Cross-Silo Federated Learning Across Divergent Domains with Iterative Parameter Alignment

Matt Gorbett
Department of Computer Science
Colorado State University
Fort Collins, CO, United States
Matt.Gorbett@colostate.edu

Hossein Shirazi
Fowler College of Business
San Diego State University
San Diego, CA, United States
hshirazi@sdsu.edu

Indrakshi Ray
Department of Computer Science
Colorado State University
Fort Collins, CO, United States
Indrakshi.Ray@colostate.edu

***Abstract*—**
Learning from the collective knowledge of data dispersed across private sources can provide neural networks with enhanced generalization capabilities. Federated learning, a method for collaboratively training a machine learning model across remote clients, achieves this by combining client models via the orchestration of a central server. However, current approaches face two critical limitations: i) they struggle to converge when client domains are sufficiently different, and ii) current aggregation techniques produce an identical global model for each client. In this work, we address these issues by reformulating the typical federated learning setup: rather than learning a single global model, we learn N models each optimized for a common objective. To achieve this, we apply a weighted distance minimization to model parameters shared in a peer-to-peer topology. The resulting framework, Iterative Parameter Alignment, applies naturally to the cross-silo setting, and has the following properties: (i) a unique solution for each participant, with the option to globally converge each model in the federation, and (ii) an optional early-stopping mechanism to elicit *fairness* among peers in collaborative learning settings. These characteristics jointly provide a flexible new framework for iteratively learning from peer models trained on disparate datasets. We find that the technique achieves competitive results on a variety of data partitions compared to state-of-the-art approaches. Further, we show that the method is robust to divergent domains (i.e. disjoint classes across peers) where existing approaches struggle.

I. Introduction

Federated Learning (FL) addresses issues of data privacy and access rights by enabling wide-scale training of machine learning models across decentralized data sources [1]–[3]. Standard FL involves clients (e.g. mobile, edge devices) training a model locally with private data and communicating their model updates back to a central server. The server aggregates client models into a global model and returns it to each client, a process that repeats iteratively until a final global model is produced. Traditional FL often addresses *cross-device* settings where clients consist of unreliable devices. Extensive research has concentrated on addressing issues related to cross-device FL such as communication constraints and heterogeneous data partitioning [1], [4]–[8].

A second setting, *cross-silo* FL, involves training a machine learning model across large organizations such as banks [9], [10] and hospitals [11]–[14]. Silos in this scenario generally have big data, extensive computational resources, and strong network communication [15]. Further, the setting often contains fewer clients compared to cross-device FL.

Motivation. In this work we identify and address two issues present in current FL algorithms. First, we identify a novel failure scenario in current FL frameworks: cross-domain global model aggregation. Specifically, when clients have divergent domains, such as completely different labels, common FL approaches fail. Figure 1 (center) highlights the issue, with existing algorithms FedAvg [2], FedDC [7], and FedDyn [6] each failing to converge to baseline test accuracy when three clients have differing labels (e.g. client one has only training samples of animals and another only vehicles). Cross-domain scenarios are important in the real-world such as those involving GDPR where an entire demographic segment is isolated, or cross-industry learning where the domains of peers are disjoint. In Section IV-A, we show that existing FL algorithms consistently have unstable results across various datasets and label splits.

In addition, we also address an overlooked characteristic of existing FL: *the global model is identical for each participant*. This property can lead to important disadvantages. In particular, the global model is exposed to all participants in the federation. In the cross-silo setting this may leave a client model unprotected against direct competitors, exposing obvious vulnerabilities such as white-box attacks [16]. Personalized FL is an alternative approach which produces individualized models for each client unique to their data distribution [17]–[19], including methods to produce joint personalized and global models [20], [21], however current approaches still produce a *single* global model (details in Section II).

Proposed Approach. To address the issues of divergent client domains and a single global model, we propose the Iterative Parameter Alignment (IPA) algorithm for merging machine learning models across silos. Unique from existing approaches the algorithm trains N different models, one for each silo. The models each have arbitrary initializations, different from current techniques which require the same initial parameters [2]. IPA works by iteratively merging the models by minimizing the distance between weights. The architecture is depicted in Figure 1 (right).

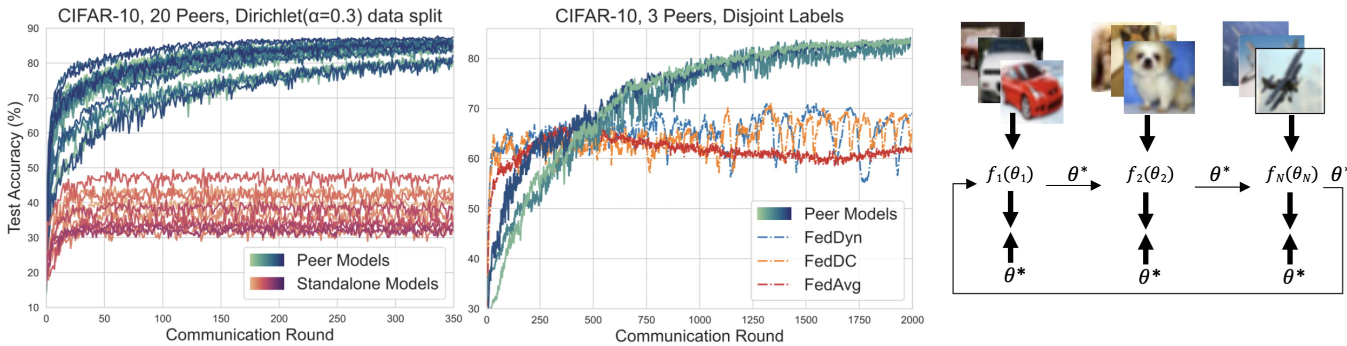

Fig. 1: **Left:** Test set accuracy across communication rounds of peers trained with Iterative Parameter Alignment compared to their standalone performance (trained only on their local data). There are twenty peers each trained with an imbalanced subset of the CIFAR-10 training set. They are split using heterogeneous data partitioning using a Dirichlet distribution with $\alpha = 0.3$. One communication round (the x-axis) equals each $peer_i$ training their model (f_i) once. **Center:** Three peers each trained with distinct CIFAR-10 training labels (one peer has 4 labels, two peers have 3 labels each). We find that when peers have sufficiently divergent domains, existing FL methods fail, creating global models that do not reach baseline accuracy. Iterative Parameter Alignment produces distinct global models for each peer that each converge to baseline accuracy (85% on the test set). **Right:** A single iteration of Parameter Alignment trained in a ring topology (random topologies are used in experiments). The method relies on parameter exchange and alignment to learn from others. $\theta_1, \theta_2, ...\theta_N$ are N peers parameters and $f_1, f_2, ...f_N$ are the models. θ^* represents all peer parameters $\{\theta_1, \theta_2, ...\theta_N\}$. Each $peer_i$ can optionally apply differential privacy to their θ_i for protection. Our code is available at https://github.com/mattgorb/iterative_parameter_alignment.

