FedConv: Enhancing Convolutional Neural Networks for Handling Data Heterogeneity in Federated Learning

Anonymous authors
Paper under double-blind review

Abstract

Federated learning (FL) is an emerging paradigm in machine learning, where a shared model is collaboratively learned using data from multiple devices to mitigate the risk of data leakage. While recent studies posit that Vision Transformer (ViT) outperforms Convolutional Neural Networks (CNNs) in addressing data heterogeneity in FL, the specific architectural components that underpin this advantage have yet to be elucidated. In this paper, we systematically investigate the impact of different architectural elements, such as activation functions and normalization layers, on the performance within heterogeneous FL. Through rigorous empirical analyses, we are able to offer the first-of-its-kind general guidance on micro-architecture design principles for heterogeneous FL.

Intriguingly, our findings indicate that with strategic architectural modifications, pure CNNs can achieve a level of robustness that either matches or even exceeds that of ViTs when handling heterogeneous data clients in FL. Additionally, our approach is compatible with existing FL techniques and delivers state-of-the-art solutions across a broad spectrum of FL benchmarks.

1 Introduction

Federated Learning (FL) is an emerging paradigm that holds significant potential in safeguarding user data privacy in a variety of real-world applications, such as mobile edge computing (Li et al., 2020a). Yet, one of the biggest challenges in FL is data heterogeneity, making it difficult to develop a single shared model that can generalize well across all local devices. While numerous solutions have been proposed to enhance heterogeneous FL from an optimization standpoint (Li et al., 2020c; Hsu et al., 2019), the recent work by Qu et al. (2022b) highlights that the selection of neural architectures also plays a crucial role in addressing this challenge. This study delves into the comparative strengths of Vision Transformers (ViTs) vis-à-vis Convolutional Neural Networks (CNNs) within the FL context and posits that the performance disparity between ViTs and CNNs amplifies with increasing data heterogeneity.

Though the study in Qu et al. (2022b) provides the first systematic and comprehensive analysis, it remains in macro level, leaving certain nuances of neural architectures unexplored. Specifically, the interplay between individual architectural elements in ViT and its robustness to heterogeneous data in FL remains unclear. While several recent studies (Bai et al., 2021; Zhang et al., 2022) suggest that self-attention-block is the main building block that contributes significantly to ViT's robustness against out-of-distribution data, its applicability to other settings, particularly heterogeneous FL, is yet to be ascertained. This gap in understanding prompts us to consider the following questions: ***which architectural elements in ViT underpin its superior performance in heterogeneous FL?*** And as a step further, ***can CNNs benefit from incorporating these architectural elements to improve their performance in this scenario?***

To this end, we take a closer analysis of main architectural elements found in ViTs differ from CNNs. Through picking these variances, several pivotal designs are uncovered for improving model robustness against heterogeneous data in FL. First, we note the design of activation functions matters, concluding that smooth and near-zero-centered activation functions consistently yield substantial improvements. Second, simplifying the architecture by completely removing all normalization layers and retraining a single activation

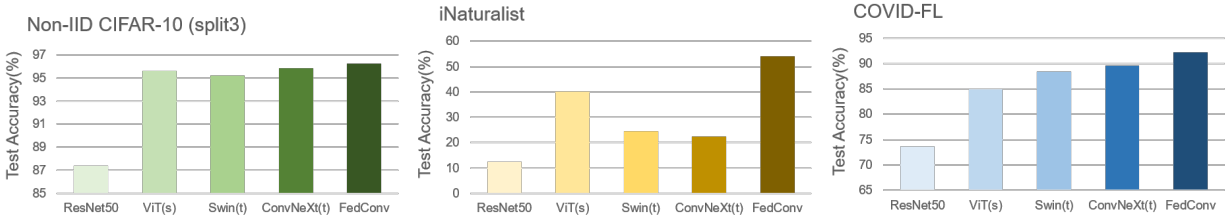

Figure 1: Performance comparison on three heterogeneous FL datasets. While a vanilla ResNet significantly underperforms Transformers and ConvNeXt when facing data heterogeneity, our enhanced CNN model, named FedConv, consistently achieves superior performance.

function in each block emerges as a beneficial design. Third, we reveal two key properties — feature extraction from overlapping patches and convolutions only for downsampling — that define a robust stem layer for handling diverse input data. Lastly, we critically observe that a large enough kernel size (*i.e.*, 9) is essential in securing model robustness against heterogeneous distributions.

