L-DAWA: Layer-wise Divergence Aware Weight Aggregation in Federated Self-Supervised Visual Representation Learning

Yasar Abbas Ur Rehman[1,*], Yan Gao[2,*], Pedro Porto Buarque de Gusmão[2], Mina Alibeigi[2,3], Jiajun Shen[1], Nicholas D. Lane[2,4]
[1]TCL AI Lab, Hong Kong, [2]University of Cambridge, United Kingdom, [3]Zenseact, Sweden, [4]Flower Labs

Abstract

The ubiquity of camera-enabled devices has led to large amounts of unlabeled image data being produced at the edge. The integration of self-supervised learning (SSL) and federated learning (FL) into one coherent system can potentially offer data privacy guarantees while also advancing the quality and robustness of the learned visual representations without needing to move data around. However, client bias and divergence during FL aggregation caused by data heterogeneity limits the performance of learned visual representations on downstream tasks. In this paper, we propose a new aggregation strategy termed Layer-wise Divergence Aware Weight Aggregation (L-DAWA) to mitigate the influence of client bias and divergence during FL aggregation. The proposed method aggregates weights at the layer-level according to the measure of angular divergence between the clients' model and the global model. Extensive experiments with cross-silo and cross-device settings on CIFAR-10/100 and Tiny ImageNet datasets demonstrate that our methods are effective and obtain new SOTA performance on both contrastive and non-contrastive SSL approaches.

1. Introduction

Federated learning (FL) has been a center of interest for the research and industrial communities due to its unique property of collaboratively learning feature representations from large-scale datasets without compromising the users' data privacy [28, 49, 20, 16]. It has been successful in joint visual representations learning from image data while preserving data privacy [28, 35, 22]. However, current practices in FL are commonly limited to supervised learning tasks that require high-quality and domain-specific labels to be available alongside the data. This requirement limits the deployment of FL in many real-world applications where access to high-quality labels at the edge is restricted [15].

Self-supervised learning (SSL) has been combined with FL, enabling it to expand its potential of learning feature representations from the vast amount of unlabeled, private, uncurated, and visual data being produced at the edge [50, 26, 21, 51]. In contrast to supervised FL [7, 10, 28], federated self-supervised learning (F-SSL) does not require high-quality labeled data, although it may require another stage of centralized fine-tuning or personalizing the model for downstream tasks with limited labeled data. F-SSL enables the re-purposing of heterogeneous, unlabeled, and uncurated real-world image data for various downstream tasks. (*viz.*, image recognition [18], object detection[14], semantic segmentation[33], facial recognition, authentication [15, 38], etc.) by collaboratively learning *intermediate* visual representations in a privacy-preserving fashion.

One of the unique challenges of F-SSL is learning visual representations from non-independently and identically distributed (*Non-iid*) data [50, 26, 36]. Recent studies in F-SSL, both image-based [50] and video-based [36], directly extend the state-of-the-art (SOTA) centralized SSL pre-training techniques (e.g., SimCLR [5], SimSiam [6], BYOL [12], Barlow Twins [45] Speed [1], VCOP [43], and CtP [40]) under the setting of FL. Generally, these F-SSL methods aggregate the participating clients' model during each FL round using FedAvg [28]. While FedAvg provides certain convergence guarantees during the FL stage, the downstream task performance is sub-optimal [3].

One of the main reasons for sub-optimal performance is the uncontrolled divergence between participating clients' caused by data heterogeneity. [49, 23, 50, 51]. The client's model divergence in FL, if not appropriately controlled, affects the aggregated global model, even if the data and the model quality of individual clients are good, which often happens when the clients' models lie in different basins of the loss landscape [42, 24]. FedAvg, used by most existing F-SSL work, aggregates the local models based on the number of data samples on each client. This leads the model to be biased towards the clients possessing more data [23], and the optimization trajectory of the global model to be dominated by those clients, thus deviating the model into local minima, further leading to sub-optimal downstream perfor-

[*] Equal contribution, authors ordered alphabetically.

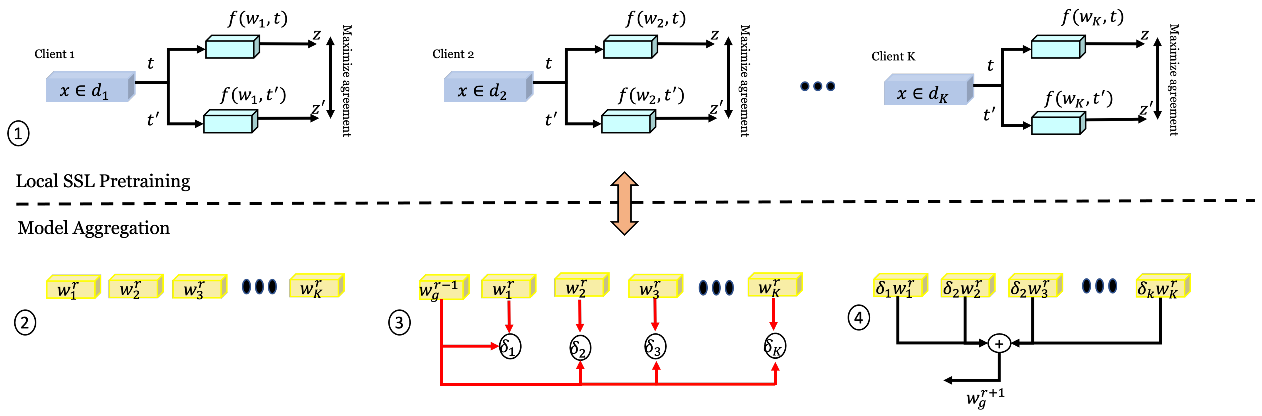


Figure 1: Pipeline of DAWA for Federated SSL. ① Local SSL pre-training. ② Received clients' models on the server. ③ Computing layer-wise divergence for each client model. ④ Aggregating the clients' models weighted by the corresponding layer-wise divergence and generating a new global model. Note that the client models are discarded after generating the global model.