Essential to cross-silo FL, IPA can protect the client's data *and* the client's final model. Data protection is a primary goal of FL (achieved via data localization [2] and differential privacy [22]–[25]), however, model protection is a more ambiguous task. Homomorphic encryption is the primary model protection technique in FL, enabling clients to encrypt model updates for protection against a central server [26], [27]. Differential privacy also achieves model protection by enabling clients to add noise to their models parameters [28]. However, in each of these scenarios the global model is still the same for each client. Some techniques emphasize fairness by producing varying models which depend on a clients data contribution [29]–[31]. However, such approaches produce client models derived from the same global model, which are produced at the server. Moreover, fairness may not be a requirement for every FL scenario, and in such cases it may still be desirable to create differing global models without this constraint.

IPA addresses issues of model protection through silo-to-silo federated learning over N unique models. This decentralized topology is favorable particularly in cross-silo scenarios where a central server may create a communication bottleneck [32]. By creating unique global models for each peer, IPA prevents peers from knowing each others parameters, an approach achieved through differential privacy. In particular, a peer may add noise to its model parameters to protect its model from the other peers (we assess IPA under differential privacy in Section V). We study the differences in peer models without differential privacy in Section IV-C.

In addition to model protection, IPA provides the flexibility for peer models to either converge to a global optimum, or decide on an early-stopping point to elicit fairness (in cases of heterogeneous data silos). For example, when one peer has more data than another, their model will converge faster using the IPA algorithm. If fairness is a requirement, peers can decide on an early stopping point so that the higher contributor achieves a stronger model. If fairness is not a requirement, all peers will still converge to a global optimum (each with unique parameters). We study this property of IPA in Section IV-D.

Contributions. We propose IPA for merging peer models trained on separate data. Different from existing approaches, the algorithm produces a unique model for each peer (silo) in the federation, each with arbitrary initializations. IPA works by iteratively merging peer models by minimizing the distance between weights. The architecture is depicted in Figure 1 (right). IPA offers several advantages compared to existing methods:

- IPA is robust in scenarios with completely segregated labels across peers, including scenarios where existing FL algorithms fail to converge.
- IPA achieves state-of-the-art convergence rates on balanced data partitions (Table I). Further, the method achieves competitive results (significantly outperforming FedAvg) with heterogeneous data sources, a known burden of standard FL [1], [2], [6]–[8].
- It produces unique peer models in a decentralized topology, providing independence from a central orchestrator and implicit collaboration with peers.
- The method produces distinct global models for each peer, which we analyze in Section IV-C.
- IPA contains *built-in* fairness: we show that model performance on classification tasks is correlated with a peers standalone model performance. We propose an early stopping mechanism to elicit fairness in Section IV-D.

II. Related Work

Federated Learning. The pioneering FL framework, Federated Averaging (FedAvg), aggregated a global model by averaging the weights of client models trained on private data [2]; heterogeneous data partitioning, inefficient communication, and variable participation across clients were identified as key challenges [33]–[35]. Subsequent work improved the convergence rate of heterogeneous client data through corrections to the gradients of local models [8], regularization of local models against the global model [1], dynamic regularization of local models [6], and correcting local model drift from the global model [7].

Cross-Silo Federated Learning. Cross-silo FL involves training machine learning models across entities with large data-silos (i.e. data centers) [5], [15]. Peer-to-peer communication has been proposed as an effective alternative to centralized orchestration in cross-silo federations with reliable participants [32]. Marfoq et al. examine the effect of topology on the duration of communication rounds in cross-silo settings, and propose algorithms for measuring network characteristics to construct a high-throughput network topology. Other works address security and personalization of cross-silo FL [22], [36]. We consider cross-silo FL a realistic application for IPA due to the large computational costs of the algorithm, as well as organizations' potential desire to maintain independent models.

Collaborative Learning. Important to cross-silo FL is designing incentive mechanisms for peers to participate in a federation, commonly referred to as collaborative learning. Participants may have concerns about contributing their data for the benefit of others. For example, if two peers are direct competitors they may be concerned that the other peer will benefit more from federated learning. As a result, *fairness* schemes have been proposed using methods such as contract theory [37], [38], monetary payouts [39], and game-theoretic approaches [40], [41]. Lyu et al. [30] propose a credibility metric so that each participant receives a different version of the global model with performance comparable to its contribution. Similar to our work the authors use a decentralized framework (they propose blockchain). Different from their approach, IPA works in cross-domain settings and produces differing global models. IPA is additionally a less complex framework. Xu et al. [29] propose a reward mechanism that specifies model updates at the server commensurate to a client's contributions. Other works utilize the Shapely value [42] and reputation lists [31] to evaluate client contributions.

Personalized Federated Learning. Personalized FL produces individualized models that are catered to a client's data distribution while also leveraging the data of the federation [17]. Clients can create personalized models via local fine-tuning of the global model [5], or from more advanced techniques such as hypernetworks [43], pruning [44], encouraging interaction between related clients [18], [21], [45], [46], and learning client-level and shared feature extractors [47], [48]. Research also addresses *fairness* in personalized FL [19], [49], identifying performance disparity across clients as a key issue.

Some methods create high performing personalized *and* global models. `FedRoD` [20] utilizes an additional local layer on a global model to create a high performing personalized model, while `FedHKD` [21] uses local "hyper knowledge" to aggregate the global model. However, these approaches create identical global models across clients. Further, the methods centrally aggregate the global model.

IPA versus personalized FL. Unique from IPA, personalized FL methods produce models *individualized to each clients data distribution*. For example, if one client has data of dogs and another has data of cats, they may not benefit from each other. Unlike existing research in personalized FL, IPA aims to learn an individualized model on a common objective for each peer in the network. In our dogs and cats example, each peer would learn a different global model that does well at classifying dogs and cats in a decentralized network topology.

III. Method

We begin by reviewing the standard federated averaging objective, followed by describing the unique approach of IPA.

Background. In standard FL there are N clients in a federation, where each client i has a local dataset $\mathcal{D}_i$. The goal is to solve a common objective over a universal dataset $\mathcal{D} = \cup_{i\in[N]}$ by aggregating each local model into a global model. The system iterates between local training on each client and global aggregation at the server. `FedAvg`, the original FL algorithm [2], involves a weighted averaging of client parameters at the server:

$$\textbf{Local}: \quad \theta_i = \arg\min_{\theta\in\mathbb{R}} \mathcal{L}_i(\mathcal{D}_i;\theta), \text{ initialized with } \theta \tag{1}$$

$$\textbf{Global:} \quad \theta = \sum_{i=1}^{N} \frac{|\mathcal{D}_i|}{|\mathcal{D}|}\theta_i \tag{2}$$

where θ_i is the local model's parameters, θ is the global model's parameters, $\mathcal{L}_i(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}_i}\left[\ell_i(f(x), y; \theta)\right]$ is the local empirical loss of model i on dataset $\mathcal{D}_i$, and x and y are the samples and labels in $\mathcal{D}_i$.

