given the probabilistic graph model $(R^{\text{spu}}, Y) \rightarrow X \rightarrow H^1 \rightarrow \cdots \rightarrow H^k \rightarrow F(X)$ (Eq. 1), where R^{inv} are ignored for simplicity, the MI between Y and subsequent transformations H^k on X satisfies a decreasing trend: $I(X;Y) \geq I(H^1;Y) \geq \cdots \geq I(H^K;Y)$; the MI between X and subsequent transformations on X satisfies a decreasing trend: $Entropy(X) \geq I(H^1;X) \geq \cdots \geq I(H^K;X)$ [122]. If $I(H^K;R^{\text{spu}}) = 0$ and H^K can predict labels, H^K is called invariant features so that the final classifier will not overfit to spurious correlations between R^{spu} and Y. The previous works [122] show that achieving the following minimal sufficient statistic provides good

[4]Without loss of generality, one can also model the SEM of test data as $Y_{\text{test}} \rightarrow R_{\text{test}}^{\text{inv}} \rightarrow X_{\text{test}}$ for simplicity.

[5]Support is the region where the probability density for continuous random variables (probability mass function for discrete random variables) is positive [3].

generalization:

$$\textbf{Sufficient statistic}: I(X;Y) = I(H(X);Y), \tag{2}$$

$$\textbf{Minimal statistic}: H(X) = \arg\min_{\tilde{H}(X)} I(\tilde{H}(X);X). \tag{3}$$

Lemma 3.1 (Invariance and minimality [2])**.** *Given spurious feature R^{spu} for the label Y, and probabilistic graph model $(R^{spu}, Y) \to X \to H(X)$, then,*

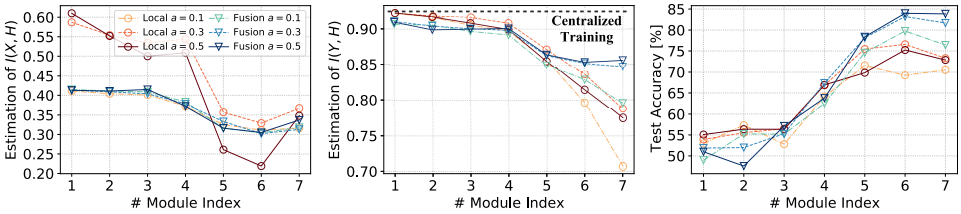
$$I(H(X);R^{spu}) \leq I(H(X);X) - I(X;Y).$$

There is a nuisance R^{spu} such that equality holds up to a residual ϵ:

$$I(H(X);R^{spu}) = I(H(X);X) - I(X;Y) - \epsilon,$$

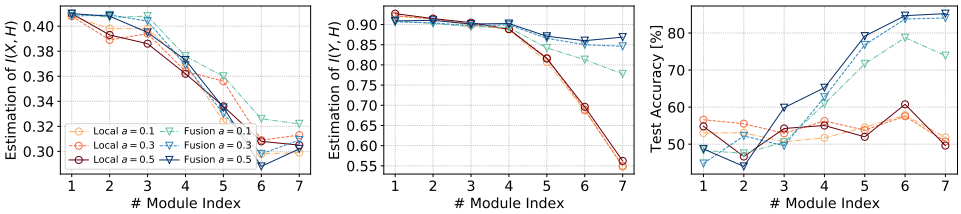
where $\epsilon \triangleq I(H(X);Y|R^{spu}) - I(X;Y)$. The sufficient statistic $H(X)$ (satisfying Eq. 3) is invariant to R^{spu} if and only if it is minimal (satisfying Eq. 2).

Remark 3.1. Based on Lemma 3.1, we can study how $I(H(X);X)$ and $I((H);Y)$ changes to study to what degree the $H(X)$ contains spurious features.



(a) Estimated MI $I(H^k;X)$. (b) Estimated MI $I(H^k;Y)$. (c) The separability of layers.

Figure 2: Estimated MI and separability of trained models with non-IID datasets.



(a) Estimated MI $I(H^k;X)$. (b) Estimated MI $I(H^k;Y)$. (c) The separability of layers.

Figure 3: Estimated MI and separability of trained models with non-IID backdoored datasets.

Empirical study. We empirically estimate the MI $I(H^k_{\text{loc}}, X)$ and $I(H^k_{\text{loc}}, Y)$ of isolated local trained features, and the $I(H^k_{\text{fus}}, X)$ and $I(H^k_{\text{fus}}, Y)$ of augmented features. As the deep neural networks (DNNs) show layer-wise feature enhancements [7], we also measure the linear separability [7; 101] of features H^k_{loc} and H^k_{fus} to see how they change. Details of MI estimation and linear separability are shown in Appendix D.3 and D.4. The experiments are conducted by training ResNet-18 with CIFAR-10 [74] partitioned across $M = 5$ clients. Figure 2 shows the local features H^k_{loc} have significantly higher $I(H^k, X)$ but lower $I(H^k, Y)$ than augmented features H^k_{fus}. With the increased non-IID degree (lower a), the $I(H^k, Y)$ decreased further, demonstrating that the local feature H^k_{loc} fits on a more anti-causal relationship between R^{spu}_m and Y_m. The fused high-level features show better linear separability. And H^k_{fus} is more robust to R^{spu}_m.

Except for the natural spurious features that exist in CIFAR-10, we also study the effect of spurious features by handcraft. Specifically, we inject backdoored data samples [10; 97] of 1 out of 5 clients as $\mathcal{D}^{\text{back}}_1$, in which the images have handcrafted textures generated according to the labels as a strong anti-causal relation. Details of backdoored datasets are introduced in Appendix D.5. Figure 3 shows that the backdoor features lead to information loss in X. And the backdoored data samples further aggravate the information loss of label Y. The isolated local trained features retain significantly less $I(H^k, Y)$ than augmented features.

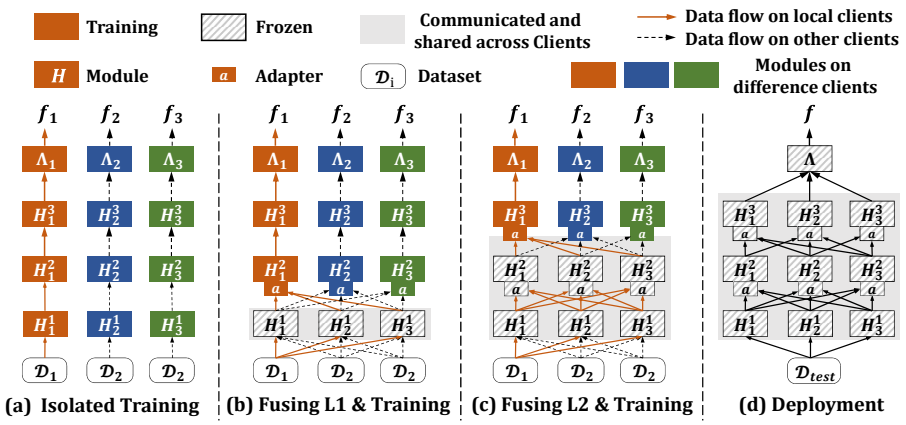


Figure 4: (a) Initially, all layers are isolated training. Note that the layer here does not only mean one or Conv layer, but generally refers to a neural network block that can consist of multiple layers. (b) Then, all first blocks (**L1**) of different clients are communicated, shared and frozen among clients. Then, the adaptors are added behind the fused block, to fuse features outputted from the concatenated local blocks. (c) Train the third blocks (**L3**) follow the similar process in (b). (d) inference process of `FuseFL`. The larger squares represent the original training block in local models. The smaller squares are adaptors that fuse features from previous modules together, which are 1×1 Conv kernels or simple average operations with little or no memory costs. Note that (a) also represents local training in ensemble FL, where different clients train models on local datasets.

4 FuseFL: Progressive FL Model Fusion

Motivated by analysis in Section 3, we propose augmenting local intermediate features $H_m^{1,\dots,K}$ on client m, which helps reduce fitting to a spurious correlation between Y_m and R_m^{spu}. However, direct fusing features together to make predictions faces the following problems.

Altered feature distribution. During local training, the local subsequent model $\Gamma_m^{k+1} \triangleq \Lambda_m \circ H_m^K \cdots \circ H_m^{k+1}$ after block k on client m is trained based on local features H_m^k. After feature fusion, changed local features lead to feature drifts [75; 134].

Mismatched semantics. Each local feature H_m^k has totally different distributions, scales, or even dimensions; thus, directly averaging may cause useful features to be overwhelmed or confused by noisy features.

4.1 Train, Fuse, Freeze, and Re-Train

As Figure 4 and Algorithm 1 (Appendix 1) show, the main loop of `FuseFL` including training, fusing, and freezing, which is repeated for all K split blocks following a progressive manner. Note that for H_m^1, there is no layer fusion, which is isolatedly trained.

Fuse. After each local training step, clients share their H_m^k with other clients, and fuse them following Eq. 4. An adaptor A is stitched before H_m^{k+1} (Section 4.2). Then the local model becomes as $F_{\text{fus}}^{k,m}$ following Eq. 5.

Freeze and re-train. To address the altered feature distribution problem ($A(H_{\text{fus}}^k(x)) \to H_m^k(x)$), for each step k on client m, the subsequent layers Γ_m^{k+1} will be trained again based on $\mathcal{D}_m$. Thus, the Γ_m^{k+1} can learn from all low-level features of all clients. Note that we do not need to train each block k with the same epochs in isolated training, because the DNNs naturally follow the layer-wise convergence [113; 48].

$$H^k_{\text{fus}}(x) = \left[H^k_1(x), ..., H^k_M(x)\right]. \tag{4}$$

$$F^{k,m}_{\text{fus}} = \Lambda_m \circ H^K_m \circ ... \circ H^{k+1}_m \circ H^k_{\text{fus}} \circ ... \circ H^1_{\text{fus}}. \tag{5}$$

$$F_{\text{fus}} = \Lambda \circ H^K_{\text{fus}} \circ H^{K-1}_{\text{fus}} \circ ... \circ H^1_{\text{fus}}. \tag{6}$$

$$A_{\text{avg}} = \text{Average}(H^k_{\text{fus}}(x)). \tag{7}$$

$$A_{\text{conv}} = \text{Conv}(\text{Concat}(H^k_{\text{fus}}(x))). \tag{8}$$

By freezing fused blocks and retraining high-level models, the another benefit is to enforce the SGD to use previous features from other clients to continue tuning the high-level model. The previous local trained high-level models may overfit on shortcut features from the noisy data. This insight is also utilized in defensing adversarial attacks [139; 107].