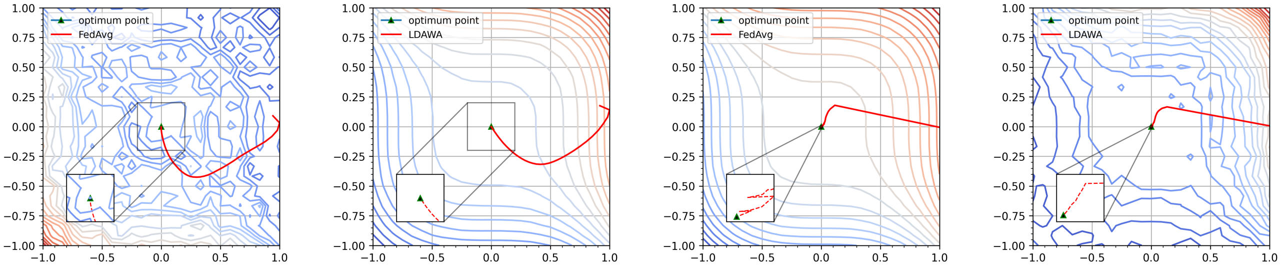
mance. While bias is common in both *cross-silo* and *cross-device* settings of FL, it can have a more significant influence in the former, especially when all clients participate in each FL round to generate the global model and clients with more data dominate the training. In the *cross-device* settings, a fraction of clients participate in every round of FL, mitigating the possibility of certain groups of clients dominating the FL training; however, client drift is higher compared to *cross-silo*, resulting in a worse global model when quality clients are not selected.

To alleviate this issue, several aggregation strategies based on model divergence among clients have been proposed. FedU [50] determines the update of the local client models' predictor based on the divergence between the backbone models of the clients and the server. Despite its effectiveness, FedU still uses FedAvg [28] to aggregate client models at the server. This overlooks the problem of clients' bias caused by FedAvg during model aggregation. Another work termed Loss [10] uses clients' local training loss as an indicator of the clients' model quality to aggregate models. However, this method is still biased toward the clients with lower training loss (similar problem as FedAvg). Additionally, the existing aggregation strategies assign a single scalar to one client model. Nevertheless, the different layers of the clients' model exhibit different levels of heterogeneity [27, 47]. Assigning a single scalar value to all layers of a client's model would exacerbate the bias.

In this paper, we propose a novel aggregation scheme termed Layer-wise Divergence Aware Weight Aggregation (L-DAWA) to mitigate the dominant effect of certain clients in F-SSL. L-DAWA is a unique method that incorporates *angular divergence* at the layer level into the aggregation process. The *angular divergence* is used as a weighting coefficient to scale the contribution of each layer from the client's models in the generation of the global model at the server. L-DAWA does not require the transmission of any clients' metadata, such as the number of samples and local training loss, to the server. Instead, it relies solely on the previous global model and the local clients' model updates. By integrating angular divergence during the aggregation process, similar to the role of momentum, it is possible to achieve smooth control of the trajectory of model optimization and accelerate convergence in the relevant direction. This would lead the model toward the global optima, resulting in better downstream performance (Figure 2, Table 2).

We further adapt L-DAWA into the existing aggregation methods (FedAvg, Loss, and FedU) to correct the optimization trajectory dominated by certain clients. We termed these modified techniques as L-DAWA$_{FedAvg}$, L-DAWA$_{FedU}$, and L-DAWA$_{Loss}$. Experimental results show that the proposed method and its variants significantly improve the performance both in *cross-silo* and *cross-device* settings. The main contributions of this work are as follows:

- We integrate for the first time the angular divergence into model aggregation, which maintains a coherent convergence trajectory during FL training, leading to better models, see Table 2.
- We propose a novel aggregation method for F-SSL, dubbed L-DAWA, which utilizes angular divergence as a weighting coefficient to scale the contribution of each layer from the client's models during FL aggregation. We further adapt L-DAWA into the existing aggregation methods (FedAvg, Loss, and FedU).
- We perform extensive experiments and compare the performance of L-DAWA-related methods for both contrastive and non-contrastive SSL approaches. We achieve the new SOTA in F-SSL obtaining at most 5.21%, 6.19%, and 6.25% improvement on CIFAR-10/100 and Tiny ImageNet, respectively under linear evaluation protocol in *cross-silo* and comparable performance in *cross-device* setting.



(a) FedAvg SimCLR (b) L-DAWA SimCLR (c) FedAvg Barlow Twins (d) L-DAWA Barlow Twins

Figure 2: Illustration of global model optimization trajectories for FedAvg [28] and L-DAWA under the loss-landscapes of FL SimCLR (a-b) and FL Barlow Twins (c-d) with *cross-silo* settings. L-DAWA assists the global model in converging smoothly into a substantially wider loss landscape compared to FedAvg (especially for FL SimCLR with a chaotic loss landscape in (a)). The optimization trajectories are effectively controlled by L-DAWA. For instance, (b) achieves a shorter path toward the optimum point than (a); (d) obtains a more smooth converge route near the optimum point compared to (c), which suffers clear oscillations.

2. Literature review

2.1. Self-supervised learning

Self-supervised learning has been explored predominantly in every computer vision domain [11]. Owing to learning representations based on exploiting the properties in the data using some generative or discriminative models that solve pseudo or pretext tasks. Examples of generative self-supervised pretext tasks include colorization [48], inpainting [34], or super-resolution [30]. On the other hand, discriminative self-supervised pretext tasks include, but are not limited to, predicting rotation [17], feature alignment [5, 13, 12, 4, 45, 44], solving jigsaw puzzle (a.k.a. predicting shuffled patch permutation) [31]. Among these discriminative SSL approaches, feature alignment approaches such as SimCLR [5], MOCO [13], BYOL [12], SWAV [4], and Barlow Twins [45] have been in the spotlight recently. Based on their loss function, these SSL approaches can be categorized into contrastive (requires negative samples), and non-contrastive (does not require negative samples) approaches. They can be combined into a single category of feature alignment because they enable the model to learn features that are invariant to the artificial transformation generated via the data augmentation process.

