Appendix

A. Experimental Details

In this section, we provide additional details on the datasets used, preprocessing steps, and experimental methodology. We include code to reproduce our experiments at https://github.com/Liangqiong/ViT-FL-main.

A.1. Detailed Image Pre-processing and Data Partitions

Kaggle Diabetic Retinopathy competition (RETINA) [27] contains a total of $17,563$ pairs of right and left color digital retinal fundus images. Each image is labeled on a scale of 0 to 4 by a well-trained clinician, indicating no, mild, moderate, severe, and proliferative diabetic retinopathy respectively. Following [7], we exclude the samples with scale 1, and then binarize the remaining labels to Healthy (scale 0) and Diseased (scale $2, 3$ or 4). Furthermore, we only use the left images in our study to remove the confounding factor of different disease status of left and right eyes for the same patient. We randomly select $6,000$ balanced ($3,000$ healthy and $3,000$ diseased) images for training, $3,000$ balanced images as the global validation dataset, and $3,000$ balanced images as the global test dataset. Other image pre-processing steps include rescaling as a radius of 300, local color averaging and image clipping, resizing to 256×256, horizontal flipping, and randomly cropping to 224×224. We choose a final 224×224 image dimension to be compatible with current work in both CNNs [19] and Vision Transformers [12].

We simulate three sets of data partitions for the RETINA dataset with each data partition containing four simulated clients: one IID-data partition (Split 1, KS-0), and two non-IID data partitions with label distribution skew (Split 2, KS-0.49, and Split 3, KS-0.57). See Figure 8 for the detailed non-IID data partitions.

CIFAR-10 [31] consists of $50,000$ training and $10,000$ testing 32×32 images in 10 classes, with $5,000$ and $1,000$ images per class in training and test dataset respectively. Following [21], we apply the $10,000$ image test dataset as the global test dataset, set aside $5,000$ images from the training dataset as the global validation dataset, and the remaining $45,000$ images as training dataset. We preprocess each image by resizing to 256×256 and cropping to 224×224.

We simulate one IID-data partition (Split 1, KS-0), one heterogeneous data partition (Split 2, KS-0.65), and one heterogeneous data partition in the extreme case (Split 3, KS-1). Each data partitions contains five clients [7]. We randomly assign each client with images sampled via a uniform distribution over the 10 classes for the IID data partition Split 1, KS-0. For Split 2, KS-0.65, one client receives images sampled from two classes, while the remaining four clients receive images sampled from four classes. Split 3, KS-1 is an extreme case where each client receives images sampled from only two classes. Please refer to Figure 9 for the detailed label distribution on each client for Split 2 and Split 3.

CelebA is a large-scale face attributes dataset with more than 200K celebrity images. The images in CelebA cover large diversities, *i.e.*, large pose variations and background clutter. We use a specially designed federated version of CelebA provided by the LEAF benchmark [5] which partitions the dataset into devices based on the celebrity in the picture (*i.e.*, each device contains only images of celebrity). Following [5], we test on the binary classification task (presence of smile), drop clients with larger than 8 samples to increase the difficult. This results in a total of 227 clients with an average of 5.34 ± 1.11 samples and a total of 1213 samples. for the histogram of the number of training samples in each client. We preprocess each image by resizing to 256×256 and cropping to 224×224.

A.2. Implementation Details and Hyperparameters

Implementation Details. All the methods are implemented with Pytorch and optimized either with SGD (with momentum as 0.9 and no weight decay) or AdamW [29] (with weight decay as 0.05). All experiments were conducted on either a TITAN V GPU or GeForce RTX 2080 GPU. For fair comparison, all the models used in this paper are pretrained from ImageNets ILSVRC-2012 [10]. We set local training epoch in all the FL methods to 1, and the total communication round to 100, unless otherwise stated. We set the local training batch size to 32, and adopt a default input image resolution 224×224 for all methods. More implementation details are shown below.

Training hyperparameters: Inherited from original Transformers training, the Swin-FL models are optimized with AdamW [29], and the ViT-FL models are optimized with SGD. As a fair comparison, the optimizers for the compared CNNs are selected from either SGD and AdamW according to parameter searching. We use linear learning rate warm-up and decay scheduler for the Transformer models. Specifically, we set the warmup steps to 500, and cosine learning rate decay to zero after the maximum round of FL training epochs is reached. The learning rate scheduler for FL with CNNs is selected from linear warm-up and decay or step decay (halved every 30 rounds of FL training). Gradient clipping (at global norm 1) is applied to stabilize the training.