Our experiments extensively verify the consistent improvements brought by these architectural designs in the context of heterogeneous FL. Specifically, by integrating these designs, we are able to build a pure CNN architecture, dubbed FedConv, that outperforms established models, including ViT (Dosovitskiy et al., 2020), Swin-Transformer (Liu et al., 2021), and ConvNeXt(Liu et al., 2022). Notably, our FedConv achieves 92.21% accuracy on COVID-FL and 54.19% on iNaturalist, outperforming the next-best solutions by 2.57% and 13.89%, respectively. Moreover, our FedConv can effectively generalize to other datasets with varying numbers of data clients, from as few as 5 to more than 2,000; when combined with existing FL methods, our FedConv consistently registers new performance records across a range of FL benchmarks.

In conclusion, our findings shed light on the pivotal role of architecture configuration in robustifying CNNs for heterogeneous FL. We experimentally provide heuristic insights into noteworthy distinctions among structural elements, and hope these findings will inspire the community to further probe the significance of architectural nuances in different FL settings.

2 Related Works

Federated Learning. FL is a decentralized approach that aims to learn a shared model by aggregating updates or parameters from locally distributed training data (McMahan et al., 2017). However, one of the key challenges in FL is the presence of data heterogeneity, or varying distribution of training data across clients, which has been shown to cause weight divergence (Zhao et al., 2018) and other optimization issues (Hsieh et al., 2020) in FL.

To address this challenge, FedProx (Li et al., 2020c) adds a proximal term in the loss function to achieve more stable and accurate convergence; FedAVG-Share (Zhao et al., 2018) keeps a small globally-shared subset amongst devices; SCAFFOLD (Karimireddy et al., 2020) introduces a variable to both estimate and correct update direction of each client. Beyond these, several other techniques have been explored in heterogeneous FL, including reinforcement learning (Wang et al., 2020a), hierarchical clustering (Briggs et al., 2020), knowledge distillation (Zhu et al., 2021; Li & Wang, 2019; Qu et al., 2022a), and self-supervised learning (Yan et al., 2023; Zhang et al., 2021).

Recently, a new perspective for improving heterogeneous FL is by designing novel neural architectures. Li et al. (2021) points out that updating only non-BatchNorm (BN) layers significantly enhances FedAVG performance, while Du et al. (2022) suggests that simply replacing BN with LayerNorm (LN) can mitigate external covariate shifts and accelerate convergence. Wang et al. (2020b) shows neuron permutation matters when averaging model weights from different clients. Our research draws inspiration from Qu et al. (2022b), which shows that Transformers are inherently stronger than CNNs in handling data heterogeneity (Qu et al., 2022b). Yet, we come to a completely different conclusion: with the right architecture designs, CNNs can be comparable to, or even more robust than, Transformers in the context of heterogeneous FL.

Vision Transformer. CNNs have been the dominant architecture in visual recognition for nearly a decade due to their superior performance (Simonyan & Zisserman, 2014; He et al., 2016; Szegedy et al., 2016a; Howard et al., 2017; Tan & Le, 2019). However, the recently emerged ViTs challenge the leading position of CNNs (Dosovitskiy et al., 2020; Touvron et al., 2021a;b; Liu et al., 2021) — by applying a stack of global self-attention blocks (Vaswani et al., 2017) on a sequence of image patches, Transformers can even show stronger performance than CNNs in a range of visual benchmarks, especially when a huge amount of training data is available (Dosovitskiy et al., 2020; Touvron et al., 2021a; Liu et al., 2021; He et al., 2022). Additionally, Transformers are shown to be inherently more robust than CNNs against occlusions, adversarial perturbation, and out-of-distribution corruptions (Bhojanapalli et al., 2021; Bai et al., 2021; Zhang et al., 2022; Paul & Chen, 2022).