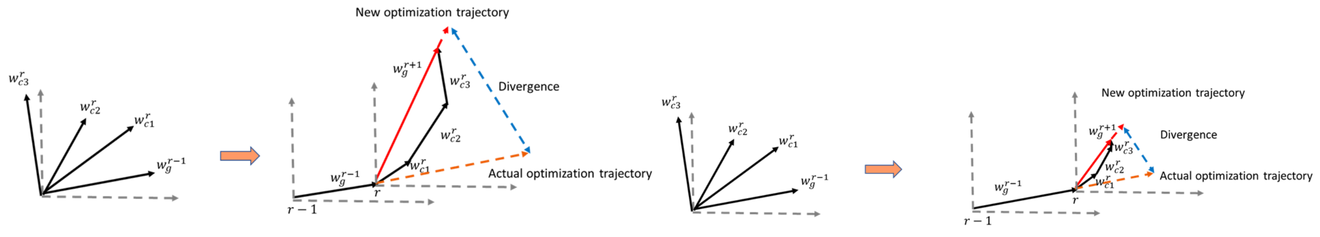
2.2. Data heterogeneity in FL

Data heterogeneity is inevitable in realistic FL settings and has been studied extensively in literature [28, 49, 26, 21, 41, 46]. McMahan et al. [28] proposed a client models aggregation strategy, FedAvg, that can work on certain Non-iid distributions. FedAvg has been considered as a baseline for many FL aggregation strategies that improve the performance of the model in Non-iid data settings [10, 35]. Data heterogeneity in FL causes weight divergence, i.e., during the aggregation, the weights of the model from different clients are not aligned causing certain weights to cancel each other [29]. Methods have been proposed to tackle the weight divergence in F-SSL, such as by introducing global dataset [49], local training loss [10], or locally updating the partial model based on the measure of divergence [50, 51].

2.3. Divergence in F-SSL

Studies have been conducted to understand the clients' model divergence in F-SSL [50, 51]. These frameworks allow to partially update the local SSL model (e.g., only updating the predictor) based on the Euclidean distance between the current model and previous global model. Although these methods are effective on the downstream task, they are limited to the only non-contrastive SSL technique, e.g., BYOL[12]. Moreover, these methods naively combine the clients' models at the server (using FedAvg) and push the computation of the model divergence on the clients' side, requiring the clients to perform extra computation.

Using angular distance for measuring the divergence or similarity between the weight vectors in supervised [32] and unsupervised FL is not a new concept. Several works use angular distance for client clustering. For example, [37] computes the cosine distance to cluster different clients for multi-task learning after fetching the learned global model. In the other works, the cosine-based similarity is used to perform cluster-based supervised FL of different clients [8], or to understand the divergence between the representations of different clients' predictor networks [7]. The work in [27] uses hypernetworks to find the similarity between the weights of the clients' models for personalized FL. However, the personalized models for each client are maintained on the server, which differs from a general FL framework, where no client's models are reserved on the server. We further note that, to the best of our knowledge, there is no existing work considering angular divergence for model aggregation in the general FL framework, which is considered in our work.



(a) FedAvg

(b) DAWA

Figure 3: Illustration of optimization trajectories for FedAvg and DAWA. (a) Without any divergence control, FedAvg can deviate from the actual optimization trajectory by a large margin (b) DAWA effectively controls the trajectory by scaling the clients based on the measure of their *angular divergence* compared to the previous global model, effectively restricting the *angular divergence* range of the clients.

3. Methodology

3.1. Federated self-supervised learning

We consider M partitions $\{d_m\}_{m=1}^{M}$ of dataset D to compose M decentralized clients with $\{n_m\}_{m=1}^{M}$ samples on each local data set. At each communication round r of FL, the server randomly selects K clients participating in the training and initializes the local models with the global model weights w_g^r. Then, each decentralized client k learns the intermediate feature representations $w_k^r = \mathcal{F}_{SSL}(d_k, w_g^r, E)$ by training with a specific SSL approach on its own local dataset d_k for E local epochs before transmitting the local model w_k^r to the server. The server then receives the local models $\{w_k^r\}_{k=1}^{K}$ and aggregate them based on a weighting factor $\beta(\cdot)$ to generate a new global model w_g^{r+1} as follows:

$$w_g^{r+1} = \sum_{k=1}^{K} \beta_k w_k^r. \tag{1}$$

This process is repeated until model convergence. A common aggregation strategy is FedAvg [28], which combines the local models based on the number of samples over selected clients, i.e., $\beta_k = \frac{n_k}{\sum_{k=1}^{K} n_k}$. In realistic FL settings with heterogeneous client data distribution, some clients may contain skewed data, not representing the global data distribution. This scenario could lead to model deviation in the aggregation step, which can not be solved by vanilla FedAvg. Several existing works [50, 19] attempt to alleviate this issue by using Euclidean distance (between the client's model and the global model) as an indicator to partially or entirely update the layers of the client models. Nevertheless, the aggregation step on the server for these methods is still based on FedAvg. Other works [10, 36] use the averaged training loss as a weighting coefficient for aggregation, i.e., $\beta_k = \frac{exp(-\mathcal{L}_k)}{\sum_{k=1}^{K} exp(-\mathcal{L}_k)}$, thus reflecting the quality of the locally trained models.