Iterative Parameter Alignment. To begin, we consider a set of N peers (rather than clients) where peer i has access to local dataset $\mathcal{D}_i$. Similar to standard FL, our goal is to solve an objective over universal dataset $\mathcal{D}$ for each peer model $f(\theta_i)$. To do this, each peer solves both an empirical learning objective, denoted $\mathcal{L}_i$, as well as an alignment objective $\mathcal{A}_i$, which together minimize the set of *all* peer parameters θ^*, where $\theta^* = \{\theta_1, ...\theta_N\}$:

$$\arg\min_{\theta^*\in\mathbb{R}} \left[\mathcal{L}_i(\mathcal{D}_i;\theta_i) + \mathcal{A}_i(\theta^*)\right]$$

where $\mathcal{L}_i(\theta_i) = \mathbb{E}_{(x,y)\sim\mathcal{D}_i}\left[\ell_i(f(x), y; \theta_i)\right]$ and x and y are samples from $\mathcal{D}_i$. For experiments in this work, we set ℓ to be a cross-entropy loss for image classification problems. Importantly, $\mathcal{D}_i$ is only seen by peer model $f(\theta_i)$, from which the empirical loss is calculated. Moreover, peers are not able to share data with each other, only model parameters. This is similar to parameter sharing among the client and server in standard FL. We can apply differential privacy to the parameter sharing similar to previous work [28], albeit in a decentralized (rather than centralized) topology.

Key to the global convergence of a peer model is the alignment of parameters during training. Specifically, model i holds parameters θ^* locally, and during each minibatch updates θ_i by minimizing the distance between θ_i and *each* θ_n. For a single weight matrix or bias for model i we denote this as a_i:

$$a_i(\theta^*) = \sum_{n=1}^{N} ||\theta_i - \theta_n||_p, \text{ where } i \neq n \tag{3}$$

where p is the L_1 or L_2 distance. In other words, a_i is the sum of distances in θ^* between θ_i and θ_n. Generalizing parameter alignment across the weights and biases of each layer $1, l, ...L$ of a neural network we achieve our alignment objective for model i:

$$\mathcal{A}_i(\theta^*) = \lambda \sum_{l=1}^{L} a_i(\theta^*) \tag{4}$$

where λ is a global scale factor on the weight alignment objective. We set λ to 1 in this work. IPA leads to a minimization of the global loss in individual models who have never seen the global dataset. In other words, when solving for the alignment objective in Equation III, we show that a peer model with access to the full parameter set θ^* iteratively converges to an objective solved over the global dataset $\mathcal{D}$: $\arg\min_{\theta_i} \mathcal{L}_i(\mathcal{D}_i;\theta_i) \rightarrow \arg\min_{\theta} \mathcal{L}(\mathcal{D};\theta)$. Compared to standard FL, the IPA algorithm only updates parameters on peer devices in a decentralized and synchronous architecture. Further, the method relies on independent (i.e. never aggregated) peer models. In the next section, we highlight the benefits of the approach in various settings.

Algorithm 1 Parameter Alignment, One Iteration.

Input:
N peers
Peer$_i$ has: dataset $\mathcal{D}_i$, model f_i with weights θ_i
θ^* is all peer parameters: $\{\theta_i, ...\theta_N\}$
Output:
Models $f_1(\theta_1), f_2(\theta_2), ...f_N(\theta_N)$

Each peer initializes θ_i, sends to $\texttt{peer}_1$
for each $\texttt{peer}_i \in N$ **do**
  for each batch $b \in \mathcal{D}_i$ **do**
    $\mathcal{L}_i = \ell(f_i(b;\theta_i)) + \textsc{ParamAlign}(f_i, \theta^*)$
    $\theta_i \leftarrow \theta_i - \nabla \mathcal{L}_i$
  Transfer θ^* to $\texttt{peer}_{i+1}$

$\textsc{ParamAlign}(f_i, \theta^*)$:
  $\mathcal{R}_i \leftarrow 0$
  for each layer $\in f_i$ **do**
    for each $\theta_j \in \theta^*, j \neq i$ **do**
      $\mathcal{R}_i \leftarrow \mathcal{R}_i + |\theta_i - \theta_j|_p$
  return $\mathcal{R}_i$

IV. Experiments

We begin by evaluating Iterative Parameter Alignment against existing methods in federated learning, including experiments merging peer models trained on segregated classes. Next, we quantify the difference between peer models, showing that each peer produces a distinct model in both parameter space and during inference. Finally, we highlight the ability for IPA to produce fair models (at epoch t), converging thereafter to globally optimized solutions.

A. Domain Divergent Silos

Unique to this work we experiment with merging peer models that have completely segregated classes. For example, Peer$_1$ may only have images of dogs while Peer$_2$ has only images of cats. Such scenarios are important in the real-world such as those involving GDPR where an entire demographic segment is isolated, or cross-industry learning where the domains of individual peers are disjoint.

The scenario also highlights the distinction between IPA and personalized FL [20], [21]. In the above example, personalized FL would aid Peer$_1$ to better generalize to its own domain (dogs) by utilizing Peer$_2$ data. However, Peer$_1$ may not gain much value from Peer$_2$'s information about cats. We further distinguish IPA from personalized FL in Section II.

IPA, in contrast, can successfully merge two or more seemingly independent domains. Figure 1 (center) shows how three peers trained with different CIFAR-10 labels can be iteratively aligned and each converge to the accuracy of a baseline model trained on all data.

We compose our experiments with simple class splits, such as a two-peer class split where one peer has all training data labeled 0 to 4 and the second peer has training data labeled 5 to 9 (in a dataset with 10 classes). We also consider imbalanced splits such as peers with an unequal number of classes.

Results. Figure 2 highlights the convergence of peer models trained using the IPA algorithm on disjoint classes. We find that compared to `FedAvg`, `FedDyn`, and `FedDC`, IPA achieves stable training. Moreover, under both balanced splits (each peer has the same number of labels) and imbalanced data splits, IPA converges consistently to baseline accuracy. Existing algorithms such as `FedDyn` and `FedDC` have very unstable training; their global model test accuracy curves were smoothed in Figure 2 for visualization purposes. `FedAvg` had more stable training, however, its naive parameter averaging technique did not converge to baseline accuracy. Rather, its performance flattened out after a few communication rounds.

We hypothesize that existing FL algorithms are unstable in the segregated class scenario because the gradient updates of local models are entirely disassociated from each other as a result of the domain discrepancy. Existing work has shown that clients with heterogeneous data partitions have inconsistent optimization directions [8], [33], which cause drifts in the local models away from a global solution. We tried over half a dozen different configurations for existing algorithms, including different seeds, reduced learning rate, and a smaller number of local epochs.