Modernized CNN. Recent works also reignite the discussion on whether CNNs can still be the preferred architecture for visual recognition. Wightman et al. (2021) find that by simply changing to an advanced training recipe, the classic ResNet-50 achieves a remarkable 4% improvement on ImageNet, a performance that is comparable to its DeiT counterpart (Touvron et al., 2021a). ConvMixer (Trockman & Kolter, 2022), on the other hand, integrates the patchify stem setup into CNNs, yielding competitive performance with Transformers. Furthermore, ConvNeXt (Liu et al., 2022) shows that, by aggressively incorporating every applicable architectural design from Transformer, even the pure CNN can attain excessively strong performance across a variety of visual tasks.

Our work is closely related to ConvNeXt (Liu et al., 2022). However, in contrast to their design philosophy which aims to build Transformer-like CNNs, our goal is to pinpoint a core set of architectural elements that can enhance CNNs, particularly in the context of heterogeneous FL.

3 FedConv

In this section, we conduct a comprehensive analysis of several architectural elements in heterogeneous FL. Our findings suggest that pure CNNs can achieve comparable or even superior performance in heterogeneous FL when incorporating specific architectural designs. Key designs including switching to smooth and near-zero-centered activation functions in Section 3.2, reducing activation and normalization layers in Section 3.3, adopting the stem layer setup in Section 3.4, and enlarging the kernel size in Section 3.5. By combining these designs, we develop a novel CNN architecture, **FedConv**. As demonstrated in Section 3.6, this architecture emerges as a simple yet effective alternative to handle heterogeneous data in FL.

3.1 Experiment Setup

Datasets. Our main dataset is COVID-FL (Yan et al., 2023), a real-world medical FL dataset containing 20,055 medical scans, sourced from real hospital sites, forming a consortium of 12 clients. The images within each client range from 120 to 3860, categorized based on different lung disease: health, pneumonia, and COVID. Note that clients (*i.e.*, hospital) in this dataset are characterized by the absence of one or more classes. This absence induces pronounced data heterogeneity in FL, driven both by the limited overlap in client partitions and the imbalanced distribution of labels. To provide a comprehensive assessment of our approach, we report performance in both the centralized training setting and the distributed FL setting.

Federated learning methods. We consider the classic Federated Averaging (FedAVG) (McMahan et al., 2017) as the default FL method, unless specified otherwise. FedAVG operates as follows: 1) the global server model is initially sent to local clients; 2) these clients next engage in multiple rounds of local updates with their local data; 3) once updated, these models are sent back to the server, where they are averaged to create an updated global model.

Training recipe. For fair comparison, we employ the same methodology for training each model. All models are first pre-trained on ImageNet using the recipe provided in (Liu et al., 2022), and then get fine-tuned on COVID-FL using FedAVG. Specifically, in fine-tuning, we set the base learning rate to 1.75e-4 with a cosine learning rate scheduler, weight decay to 0.05, batch size to 64, and warmup epoch to 5. Following

(Qu et al., 2022b; Yan et al., 2023), we apply AdamW optimizer (Loshchilov & Hutter, 2017) to each local client and maintain their momentum term locally. For FedAVG, we set the total communication round to 100, with a local training epoch of 1. Note that all 12 clients are included in every communication round.

Computational cost. We hereby use FLOPs as the metric to measure the model scale. To ensure a fair comparison, all models considered in this study are intentionally calibrated to align with the FLOPs scale of ViT-Small (Dosovitskiy et al., 2020), unless specified otherwise.

To improve the performance-to-cost ratio of our models, we pivot from the conventional convolution operation in ResNet, opting instead for depth-wise convolution. This method, as corroborated by prior research (Howard et al., 2017; Sandler et al., 2018; Howard et al., 2019; Tan & Le, 2019), exhibits a better balance between performance and computational cost. Additionally, following the design philosophy of ResNeXt (Xie et al., 2017), we adjust the base channel configuration across stages, transitioning from (64, 128, 256, 512) to (96, 192, 384, 768); we further calibrate the block number to keep the total FLOPs closely align with the ViT-Small scale.