However, the above approaches are generally biased toward the clients with a larger number of data samples or lower training loss. In this case, the trajectory of the global model optimization is determined by certain clients [23], thus deviating into some local minima [49, 25] resulting in sub-optimal performance (Figure 2). On the model level, the dominant effect of certain clients during FL training leads to oscillations in the divergence of the clients' models with respect to the global model (Figure 4). This could deviate the model optimization trajectory, which results in a decrease in performance on the downstream tasks (see Table 2).

3.2. Divergence aware weight aggregation

A smooth control of the trajectory of model optimization would assist model convergence and leading to higher downstream performance. Here, we propose a divergence aware weight aggregation (DAWA) method by introducing the *angular divergence* δ_k between the clients' model and the previous global model into aggregation process (Figure 1). δ_k can be calculated as follows:

$$\delta_k = cos\theta_{g,k} = \frac{w_g^r . w_k^r}{||w_g^r||.||w_k^r||}. \tag{2}$$

The angular divergence determines whether the model weights are aligned or orthogonal to each other. The range of δ_k is naturally restricted to [-1, 1]. To integrate angular divergence into aggregation, we set $\beta_k = \delta_k / K$, then Eq. 1 can be re-written as:

$$w_g^{r+1(M-DAWA)} = \frac{1}{K} \sum_{k=1}^{K} \delta_k w_k^r. \tag{3}$$

The integration of angular divergence, as a similar role of momentum, would accelerate convergence in the relevant direction while dampening oscillations during FL optimization (Figure 2 & 3 & 4). For instance, some client models w_k^r may diverge by a large angle (e.g., $[180°, 90°)$) from the global model w_g^r. In this case, δ_k in Eq. 2 falls into the range of $[-1, 0)$. The subsequent multiplication of δ_k with w_k^r in Eq. 3 could correct the alignment with the direction

Algorithm 1 *L-DAWA*: Let us consider the server randomly selecting K clients at the given round. The clients train the SSL model with L layers for E local epochs on its dataset d_k with n_k number of samples. The FL optimization lasts R rounds.

```
Input: $K, R, n_k, d_k, E, L$
Output: $w_g^R$
Central server does:
  for $r = 1,..., R$ do
      Server randomly selects $K$ clients.
      for $k = 1,..., K$ do
          $w_k^r, n_k, \mathcal{L}_k$ = TrainLocally$(k, w_g^r, E)$
      Aggregation:
      for $l = 1,..., L$ do
          Compute divergence $\delta_k^{(l)}$ based on Eq. 2.
      If M-DAWA then: Compute $w_g^{r+1}$ based on Eq. 3.
      If L-DAWA then: Compute $w_g^{r+1}$ based on Eq. 4.
      If L-DAWA$_{FedAvg}$ or L-DAWA$_{FedU}$ then:
         Compute $w_g^{r+1}$ based on Eq. 5.
      If L-DAWA$_{Loss}$ then:
         Compute $w_g^{r+1}$ based on Eq. 6.
TrainLocally $(k, w_g^r)$:
  $w_k^r, \mathcal{L}_k = \mathcal{F}_{SSL}(d_k, w_g^r, E)$
  Upload $w_k^r, n_k, \mathcal{L}_k$ to the server.
```

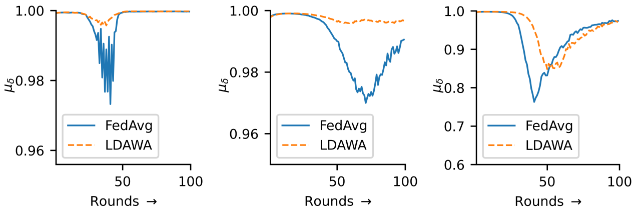
of w_g^r by effectively reducing the theoretical angular range of divergence from $[180°, 0°]$ to $[90°, 0°]$.

However, the divergence at layer-level may dramatically vary, which cannot be represented by a single value δ_k of the whole model. Additionally, directly computing δ_k on the entire model is prohibitively expensive in practice imposing large memory footprints and increase the computation time (see Table 1). As a result, we further calculate angular divergence $\delta_k^{(l)}$ based on Eq. 2 for each layer. Then, the layer-wise divergence aware weight aggregation (L-DAWA) method can be represented as follows:

$$w_g^{r+1(L-DAWA)} = \frac{1}{K} \sum_{l=1}^{L} \sum_{k=1}^{K} \delta_k^{(l)} w_k^{r(l)}. \tag{4}$$

3.3. Layer-wise divergence adaptation with SOTA aggregation strategies

The layer-wise angular divergence can be easily integrated to the existing aggregation methods in order to correct the trajectory of model optimization. Here, we select three SOTA aggregation strategies, namely FedAvg [28], Loss [10], and FedU [50]. By introducing divergence, we term the new variations as L-DAWA$_{FedAvg}$ (Eq. 5), L-DAWA$_{Loss}$ (Eq. 6) and L-DAWA$_{FedU}$ (Eq. 5). Note that FedU conducts aggregation with FedAvg, and partially update the model based on Euclidean distance. The overall algorithm is summarised in Algo. 1.



(a) E=1 (b) E=5 (c) E=10

Figure 4: The mean angular divergence between the clients' models with respect to the previous global model over $R = 100$ rounds, computed by the following equation: $\mu_\delta = \frac{1}{K}\sum_{k=1}^{K} \delta_k$, where $K = 10$. The higher δ value means lower divergence. L-DAWA has a good control of oscillations, maintaining the angular divergence in a lower level over FL rounds than FedAvg.

$$w_g^{r+1(FedAvg)} = \sum_{l=1}^{L} \sum_{k=1}^{K} \frac{n_k}{\sum_{k=1}^{K} n_k} \delta_k^{(l)} w_k^{r(l)}. \tag{5}$$

$$w_g^{r+1(Loss)} = \sum_{l=1}^{L} \sum_{k=1}^{K} \frac{exp(-\mathcal{L}_k)}{\sum_{k=1}^{K} exp(-\mathcal{L}_k)} \delta_k^{(l)} w_k^{r(l)}. \tag{6}$$

4. Experimental setup

4.1. Federated datasets

For both F-SSL pre-training and downstream tasks evaluation, we conduct experiments on CIFAR-10/100 and Tiny ImageNet datasets [18]. To simulate a realistic FL environment, we generate Non-iid versions of datasets based on actual class labels using a Dirichlet coefficient α, where a lower value indicates greater heterogeneity. As a result, the datasets are randomly partitioned into 10/100 shards for *cross-silo*/*cross-device* setting to mimic the setup of having 10/100 disjoint clients participating in FL. Additionally, we scale the value of α to generate various Non-iid levels w.r.t. class labels and the number of samples.