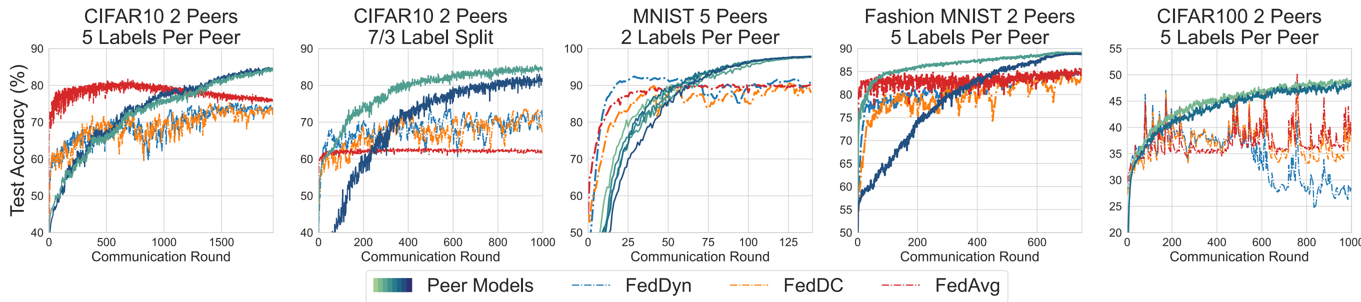

Fig. 2: **Aligning Peer Models Trained on Disjoint Classes:** We find that existing federated learning approaches such as `FedAvg` struggle when trying to merge models trained with divergent (rather than heterogeneous) data partitions. We show how IPA achieves stable training compared to existing approaches, with IPA eventually converging to baseline accuracy compared to other methods which create global models with unstable performance.

Dataset	Target Acc. (%)	FedAvg	FedProx	Scaffold	FedDyn	FedDC	IPA
IID, 20 Peers, $p = 2$							
MNIST	98	49	46	50	20	33	3
Fashion	89	148	151	165	35	100	14
CIFAR-10	85	42	46	31	20	20	15
CIFAR-100	50	82	84	45	60	43	30
Dir. ($\alpha = 0.6$), 20 Peers, $p = 1$							
MNIST	98	147	140	52	20	35	28
Fashion	87	60	67	62	15	40	60
CIFAR-10	85	64	65	44	22	24	44
CIFAR-100	50	105	105	56	61	55	97
Dir. ($\alpha = 0.3$), 20 Peers, $p = 1$							
MNIST	98	139	199	57	45	39	70
Fashion	87	98	93	92	25	50	90
CIFAR-10	85	133	144	58	28	29	95
CIFAR-100	50	111	110	64	74	55	103

TABLE I: **Communication rounds required to achieve target accuracy:** We compare the number of communication rounds required for IPA and other state-of-the-art FL algorithms to reach a target accuracy. IPA converges quickly on IID data, with competitive results on heterogeneous splits. IPA does not achieve state-of-the-art performance in heterogeneous experiments, however, communication is less of a constraint in cross-silo settings.

B. Comparison to Existing Approaches

Our second empirical study compares the convergence rate of Iterative Parameter Alignment against existing FL algorithms. McMahan et al. [2] noted the slow convergence of `FedAvg` when clients had heterogeneous data partitions. Since the initial research, much effort has been put into improving this convergence rate, which is measured by the *number of communication rounds* between the clients and the server until the global model reaches some target accuracy on the test set. We test IPA in a similar fashion, where one communication round equals each peer performing their allotted training.

Experimental Setup. We construct our experiments from a set of scenarios with homogeneous and heterogeneous data partitions consistent with previous research. In heterogeneous settings, our label ratios follow the Dirichlet distribution with $\alpha = 0.3$ and $\alpha = 0.6$, similar to previous works. Lower α indicates a higher data heterogeneity. We compare Iterative Parameter Alignment to the standard FL algorithm `FedAvg` [2] as well as state-of-the-art approaches `FedProx` [1], `Scaffold` [8], `FedDyn` [6], and `FedDC` [7]. The original hyperparameters are used for each algorithm. We compare algorithms using MNIST, FashionMNIST, CIFAR-10, and CIFAR-100 datasets. We use the same architecture as previous works for the MNIST and FashionMNIST datasets; for the CIFAR-10 and CIFAR-100 datasets, we use a larger CNN model which includes four convolutional layers followed by three linear layers. We consider one round of communication as each client training the model and sending it back to the server for aggregation (100% client participation). For IPA, we report the number of communication rounds it takes for the first peer to reach a target accuracy.

Unique to Iterative Parameter Alignment, we report the convergence rates of peer models with *different initializations*, i.e. each peer model is initialized from a different random seed. In the original FL work, the authors highlighted the success of naive parameter averaging when models had the same initial weights. Averaging did not perform as well when the models were initialized differently. This phenomenon was also reported in model merging literature [50], where the authors required models trained from the same initial weights. Research has suggested permutation invariance of neural networks as a driving force for this observation, i.e. a neural network has many variants which differ only in the ordering of its parameters [51].

Results. Table I highlights the results of IPA against five state-of-the-art methods. Unsurprisingly, under IID settings IPA converges quickly towards the target accuracy on all four datasets. While the algorithm only feeds dataset $\mathcal{D}_i$ to $f(\theta_i)$, it has $20 \times \bar{\theta}$ parameters, optimizing $19 \times \bar{\theta}$ parameters using alignment and the final θ_i using alignment plus empirical loss. As a result, the balanced, overparameterized networks converge quickly despite only having access to a fraction of the training samples. Compared to existing approaches, IPA achieves state-

	FashionMNIST		CIFAR-10	
Distance	Dir(0.3)	IID	Dir(0.3)	IID
$\|\|\theta_i - \theta_j\|\|_1$	196.5	1352.6	1.8×10^3	3.3×10^4
$\|\|\theta_i - \theta_j\|\|_2$	0.7	4.9	2.0	35.9
$\mathcal{H}(f_i, f_j)$	1,990	650	2,504	1,043
$f_i \wedge f_j$	7,358	8,603	6,871	8,214
$\overline{f_i} \wedge \overline{f_j}$	947	837	1,094	947

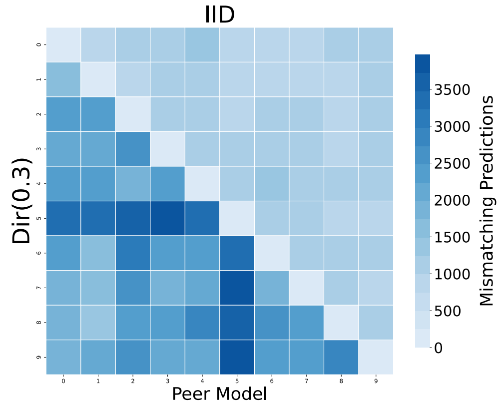

Fig. 3: **Comparing Peer Models:** We measure the distance between peer models across a variety of metrics. Each experiment contains ten peers and is aggregated across three runs, with the mean presented for each. **Left:** Measuring the distance between models across parameters (first two rows) and model predictions (the last three rows). The last three rows denote the Hamming distance between predictions, mutual correct predictions, and mutual incorrect predictions on the test set. Test set size for both datasets is 10k. **Right:** A similarity matrix of Hamming distances between peer model predictions for: 1) heterogeneous data partition (bottom triangle) and 2) homogeneous (IID) data partition (top triangle). The distances represent the number of mismatching predictions in the test set for each model. For reference, the lowest (averaged) Hamming distance between models in the IID setting is 880, with a test set size of 10k.

of-the-art performance.