Hyperparameter selection: We tune the best parameters (including learning rate scheduler, and penalty constant μ in the proximal term of FedProx) for FL with CNNs on Split-2 of RETINA and CIFAR-10 dataset with grid search, and apply the same parameters to all the remaining data par-

titions, including the extreme large-scale edge case setting. The detailed hyperparameters of different models for RETINA and CIFAR-10 are shown in Table 5.

FL hyperparameters: For RETINA and CIFAR-10, we set the number of local training epochs E on each client to 1 (unless otherwise stated) and the total number of communication rounds to 100, with all local clients participating in FL training in each round. β is selected from $\{0.1, 0.3, 0.5, 0.7, 0.9, 0.97, 0.99, 0.997\}$ for FedAVGM [22], and is set to 0.5 and 0.3 for Retina and CIFAR-10 dataset, respectively. In FedProx [37], μ is set to 0.001 for Retina dataset and 0.1 for CIFAR-10 dataset by selecting from $\{0.001, 0.01, 0.1, 1\}$.

For the CelebA dataset, we randomly sample 10 clients in each round of FL learning for parallel FL methods. We set E to 1, the maximum train round to 30 for CWT, and 1000 for all the other parallel FL methods, to ensure each local client joins in FL training for around 30 rounds. μ of FedProx is set to 0.001 for CelebA dataset. We allow each client to share 5% percentage of their data among each other for FedAVG-Share on all the compared datasets. The detailed hyperparameters are shown in Table 5 and Table 6. Please refer to our anonymous project page https://github.com/ViT-FL/ViT-FL-main for an implementation to reproduce our results.

B. Additional Results

B.1. Take-aways for Practical Usage

The training strategy of VIT in FL can be directly inherited from VIT training, such as using linear warm-up and learning rate decay, and gradient clip. We also notice that gradient clip stabilizes training for most FL methods on the highly heterogeneous data partition, and therefore can be applied as a general technique in FL applications (see Figure 10 of ViT(B)-FL and ResNet(50)-FedAVG with and without gradient clip). The training of VIT-CWT favors a relatively smaller learning rate on heterogeneous data partitions, whereas using a smaller learning rate for CNN counterparts leads to worse performance. In real-world applications, users can use a large learning rate for IID or mild-skewed data partitions for VIT-CWT, but a smaller learning rate is necessary to stabilize training for highly heterogeneous data partitions.

B.2. Experiments on Real-World Federated Datasets

We further evaluate on a large-scale real-world dataset, OpenImage image classification [32] collected from Flickr, containing 1.3M images spanning 600 categories across 14k clients. We select the categories with #samples per class between 20 and 800 from the dataset, resulting in 81,088 images spanning 365 categories across 9,265 clients. We use similar training parameters to CelebA for OpenImage, i.e., we randomly sample 10 clients in each round of FL learning for parallel FL methods. We set E to 1, the maximum train round to 30 for CWT, and 27,000 for all the other parallel FL methods, to ensure each local client joins in FL training for around 30 rounds. From Table 8, VIT significantly outperforms ResNets on this heterogeneous large-scale real-world data partition, even outperforming ideal centrally-hosted models (60.56% for ResNet and 63.50% for VIT on centrally-hosted dataset)

R-CWT	ViT-CWT	R-FedAVG	R-FedProx	R-FedAVG-Share	ViT-FedAVG
41.62	**64.39**	50.92	51.39	55.34	**67.95**

Table 8. Prediction accuracy (%) on a large-scale real world dataset OpenImage [Ref.A], covering 365 categories across 9,265 clients. VITs significantly outperform their ResNet (R in Table) counterparts.

B.3. Investigating the Influence of Normalization Technique in VIT-FL

The batch normalization layer has been shown to be one of the major factors that deteriorate the performance of federated learning methods on non-IID data partitions [16, 21]. Hsieh *et al.* [21] demonstrate that group normalization (or layer normalization) can avoid the skew-induced accuracy loss of batch normalization on non-IID data. This may raise the question: does the promising performance of VIT-FL come purely from not using a batch normalization layer? To answer this question, we compare VIT-FL with FL-ResNet50 (GN) by replacing all batch normalization layers in ResNet(50) with group normalization. As shown in Table 7, group normalization indeed helps to obtain better performance for both CWT and FedAVG on mildly skewed data partitions than their batch normalization counterparts. For example, the performance on Split-2 of CIFAR-10 is improved from original 56.46% to 93.87%. However, it still suffers performance loss on highly skewed data partitions. In contrast, VIT-FL consistently shows promising results on both mildly skewed and extremely highly skewed data partitions (see Figure 3 in main body paper for our results), indicating that the effectiveness VIT-FL does not arise purely from different normalization techniques.