4.2. Federated SSL pre-training

To perform a side-by-side comparison of the contrastive and non-contrastive SSL methods, we select SimCLR [5] and Barlow Twins [45]. Both SSL approaches share the same network architecture (ResNet-18) [14, 39] as a backbone while trained with different loss functions. To evaluate these methods on common ground, we maintain the same hyperparameters for both these methods throughout the FL pre-training, except the hyperparameters τ in SimCLR and λ in Barlow Twins. We adopted the setup from Tamkin et al. [39], using a batch size of 256, a learning rate of 0.03, a weight decay of 1e-4, and an SGD momentum of 0.9. Unless otherwise stated, we set the number of local epochs $E = 10$. For *cross-silo* FL, all clients participate

Method	Execution Time (Sec)	E=1	E=5	E=10
FedAvg (Baseline)	0.29	52.90	63.76	67.64
M-DAWA	22.65	53.87	**65.80**	68.90
L-DAWA	0.38	**53.94**	65.44	**69.34**

Table 1: Ablation study: Linear-probe accuracy on downstream task and average aggregation execution time for FedAvg, M-DAWA and L-DAWA. Each method is pre-trained with SimCLR on the Non-iid version (α=0.1) of CIFAR-10 for R=10 rounds under the *cross-silo (K=10)* settings.

in the training each round, while 10 clients participate per round in *cross-device* setting. The FL pre-training lasts for R=200 rounds for *cross-silo* and R=100 rounds for *cross-device* settings. All training schemes are implemented with PyTorch-Lightning [9], and Flower [2].

4.3. Evaluation protocol

A standard linear-probe protocol [5, 26] is utilized to evaluate the pre-trained SSL models. In this protocol, the pre-trained SSL model is frozen, and a linear classifier is learned on top of it. In addition to fine-tuning on the whole training set, we also conduct semi-supervised learning with limited labeled data (1% and 10%). Following the setup in [39], we set the batch size to 128, momentum to 0.9, and the learning rate to 0.01, which is decayed by a factor of 0.1 after 60 and 80 epochs. We perform the training for 100 epochs and report the test results for the last epoch.

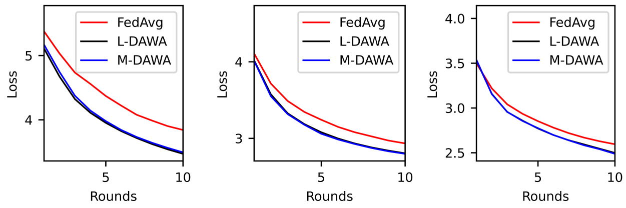
5. Experimental results and discussion

5.1. Ablation studies

We first perform experiments by comparing L-DAWA and M-DAWA against the baseline FedAvg under the *cross-silo* settings with Non-iid data using the SimCLR method and training them for 10 communication rounds. L-DAWA and M-DAWA consistently achieve higher performance (Table 1) and converge faster (Figure 5) compared to FedAvg. Particularly, L-DAWA obtains the best results on E=1 and E=10 settings. On the other hand, we find that M-DAWA takes 22.65 seconds to complete the aggregation process, which is around 60× and 78× slower than L-DAWA and FedAvg, respectively. This computational delay mainly results from the divergence calculation of the entire model. Therefore, we stick to L-DAWA for all of the rest experiments due to its stable performance and lower resource consumption compared to M-DAWA.

5.2. Communication efficiency

In this experiment, we analyze the computational efficiency and model convergence behavior of pre-training SimCLR and Barlow Twins in FL with FedAvg and L-DAWA. Figure 6 shows that L-DAWA offers a faster convergence and requires less number of communication rounds to achieve a given target accuracy for both SimCLR and Barlow Twins compared to FedAvg. Additionally, L-DAWA offers an improved and stable performance for both SSL methods. In contrast, FedAvg requires more rounds to reach a given target accuracy for SimCLR and Barlow Twins. This is evident from the corresponding optimization trajectories for SimCLR and Barlow Twins as shown in Figure 2. One can see from Figure 2 that the optimization trajectory of FL pre-training of SimCLR with FedAvg lags behind SimCLR with L-DAWA. Whereas the optimization trajectory of FL pre-training of Barlow Twins with FedAvg oscillates more near the optimization point compared to the one with L-DAWA, which also provides a possible reason for the deterioration of the performance near the end of FL pre-training (Figure 6).



(a) E=1 (b) E=5 (c) E=10

Figure 5: Average local loss of SimCLR for FedAvg, M-DAWA, and L-DAWA: Each method is pre-trained with SimCLR on the Non-iid version of CIFAR-10 for R=10 rounds under the cross-silo (K=10) setting.

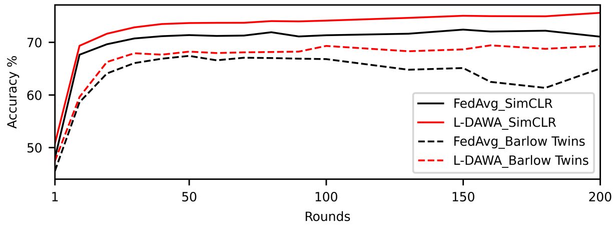


Figure 6: Variation of linear-probe accuracy (%) with communication rounds for SimCLR and Barlow Twins. L-DAWA shows stable and improved convergence compared to FedAvg.