Under increasingly heterogeneous settings (from top to bottom) we observe a longer convergence rate for IPA compared to other algorithms. IPA remains competitive for MNIST and FashionMNIST, however, has a slightly longer convergence rate for CIFAR-10 at Dirichlet ($a = 0.3$) as well as CIFAR-100. We argue that convergence rate is less of a concern in cross-silo settings since large companies likely have adequate computation. Moreover, it achieves better accuracy than baseline models FedAvg and FedProx.

C. Peer Model Comparison

We look at the quantitative differences between peer models across a variety of metrics to assess whether IPA creates *sufficiently unique* models. Existing literature has found that neural networks are known to be sensitive to small changes in their parameters [52], causing drastic changes in model inference and generalization. There is a rich area of research examining this phenomenon for injecting adversarial attacks [52]–[54], evaluating the generalization gap of model minima [55], [56], and assessing the effects of model quantization [57]. As a result, even the smallest differences in the weights of peer models can create unique results.

Experiments. To quantify the difference between two peer neural networks we compare both the network parameters as well as the predictions. We measure the distance between two models' parameters as $||\theta_i - \theta_j||_p$, where $p = \{1, 2\}$. To measure the difference between model predictions, we compute the Hamming distance between models' outputs on the test set, which we denote $\mathcal{H}(f_i, f_j)$. We also present a count of when both models' predictions are correct (denoted $f_i \wedge f_j$), as well as both incorrect ($\overline{f_i} \wedge \overline{f_j}$).

We test heterogeneous (Dirichlet with $\alpha = 0.3$) and homogeneous scenarios with both the FashionMNIST and CIFAR10 datasets. All experiments use ten peers and are averaged over three runs. We choose a lower number of peers compared to previous experiments in order to magnify potential similarities between models. Heterogeneous experiments are trained for 200 epochs and homogeneous experiments are trained for 50 epochs. The FashionMNIST experiment on homogeneous data had a test accuracy of 88.9%$\pm$0.24, and the heterogeneous scenario 82.5%$\pm$3.42. The CIFAR10 experiment on homogeneous (IID) data had a mean test accuracy of 86.4%$\pm$0.44, and the heterogeneous scenario 79.5%$\pm$4.12.

Results. Table 3 highlights the differences between peer models across four experiments. The first two rows indicate a dissimilarity between peer model parameters across L_1 distance, with a smaller discrepancy when measured with L_2 distance. We hypothesized that IID data experiments would have closer parameters, however, the heterogeneous experiments yielded smaller values. We speculate this is because we train heterogeneous data for 200 epochs compared to just 50 epochs for IID data.

The bottom three rows measure the difference in test inference between peer models, with both datasets having a test set size of 10k. The smallest Hamming distance was between IID models, with 650 for FashionMNIST and 1,043 for CIFAR-10. We argue that these values indicate a significant difference from each other since IID models achieve 88.9% and 86.4% accuracy on the test set. Finally, we note that the standard error was negligible across all experiments.

D. Fairness through Early Stopping

In cross-silo settings organizations may be competing against each other, hence the contribution of participants becomes

	MNIST 20 Peers		CIFAR-10 10 Peers	
Algorithm	CLA	POW	CLA	POW
q-FFL [58]	38.7	48.07	51.33	94.06
CFFL [31]	94.7	85.71	72.55	81.31
ECI [59]	99.41	95.21	79.5	99.55
CGSV (β=1) [29]	96.39	97.23	98.78	99.89
CGSV (β=2) [29]	91.33	94.32	88.78	93.39
IPA (Ours)	96.44	95.98	95.86	92.22

TABLE II: **Fairness of IPA compared to existing approaches:** We compare the fairness of various collaborative learning approaches against IPA by measuring the *correlation* of each clients model performance compared to its standalone models' performance. Correlation is scaled between -100 and 100.

a critical measure. Designing proper incentive mechanisms and rewards for participation can encourage peers to join a federation. Previous work has proposed fairness schemes using methods such as contract theory [37], [38], monetary payouts [39], game-theoretic approaches [40], [41], the Shapely value [29], [42], and reputation lists [31]. Most of these methods produce variations of the single global model, i.e. models for each client whose performance is commensurate to its data contribution.

IPA takes a different approach: Figure 1 highlights the variable convergence rates of peer models with heterogeneous data partitions. We find that the convergence of a peer model trained with IPA is a function of the peers' standalone model performance. We enable fairness in the IPA algorithm through the *early stopping* of training at some iteration $t < T$, where T is the number of iterations it takes for *all* peer models to converge to some target accuracy.

To test our approach, we conduct experiments designed from benchmarks in previous works. We measure fairness using a scaled Pearson's coefficient: $100 \times \rho(\varphi, \xi) \in [-100, 100]$ [29]–[31]. Specifically, we measure the correlation between the test set accuracy of the set of standalone models (φ) compared to the test set accuracy of the set of models generated by IPA (ξ) . *The intuition is that peers should have a federated model with similar capabilities to their standalone model relative to others peers.*

Our first experiments involve comparing our method with the benchmarks of Xu et al. [29] since their approach provides theoretically guaranteed fairness metrics. Additionally we compare q-FFL [58], CFFL [31], and ECI [59]. Experiments use the CIFAR10 and MNIST datasets and apply class-imbalanced (CLA) and size-imbalanced (POW) data partitioning each using 600 samples per peer. For IPA, we run each model in a random topology with one local training epoch per peer since there is a limited amount of training data. For CIFAR10, we average peer model performances on the test set for epochs 5-15, while in MNIST we average peer model performances in epoch 1-5. Results in Table II show that IPA has a correlation above 86 for each of the four tests, with three of the results above 95. These metrics are on average stronger than each prior method except for CGSV at $\beta = 1$.