B.4. Comparisons to Existing FL Methods

We compare VIT-FL to several state-of-the-art optimization based FL methods: FedAVGM [22], FedProx [37], and FedAVG-Share [67]. We use ResNet-50 as the backbone network for all the compared FL methods. We tune the best parameters (including learning rate, momentum parameter β for FedAVGM, and penalty constant μ in the proximal term of FedProx) on Split-2 dataset with grid search, and apply the same parameters to all the remaining data partitions. We allow each client to share 5% percentage of their data among each other for FedAVG-Share.

As shown in Figure 11, VIT-FL outperforms all the other FL methods in non-IID data partitions. Both Fed-

Models	Dataset	Split type	Total Round	Optimizer type	Warm-steps	LR decay	Base LR
ResNets-CWT	Retina & CIFAR-10	All	100	SGD	500	cosine	0.03
EfficientNets-CWT	Retina	All	100	AdamW	500	cosine	0.0005
EfficietNets-CWT	CIFAR-10	All	100	SGD	500	cosine	0.03
ViTs-CWT	Retina & CIFAR-10	All	100	SGD	500	cosine	0.003
Swins-CWT	Retina & CIFAR-10	All	100	AdamW	500	cosine	3.125×10^{-5}
ResNets-FedAVG	Retina & CIFAR-10	All	100	SGD	500	cosine	0.03
EfficientNets-FedAVG	Retina	All	100	AdamW	500	cosine	0.0005
EfficientNets-FedAVG	CIFAR-10	All	100	SGD	500	cosine	0.03
ViTs-FedAVG	Retina & CIFAR-10	All	100	SGD	500	cosine	0.03
Swins-FedAVG	Retina & CIFAR-10	All	100	AdamW	500	cosine	3.125×10^{-5}
ResNet(50)-FedAVGM [22]	Retina	All	100	SGD	0	step	0.03
ResNet(50)-FedAVGM [22]	CIFAR-10	All	100	SGD	500	cosine	0.03
ResNet(50)-FedProx [37]	Retina	All	100	SGD	0	step	0.03
ResNet(50)-FedProx [37]	CIFAR-10	All	100	SGD	500	cosine	0.03
ResNet(50)-FedAVG-Share [67]	Retina & CIFAR-10	All	100	SGD	500	cosine	0.03

Table 5. Table of hyperparameters for experiments on RETINA and CIFAR-10 with ResNets [19], EfficientNets [59], ViTs and Swins [41]. Gradient clip at global norm 1 are applied to all models to stabilize the training. The learning rate is halved every 30 epochs in the step decay scheduler.

Models	Avg. Total Round	Warm-steps	Optimizer type	LR decay	Base LR
ResNet(50)-CWT	30	500	SGD	cosine	0.03
ResNet(50)-FedAVG	30	500	SGD	cosine	0.03
ResNet(50)-FedProx	30	500	SGD	cosine	0.03
ResNet(50)-FedAVG-Share	30	500	SGD	cosine	0.03
ViT(S)-CWT	30	500	SGD	cosine	0.003
ViT(S)-FedAVG	30	500	SGD	cosine	0.03

Table 6. Table of hyperparameters for experiments on CelebA and OpenImage. All methods are optimized with SGD (momentum 0.9 and no weight decay), and gradient clip at global norm 1.

Prox [37] and FedAVGM [22] suffer severe performance drops on highly heterogeneous data partitions despite carefully tuned optimization parameters. Similarly, FedAVG-Share also suffers from performance drops on highly heterogeneous data partition Split-3 even when 5% percentage of the local data is shared among all clients (94.2% of Split-3 on CIFAR-10 dataset compared to 96% on Split-1). We conclude that simply using Transformers achieve superior performance than several recent methods designed for federated optimization, which often require careful tuning of optimization parameters.