5.3. Fairness analysis

In this section, we provide a fairness analysis of our proposed method regarding the number of local data samples on the clients. To offer a formal illustration, we conduct two Non-iid federated datasets with 10 clients using CIFAR-10 at a similar level of difficulty for SSL: (1) each client contains the same number of samples with only one single class (W/Sc) without overlap between the classes; (2) the Dirichlet coefficient α is set to 0.1 with an uneven distribution of the samples among the clients. Table 3 shows the linear-probe performance comparison of L-DAWA with FedAvg on these two settings. One can see that FedAvg

Method	CIFAR-10						CIFAR-100					
	SimCLR			Barlow Twins			SimCLR			Barlow Twins		
	100%	1%	10%	100%	1%	10%	100%	1%	10%	100%	1%	10%
FedAvg [28]	71.07	53.32	66.13	65.02	43.29	57.06	43.85	21.35	36.49	35.70	12.93	27.04
Loss [10]	71.34	60.14	69.91	57.12	42.70	54.10	44.69	20.90	37.14	34.76	11.08	23.88
FedU [50]	70.36	61.74	69.77	64.55	50.95	61.96	44.31	20.86	36.76	35.25	12.57	26.72
EUC [19]	70.51	60.77	69.37	63.60	49.62	62.61	44.89	21.96	36.90	35.44	12.43	27.07
L-DAWA	75.60	62.19	71.40	69.31	57.62	**68.38**	49.88	21.53	39.41	41.85	16.92	32.12
L-DAWA$_{FedAvg}$	75.72	63.74	**73.29**	69.14	**58.98**	67.61	49.99	**21.87**	39.57	41.49	**17.35**	**32.20**
L-DAWA$_{Loss}$	**76.55**	63.86	72.82	69.46	58.74	66.82	50.29	21.43	39.72	**41.89**	16.36	31.88
L-DAWA$_{FedU}$	76.23	**64.06**	72.89	**69.50**	58.67	67.42	**50.59**	21.72	**39.76**	41.72	17.28	31.91

Table 2: Comparison of L-DAWA with SOTA methods on the Non-iid version (α=0.1) of CIFAR-10 and CIFAR-100 under *cross-silo (K=10)* settings. The pre-trained SSL models are separately fine-tuned on 100%, 10%, and 1% data samples of the training set.

SSL Type	Aggregation Type	Non-iid (W/Sc)			Non-iid (α=0.1)		
		E1	E5	E10	E1	E5	E10
SimCLR	FedAvg	49.06	60.06	66.13	50.93	65.42	71.36
	L-DAWA	**60.45**	**69.08**	**73.49**	**60.29**	**70.65**	**75.60**
Barlow Twins	FedAvg	49.47	57.14	60.51	51.65	58.84	65.02
	L-DAWA	**50.54**	**63.46**	**65.70**	**54.84**	**65.07**	**69.31**

Table 3: Fairness Evaluation of FedAvg and L-DAWA for SimCLR and Barlow Twins. W/Sc means single class per client. Each method is pre-trained on the Non-iid version of CIFAR-10 for R=200 rounds under the *cross-silo (K=10)* settings.

performs worse on the downstream task under W/Sc setting compared to the case when the clients contain an unequal number of samples Dir(α = 0.1). This is mainly because FedAvg is biased toward the clients with more samples than others (e.g., α=0.1 setting). The model optimized with FedAvg would deviate to the dominant clients while the contribution from other clients would be diminished [23]. This causes the model converges in a local optimum with lower performance on downstream tasks.

Compared to FedAvg, L-DAWA offers higher and more fair performance in these two Non-iid settings, especially for SimCLR. One can observe that L-DAWA with E=5 produces nearly the same results with both configurations. This demonstrates that our proposed method is agnostic to the number of data samples on each client. L-DAWA provides a more balanced optimization over the clients by fairly aggregating the local model weights. Indeed, this would help the model move towards global optima during FL training.

5.4. L-DAWA in linear fine-tuning

In this section, we compare the performance of our proposed methods against the state-of-the-art aggregation strategies, *viz.*, FedAvg[28], Loss [10], FedU [50] and EUC [19] with supervised and semi-supervised fine-tuning schemes under *cross-silo* and *cross-device* settings.

5.4.1 Cross-silo performance

In Table 2, we compare the linear-probe accuracy of our proposed L-DAWA-related methods against state-of-the-art approaches under *cross-silo (K=10)* setting, fine-tuned with supervised (100% training set) and semi-supervised (partial training set) schemes on the datasets of CIFAR-10/100.

First, our methods achieve new SOTA results in all settings. Notably, it gains 5.21% and 6.19% improvement with 100% data fine-tuning on CIFAR-10/100 datasets, respectively. As for the more difficult semi-supervised evaluation, the SOTA accuracy is increased by 2.32% (1% data setting), 3.38% (10% data setting) and 0.52% (1% data setting), 2.62% (10% data setting) on CIFAR-10/100, respectively.

Second, L-DAWA-related approaches obtain similar improvements for both SimCLR and Barlow Twins. This indicates that our methods are agnostic to contrastive or non-contrastive SSL models.

Third, L-DAWA achieves higher performance on all of the settings compared to the baselines (FedAvg, Loss, FedU, and EUC). This highlights the importance of integrating divergence into aggregation during FL training. The angular divergence plays a similar role of momentum, accelerating convergence toward the global optima in the relevant direction while diminishing oscillations during FL optimization. In contrast, FedAvg and Loss-based aggregation would be dramatically influenced by the dominant clients with larger amounts of data or lower loss, leading to a deviation of the optimization trajectory.