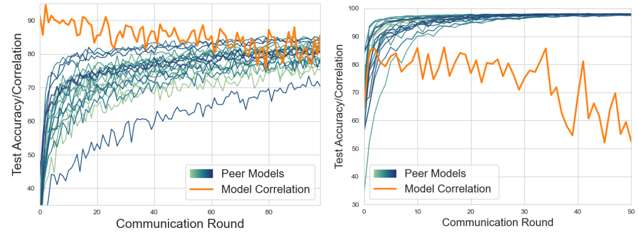

Fig. 4: **Fairness across iterations:** We show that the IPA algorithm creates fair models earlier in training before the global convergence of peer models. CIFAR-10 (Left) and MNIST (Right) performance across models and communication rounds, overlaid with peer models' correlation with their standalone performances (orange). We find that early in training, peer model performances are correlated with the performance of their standalone models relative to other peers. As training proceeds and the peer models globally converge, model fairness decreases, as can be seen in the MNIST figure.

Next we experiment with a more robust and realistic data partitioning by using the full CIFAR-10 and MNIST datasets, 20 peers, and a Dirichlet split with $\alpha = 0.25$. We run each experiment four times in a random topology and test the correlation between IPA and standalone model performance. In these experiments, we find that the *test loss* (rather than the test accuracy) is a stronger metric for correlation. For CIFAR-10, we average the test loss between communication rounds 50 and 100 to gain a thorough picture of the correlation, and to counter the variance of individual communication rounds. For MNIST, we average communication rounds 5 to 25. Overall our CIFAR-10 experiments have a correlation of 86.3 $\pm$2.2, while our MNIST experiments have a correlation of 80.5 $\pm$3.4. Figure 4 depicts the test accuracy of peer models across communication rounds overlaid with the correlation (orange) of the group of peer models compared to the group of standalone models.

Finally, we would like to note that IPA offers a distinct advantage in homogeneous settings: in existing fairness approaches, peer models will be more or less identical as a result of being derived from the same global model. IPA, however, will produce a unique solution for each peer in the homogeneous setting.

V. Additional Analysis

Simulating Differential Privacy To simulate differential privacy we run two experiments which test the effect of adding noise to peer model parameters. Our motivation is to test whether peer models still converge to a high test set accuracy when differential privacy is applied to each peer. Specifically, when one peer is finished with a training iteration, we add a small amount of noise to their parameters (θ_i) prior to sharing with others. We add random noise with $\mu = 0$ and $\sigma = 0.0005$.

Our first experiment uses the CIFAR-10 dataset with two silos, each with half of the labels. Both models converge to 85% test accuracy after roughly 4,000 rounds. Our second experiment uses CIFAR-10 with 20 peers using a Dirichlet data split with $\alpha = 0.6$. The first silo converges to 85% test accuracy after 197 rounds. While both of these are significantly longer than IPA without differential privacy, differential privacy provides security guarantees for each silo.

L_1 versus L_2 Parameter Alignment. In Figure 5 (left) we show that split label experiments exhibit instability when using squared error (L_2) alignment, while absolute error (L_1) alignment achieves smooth convergence. We observed similar results in all experiments using heterogeneous and disjoint data partitions.

Effect of Initialization Strategy. We show how models are able to be aligned even when they have different initializations. In Figure 5 (right), we show the convergence of a CIFAR-10 experiment with ten peers split with Dirichlet with $\alpha = 0.25$. Both the green (same initialization) and orange (different initialization) converge at similar rates.

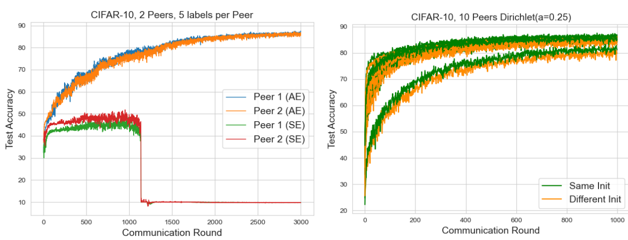

Fig. 5: **Left:** We show the instability of squared error alignment compared to absolute error in a CIFAR-10 experiment with two peers with 5 labels each. **Right:** We discover similar convergence rates of peers when models have the same initialization (green) and different initializations (orange).

VI. Discussion

Limitations. IPA is feasible specifically in cross-silo settings where peers have an adequate amount of computational capacity. It does not scale well to many peers as a result of requiring $N \times \bar{\theta}$ parameters during training unless all peers have large computational capacity. For example, IPA worked well in our experimental settings with up to 20 peers on a single GPU where each model had 2-3 million parameters. We note that advances in neural network pruning and quantization may enable the method to scale well in the future [60]–[62].

Additionally, we note that IPA works well under settings with *reliable* peers. Standard FL considers scenarios where peers go offline; we do not consider this scenario in this paper. However, if one peer drops out during the IPA training process, their latest parameters will still be available for others to continue.

There are additional settings we have not considered in this paper such as tasks other than image classification and vertically aligned FL [5].

Additional Security Considerations. Key to our approach is sharing model parameters across peers during the IPA training process. While each peer produces an independent global model, each peer has access to others parameters during training. This may lead to inadequate security and potential for misuse. To counter this security flaw, we propose differential privacy on top of IPA, which provides formalized privacy guarantees [63]. Using this approach, each peer may add noise to their parameters before sharing with others. Differential privacy is commonly applied to training data, however, it can also be applied to model training [64]. We perform experiments on IPA with differential privacy in Section V. Differential privacy has been applied to the FL pipeline [23]–[25], [65] including in the cross-silo setting [36] where additional considerations need to be made such as securing the privacy of sample-level (rather than client level) data [22].

Homomorphic encryption [66] and garbled circuits [67] are other protection techniques that enable peers to encrypt their models for enhanced protection; such techniques have been applied to FL systems [26], [27], [68]. For example, homomorphic encryption allows clients to encrypt their model parameters before sending their updates to the server [69], effectively protecting their model against a potential malicious server. Homomorphic encryption can be applied in a similar fashion in the IPA algorithm, where peers send encrypted models to each other to hide the true values.

Applications Beyond Federated Learning. IPA may be of interest to other fields such as domain adaptation and transfer learning [70]–[74], model merging [50], [75], model fusion [76]–[78], ensembling [79], and other contexts with variable data distributions. For example, fine-tuning has been found to cause reduced robustness on source domain distribution shift benchmarks [80], [81]. Wortsmann et al. proposed ensembling the pre-trained and fine-tuned models for increased performance on source domain robustness [72]. Similar insights could potentially be gleaned from IPA, where iteratively merging the parameters of segregated domains provides enhanced performance. Domain divergence is also an active area of research in negative transfer learning [82], [83], where source domain knowledge negatively effects a target domain's ability to learn. Exchanging and aligning models trained on divergent domains can enable opposing models to learn from each other, thereby enhancing generalization.

Conclusion We propose a new method for iteratively aligning the parameters of peers models trained on independent data. IPA is favorable in segregated class settings, achieves state-of-the-art performance on homogeneous data partitions, and has competitive convergence under heterogeneous data partitions. We assess our approach across novel and existing benchmarks and show that the method generates unique peer models that converge at a rate correlated to their standalone performance.