4.2 Feature Adaptation

To address the mismatched semantics problem, the intuitive approach is to preserve the original feature structures through the concatenation of all features to the next block. However, this leads to new problems: (1) requiring modification of subsequent modules; (2) feature size explosion of subsequent blocks by $O(M^K)$. To address these two problems, we introduce an adaptor stitched before local modules ($k > 1$), and training together with Γ^k_m. As an initial trial to operationalize `FuseFL`, we utilize conv1×1 as the adapter as Eq. 8. We also verify the use of average as an adapter (Eq. 7) in experiments (Section 6).

4.3 Benefits of `FuseFL` Design

Mitigating fitting on spurious correlations. During the local training on datasets with spurious features, the final learned representations with ERM still contain some invariant features [3; 22]. Thus, some work proposes to finetune the classifier based on data samples with invariant features to let the classifier make predictions based on invariant features [70; 61; 106]. Similar to this motivation, we hope to incorporate other client modules as auxiliary feature extractors to generate more invariant features of local data during training subsequent layers. Figure 1 describes the mechanism that using other local models $H_{i\neq m}(X_m)$ help to filter out spurious features R^{spu}_m, but retain R^{inv}_m. As other clients $\{i \neq m\}$ cannot see R^{spu}_m in their dataset $\mathcal{D}_{i\neq m}$ during local training, only invariant features can pass through $H_{i\neq m}(X_m)$. This method can be seen as the invariant feature augmentation [22].

Saving storage and communication costs than ensemble FL. Similar to ensemble learning, directly collecting and fusing local models together will enlarge the total model size from S to $S \times M$. Note that `FuseFL` actually builds up a global model with blocks fused together, with the hidden dimensions (channels) enlarged from $n_s \rightarrow n_s \times M$. Thus, intuitively, we can reduce the hidden dimension of the local model n_f to reduce the memory requirements. Interestingly, with a scaling ratio γ, when scaling all local linear or convolutional layers, each matrix should be scaled on *both input and output* dimension as $n_f = \gamma n_s$. The ratio of memory costs between `FuseFL` and the original single model is $r_m = M \times n_s^2/(\gamma n_s)^2$. To obtain $r_m = 1$, we obtain the scaling ratio $\gamma = \sqrt{M}$, which means that `FuseFL` can keep similar memory requirements with the original model size S with reducing hidden dimensions as ratio $\sqrt{M}$, demonstrating good theoretical scalability to the M. We will verify this in experiments (Section 6.2).

Privacy concerns. `FuseFL` only shares layers between clients, which aligns with other classic and advanced FL methods in all directions mentioned in Section 5.

Support of heterogeneous models. The block in `FuseFL` does not mean a single linear or convolution layer, but a general module that can consist of any DNN, thus supporting FL with heterogeneous models (see experiments 6.2). The adaptor can be designed to transform features of different shapes to align with the input of the next local block.

Layer-wise training to reduce training epochs. Because each communication round means multiple local training epochs. To keep the total training epochs the same as the one-shot FedAVG (represented as E), we assign the local training epochs of the `FuseFL` as E/K. Thus, the number of total training epochs of FuseFL is the same as other OFL methods. The core insight of this design can be referred to as the progressive freezing during training DNNs [113].

5 Related Works

5.1 Data Heterogeneity in FL

The notorious non-IID data distribution in FL severely harms the convergence rate and model performance of FL. The **model regularization** proposes to add a penalty of distances between local and global models [117; 1]. **Feature calibration** aligns feature representations of different clients in similar spaces [29; 134; 60; 135]. FedMA [141] exploits a layer-wise communication and averaging methods, in which the aggregation is conducted on fine-grained layer. Thus, its linear dependence of computation and communication on the network's depth, is not suitable for deeper models, which introduces large computation costs in re-training and more communication rounds [19; 58]. Unlike FedMA, FuseFL introduces block-wise communication and aggregation with much less communication rounds and computation costs. Furthermore, due to the matching and averaging aggregation, FedMA only supports linear or Conv layers, which severely limits its practical usage. By viewing the separated block as a black box and concatenating output features, FuseFL can successfully support merging any kind of neural layer. Some works on fairness analysis in FL also relate to this work in perspective of local and global characteristics [42; 32].

Table 1: Demystifying different FL algorithms. T represents communication rounds, S the model size, M the number of clients. The "Centralized" means training the model with all datasets aggregated with SGD. "Comm." means communication.

Method	Comm. cost	Storage cost	Support model heterogeneity	Not require extra data
FedAvg	$T \times S$	S	✓	✓
Average-based OFL	S	S	✗	✗
Ensemble-based OFL	S	$S \times M$	✓	✗
Model distillation OFL	S	S	✓	✗
FuseFL	S	S	✓	✓

5.2 One-shot FL

One-shot FL [159; 38; 81; 164; 25] reduces communication costs from $T \times S$ to S by communicating with only one round. **Average-based** methods focus on better averaging client models, like Fisher information [63; 112], bayesian optimization [90; 6] or matching neurons [5; 141]. However, the non-linear structure of DNNs makes it difficult to obtain a comparable global model through averaging. **Ensemble-based** methods make prediction based on all or selected client models [26; 52; 143], but requires additional datasets with privacy concerns. And they have low scalability of the number of clients due to the storage of client models. **Model distillation** uses the public [81] or synthesized datasets [164; 25] to distill a new model based on ensemble models [38; 81]. These methods may be impractical in data-sensitive scenarios, such as medical and education, or continuous learning scenarios [28; 27]. Furthermore, there exists a large performance gap between these methods and the ensemble learning.

Due to the limited space, we leave the detailed reviews in Table 8 and Appendix C. Table 1 concisely demystifies different FL methods in terms of communication cost, storage cost, model performance, and whether supporting model heterogeneity or requiring extra data.

6 Experiments

6.1 Experiment Setup

Federated Datasets and Models. In order to validate the efficacy of FuseFL, we conduct comprehensive experiments with commonly used datasets in FL, including MNIST [77], CIFAR-10 [73], FMNIST [147], SVHN [100], CIFAR-100 [73] and Tiny-Imagenet [76]. For studying the non-IID problem in FL, we partitioned the datasets through a widely-used non-IID partition method, namely Latent Dirichlet Sampling [56; 67; 114; 80], in which the coefficient a represents the non-IID degree. Lower a generates more non-IID datasets, and vice versa. Consistent with established practices in the field [159; 114; 93; 80], each dataset was divided with three distinct degrees of non-IID with $a \in \{0.1, 0.3, 0.5\}$. If there is no additional explanation, the non-IID degree a is set to 0.5 by default.

Following other classic and advanced FL works studying non-IID problems and communication-efficient FL [125; 52; 93; 159], we train ResNet-18 [51] on all datasets in main experiments. And we reduce and increase the number of layers as ResNet-10 and ResNet-26 to verify the effect of FuseFL

in model-heterogeneity FL. The number of clients is set as $M = 5$ by default. Moreover, we study the scalability of our methods with different $M \in \{5, 10, 20, 50\}$.

We use SGD optimizer with momentum coefficient as 0.9, and the batch size is 128. The number of local training epochs $E = 200$. We search learning rates in $\{0.0001, 0.001, 0.01, 0.1\}$ and report the best results. The detailed hyper-parameters of different settings are shown in Table 9 of Appendix D.

Table 2: Accuracy of different methods across $\alpha = \{0.1, 0.3, 0.5\}$ on different datasets. Ensemble means ensemble learning with local trained models, which is an upper bound of all previous methods but impractical in FL due to the large memory costs and the weak scalability of clients. Thus, we highlight the best results in **bold font** except Ensemble.

Dataset	MNIST			FMNIST			CIFAR-10			SVHN			CIFAR-100			Tiny-Imagenet		
Method	α=0.1	α=0.3	α=0.5	α=0.1	α=0.3	α=0.5	α=0.1	α=0.3	α=0.5	α=0.1	α=0.3	α=0.5	α=0.1	α=0.3	α=0.5	α=0.1	α=0.3	α=0.5
FedAvg	48.24	72.94	90.55	41.69	82.96	83.72	23.93	27.72	43.67	31.65	61.51	56.09	4.58	11.61	12.11	3.12	10.46	11.89
FedDF	60.15	74.01	92.18	43.58	80.67	84.67	40.58	46.78	53.56	49.13	73.34	73.98	28.17	30.28	36.35	15.34	18.22	27.43
Fed-DAFL	64.38	74.18	93.01	47.14	80.59	84.02	47.34	53.89	58.59	53.23	76.56	78.03	28.89	34.89	38.19	18.38	22.18	28.22
Fed-ADI	64.13	75.03	93.49	48.49	81.15	84.19	48.59	54.68	59.34	53.45	77.45	78.85	30.13	35.18	40.28	19.59	25.34	30.21
DENSE	66.61	76.48	95.82	50.29	83.96	85.94	50.26	59.76	62.19	55.34	79.59	80.03	32.03	37.32	42.07	22.44	28.14	32.34
Ensemble	86.81	96.76	97.22	67.71	87.25	89.42	57.5	77.35	79.91	65.29	88.31	85.7	35.69	49.41	53.39	30.85	39.43	45.8
FuseFL $K = 2$	97.02	**98.43**	**98.54**	83.15	**89.94**	89.47	70.85	81.41	**84.34**	76.88	**91.07**	**90.87**	34.07	**45.12**	46.12	**29.28**	31.11	**34.34**
FuseFL $K = 4$	**97.19**	98.34	98.29	83.05	84.58	**90.50**	**73.79**	**84.58**	81.15	78.08	89.63	89.34	**36.86**	42.79	**49.30**	27.63	**33.04**	34.28
FuseFL $K = 8$	96.66	98.35	98.16	**83.2**	88.57	88.24	70.46	80.70	74.99	**80.31**	88.88	89.94	34.97	39.08	40.73	25.21	32.59	33.82

Baselines. Except the classic baseline FedAvg [96] and advanced OFL method DENSE [159], we apply two prevailing data-free KD methods DAFL [18] and ADI [153] into OFL. We choose FedDF [88] as its high efficiency in few-round FL. We conduct ensemble FL as it is the upper bound across ensemble-and-distillation based methods, yet impractical in real-world scenarios. The communication round for all baseline methods is only 1. For our method FuseFL, the number of communication rounds is equal to the number of splitted blocks K. However, the actual communication cost is as same as one-shot FL. Because FuseFL only communicates a part of the model. After all rounds, the total communication cost is SM, where S is the model size, M the number of clients.