Interestingly, the modified versions of FedAvg, Loss, and FedU (L-DAWA$_{FedAvg}$, L-DAWA$_{Loss}$ and L-DAWA$_{FedU}$) provide better and unbiased results once the layer-wise divergence is introduced into the aggregation. This is mainly because the integration of angular divergence offers a corrective effect with respect to the original optimization trajectory (Figure 2). Indeed, the combination of L-DAWA with the existing SOTA methods boosts the performance at a large scale and provides various candi-

		α values			
Non-iid levels		0.1	0.2	0.4	0.6
SimCLR	FedAvg	50.92	53.26	53.98	53.94
	L-DAWA	**60.29**	**63.41**	**62.98**	**63.40**
Barlow Twins	FedAvg	53.45	54.72	54.18	54.82
	L-DAWA	**54.84**	**57.90**	**54.61**	**57.89**

Table 4: Performance of FedAvg and L-DAWA for SimCLR and Barlow Twins within different Non-iid levels (α values). Each method is pre-trained on CIFAR-10 with 1 local epoch for R=200 rounds under the *cross-silo (K=10)* settings.

Aggregation Type	E1	E5	E10
FedAvg	16.87	27.70	32.92
Loss	15.25	28.47	33.37
FedU	16.41	27.46	32.63
L-DAWA	**23.12**	**31.97**	**37.72**

Table 5: Evaluation with SimCLR on the Non-iid version of Tiny ImageNet under the *cross-silo (K=10)* setting.

dates for different settings. For instance, L-DAWA$_{Loss}$ and L-DAWA$_{FedU}$ achieve the highest accuracy in 100% data setting, while L-DAWA$_{FedAvg}$ performs best in the most challenging setup (1% semi-supervised evaluation).

We further evaluate L-DAWA on a larger dataset Tiny ImageNet, compared with SOTA approaches (Table 5). A large margin of improvements (6.25%, 3.50%, 4.35%) is obtained by L-DAWA for the settings of 1/5/10 local epochs, respectively. This demonstrates that our proposed method has a good generalization on a large dataset.

We also show the results of our proposed method on different Non-iid levels of *cross-silo* by scaling Dirichlet coefficient α from 0.1 to 0.6 (Table 4). One can see that the performance of all methods shows a slight increase trend with Non-iid levels decreasing. Noticeably, L-DAWA obtains higher performance at all levels of Non-iid settings, especially for SimCLR ($\alpha = 0.1$) with 9.37% improvement. Interestingly, the results on the $\alpha = 0.2$ setting are better than others. This indicates that the Non-iid level for image-SSL may be partially determined by actual class labels.

5.4.2 Cross-device performance

Compared to *cross-silo*, *cross-device* setting is more challenging due to its nature of heterogeneous data distribution. One can observe from Table 6 that our proposed methods still perform better than all baselines with a slight improvement of 1.26%/0.25% (SimCLR) and 0.25%/0.85% (Barlow Twins) on CIFAR-10/100, respectively. Also, the combination of L-DAWA with the existing methods (L-DAWA$_{FedAvg}$ and L-DAWA$_{Loss}$) provide the best performance at most of the settings, suggesting that in *cross-device* settings the integration of angular divergence is necessary to boost performance.

Method	CIFAR-10		CIFAR-100	
	SimCLR	Barlow Twins	SimCLR	Barlow Twins
FedAvg	68.66	62.07	44.59	32.65
Loss	66.09	56.40	44.83	33.27
FedU	68.52	61.43	44.56	32.89
L-DAWA	68.20	58.25	45.04	**34.12**
L-DAWA$_{FedAvg}$	**69.92**	**62.32**	44.19	33.20
L-DAWA$_{Loss}$	68.79	61.36	**45.08**	31.93
L-DAWA$_{FedU}$	69.69	62.19	44.97	32.84

Table 6: Comparison of the proposed aggregation strategy with state-of-the-art methods on the Non-iid version (α=0.1) of CIFAR-10 and CIFAR-100 under *cross-device (K=100)* setting.

Methods	Tiny ImageNet → CIFAR-10	Tiny ImageNet → CIFAR-100
FedAvg	77.46	52.11
Loss	77.52	52.68
FedU	76.70	52.25
L-DAWA	**81.87**	**57.81**
L-DAWA-W$_{FedAvg}$	81.61	57.75
L-DAWA-W$_{Loss}$	81.87	57.86
L-DAWA-W$_{FedU}$	81.62	57.34

Table 7: Comparison of the proposed aggregation strategy with state-of-the-art methods for transfer learning with cross-dataset evaluation under *cross-silo (K=10)* setting.

5.5. Evaluation on transfer learning

We further evaluate the generalization of the learned features from FL pre-training by fine-tuning the resulting model on a different dataset. Such evaluation helps in assessing whether the pre-trained representations can be transferred to different downstream tasks. We follow the same procedure that is adopted for linear evaluation. Specifically, we first perform FL pre-training on Tiny ImageNet followed by linear-probe evaluation on CIFAR10/100.

One can see from Table 7 that L-DAWA generalizes well for both CIFAR-10/100 compared to other aggregation strategies in the *cross-silo* settings. Particularly, it obtains 4.4% and 5.7% improvements under these two cross-dataset settings, respectively. Additionally, with the integration of angular divergence into FedAvg, Loss, and FedU, a significant performance boost is obtained for these methods.

6. Conclusion

In this paper, we proposed layer-wise divergence aware weight aggregation (L-DAWA) for SSL pre-training in FL. We empirically show that the SOTA methods get biased towards the clients' metadata (number of samples and loss). To reduce such bias, L-DAWA scales the weighting of each layer of clients' models, based on the measure of *layer-wise angular divergence* with previous global model. Extensive experiments show that L-DAWA obtained a new SOTA performance in the *cross-silo* and *cross-device* settings with both contrastive and non-contrastive SSL methods.

References

[1] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T. Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[2] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Titouan Parcollet, and Nicholas D Lane. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*, 2020.