Acknowledgement

This work was supported in part by NSF under award numbers ATD 2123761, CNS 2335687, CNS 1822118 and from NIST, Statnett, Newpush, Cyber Risk Research, AMI, and ARL.

References

[1] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[2] 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*. PMLR, 2017, pp. 1273–1282.

[3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[4] Y. Guo, Y. Sun, R. Hu, and Y. Gong, "Hybrid local sgd for federated learning with heterogeneous communications," in *International Conference on Learning Representations*, 2021.

[5] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[6] 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. [Online]. Available: https://openreview.net/forum?id=B7v4QMR6Z9w

[7] L. Gao, H. Fu, L. Li, Y. Chen, M. Xu, and C.-Z. Xu, "Feddc: Federated learning with non-iid data via local drift decoupling and correction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 112–10 121.

[8] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5132–5143.

[9] "FedSyn: Federated learning meets Blockchain." [Online]. Available: https://www.jpmorgan.com/technology/federated-learning-meets-blockchain/

[10] editor2fedai, "WeBank and Swiss Re signed Cooperation MoU." [Online]. Available: https://www.fedai.org/news/webank-and-swiss-re-signed-cooperation-mou/

[11] I. Dayan, H. R. Roth, A. Zhong, A. Harouni, A. Gentili, A. Z. Abidin, A. Liu, A. B. Costa, B. J. Wood, C.-S. Tsai *et al.*, "Federated learning for predicting clinical outcomes in patients with covid-19," *Nature medicine*, vol. 27, no. 10, pp. 1735–1743, 2021.

[12] J. Ogier du Terrail, A. Leopold, C. Joly, C. Béguier, M. Andreux, C. Maussion, B. Schmauch, E. W. Tramel, E. Bendjebbar, M. Zaslavskiy, G. Wainrib, M. Milder, J. Gervasoni, J. Guerin, T. Durand, A. Livartowski, K. Moutet, C. Gautier, I. Djafar, A.-L. Moisson, C. Marini, M. Galtier, F. Balazard, R. Dubois, J. Moreira, A. Simon, D. Drubay, M. Lacroix-Triki, C. Franchet, G. Bataillon, and P.-E. Heudel, "Federated learning for predicting histological response to neoadjuvant chemotherapy in triple-negative breast cancer," *Nature Medicine*, vol. 29, no. 1, pp. 135–146, Jan. 2023, number: 1 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41591-022-02155-w

[13] S. Silva, B. A. Gutman, E. Romero, P. M. Thompson, A. Altmann, and M. Lorenzi, "Federated Learning in Distributed Medical Databases: Meta-Analysis of Large-Scale Subcortical Brain Data," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, Apr. 2019, pp. 270–274, iSSN: 1945-8452.

[14] M. Flores, "NVIDIA Blogs: NVIDIA Blogs: AI models for Mammogram Assessment," Apr. 2020. [Online]. Available: https://blogs.nvidia.com/blog/2020/04/15/federated-learning-mammogram-assessment/

[15] C. Huang, J. Huang, and X. Liu, "Cross-silo federated learning: Challenges and opportunities," *arXiv preprint arXiv:2206.12949*, 2022.

[16] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[17] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning: A meta-learning approach," *arXiv preprint arXiv:2002.07948*, 2020.

[18] M. Zhang, K. Sapra, S. Fidler, S. Yeung, and J. M. Alvarez, "Personalized federated learning with first order model optimization," *arXiv preprint arXiv:2012.08565*, 2020.

[19] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *International Conference on Machine Learning*. PMLR, 2021, pp. 6357–6368.

[20] H.-Y. Chen and W.-L. Chao, "On bridging generic and personalized federated learning for image classification," *arXiv preprint arXiv:2107.00778*, 2021.

[21] H. Chen, C. Wang, and H. Vikalo, "The best of both worlds: Accurate global and personalized models through federated learning with data-free hyper-knowledge distillation," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=29V3AWjVAFi

[22] K. Liu, S. Hu, S. Wu, and V. Smith, "On privacy and personalization in cross-silo federated learning," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=Oq2bdIQQOIZ

[23] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[24] N. Agarwal, P. Kairouz, and Z. Liu, "The skellam mechanism for differentially private federated learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 5052–5064, 2021.

[25] P. Kairouz, Z. Liu, and T. Steinke, "The distributed discrete gaussian mechanism for federated learning with secure aggregation," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5201–5212.

[26] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning," in *Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC 2020)*, 2020.

[27] Z. Jiang, W. Wang, and Y. Liu, "Flashe: Additively symmetric homomorphic encryption for cross-silo federated learning," *arXiv preprint arXiv:2109.00675*, 2021.

[28] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[29] X. Xu, L. Lyu, X. Ma, C. Miao, C. S. Foo, and B. K. H. Low, "Gradient driven rewards to guarantee fairness in collaborative machine learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 16 104–16 117, 2021.

[30] L. Lyu, J. Yu, K. Nandakumar, Y. Li, X. Ma, J. Jin, H. Yu, and K. S. Ng, "Towards fair and privacy-preserving federated deep models," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 11, pp. 2524–2541, 2020.

[31] L. Lyu, X. Xu, Q. Wang, and H. Yu, "Collaborative fairness in federated learning," *Federated Learning: Privacy and Incentive*, pp. 189–204, 2020.

[32] O. Marfoq, C. Xu, G. Neglia, and R. Vidal, "Throughput-optimal topology design for cross-silo federated learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19 478–19 487, 2020.

[33] A. Khaled, K. Mishchenko, and P. Richtárik, "Tighter theory for local sgd on identical and heterogeneous data," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 4519–4529.

[34] B. E. Woodworth, J. Wang, A. Smith, B. McMahan, and N. Srebro, "Graph oracle models, lower bounds, and gaps for parallel stochastic optimization," *Advances in neural information processing systems*, vol. 31, 2018.

[35] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[36] M. A. Heikkilä, A. Koskela, K. Shimizu, S. Kaski, and A. Honkela, "Differentially private cross-silo federated learning," *arXiv preprint arXiv:2007.05553*, 2020.

[37] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 700–10 714, 2019.

[38] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y.-C. Liang, and D. I. Kim, "Incentive design for efficient federated learning in mobile networks: A contract theory approach," in *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*. IEEE, 2019, pp. 1–5.

[39] H. Yu, Z. Liu, Y. Liu, T. Chen, M. Cong, X. Weng, D. Niyato, and Q. Yang, "A sustainable incentive scheme for federated learning," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 58–69, 2020.

[40] K. Donahue and J. Kleinberg, "Optimality and stability in federated learning: A game-theoretic approach," *Advances in Neural Information Processing Systems*, vol. 34, pp. 1287–1298, 2021.

[41] A. Blum, N. Haghtalab, R. L. Phillips, and H. Shao, "One for one, or all for all: Equilibria and optimality of collaboration in federated learning," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1005–1014.