	RETINA			CIFAR-10		
	Split-1	Split-2	Split-3	Split-1	Split-2	Split-3
ResNet(50)-CWT	79.44	77.01	71.30	96.08	56.46	19.92
ResNet(50)(GN)-CWT	82.21	81.13	77.05	95.10	93.87	87.70
ResNet(50)-FedAVG	80.48	76.36	75.99	96.51	93.14	59.68
ResNet(50)(GN)-FedAVG	82.40	80.13	80.57	96.39	95.12	86.20

Table 7. Prediction accuracy (%) of CWT and FedAVG on RETINA and CIFAR-10 when using ResNet50 and ResNet50(GN) as the backbone network. Replacing the batch normalization layer with group normalization in ResNet50 still suffers performance loss on highly heterogeneous data partitions, indicating that the promising performance of VIT-FL does not come purely from not using batch normalization.

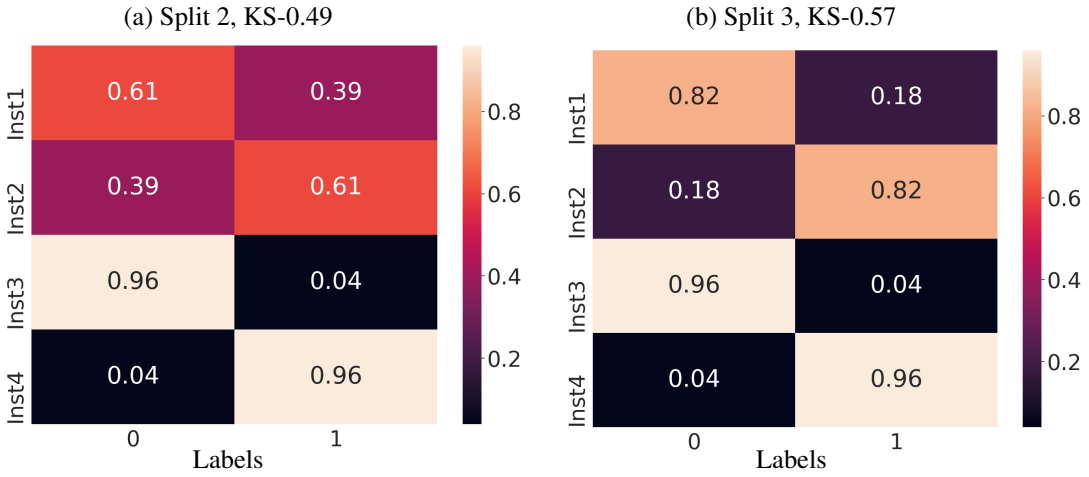


Figure 8. Detailed non-IID data partitions on RETINA with label distribution skew. The value in each rectangle shows the fraction of data samples of a class over their total number.

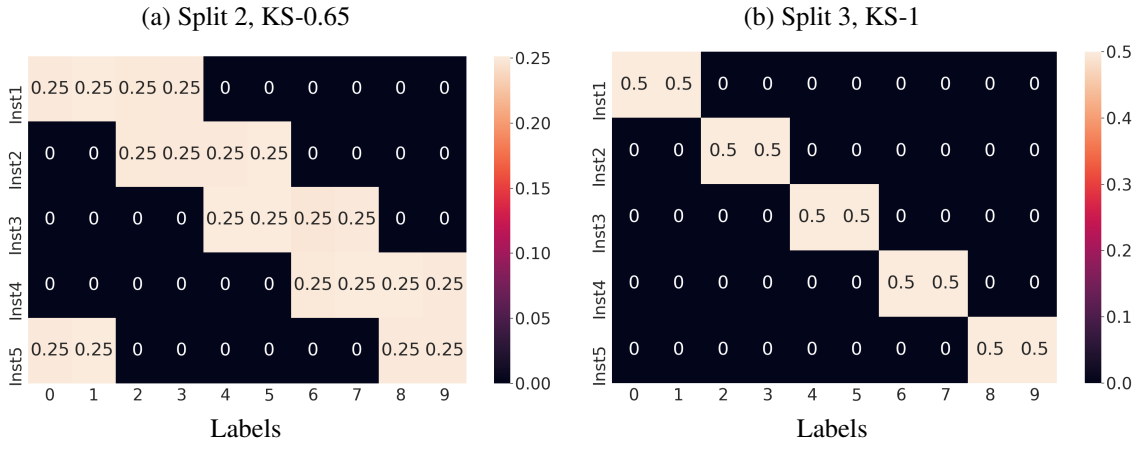


Figure 9. Detailed non-IID data partitions on CIFAR-10 with label distribution skew. The value in each rectangle shows the fraction of data samples in a class over their total number.