6.2 Experimental Results

Main Results. Table 2 shows that FuseFL generally outperforms all other baselines except for ensemble FL. All ensemble-and-distillation baselines have lower performance than ensemble FL. Nevertheless, by the insights from causality (Section 3) and our innovative design (Section 4), FuseFL can significantly outperform ensemble FL for almost all cases except for CIFAR-100 with $a = 0.3, 0.5$ and Tiny-Imagenet. We suppose the reason is that CIFAR-100 and Tiny-Imagenet has much more data divergence between different classes, thus the overlap between R_m^{inv} is much less than other datasets. Recall that the benefits of FuseFL come from fusing sub models training on other clients, thus filtering out R_m^{spu} and collecting R_m^{inv} of the same class to improve the generalization performance. The large data divergence in CIFAR-100 and Tiny-Imagenet limits benefits of FuseFL.

Table 3: Accuracy with FuseFL with conv1×1 or averaging to support heterogeneous model design on CIFAR-10.

non-IID degree	$a = 0.1$	$a = 0.3$	$a = 0.5$
Ensemble	57.5	77.35	79.91
FuseFL	**73.79**	**84.58**	81.15
FuseFL (Avg)	68.08	71.49	80.35
FuseFL-Hetero	75.33	81.71	**82.71**
FuseFL (Avg)-Hetero	68.31	76.27	79.74

Table 4: Memory Occupation. For different number of clients, the number of basic channels in ResNet-18 of FuseFL is set as 32, 20, 14, 9 with $M \in \{5, 10, 20, 50\}$, respectively. Other OFLs refer to FedAvg, FedDF, Fed-DAFL, Fed-ADI, DENSE.

# Clients	$M = 5$	$M = 10$	$M = 20$	$M = 50$
Single Model	42.66MB	42.66MB	42.66MB	42.66MB
Other OFLs	42.66MB	42.66MB	42.66MB	42.66MB
Ensemble	213.31MB	426.62MB	853.24MB	2133.10MB
FuseFL $K = 2$	53.71MB	42.32MB	42.13MB	45.48MB
FuseFL $K = 4$	55.38MB	44.92MB	47.22MB	58.63MB
FuseFL $K = 8$	68.08MB	64.77MB	86.11MB	159.08MB
FuseFL (Avg)	**53.32MB**	**41.66MB**	**40.83MB**	**42.18MB**

Support of heterogeneous model design. Table 3 shows training heterogeneous model using FuseFL. In all $M = 5$ clients, 2 clients train ResNet-10 and other 2 clients train ResNet-26, the left 1 client trains ResNet-18. We set $K = 4$ for FuseFL. Results show that training with heterogeneous models has similar or even better results than homogeneous models, demonstrating that FuseFL supports training heterogeneous model well. This will be very useful in heterogeneous computation environments.

Table 5: Accuracy of different methods across $M = \{5, 10, 20, 50\}$.

Dataset	CIFAR-10				SVHN			
Method	$M = 5$	$M = 10$	$M = 20$	$M = 50$	$M = 5$	$M = 10$	$M = 20$	$M = 50$
FedAvg	43.67	38.29	36.03	23.01	56.09	45.34	47.79	36.53
FedDF	53.56	54.44	43.15	29.52	73.98	62.12	60.45	51.44
Fed-DAFL	55.46	56.34	45.98	29.41	78.03	63.34	62.19	54.23
Fed-ADI	58.59	57.13	46.45	27.45	78.85	65.45	63.98	57.35
DENSE	62.19	61.42	52.71	28.51	80.03	67.57	66.42	59.27
Ensemble	79.91	77.25	59.69	55.63	85.7	73.45	68.76	54.96
FuseFL $K = 2$	**84.34**	73.71	62.85	**42.18**	**90.87**	88.52	85.18	**72.25**
FuseFL $K = 4$	81.15	**78.28**	62.57	37.08	89.34	**89.31**	**86.94**	45.49
FuseFL $K = 8$	74.99	67.35	**63.19**	28.28	89.94	72.65	64.11	42.19

Memory occupation. Table 4 shows the memory occupation with varying number of clients and split modules. Results show that FuseFL requires similar memory with the single model when $K = 2$, while the ensemble FL requires the $S \times M$ storage cost. And using averaging as feature fusion method can further reduce the memory cost. Table 12 shows that the FuseFL with $K = 2$ or FuseFL with averaging has comparable or better performance than other variants of FuseFL.

Scalability of the number of clients. We empirically prove that the memory occupation increases a little along with the increased M (Table 4), and keeping performance outperforms than baselines (Table 5).

Influence on local models of backdoored datasets. As shown in Section 3, the backdoored features lead to information loss in X. Here we further show how the FL performance is influenced by the backdoored features. Table 6 shows that the backdoored (BD) clients fit on the handcrafted spurious features, thus having lower global accuracy than normal clients.

Table 6: Local and global accuracy of 5 local client models. BD_0 and BD_1 represent two clients trained on backdoored datasets. $Normal_0$, $Normal_1$, and $Normal_2$ represent three clients trained on clean datasets.

Client	BD_0	BD_1	$Normal_0$	$Normal_1$	$Normal_2$
Local Acc.	100.0	100.0	99.7	99.9	100.0
Global Acc.	32.6	27.1	41.2	42.3	38.4

Test accuracy on backdoored datasets. Table 7 provides the test accuracy of different methods training with backdoored CIFAR-10. The test dataset is the original clean test set. We test different numbers of backdoored clients $M_{bd} = 1, 2$ out of a total of 5 clients, to see how the degree of the backdoor influences training. Results show FuseFL outperforms ensemble FL in all cases. demonstrating that FuseFL can defend better against the backdoor samples than ensemble FL.

Table 7: Comparing accuracy on backdoored CIFAR-10.

# Backdoored clients	$M_{bd} = 1$		
Non-IID degree	$a = 0.1$	$a = 0.3$	$a = 0.5$
Ensemble	52.76	74.88	73.46
FuseFL $K = 2$	43.99	75.61	**84.79**
FuseFL $K = 4$	**55.99**	**78.39**	80.52
FuseFL $K = 8$	45.66	74.01	82.12
# Backdoored clients	$M_{bd} = 2$		
Non-IID degree	$a = 0.1$	$a = 0.3$	$a = 0.5$
Ensemble	52.67	68.42	75.83
FuseFL $K = 2$	54.24	**71.49**	**82.61**
FuseFL $K = 4$	**54.77**	67.53	72.99
FuseFL $K = 8$	51.70	68.31	76.95

7 Conclusion

In this work, we draw inspiration from the causality and the information bottleneck to analyze the cause of low performance of ensemble FL and OFL. Specifically, the local isolatedly trained models are easily to fit spurious features, as local clients cannot learn more invariant features and remove spurious features from other datasets. Built upon this insight, we provide a novel approach FuseFL to augment features by fusing client layers in a bottom-up manner, thus mitigating the spurious fitting and encourage learning of invariant features. FuseFL achieves OFL with extremely low communication costs with significantly higher performance than current OFL and ensemble FL methods.

Acknowledgments and Disclosure of Funding

This work was partially supported by National Natural Science Foundation of China under Grant No. 62272122, the Guangzhou Municipal Joint Funding Project with Universities and Enterprises under Grant No. 2024A03J0616, the Hong Kong RIF grant under Grant No. R6021-20, and Hong Kong CRF grants under Grant No. C2004-21G and C7004-22G. BH was supported by NSFC General Program No. 62376235, Guangdong Basic and Applied Basic Research Foundation Nos. 2022A1515011652 and 2024A1515012399, HKBU Faculty Niche Research Areas No. RC-FNRA-IG/22-23/SCI/04, and HKBU CSD Departmental Incentive Scheme. YGZ and YMC were supported in part by the NSFC / Research Grants Council (RGC) Joint Research Scheme under the grant: N-HKBU214/21, the General Research Fund of RGC under the grants: 12201321, 12202622, 12201323, the RGC Senior Research Fellow Scheme under the grant: SRFS2324-2S02, and the Initiation Grant for Faculty Niche Research Areas of Hong Kong Baptist University under the grant: RC-FNRA-IG/23-24/SCI/02.

References

[1] D. A. E. Acar, Y. Zhao, R. Matas, M. Mattina, P. Whatmough, and V. Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021.

[2] A. Achille and S. Soatto. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research*, 2018.

[3] K. Ahuja, E. Caballero, D. Zhang, J.-C. Gagnon-Audet, Y. Bengio, I. Mitliagkas, and I. Rish. Invariance principle meets information bottleneck for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 34:3438–3450, 2021.

[4] K. Ahuja, J. Wang, A. Dhurandhar, K. Shanmugam, and K. R. Varshney. Empirical or invariant risk minimization? a sample complexity perspective. In *International Conference on Learning Representations*, 2020.

[5] S. Ainsworth, J. Hayase, and S. Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *The Eleventh International Conference on Learning Representations*, 2022.

[6] M. Al-Shedivat, J. Gillenwater, E. Xing, and A. Rostamizadeh. Federated learning via posterior averaging: A new perspective and practical algorithms. In *International Conference on Learning Representations*, 2020.

[7] G. Alain and Y. Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

[8] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[9] S. Babakniya, S. Kundu, S. Prakash, Y. Niu, and S. Avestimehr. Federated sparse training: Lottery aware model compression for resource constrained edge. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022.

[10] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2938–2948, 2020.

[11] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pages 531–540. PMLR, 2018.

[12] S. Bibikar, H. Vikalo, Z. Wang, and X. Chen. Federated dynamic sparse training: Computing less, communicating less, yet learning better. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6080–6088, 2022.

[13] I. Bistritz, A. Mann, and N. Bambos. Distributed distillation for on-device learning. *Advances in Neural Information Processing Systems*, 33:22593–22604, 2020.

[14] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[15] K. Cai, X. Lei, J. Wei, and X. Xiao. Data synthesis via differentially private markov random fields. *Proc. VLDB Endow.*, 14(11):2190–2202, jul 2021.

[16] H. Chang, V. Shejwalkar, R. Shokri, and A. Houmansadr. Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer. *arXiv preprint arXiv:1912.11279*, 2019.

[17] A. Chatalic, V. Schellekens, F. Houssiau, Y. A. de Montjoye, L. Jacques, and R. Gribonval. Compressive learning with privacy guarantees. *Information and Inference: A Journal of the IMA*, 05 2021. iaab005.

[18] H. Chen, Y. Wang, C. Xu, Z. Yang, C. Liu, B. Shi, C. Xu, C. Xu, and Q. Tian. Data-free learning of student networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3514–3522, 2019.

[19] H.-Y. Chen and W.-L. Chao. Fedbe: Making bayesian model ensemble applicable to federated learning. In *NeurIPS*, 2020.

[20] H.-Y. Chen and W.-L. Chao. On bridging generic and personalized federated learning for image classification. In *International Conference on Learning Representations*, 2021.

[21] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[22] Y. Chen, W. Huang, K. Zhou, Y. Bian, B. Han, and J. Cheng. Understanding and improving feature learning for out-of-distribution generalization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[23] R. Dai, Y. Zhang, A. Li, T. Liu, X. Yang, and B. Han. Enhancing one-shot federated learning through data and ensemble co-boosting. In *The Twelfth International Conference on Learning Representations*, 2024.