[3] Debora Caldarola, Barbara Caputo, and Marco Ciccone. Improving generalization in federated learning by seeking flat minima. In *European Conference on Computer Vision*, pages 654–672. Springer, 2022.

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[6] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[7] Xin Dong, Sai Qian Zhang, Ang Li, and HT Kung. Spherefed: Hyperspherical federated learning. *arXiv preprint arXiv:2207.09413*, 2022.

[8] Moming Duan, Duo Liu, Xinyuan Ji, Yu Wu, Liang Liang, Xianzhang Chen, Yujuan Tan, and Ao Ren. Flexible clustered federated learning for client-level data distribution shift. *IEEE Transactions on Parallel and Distributed Systems*, 2021.

[9] William Falcon et al. Pytorch lightning. *GitHub. Note: https://github. com/PyTorchLightning/pytorch-lightning*, 3(6), 2019.

[10] Yan Gao, Titouan Parcollet, Salah Zaiem, Javier Fernandez-Marques, Pedro PB de Gusmao, Daniel J Beutel, and Nicholas D Lane. End-to-end speech recognition from federated acoustic models. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7227–7231. IEEE, 2022.

[11] Priya Goyal, Quentin Duval, Isaac Seessel, Mathilde Caron, Mannat Singh, Ishan Misra, Levent Sagun, Armand Joulin, and Piotr Bojanowski. Vision models are more robust and fair when pretrained on uncurated images without supervision. *arXiv preprint arXiv:2202.08360*, 2022.

[12] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

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

[15] Anil K Jain, Debayan Deb, and Joshua J Engelsma. Biometrics: Trust, but verify. *arXiv preprint arXiv:2105.06625*, 2021.

[16] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[17] Nikos Komodakis and Spyros Gidaris. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations (ICLR)*, 2018.

[18] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[19] Sunwoo Lee, Tuo Zhang, Chaoyang He, and Salman Avestimehr. Layer-wise adaptive model aggregation for scalable federated learning. *arXiv preprint arXiv:2110.10302*, 2021.

[20] Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854, 2020.

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

[22] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[23] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019.

[24] Ziwei Li, Hong-You Chen, Han Wei Shen, and Wei-Lun Chao. Understanding federated learning through loss landscape visualizations: A pilot study. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*.

[25] Wei Liu, Li Chen, Yunfei Chen, and Wenyi Zhang. Accelerating federated learning via momentum gradient descent. *IEEE Transactions on Parallel and Distributed Systems*, 31(8):1754–1766, 2020.

[26] Ekdeep Singh Lubana, Chi Ian Tang, Fahim Kawsar, Robert P Dick, and Akhil Mathur. Orchestra: Unsupervised federated learning via globally consistent clustering. *arXiv preprint arXiv:2205.11506*, 2022.

[27] Xiaosong Ma, Jie Zhang, Song Guo, and Wenchao Xu. Layer-wised model aggregation for personalized federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10092–10101, June 2022.

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

[29] Matias Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, Zhengming Ding, and Chen Chen. Local learning matters: Rethinking data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8397–8406, June 2022.

[30] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[31] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

[32] Wei-Feng Ou, Lai-Man Po, Chang Zhou, Yu-Jia Zhang, Li-Tong Feng, Yasar Abbas Ur Rehman, and Yu-Zhi Zhao. Lincos-softmax: Learning angle-discriminative face representations with linearity-enhanced cosine logits. *IEEE Access*, 8:109758–109769, 2020.

[33] Hyojin Park, Lars Sjosund, YoungJoon Yoo, Nicolas Monet, Jihwan Bang, and Nojun Kwak. Sinet: Extreme lightweight portrait segmentation networks with spatial squeeze module and information blocking decoder. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2066–2074, 2020.

[34] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[35] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

[36] Yasar Abbas Ur Rehman, Yan Gao, Jiajun Shen, Pedro Porto Buarque de Gusmao, and Nicholas Lane. Federated self-supervised learning for video understanding. *arXiv preprint arXiv:2207.01975*, 2022.

[37] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722, 2020.

[38] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[39] Alex Tamkin, Mike Wu, and Noah Goodman. Viewmaker networks: Learning views for unsupervised representation learning. In *International Conference on Learning Representations*, 2021.

[40] Guangting Wang, Yizhou Zhou, Chong Luo, Wenxuan Xie, Wenjun Zeng, and Zhiwei Xiong. Unsupervised visual representation learning by tracking patches in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2563–2572, 2021.

[41] Heqiang Wang and Jie Xu. Friends to help: Saving federated learning from client dropout. *arXiv preprint arXiv:2205.13222*, 2022.

[42] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pages 23965–23998. PMLR, 2022.

[43] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10334–10343, 2019.

[44] Ceyuan Yang, Zhirong Wu, Bolei Zhou, and Stephen Lin. Instance localization for self-supervised detection pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3987–3996, 2021.

[45] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

[46] Fengda Zhang, Kun Kuang, Zhaoyang You, Tao Shen, Jun Xiao, Yin Zhang, Chao Wu, Yueting Zhuang, and Xiaolin Li. Federated unsupervised representation learning. *arXiv preprint arXiv:2010.08982*, 2020.

[47] Jie Zhang, Song Guo, Xiaosong Ma, Haozhao Wang, Wenchao Xu, and Feijie Wu. Parameterized knowledge transfer for personalized federated learning. *Advances in Neural Information Processing Systems*, 34:10092–10104, 2021.

[48] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[49] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[50] Weiming Zhuang, Xin Gan, Yonggang Wen, Shuai Zhang, and Shuai Yi. Collaborative unsupervised visual representation learning from decentralized data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4912–4921, 2021.

[51] Weiming Zhuang, Yonggang Wen, and Shuai Zhang. Divergence-aware federated self-supervised learning. *arXiv preprint arXiv:2204.04385*, 2022.