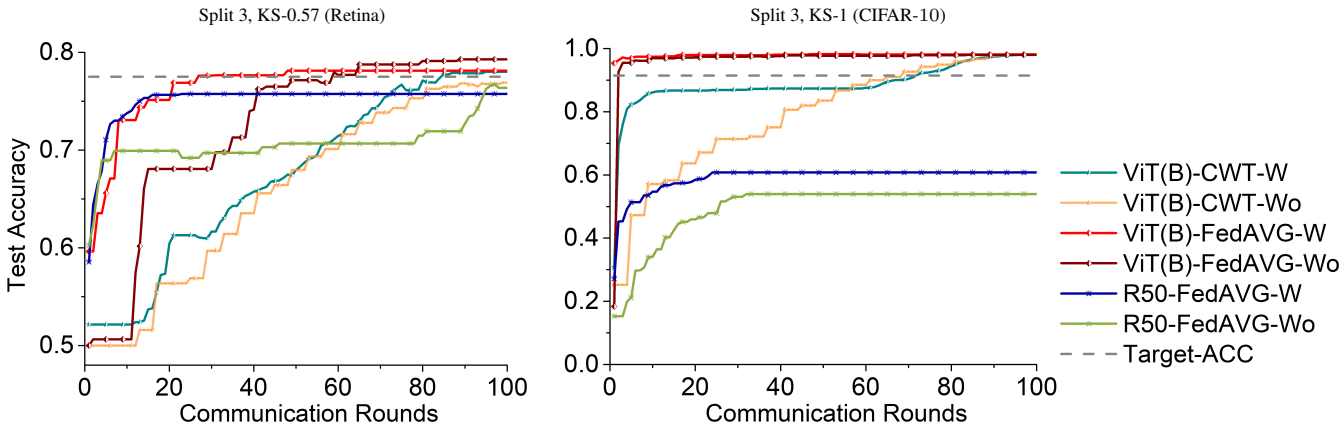
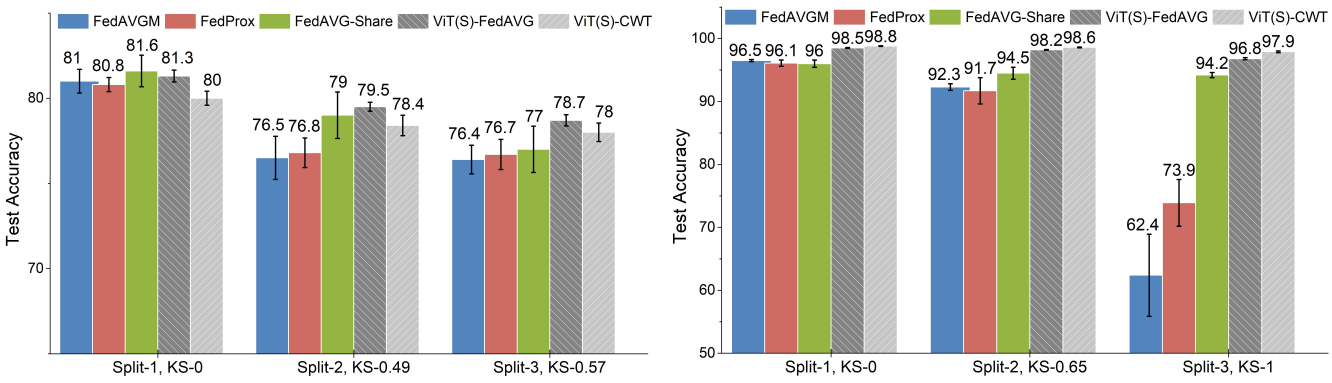


Figure 10. Influence of gradient clip on different FL methods with ViT(B) and ResNet-50(R50) as the backbone networks, respectively. In the legend, **-W** denotes with gradient clip and **-Wo** denotes without gradient clip. We find that gradient clip stabilizes training and accelerates convergence speed on highly heterogeneous data splits.



Retina dataset CIFAR-10 dataset

Figure 11. Comparisons with state-of-the-art optimization based federated learning methods with ResNet-50 as backbone. Vision Transformer-based FL methods (ViT(S)-CWT and ViT(S)-FedAVG) outperform other methods in non-IID data partitions.