[24] Y. Dai, Z. Chen, J. Li, S. Heinecke, L. Sun, and R. Xu. Tackling data heterogeneity in federated learning with class prototypes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[25] D. K. Dennis, T. Li, and V. Smith. Heterogeneity for the win: One-shot federated clustering. In *International Conference on Machine Learning*, 2021.

[26] Y. Diao, Q. Li, and B. He. Towards addressing label skews in one-shot federated learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[27] J. Dong, H. Li, Y. Cong, G. Sun, Y. Zhang, and L. Van Gool. No one left behind: Real-world federated class-incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4):2054–2070, 2024.

[28] J. Dong, L. Wang, Z. Fang, G. Sun, S. Xu, X. Wang, and Q. Zhu. Federated class-incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[29] X. Dong, S. Q. Zhang, A. Li, and H. Kung. Spherefed: Hyperspherical federated learning. In *European Conference on Computer Vision*, 2022.

[30] R. Dorfman, S. Vargaftik, Y. Ben-Itzhak, and K. Y. Levy. Docofl: downlink compression for cross-device federated learning. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[31] S. Dou, E. Zhou, Y. Liu, S. Gao, W. Shen, L. Xiong, Y. Zhou, X. Wang, Z. Xi, X. Fan, S. Pu, J. Zhu, R. Zheng, T. Gui, Q. Zhang, and X. Huang. LoRAMoE: Alleviating world knowledge forgetting in large language models via MoE-style plugin. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Aug. 2024.

[32] Y. H. Ezzeldin, S. Yan, C. He, E. Ferrara, and A. S. Avestimehr. Fairfed: Enabling group fairness in federated learning. In *Proceedings of the AAAI conference on artificial intelligence*, pages 7494–7502, 2023.

[33] J. Frankle and M. Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

[34] J. Frankle, G. K. Dziugaite, D. Roy, and M. Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th International Conference on Machine Learning*, pages 3259–3269, 2020.

[35] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[36] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2018.

[37] J. Goetz and A. Tewari. Federated learning via synthetic data. *arXiv preprint arXiv:2008.04489*, 2020.

[38] N. Guha, A. Talwalkar, and V. Smith. One-shot federated learning. *arXiv preprint arXiv:1902.11175*, 2019.

[39] I. Gulrajani and D. Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2020.

[40] C. Guo, M. Rana, M. Cisse, and L. van der Maaten. Countering adversarial images using input transformations. In *ICLR*, 2020.

[41] K. Gupta, M. Fournarakis, M. Reisser, C. Louizos, and M. Nagel. Quantization robust federated learning for efficient inference on heterogeneous devices. *Transactions on Machine Learning Research*, 2023.

[42] F. Hamman and S. Dutta. Demystifying local & global fairness trade-offs in federated learning using partial information decomposition. In *The Twelfth International Conference on Learning Representations*, 2024.

[43] W. Hao, M. El-Khamy, J. Lee, J. Zhang, K. J. Liang, C. Chen, and L. C. Duke. Towards fair federated learning with zero-shot data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3310–3319, 2021.

[44] M. Hardt, K. Ligett, and F. Mcsherry. A simple and practical algorithm for differentially private data release. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[45] M. Hardt and G. N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 61–70, 2010.

[46] C. He, M. Annavaram, and S. Avestimehr. Group knowledge transfer: Federated learning of large cnns at the edge. In *Advances in Neural Information Processing Systems 34*, 2020.

[47] C. He, S. Li, J. So, M. Zhang, H. Wang, X. Wang, P. Vepakomma, A. Singh, H. Qiu, L. Shen, P. Zhao, Y. Kang, Y. Liu, R. Raskar, Q. Yang, M. Annavaram, and S. Avestimehr. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020.

[48] C. He, S. Li, M. Soltanolkotabi, and S. Avestimehr. Pipetransformer: Automated elastic pipelining for distributed training of large-scale models. In *Proceedings of the 38th International Conference on Machine Learning*, pages 4150–4159. PMLR, 18–24 Jul 2021.

[49] C. He, A. D. Shah, Z. Tang, D. F. N. Sivashunmugam, K. Bhogaraju, M. Shimpi, L. Shen, X. Chu, M. Soltanolkotabi, and S. Avestimehr. Fedcv: A federated learning framework for diverse computer vision tasks. *arXiv preprint arXiv:2111.11066*, 2021.

[50] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[51] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[52] C. E. Heinbaugh, E. Luz-Ricca, and H. Shao. Data-free one-shot federated learning under very high statistical heterogeneity. In *The Eleventh International Conference on Learning Representations*, 2023.

[53] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021.

[54] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019.

[55] C.-H. Ho and N. Nvasconcelos. Contrastive learning with adversarial examples. *Advances in Neural Information Processing Systems*, 33:17081–17093, 2020.

[56] T. Hsu, H. Qi, and M. Brown. Measuring the effects of non-identical data distribution for federated visual classification. *ArXiv*, abs/1909.06335, 2019.

[57] T.-M. H. Hsu, H. Qi, and M. Brown. Federated visual classification with real-world data distribution. In A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, editors, *Computer Vision – ECCV 2020*, pages 76–92, Cham, 2020. Springer International Publishing.

[58] S. Hu, Q. Li, and B. He. Communication-efficient generalized neuron matching for federated learning. In *Proceedings of the 52nd International Conference on Parallel Processing*, ICPP '23, page 254–263, New York, NY, USA, 2023. Association for Computing Machinery.

[59] C. Huang, Q. Liu, B. Y. Lin, C. Du, T. Pang, and M. Lin. Lorahub: Efficient cross-task generalization via dynamic loRA composition, 2024.

[60] W. Huang, M. Ye, Z. Shi, H. Li, and B. Du. Rethinking federated learning with domain shift: A prototype view. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[61] P. Izmailov, P. Kirichenko, N. Gruver, and A. G. Wilson. On feature learning in the presence of spurious correlations. *Advances in Neural Information Processing Systems*, 35:38516–38532, 2022.

[62] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*, 2018.

[63] D. Jhunjhunwala, S. Wang, and G. Joshi. Towards a theoretical and practical understanding of one-shot federated learning with fisher information. In *Federated Learning and Analytics in Practice: Algorithms, Systems, Applications, and Opportunities*, 2023.

[64] Y. Jiang, S. Wang, V. Valls, B. J. Ko, W.-H. Lee, K. K. Leung, and L. Tassiulas. Model pruning enables efficient federated learning on edge devices. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[65] X. Jin, X. Ren, D. Preotiuc-Pietro, and P. Cheng. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*, 2023.

[66] N. Johnson, J. P. Near, and D. Song. Towards practical differential privacy for sql queries. *Proceedings of the VLDB Endowment*, 11(5):526–539, 2018.

[67] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, H. Eichner, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao. Advances and open problems in federated learning, 2021.

[68] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh. Scaffold: Stochastic controlled averaging for federated learning. *arXiv preprint arXiv:1910.06378*, 2019.

[69] H. Kim, Y. Kwak, M. Jung, J. Shin, Y. Kim, and C. Kim. Protofl: Unsupervised federated learning via prototypical distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023.

[70] P. Kirichenko, P. Izmailov, and A. G. Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In *The Eleventh International Conference on Learning Representations*, 2023.

[71] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 2017.

[72] J. Konečný, H. Brendan McMahan, F. X. Yu, P. Richtárik, A. Theertha Suresh, and D. Bacon. Federated Learning: Strategies for Improving Communication Efficiency. *arXiv e-prints*, 2016.

[73] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis*, 2009.

[74] A. Krizhevsky, V. Nair, and G. Hinton. Cifar-10 (canadian institute for advanced research). *URL http://www.cs.toronto.edu/kriz/cifar.html*, 2010.

[75] A. Kumar, A. Raghunathan, R. M. Jones, T. Ma, and P. Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022.

[76] Y. Le and X. Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[77] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 1998.

[78] A. Li, J. Sun, B. Wang, L. Duan, S. Li, Y. Chen, and H. Li. Lotteryfl: Empower edge intelligence with personalized and communication-efficient federated learning. In *2021 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 68–79, 2021.

[79] D. Li and J. Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

[80] Q. Li, Y. Diao, Q. Chen, and B. He. Federated learning on non-iid data silos: An experimental study, 2021.

[81] Q. Li, B. He, and D. Song. Practical one-shot federated learning for cross-silo setting. *arXiv preprint arXiv:2010.01017*, 2020.

[82] Q. Li, B. He, and D. Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10713–10722, 2021.

[83] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2020.

[84] X. Li, Z. Zhou, J. Zhu, J. Yao, T. Liu, and B. Han. Deepinception: Hypnotize large language model to be jailbreaker. *arXiv preprint arXiv:2311.03191*, 2023.

[85] Y. Li, B. Luo, Q. Wang, N. Chen, X. Liu, and B. He. A reflective llm-based agent to guide zero-shot cryptocurrency trading. *arXiv preprint arXiv:2407.09546*, 2024.

[86] Z. Li, X. Shang, R. He, T. Lin, and C. Wu. No fear of classifier biases: Neural collapse inspired federated learning with synthetic and fixed classifier. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5319–5329, October 2023.

[87] P. P. Liang, T. Liu, L. Ziyin, N. B. Allen, R. P. Auerbach, D. Brent, R. Salakhutdinov, and L.-P. Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

[88] T. Lin, L. Kong, S. U. Stich, and M. Jaggi. Ensemble distillation for robust model fusion in federated learning. In *NeurIPS*, 2020.

[89] J. Liu, Z. Shen, Y. He, X. Zhang, R. Xu, H. Yu, and P. Cui. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021.

[90] L. Liu, X. Jiang, F. Zheng, H. Chen, G.-J. Qi, H. Huang, and L. Shao. A bayesian federated learning framework with online laplace approximation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[91] Y. Long, B. Wang, Z. Yang, B. Kailkhura, A. Zhang, C. Gunter, and B. Li. G-pate: Scalable differentially private data generator via private aggregation of teacher discriminators. *Advances in Neural Information Processing Systems*, 34, 2021.

[92] B. Luo, Z. Zhang, Q. Wang, A. Ke, S. Lu, and B. He. Ai-powered fraud detection in decentralized finance: A project life cycle perspective. *arXiv preprint arXiv:2308.15992*, 2023.

[93] M. Luo, F. Chen, D. Hu, Y. Zhang, J. Liang, and J. Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-IID data. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[94] Z. Luo, Y. Wang, Z. Wang, Z. Sun, and T. Tan. Disentangled federated learning for tackling attributes skew via invariant aggregation and diversity transferring. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 14527–14541. PMLR, 17–23 Jul 2022.