[42] Z. Liu, Y. Chen, H. Yu, Y. Liu, and L. Cui, "Gtg-shapley: Efficient and accurate participant contribution evaluation in federated learning," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, may 2022. [Online]. Available: https://doi.org/10.1145/3501811

[43] A. Shamsian, A. Navon, E. Fetaya, and G. Chechik, "Personalized federated learning using hypernetworks," in *International Conference on Machine Learning*. PMLR, 2021, pp. 9489–9502.

[44] S. Vahidian, M. Morafah, and B. Lin, "Personalized federated learning by structured and unstructured pruning under data heterogeneity," in *2021 IEEE 41st International Conference on Distributed Computing Systems Workshops (ICDCSW)*, 2021, pp. 27–34.

[45] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," *Advances in neural information processing systems*, vol. 30, 2017.

[46] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang, "Personalized cross-silo federated learning on non-iid data," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 9, 2021, pp. 7865–7873.

[47] 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.

[48] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *International Conference on Machine Learning*. PMLR, 2021, pp. 2089–2099.

[49] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 4615–4625.

[50] M. S. Matena and C. A. Raffel, "Merging models with fisher-weighted averaging," *Advances in Neural Information Processing Systems*, vol. 35, pp. 17 703–17 716, 2022.

[51] J. Wang, A. K. Sahu, Z. Yang, G. Joshi, and S. Kar, "Matcha: Speeding up decentralized sgd via matching decomposition sampling," in *2019 Sixth Indian Control Conference (ICC)*. IEEE, 2019, pp. 299–300.

[52] T.-W. Weng, P. Zhao, S. Liu, P.-Y. Chen, X. Lin, and L. Daniel, "Towards certificated model robustness against weight perturbations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 6356–6363.

[53] P. Zhao, S. Wang, C. Gongye, Y. Wang, Y. Fei, and X. Lin, "Fault sneaking attack: A stealthy framework for misleading deep neural networks," in *Proceedings of the 56th Annual Design Automation Conference 2019*, ser. DAC '19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3316781.3317825

[54] Y. Liu, L. Wei, B. Luo, and Q. Xu, "Fault injection attack on deep neural network," in *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2017, pp. 131–138.

[55] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, "Exploring generalization in deep learning," *Advances in neural information processing systems*, vol. 30, 2017.

[56] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," *arXiv preprint arXiv:1609.04836*, 2016.

[57] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6869–6898, 2017.

[58] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning," *arXiv preprint arXiv:1905.10497*, 2019.

[59] T. Song, Y. Tong, and S. Wei, "Profit allocation for federated learning," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 2577–2586.

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

[61] M. Gorbett, H. Shirazi, and I. Ray, "Sparse binary transformers for multivariate time series modeling," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 544–556.

[62] M. Gorbett and D. Whitley, "Randomly initialized subnetworks with iterative weight recycling," *arXiv preprint arXiv:2303.15953*, 2023.

[63] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[64] M. Jagielski, J. Ullman, and A. Oprea, "Auditing differentially private machine learning: How private is private sgd?" *Advances in Neural Information Processing Systems*, vol. 33, pp. 22 205–22 216, 2020.

[65] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," *arXiv preprint arXiv:1710.06963*, 2017.

[66] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology—EUROCRYPT'99: International Conference on the Theory and Application of Cryptographic Techniques Prague, Czech Republic, May 2–6, 1999 Proceedings 18*. Springer, 1999, pp. 223–238.

[67] I. Lazrig, T. Ong, I. Ray, I. Ray, X. Jiang, and J. Vaidya, "Privacy preserving probabilistic record linkage without trusted third party," in *2018 16th Annual Conference on Privacy, Security and Trust, PST 2018*, ser. 2018 16th Annual Conference on Privacy, Security and Trust, PST 2018, R. Deng, S. Marsh, J. Nurse, R. Lu, S. Sezer, P. Miller, L. Chen, K. McLaughlin, and A. Ghorbani, Eds. United States: Institute of Electrical and Electronics Engineers Inc., Oct. 2018, funding Information: This work was supported by grants from UC Anschutz Medical Center, NSF under award no. CNS 1650573, AFRL, CableLabs, Furuno Electric Company, and SecureNok. Publisher Copyright: © 2018 IEEE.; 16th Annual Conference on Privacy, Security and Trust, PST 2018 ; Conference date: 28-08-2018 Through 30-08-2018.

[68] G. Xu, H. Li, Y. Zhang, S. Xu, J. Ning, and R. H. Deng, "Privacy-preserving federated deep learning with irregular users," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 1364–1381, 2022.

[69] "Federated learning with homomorphic encryption." [Online]. Available: https://developer.nvidia.com/blog/federated-learning-with-homomorphic-encryption/

[70] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[71] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Advances in neural information processing systems*, vol. 27, 2014.

[72] M. Wortsman, G. Ilharco, J. W. Kim, M. Li, S. Kornblith, R. Roelofs, R. G. Lopes, H. Hajishirzi, A. Farhadi, H. Namkoong *et al.*, "Robust fine-tuning of zero-shot models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7959–7971.

[73] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[74] Y. Pruksachatkun, J. Phang, H. Liu, P. M. Htut, X. Zhang, R. Y. Pang, C. Vania, K. Kann, and S. R. Bowman, "Intermediate-task transfer learning with pretrained models for natural language understanding: When and why does it work?" 2020.

[75] S. K. Ainsworth, J. Hayase, and S. Srinivasa, "Git re-basin: Merging models modulo permutation symmetries," *arXiv preprint arXiv:2209.04836*, 2022.

[76] N. Hoang, T. Lam, B. K. H. Low, and P. Jaillet, "Learning task-agnostic embedding of multiple black-box experts for multi-task model fusion," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 4282–4292. [Online]. Available: https://proceedings.mlr.press/v119/hoang20b.html

[77] T. C. Lam, N. Hoang, B. K. H. Low, and P. Jaillet, "Model fusion for personalized learning," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 5948–5958. [Online]. Available: https://proceedings.mlr.press/v139/lam21a.html

[78] S. P. Singh and M. Jaggi, "Model fusion via optimal transport," *Advances in Neural Information Processing Systems*, vol. 33, pp. 22 045–22 055, 2020.

[79] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith *et al.*, "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," in *International Conference on Machine Learning*. PMLR, 2022, pp. 23 965–23 998.

[80] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[81] H. Pham, Z. Dai, G. Ghiasi, H. Liu, A. W. Yu, M.-T. Luong, M. Tan, and Q. V. Le, "Combined scaling for zero-shot transfer learning," *arXiv preprint arXiv:2111.10050*, 2021.

[82] Z. Wang, Z. Dai, B. Póczos, and J. Carbonell, "Characterizing and avoiding negative transfer," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11 293–11 302.

[83] W. Zhang, L. Deng, L. Zhang, and D. Wu, "A survey on negative transfer," *IEEE/CAA Journal of Automatica Sinica*, 2022.