[95] M. S. Matena and C. A. Raffel. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716, 2022.

[96] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

[97] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706. IEEE, 2019.

[98] V. Mugunthan, E. Lin, V. Gokul, C. Lau, L. Kagal, and S. Pieper. Fedltn: Federated learning for sparse and personalized lottery ticket networks. In *European Conference on Computer Vision*, pages 69–85. Springer, 2022.

[99] M. Naseer, S. Khan, M. Hayat, F. S. Khan, and F. Porikli. A self-supervised approach for adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 262–271, 2020.

[100] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[101] B. Neyshabur, S. Bhojanapalli, D. Mcallester, and N. Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, 2017.

[102] B. Neyshabur, H. Sedghi, and C. Zhang. What is being transferred in transfer learning? *Advances in neural information processing systems*, 2020.

[103] J. Nguyen, K. Malik, M. Sanjabi, and M. Rabbat. Where to begin? exploring the impact of pre-training and initialization in federated learning. *arXiv preprint arXiv:2206.15387*, 2022.

[104] N.-H. Nguyen, T.-A. Nguyen, T. Nguyen, V. T. Hoang, D. D. Le, and K.-S. Wong. Towards efficient communication federated recommendation system via low-rank training. *arXiv preprint arXiv:2401.03748*, 2024.

[105] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[106] M. Pagliardini, M. Jaggi, F. Fleuret, and S. P. Karimireddy. Agree to disagree: Diversity through disagreement for better transferability. In *The Eleventh International Conference on Learning Representations*, 2023.

[107] N. Papernot and P. McDaniel. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*, 2018.

[108] Y. Park, D.-J. Han, D.-Y. Kim, J. Seo, and J. Moon. Few-round learning for federated learning. *Advances in Neural Information Processing Systems*, 34:28612–28622, 2021.

[109] J. Pearl. *Causality*. Cambridge university press, 2009.

[110] J. Peters, D. Janzing, and B. Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[111] X. Qiu, J. Fernandez-Marques, P. P. Gusmao, Y. Gao, T. Parcollet, and N. D. Lane. ZeroFL: Efficient on-device training for federated learning with local sparsity. In *International Conference on Learning Representations*, 2022.

[112] Z. Qu, X. Li, R. Duan, Y. Liu, B. Tang, and Z. Lu. Generalized federated learning via sharpness aware minimization. In *International Conference on Machine Learning*, pages 18250–18280. PMLR, 2022.

[113] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein. Svcca: singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, 2017.

[114] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021.

[115] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*, pages 2021–2031. PMLR, 2020.

[116] S. Sagawa, A. Raghunathan, P. W. Koh, and P. Liang. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pages 8346–8356. PMLR, 2020.

[117] A. K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith. On the convergence of federated optimization in heterogeneous networks. *ArXiv*, abs/1812.06127, 2018.

[118] A. M. Saxe, Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, B. D. Tracey, and D. D. Cox. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020, 2019.

[119] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

[120] V. Sehwag, S. Mahloujifar, T. Handina, S. Dai, C. Xiang, M. Chiang, and P. Mittal. Improving adversarial robustness using proxy distributions. *arXiv preprint arXiv:2104.09425*, 2021.

[121] M. Shin, C. Hwang, J. Kim, J. Park, M. Bennis, and S.-L. Kim. Xor mixup: Privacy-preserving data augmentation for one-shot federated learning. *arXiv preprint arXiv:2006.05148*, 2020.

[122] R. Shwartz-Ziv and N. Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[123] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, jan 2014.

[124] Y. Sun, L. Shen, S. Chen, L. Ding, and D. Tao. Dynamic regularized sharpness aware minimization in federated learning: Approaching global consistency and smooth landscape. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[125] Y. Sun, L. Shen, T. Huang, L. Ding, and D. Tao. Fedspeed: Larger local interval, less communication round, and higher generalization accuracy. In *The Eleventh International Conference on Learning Representations*, 2023.

[126] A. Z. Tan, H. Yu, L. Cui, and Q. Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–17, 2022.

[127] Z. Tang, X. Chu, R. Y. Ran, S. Lee, S. Shi, Y. Zhang, Y. Wang, A. Q. Liang, S. Avestimehr, and C. He. Fedml parrot: A scalable federated learning system via heterogeneity-aware scheduling on sequential and hierarchical training. *arXiv preprint arXiv:2303.01778*, 2023.

[128] Z. Tang, J. Huang, R. Yan, Y. Wang, Z. Tang, S. Shi, A. C. Zhou, and X. Chu. Bandwidth-aware and overlap-weighted compression for communication-efficient federated learning. In *53rd International Conference on Parallel Processing*, Gotland, Sweden, 12–15 August 2024.

[129] Z. Tang, X. Kang, Y. Yin, X. Pan, Y. Wang, X. He, Q. Wang, R. Zeng, K. Zhao, S. Shi, A. C. Zhou, B. Li, B. He, and X. Chu. Fusionllm: A decentralized llm training system on geo-distributed gpus with adaptive compression, 2024.

[130] Z. Tang, S. Shi, and X. Chu. Communication-efficient decentralized learning with sparsification and adaptive peer selection. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pages 1207–1208. IEEE, 2020.

[131] Z. Tang, S. Shi, X. Chu, W. Wang, and B. Li. Communication-efficient distributed deep learning: A comprehensive survey. *arXiv preprint arXiv:2003.06307*, 2020.

[132] Z. Tang, S. Shi, B. Li, and X. Chu. Gossipfl: A decentralized federated learning framework with sparsified and adaptive communication. *IEEE Transactions on Parallel and Distributed Systems*, pages 1–13, 2022.

[133] Z. Tang, Y. Wang, X. He, L. Zhang, X. Pan, Q. Wang, R. Zeng, K. Zhao, S. Shi, B. He, et al. Fusionai: Decentralized training and deploying llms with massive consumer-level gpus. *The 32nd International Joint Conference on Artificial Intelligence, Symposium on Large Language Models (LLM@IJCAI 2023)*, 2023.

[134] Z. Tang, Y. Zhang, S. Shi, X. He, B. Han, and X. Chu. Virtual homogeneity learning: Defending against data heterogeneity in federated learning. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 21111–21132. PMLR, 17–23 Jul 2022.

[135] Z. Tang, Y. Zhang, S. Shi, X. Tian, T. Liu, B. Han, and X. Chu. Fedimpro: Measuring and improving client update in federated learning. In *The Twelfth International Conference on Learning Representations*, 2024.

[136] C. Thapa, M. A. P. Chamikara, S. Camtepe, and L. Sun. Splitfed: When federated learning meets split learning. *arXiv preprint arXiv:2004.12088*, 2020.

[137] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020.

[138] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

[139] F. Utrera, E. Kravitz, N. B. Erichson, R. Khanna, and M. W. Mahoney. Adversarially-trained deep nets transfer better: Illustration on image classification. In *International Conference on Learning Representations*, 2020.

[140] B. Vivek and R. V. Babu. Single-step adversarial training with dropout scheduling. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 947–956. IEEE, 2020.

[141] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020.

[142] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *arXiv preprint arXiv:2007.07481*, 2020.

[143] N. Wang, W. Feng, yuchen deng, M. Duan, F. Liu, and S.-K. Ng. Data-free diversity-based ensemble selection for one-shot federated learning. *Transactions on Machine Learning Research*, 2023.

[144] Q. Wang, Z. Zhang, Z. Liu, S. Lu, B. Luo, and B. He. Ex-graph: A pioneering dataset bridging ethereum and x. In *The Twelfth International Conference on Learning Representations*, 2024.

[145] Y. Wang, Z. Ni, S. Song, L. Yang, and G. Huang. Revisiting locally supervised learning: an alternative to end-to-end training. In *International Conference on Learning Representations*, 2020.

[146] B. E. Woodworth, K. K. Patel, and N. Srebro. Minibatch vs local sgd for heterogeneous distributed learning. *Advances in Neural Information Processing Systems*, 33:6281–6292, 2020.

[147] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[148] P. Yadav, D. Tam, L. Choshen, C. Raffel, and M. Bansal. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[149] K. Yang, T. Zhou, Y. Zhang, X. Tian, and D. Tao. Class-disentanglement and applications in adversarial detection and defense. *In NeurIPS*, 2021.

[150] Z. Yang, Y. Zhang, Y. Zheng, X. Tian, H. Peng, T. Liu, and B. Han. Fedfed: Feature distillation against data heterogeneity in federated learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[151] H. Ye, C. Xie, T. Cai, R. Li, Z. Li, and L. Wang. Towards a theoretical framework of out-of-distribution generalization. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[152] M. Yi, L. Hou, J. Sun, L. Shang, X. Jiang, Q. Liu, and Z. Ma. Improved ood generalization via adversarial training and pretraing. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11987–11997. PMLR, 18–24 Jul 2021.

[153] H. Yin, P. Molchanov, J. M. Alvarez, Z. Li, A. Mallya, D. Hoiem, N. K. Jha, and J. Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020.

[154] T. Yoon, S. Shin, S. J. Hwang, and E. Yang. Fedmix: Approximation of mixup under mean augmented federated learning. In *International Conference on Learning Representations*, 2021.

[155] F. Yu, A. S. Rawat, A. Menon, and S. Kumar. Federated learning with only positive labels. In *International Conference on Machine Learning*, 2020.

[156] B. Yuan, Y. He, J. Q. Davis, T. Zhang, T. Dao, B. Chen, P. Liang, C. Re, and C. Zhang. Decentralized training of foundation models in heterogeneous environments. In *NeurIPS*, 2022.

[157] H. Yuan, W. R. Morningstar, L. Ning, and K. Singhal. What do we mean by generalization in federated learning? In *International Conference on Learning Representations*, 2022.

[158] D. Zhang, K. Ahuja, Y. Xu, Y. Wang, and A. Courville. Can subnetwork structure be the key to out-of-distribution generalization? In *International Conference on Machine Learning*, pages 12356–12367. PMLR, 2021.

[159] J. Zhang, C. Chen, B. Li, L. Lyu, S. Wu, S. Ding, C. Shen, and C. Wu. Dense: Data-free one-shot federated learning. *Advances in Neural Information Processing Systems*, 35, 2022.

[160] J. Zhang, Z. Li, B. Li, J. Xu, S. Wu, S. Ding, and C. Wu. Federated learning with label distribution skew via logits calibration. In *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 2022.

[161] Y. Zhang, M. Gong, T. Liu, G. Niu, X. Tian, B. Han, B. Schölkopf, and K. Zhang. Adversarial robustness through the lens of causality. In *International Conference on Learning Representations*, 2021.

[162] Y. Zhang, Z. Yang, X. Tian, N. Wang, T. Liu, and B. Han. Robust training of federated models with extremely label deficiency. In *The Twelfth International Conference on Learning Representations*, 2024.

[163] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[164] Y. Zhou, G. Pu, X. Ma, X. Li, and D. Wu. Distilled one-shot federated learning. *arXiv preprint arXiv:2009.07999*, 2020.

[165] Z. Zhou, R. Tao, J. Zhu, Y. Luo, Z. Wang, and B. Han. Can large language models reason robustly with noisy rationales? In *ICLR 2024 Workshop on Reliable and Responsible Foundation Models*.

Appendix / supplemental material

A Details of Algorithm

Details of our algorithm is shown in Algorithm 1.

Algorithm 1 FL with `FuseFL`

Input: The number of split modules K; Initialized local modules $H_m^1, ..., H_m^K$ and classifier Λ_m.
Output: The fused model F_{fus} (Eq. 6).

 # Train and fuse
for module $k = 1, \cdots, K$ **do**
  for each client $m \in \mathcal{M}$ **in parallel do do**
    $\theta_m^k \leftarrow$ `ClientTrain`(k);
  end for
  Fuse $H_{\text{fus}}^k \leftarrow (H_1^k, ..., H_M^k)$ as Eq. 4;
  for each client $m \in \mathcal{M}$ **in parallel do do**
    `ClientAdapt`(k, H_{fus}^k);
  end for
end for
 # Calibrate classifier
Averaging and calibrating final classifier Λ using CCVR [93] with object $\ell(F_{\text{fus}}; x, y)$ (Eq. 6);
Return F_{fus}.

`ClientTrain`(k):
Build $F_{\text{fus}}^{k,m}$ follows Eq. 5;
Freeze $H_m^{k-1}, ..., H_m^1$;
for each local iteration $j = 0, \cdots, E$ **do**
  $\theta_{m,j+1} \leftarrow \theta_{m,j} - \eta_{k,j} \nabla_\theta \ell(F_{\text{fus}}^{k,m}; x, y), x, y \sim \mathcal{D}_m$;
end for
Return H_m^k parameterized with $\theta_{m,E}^k$;

`ClientAdapt`(k, H_{fus}^k):
Build adaptor A_m^{k+1} following Eq 7 or 8;
Stitch $H_m^{k+1} \triangleq H_m^{k+1} \circ A_m^{k+1}$;

B Broader Impact

This work provides a novel OFL approach, aiming at advancing the field of federated learning. There are many potential societal consequences of our work. For instance, our work can extremely reduce the communication cost of FL, reducing energy consumption. Under the low-bandwidth communication environments like Internet with 1$\sim$ 10 MB/s [133; 156; 129], this method provides possibility to train super large models like large language models with low communication time. Furthermore, to the best of our knowledge, this is the first work that understands FL from the view of causality. There exist a large space for future works to study along this direction.

`FuseFL` can further inspire more research and applications in the communication-constrained scenarios, like extremely low communication bandwidth, and training large language models (LLM) in FL [129; 84; 165]. We will try to extend `FuseFL` into training LLMs or MoE in FL scenarios with its low communication costs.

To deploy transformer-based frameworks like current LLMs with the core idea of `FuseFL` in one-shot FL, we envision two methods here.

- **Concat-and-freeze.** Similar to training ResNet in `FuseFL`, we can block-wisely train and collect the transformer blocks together for each round; during local training, the output features of all

transformer blocks are concatenated to feed into the subsequent layers. Due to the large resource consumption of pretraining, we do not evaluate this idea here.

- **Averaging-and-freeze LoRA.** Here, we consider the finetuning scenarios with LoRA [59]. LoRA blocks can be seen as additional matrix mapping applied on the local Q V attentions and MLP layers. The output is the original feature plus the LoRA output. To use LoRA in `FuseFL`, we can follow the MoE style [31] or the averaging style [59]. Specifically, we consider averaging LoRAs on different clients together, then averaging and freezing all LoRAs in each transformer block to freeze the obtained aggregated features in each communication round.

C More related works

We provide the comprehensive introduction of the related works in this section. The data heterogeneity problem in FL is introduced in Section C.1, and the communication compression in FL is introduced in Section C.3. Further, by seeing local datasets from other clients as the OOD datasets with respect to the isolated locally trained models, we review some methods in OOD generalization (Section C.5 and mutual information (Section C.4) to build a connection between them and FL with extremely low communication costs. We demystify our work from current works in Table 8, which is a detailed version of Table 1.

Table 8: Demystifying different FL algorithms. T represents communication rounds, S the model size, M the number of clients. Practically, in communication-compression FL, the minimal sparsification ratio Q_{Spar} is 0.1, the quantization ratio Q_{Quant} is 0.125 $(32 \to 4\text{bits})$, and the low rank ratio Q_{LowR} is 0.1, otherwise the convergence is difficult to achieve and significantly harming model performance. The "Centralized" means training the model with all datasets aggregated with SGD. Due to the data heterogeneity, the performance usually is: *Centralized* $\geq$ *FedAvg* $\geq$ *Ensemble*.

Method type	Core technique	Communication cost	Storage cost	Performance upper bound	Support model heterogeneity	Not require extra data
FedAvg	No compression	$T \times S$	S	Centralized	✓	✓
	Sparsification	$T \times S \times Q_{\text{Spar}}$	S	FedAvg	✓	✓
	Quantization	$T \times S \times Q_{\text{Quant}}$	S	FedAvg	✓	✓
	Low-rank	$T \times S \times Q_{\text{LowR}}$	S	FedAvg	✓	✓
One-shot FL	Average-based	S	S	Ensemble	✗	✗
	Ensemble-based	S	$S \times M$	Centralized	✓	✗
	Model distillation	S	S	Ensemble	✓	✗
`FuseFL`		S	S	Centralized	✓	✓

C.1 Data Heterogeneity in FL

The Non-IID data distribution is the notorious problem in FL, which severely harms the convergence rate and model performance of FL. The sharp Non-IID data makes local clients learn much different weights [68; 146], resulting in heterogeneous feature [160; 134] and classifiers [93]. Current methods that address Non-IID data problems include following typical directions.

Some works design new **FL optimizers** to stabilize and accelerate convergence [114; 125]. **Personalized FL** aims to optimize different client models by learning knowledge from other clients and adapting to their own datasets [126; 88]. Distinguished from these works, how to alleviate data heterogeneity within extremely low communication costs is a new challenging problem, as clients have little possibility of communicating information with other clients.

Model regularization. This direction proposes to add a penalty of distances between local trained models and the server model. FedProx [117] directly uses the L2 distance between local models to the server model to constrain the local models not moving too far. SCAFFOLD [68] utilizes the historical local updates to correct update directions of local clients during local training. FedDyn [1] dynamically updates the objective functions to ensure the local optima between devices are asymptotically consistent. FedIR [57] claims that applying important weight to the client's local objectives helps to obtain an unbiased estimator of the global loss objective function.

Feature calibration. Some works focuses on align feature representations of different clients in similar spaces [82; 29; 134; 60]. MOON [82] adds the contrastive loss to between local and global

models to learn a similar representation between clients, in which the global model acts as an intermediate agent to communicate between clients. It is found the local features of the same data largely shift between client models during local training [134]. To address this problem, virtual homogeneous learning [134] proposes to use a homogeneous dataset which can contain completely no information of original datasets, to calibrate the feature representations between clients. This technique improves the generalization performance and convergence speed of federated learning. SphereFed [29] adds constraints on learned representations of input data to be in a unit hypersphere shared by clients. Besides, SphereFed discovers that the non-overlapped feature distributions for the same class lead to weaker consistency of the local learning targets from another perspective. The prototype [60] methods propose to utilize a pre-defined vector of each class in the representation space, then align client feature representations with such prototype, which is also called virtual feature transfer learning [134]. FedImpro [135] estimates and share the feature distribution to alleviate the gradient diversity and enhance high-level model training.

Classifier calibration. Due to the shifted features between clients, the classifier is usually trained with bias. And the final obtained classifier with FedAvg is normally prone to some specical classes. CCVR [93] is the first work to transmit the statistics of logits and label information of data samples to calibrate the classifier. In SphereFed [29], the classifier is fiexed with weights spanning the unit hypersphere, and calibrated by a mean squared loss. Some works also calibrate the classifier during or after training [24; 86; 69].

Optimization schemes. From the optimization perspective, some methods regard local updates at clients as pseudo gradients [114] and design new FL optimizers to stabilize and accelerate convergence. FedNova [142] normalizes the local updates to reduce the inconsistency between the local and global optimization objective functions. FedAvgM [56] exploits the history updates of server model to avoid the overfits on the selected clients in each round. FedOpt [114] generalizes the centralized optimizers into FL scenario, and proposes FedAdaGrad, FedYogi, FedAdam. FedSpeed [125] utilizes a correction term on local updates to reduce the biases during training. FedSpeed also merges the stochastic gradient with a perturbation computed from an extra gradient ascent step in the neighborhood, to reduce the gradient heterogeneity.

Data&Feature sharing. The phenomenon of client drift primarily originates from data heterogeneity. To address this issue, researchers have discovered that sharing a subset of private data can markedly enhance convergence speed and generalization capability, as highlighted in [163]. However, this approach entails a compromise on the privacy of client data.

Consequently, to simultaneously mitigate data heterogeneity and uphold data privacy, a range of studies [45; 44; 17; 66; 85; 15] have proposed the addition of noise to data. This method allows for the sharing of data while providing a certain level of privacy protection. Alternatively, other research efforts concentrate on disseminating synthetic data portions [62; 91; 37; 43] or data statistics [121; 154], as opposed to raw data, in order to alleviate data heterogeneity. FedDF [88] employs external data and engages in knowledge distillation based on these data to facilitate the transfer of model knowledge between the server and clients. The fundamental concept of FedDF involves fine-tuning the aggregated model through knowledge distillation using newly shared data.

Additionally, to address feature shift, certain methodologies advocate for the sharing of features to enhance the convergence rate. Cronus [16] suggests the sharing of logits as a means to counteract poisoning attacks. CCVR [93] transmits the statistical data of logits to calibrate the final layer of federated models. CCVR [93] also shares parameters representative of local feature distribution. Importantly, this approach does not necessitate sharing the count of different labels with the server, thereby preserving the privacy of clients' label distribution. Furthermore, our method serves as a framework to leverage shared features in reducing gradient dissimilarity. The feature estimator employed need not be confined to the Gaussian distribution of local features; alternative estimators or even features from additional datasets, rather than private ones, may be utilized. In this direction, VHL [134] only requires to share the pure noise dataset, which can have completely no related information of the local private datasets, largely reducing the requirements of the shared datasets and can be exploited in practical scenarios.

Personalized FL. Personalized Federated Learning (PFL) aims to enable clients to optimize distinct personal models that can absorb knowledge from other clients and tailor it to their individual datasets, as detailed in [126]. The process of knowledge transfer in personalization primarily

involves the introduction of personalized parameters [87; 136; 78], or the employment of knowledge distillation [46; 88; 79; 13] utilizing shared local features or supplementary datasets.

However, due to the tendency of personalized federated models to prioritize optimizing local objective functions, they often do not achieve generic performance (as evaluated on a global test dataset) that is on par with conventional Federated Learning (FL) methodologies [20]. Given our primary objective of learning an enhanced generic model, we have chosen not to include comparisons and improve performance of personalized FL algorithms in our work. While achieving extremely low communication costs in PFL can be an interesting future work.

C.2 One-shot FL

Current FL methods with communication costs of only one model size are referred as one-shot FL, different one-shot learning, the "one-shot" here means one communicating round [159; 38; 81; 164; 25]. Thus, the communication cost is limited as the model size $\mathcal{M}$ for each client, less than FedAvg-style algorithms for R times. While there are works that introduce additional communication costs to improve the performance of one-shot FL. Existing one-shot FL works can be categorized into average-based [63], ensemble-based [143] or model distillation based [52; 26],

Average-based methods include the basic baseline, one-shot FedAvg, which shows a severely bad performance, and some other advanced average methods [95; 65; 148]. FedFisher [63] proposes to utilize fisher information to avoid harming knowledge of local models when averaging on the server. Note that the similar methodology of using fisher information is also utilized to enhance the FedAvg [112] or bayesian FL [90; 6], which belongs to multi-round FL. All of them aims to avoid forgetting local learned knowledge like the classical exploitation of fisher information in continual learning [71]. Matching permutations between the weights of different models [5; 141] is another advanced method for model averaging. Linear mode connectivity [34; 102; 5] helps to explain to the part of the success of model averaging. However, due to the highly non-linear structure of deep neural networks, it is extremely difficult to find a method to directly average local clients through one round to obtain a perfect server model. Therefore, one-shot average method usually fails to achieve a good performance.

Ensemble-based methods are based on ensemble learning [38]. Intuitively, the server aggregates multiple local client models. Then, the direct way of deploying these models is to average outputed logits of them and make predictions [143]. Some methods propose to find a better model selection method to select local models that might be more familiar with the given inputs [143]. Some methods utilize prototype data [26] or generated datasets [52] to conduct better model selection in ensemble learning.

Model distillation methods are built upon the ensemble models [38; 81]. The core motivation is that the ensemble models occupy too much storage and require much extra computation costs. Thus [81] involved a public dataset for training. As a replacement of using global data, [164] transmits the distilled local datasets to server for model distillation. [25] clusters clients and communicates the mean data of each cluster. These methods require extra datasets, which may be impractical in some data-sensitive scenarios like medical and education data. And the large storage costs still exist when many clients upload their models for ensemble learning.

Note that there is also some work [108] that proposes to find a good initial model of FL in few communication rounds. Thus the convergence rate of FedAvg can be accelerated by this good initialization [103; 49]. The objective of this work is different from one-shot FL or ours.

C.3 Communication compressed FL

Different from one-shot FL which reduces the communication frequency, communication compression methods aim to reduce the communication size in each round. Typical communication-compression methods [131] include sparsification [12; 64; 130; 132], quantization [115; 41] and low-rank decomposition [104; 72].

Sparsification. Studies in [30; 12; 111; 132] have introduced a significant level of sparsity in the local model training stage, effectively reducing the number of parameters that need to be transmitted. The works represented in [33] have put forth the Lottery Ticket Hypothesis, suggesting the existence of trainable sub-networks within over-parameterized networks that can be independently trained

without accuracy loss. Inspired by this concept, LotteryFL [78] and FedLTN [98] aim to identify and exchange these personalized lottery ticket networks between the server and clients. Moreover, Hetero-FLASH [9] employs adaptive sparsity, with objectives extending beyond identifying the optimal sub-network, to fully leveraging clients' resources. In contrast to focusing on personalized Federated Learning (FL), this paper primarily considers generic FL, where all clients share the same model structure.

Quantization. Orthogonally to Sparsification, quantization emerges as an additional pivotal strategy to alleviate the communication bottleneck in Federated Learning. This method represents model updates with fewer bits than the conventional 32 or 64 bits, thus reducing numerical precision. FedPAQ [115] adopts periodic averaging of low-bit representations of local model updates to minimize both the frequency of communication and the overhead per round. Advancements in this realm have been furthered by [41], which introduces a variant of Quantization-Aware Training (QAT) robust to multiple bit-widths, eliminating the necessity for retraining in the FL context.

Low-rank decomposition. Concurrently, research in [104; 72] has utilized a low-rank decomposition of matrices to cultivate sparse models. Specifically, the weights of local trained models are decomposed into smaller matrices. Then these smaller matrices are communicated to the server for recovering the original weights.

While these methods also reduce the communication costs, they are far from reducing costs at an extremely low level of the one-shot FL and ours. Because these methods still require many even more communication rounds than FedAvg to achieve training convergence.

C.4 Mutual Information

The mutual information (MI) has garnered increasing attention in recent years with its explanation of how deep neural networks learn intermediate representations of the raw data. The Information Bottleneck (IB) principle [122; 118; 138] provides insights into the training dynamics of deep neural networks. Specifically, the neural network reduces the mutual information between raw data and representations layer by layer, while maximizing the mutual information between labels and representations. [2] introduces the nuisance variable into the mutual information, and proposes that mutual information between representation and the nuisance should be as less as possible. And it is proved that the empirical risk minimization (ERM) with stochastic gradient descent (SGD) has implicitly achieved the IB principle [2]. Some methods in unsupervised learning exploit maximization of mutual information [105; 137; 54] to enhance the feature representation. The contrastive learning [21; 50] maximizes the mutual information across varying views of the same input.

C.5 Out-of-Distribution Generalization

Out-of-Distribution (OOD) generalization refers to the model performance on the unseen data distribution, which is called OOD data [89; 22; 3]. The spurious features widely exist in the real-world datasets, like the textures, shapes, colors of objects [53; 36; 35; 144; 92]. Classification with original labels with these properties often leads overfitting on these variables instead of learning the real mapping relationship $X \rightarrow Y$. It is found that the out-of-distribution generalization performance of a model learned by ERM [22; 151] is connected with the spurious features. It is found that the over-parameterization, increasing model size well beyond the point of zero training error, may hurt the test error on some data samples, while it improving the average test error on all data samples [116]. These aggravated test errors are called worst-group error [89].

Invariant risk minimization (IRM) [8; 3] is proposed to address inheriting spurious correlations in trained models. It is shown that exploiting invariant causal relationships between datasets gathered from multiple environments rather than relying on varying spurious relationships appearing in isolated local datasets, help to improve the robustness of the learned predictors. [158] proposes that there exists a sub-network hidden in the full neural network that is unbiased functional (not focusing on spurious correlation), thus achieving better OOD performance.

However, some studies show that there is no clear winner between ERM and IRM when covariate shift happens [4], and ERM is still the state-of-the-art on many problems of OOD generalization [39].

The colored MNIST (CMNIST) is also called anti-causal CMNIST proposed by [8; 4] is often used to study the OOD performance of different training methods. In our work, we directly add different

geometry shapes with random colors on local clients as the spurious features (also called adversarial samples [149; 120; 55]) to explore how isolated local training and our algorithm are influenced by them.

It is proven that the self-supervised or unsupervised training with contrastive learning [55; 99] can help reduce the overfitting on spurious features. And dropout [123] also helps learning spurious relationships between label and the spurious features [140].

D Details of Experiment Configuration

D.1 Hyper Parameters

Table 9: Learning rate of all experiments.

Dataset	MNIST	FMNIST	CIFAR10	SVHN	CIFAR100	Tiny-Imagenet
$a = 0.1$	0.001	0.001	0.01	0.001	0.01	0.01
$a = 0.3$	0.001	0.01	0.01	0.01	0.01	0.01
$a = 0.5$	0.001	0.01	0.01	0.01	0.01	0.01

The learning rate configuration has been listed in Table 9. We report the best results and their learning rates (searched in $\{0.0001, 0.001, 0.01, 0.1\}$). During local updating, the optimizer is SGD with 0.9 momentum following most FL baselines []

D.2 Hardware and software

The FedAvg baseline is conducted based on the standard commonly used FL library FedML [47; 127]. While other OFL baselines are implemented following [159]. All experiments are conducted based on NVIDIA 2080 Ti or NVIDIA V100 GPU for several hours. Users just need one single GPU to run these experiments.

D.3 Mutual Information Estimation

Due to the extremely high dimension of intermediate features, MI estimation is very difficult [11]. Simple neural networks may fail to accurately estimate MI. Thus, we follow [145], to use the reconstruction error from $\boldsymbol{h}$ to $\boldsymbol{x}$ to estimate the $I(\boldsymbol{h}, \boldsymbol{x})$, and the classification error to estimate $I(\boldsymbol{h}, y)$. The details are provided as follows.

Estimating $I(\boldsymbol{h}, \boldsymbol{x})$. Assume that $\mathcal{R}(\boldsymbol{x}|\boldsymbol{h})$ denotes the expected error for reconstructing $\boldsymbol{x}$ from $\boldsymbol{h}$. It has been widely known that $\mathcal{R}(\boldsymbol{x}|\boldsymbol{h})$ follows $I(\boldsymbol{h}, \boldsymbol{x}) = H(\boldsymbol{x}) - H(\boldsymbol{x}|\boldsymbol{h}) \geq H(\boldsymbol{x}) - \mathcal{R}(\boldsymbol{x}|\boldsymbol{h})$, where $H(\boldsymbol{x})$ denotes the marginal entropy of $\boldsymbol{x}$, as a constant [54]. We estimate $I(\boldsymbol{h}, \boldsymbol{x})$ by training a decoder parameterized by w to reconstruct the original image x, namely $I(\boldsymbol{h}, \boldsymbol{x}) \approx \max_w [H(x) - \mathcal{R}_w(x|h)]$. For estimating $I(\boldsymbol{h}, \boldsymbol{x})$, the decoders are multiple up-sampling convolutional layers following [145].

Estimating $I(\boldsymbol{h}, y)$. Since $I(h, y) = H(y) - H(y|h) = H(y) - \mathbb{E}(h, y)[-\log\, p(y|h)]$, we can directly train an auxiliary classifier $q\psi(y|h)$ with parameters ψ to approximate $p(y|h)$, such that we have $I(h, y) \approx \max \psi H(y) - \mathbb{E}h[\sum_y -p(y|h) \log\, q_\psi(y|h)]$. To summarize, given the split features H^k at layer k, we freeze previous layers that with index $i \leq k$, and train a new inserted linear layer as classifier as to estimate $I(h, y)$. For estimating $I(h, x)$, the decoder follows [145].

D.4 Linear Separability

Following [7; 101], for each layer k to be examined, we stitch and train a linear classifier MLP_k following it, while freezing previous layers. Thus, the linear separability of feature H^k is shown by the classification error of MLP_k. The MLP is trained by 10 epochs.

D.5 Backdoored Datasets

Figure 5 shows the original images and the backdoored images of CIFAR-10. The shapes are added on images according to label indexes but with random colors. By training on backdoored images, the

local model is easily to fit on the shapes instead of original images. Each shape occupies 10×10 image size. Table 7 provides the test accuracy of different methods training with backdoored CIFAR-10, showing that the performance is severely harmed by the backdoored datasets. While FuseFL is not severely affected by the backdoor.

Figure 5: Each row is a class of original (upper) and backdoored (lower) images of CIFAR-10. The shapes are added on images according to label indexes.

E More Results

To validate the effect of FuseFL when training heterogeneous models with more clients, we further conduct experiments with $M = 10$ and compare results with $M = 5$ as shown in Table 10. Results show that the FuseFL well supports training heterogeneous models, of which the performance is still comparable with Ensemble FL, but requires much less storage cost than Ensemble.

E.1 Heterogeneous Model with Different Number of Clients

Table 10 shows how number of clients influences the performance of FuseFL with average or conv1$\times$1 as adapter. Results show that with increased $M = 10$, FuseFL still provide benefits to training heterogeneous model.

Table 10: Accuracy with FuseFL with conv1$\times$1 or averaging to support heterogeneous model design on CIFAR-10 with different number of clients.

# Clients	$M = 5$	$M = 10$
Ensemble	79.91	77.25
FuseFL	81.15	**78.28**
FuseFL (Avg)	80.35	76.52
FuseFL Hetero	**82.71**	76.47
FuseFL (Avg) Hetero	79.74	75.90

E.2 Comparisons with FedMA

FedMA [141] does not support training ResNet-18 and aggregating multi-layers blocks. Thus, we compare it with FedAvg and our methods on training VGG-9 on CIFAR-10. For FedAvg, we run

it for 10 communication rounds. For ensemble FL methods, the training epoch is set as 200 and communication with only one round. For fair comparison, to keep the same computation burden, we divide the 200 epochs to each communication round in FuseFL and FedMA. Thus, FedMA, FuseFL and Ensemble FL have the same communication costs and $10\times$ less than FedAvg. Note that the FuseFL can have different K to decide its communication rounds, while FedMA has a fixed number of communication rounds, i.e. the number of layers in VGG-9. Thus, the total computation and communication costs of different baselines are same except for FedAvg. Results in Table 11 shows that FuseFL successfully outperforms other methods. Note that to achieve the comparable performance to hundreds-of-rounds FedAvg, the FedMA actually requires communicating multiple model size. UNder the limited communication constraints, the matching and averaging ways in FedMA show inferior performance than the feature concatenation and merging as used in FuseFL.

Table 11: Comparing accuracy on CIFAR-10 with FedMA.

Algorithm	$a = 0.1$	$a = 0.3$	$a = 0.5$
FedAvg	21.19	26.16	38.63
FedMA	66.28	71.95	73.34
Ensemble	65.72	67.29	69.58
FuseFL $K = 2$	73.99	75.58	76.66
FuseFL $K = 3$	72.31	75.29	76.10
FuseFL $K = 4$	71.34	76.87	75.71

E.3 Different Feature Fusion Methods

Table 12: Accuracy with FuseFL with conv1$\times$1 or averaging as adapters of different non-IID degree on CIFAR-10.

non-IID degree	$a = 0.1$	$a = 0.3$	$a = 0.5$
Ensemble	57.5	77.35	79.91
FuseFL $K = 2$	70.85	81.41	**84.34**
FuseFL $K = 4$	**73.79**	**84.58**	81.15
FuseFL $K = 8$	70.46	80.7	74.99
FuseFL (Avg) $K = 2$	70.79	81.5	83.56
FuseFL (Avg) $K = 4$	68.08	71.49	80.35
FuseFL (Avg) $K = 8$	71.58	81.87	83.29

We verify the effect of using conv1$\times$1 or simply averaging features as feature fusion in Table 12. Results shows that using conv1$\times$1 is generally better than averaging. With the decreased non-IID degree, the gap between conv1$\times$1 and averaging is smaller, demonstrating that more similar features require less feature adaptation through learning a mapping i.e. training a conv1$\times$1.

E.4 Communication Cost Comparison

We provide the detailed communication costs of different methods in Table 13. The Other OFLs refer to advanced OFL method including DENSE [159], data-free KD methods DAFL [18] and ADI [153]. Table 13 shows that FuseFL does not increase the communication costs, while largely improving the model performance.

E.5 Higher Heterogeneity and More Baselines

Table 14 provides more results of the higher heterogeneity ($a = 0.05$) and more baselines including DENSE [159] and CoBoosting [23]. Results show that the CoBoosting can improve the performance than other baseline methods but fail to outperform FuseFL.

Both CoBoosting [23] and FEDCVAE-KD [52] focus on exploiting knowledge distillation methods to improve the performance of global models, while our method focuses on how to aggregate models

Table 13: Communication costs. For different number of clients, the number of basic channels in ResNet-18 of FuseFL is set as 32, 20, 14, 9 with $M \in \{5, 10, 20, 50\}$, respectively.

# Clients	$M = 5$	$M = 10$	$M = 20$	$M = 50$
Single Model	42.66MB	42.66MB	42.66MB	42.66MB
FedAvg (10 rounds)	2133.1MB	4266.2MB	8532.4MB	21331.0MB
Other OFLs	213.31MB	426.62MB	853.24MB	2133.10MB
Ensemble	213.31MB	426.62MB	853.24MB	2133.10MB
FuseFL $K = 2$	268.55 MB	423.2 MB	842.6 MB	2274.0 MB
FuseFL $K = 4$	276.9 MB	449.2 MB	944.4 MB	2931.5 MB
FuseFL $K = 8$	340.4 MB	647.7 MB	1722.2 MB	7954.0 MB
FuseFL (Avg)	**266.6 MB**	**416.6 MB**	**816.6 MB**	**2109.0 MB**

together. Thus, the knowledge distillation is orthogonal with our method and may be utilized to enhance FuseFL. For example, one can consider running FuseFL first to obtain a fused global model. Then, this model can be used to conduct knowledge distillation to guide local model training with the FuseFL once again.

Table 14: Results of higher data heterogeneiry ($a = 0.05$).

Datasets	MNIST	FMNIST	SVHN	CIFAR-10	CIFAR-100
Ensemble	58.06	66.19	62.22	53.33	32.25
FedAvg	46.35	20.07	39.41	17.49	6.45
FedDF	80.73	44.73	60.79	37.53	16.07
F-ADI	80.12	42.25	56.58	36.94	13.75
F-DAFL	78.49	41.66	59.38	37.82	15.79
DENSE	81.06	44.77	60.24	38.37	16.17
CoBoosting	93.93	50.62	65.40	47.20	19.24
FuseFL $K = 2$	95.23	83.23	75.08	46.38	29.98
FuseFL $K = 4$	**95.37**	**83.65**	**75.53**	**51.59**	**32.71**

F Limitation

Finding invariant features. With our analysis in Section 3, it will be useful to choose the invariant features R_m^{inv} during local training for client n. Currently, under the communication and privacy constrains, it is further difficult to identify which features are spurious. We left this as the future work to explore. We do not explore and exploit this technique in this paper, as it is orthogonal to our core innovation technique, augmenting features by layer-wise model fusion. Nevertheless, by manually crafting spurious correlations by adding backdoored data samples into local dataset, we empirically prove FuseFL can effectively avoid the influences of spurious features.

Limited feature adaptation. In this work, as an initial trial, we only explore using averaging and conv1$\times$1 as the feature adaptation. Further methods can consider exploring better feature adaptation methods like using non-linear models.

Security issues. In this work, we do not explicitly consider the security issue. However, the vulnerability to attacks of FuseFL will not be higher than previous multi-round FedAvg, which requires many more communication rounds to achieve the same model performance as FuseFL. For example, FedAvg may require more than 100 rounds to achieve 70% test accuracy as FuseFL with $2 \sim 4$ rounds, which introduces more communicated information and a higher possibility of attacks. And FuseFL has the same communication size with other OFL methods.

However, while the size of the communication does not increase, FuseFL increases the number of communication rounds compared to other OFL methods. There are some possible mitigations to address the security issues:

- **Adversarial attacks.** Some malicious clients might upload adversarial modules or backdoored modules that are used to misguide the aggregated model to generate incorrect or handcrafted predictions. For these attacks, the possible solution is to detect and reject such malicious uploading also through the lens of causality. Specifically, some images with the invariant features can be fed into the uploaded modules to see whether the output feature can be used to correctly classify images;
- **Model inversion or Membership attack.** Some malicious clients or the server may consider to conduct model inversion or membership attack to obtain the raw data of clients, thus threatening the user privacy. In this case, the learned module can be protected with differential privacy to enhance its security.

Multiple communication rounds. To enable concatenating local modules together and training based on aggregated features from these modules, multiple communication rounds are required in `FuseFL`. In this sense, `FuseFL` belongs to the few-shot FL. However, the communication cost of `FuseFL` is as same as other OFL methods, which is the main claim in the introduction (extremely low communication costs).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations have been discussed in Section F. The communication cost and storage costs are compared in Table 4 and 13. The scalability on datasets and clients, computation efficiency, and whether supporting personalized model design have been discussed in Experiment Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: We build the causal graph without giving the theorem. The proof of Lemma 3.1 comes from [2]

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We have provided the experiment setting, specific baselines (Section 6), hyperparameters (Section D), the algorithm details (Algorithm A, hardware and software details (Section D.2). We have provided our open-sourced code link.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We have provided our open-sourced code link. The library has provided the experiment and code instructions. And the datasets are public datasets as listed in Section 6, which can be downloaded online, or using Pytorch.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have provided the experiment setting, specific baselines, optimizers(Section 6), hyper-parameters (Section D), the algorithm details (Algorithm A, hardware and software details (Section D.2).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report the error bars in our experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided the experiment setting (Section 6) and hardware and software details (Section D.2).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work does not incorporate any ethic concerns of NeurIPS. The datasets and models are commonly used in the community, and the method does not incorporate potential concerns.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: We have discussed the broader impact in Section B.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The datasets and baselines, used libraries are properly credited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The datasets and baselines, used libraries are well documented and cited.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.