An improved ResNet baseline. Building upon the adoption of depth-wise convolutions, we demonstrate that further incorporating two simple architectural elements from ViT enables us to build a much stronger ResNet baseline. Firstly, we replace BN with LN. This shift is motivated by prior studies which reveal the adverse effects of maintaining the EMA of batch-averaged feature statistics in heterogeneous FL (Li et al., 2021; Du et al., 2022); instead, they advocate that batch-independent normalization techniques like LN can improve both the performance and convergence speed of the global model (Du et al., 2022). Secondly, we replace the traditional ReLU activation function with GELU (Hendrycks & Gimpel, 2016). As discussed in Section 3.2, this change can substantially increase the classification accuracy in COVID-FL by 4.52%, from 72.92% to 77.44%.

We refer to the model resulting from these changes as **ResNet-M**, which will serve as the default baseline for our subsequent experiments. However, a critical observation is that, despite its enhancements, the accuracy of ResNet-M still lags notably behind its Transformer counterpart in heterogeneous FL, which registers a much higher accuracy at 88.38%.

3.2 Activation function

We hereby investigate the impact of activation function selection on model performance in the context of heterogeneous FL. Our exploration begins with GELU, the default activation function in Transformers (Dosovitskiy et al., 2020; Touvron et al., 2021a; Liu et al., 2021). Previous studies show that GELU consistently outperforms ReLU in terms of both clean accuracy and adversarial robustness (Hendrycks & Gimpel, 2016; Elfwing et al., 2018; Clevert et al., 2016; Xie et al., 2020). In our assessment of GELU's efficacy for heterogeneous FL, we observe a similar conclusion: as shown in Table 1, replacing ReLU with GELU can lead to a significant accuracy improvement of 4.52% in COVID-FL, from 72.92% to 77.44%.

Building upon the insights from (Xie et al., 2020), we next explore the generalization of this improvement to smooth activation functions, which are defined as being $\mathcal{C}^1$ smooth. Specifically, we assess five activation functions: two that are non-smooth (Parametric Rectified Linear Unit (PReLU) (He et al., 2015) and Leaky ReLU (LReLU)), and three that are smooth (SoftPlus (Dugas et al., 2000), Exponential Linear Unit (ELU) (Clevert et al., 2016), and Sigmoid Linear Unit (SiLU) (Elfwing et al., 2018)). The curves of these activation functions are shown in Figure 2, and their performance on COVID-FL is reported in Table 1. Our results show that smooth activation functions generally outperform their non-smooth counterparts in heterogeneous FL. Notably, both SiLU and ELU achieve an accuracy surpassing 78%, markedly superior to the accuracy of LReLU (73.41%) and PReLU (74.24%). Yet, there is an anomaly in our findings: SoftPlus, when replacing ReLU, fails to show a notable improvement. This suggests that *smoothness alone is not sufficient for achieving strong performance in heterogeneous FL.*

Upon conducting various examination of characteristics of various smooth activation functions, a noteworthy divergence emerges, particularly in the case of SoftPlus compared to the others: while SoftPlus consistently generates positive outputs, the other three (GELU, SiLU and ELU) hold the capability to produce negative

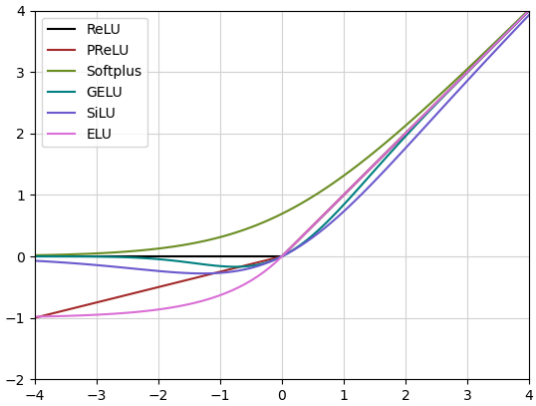

Figure 2: Plots of various activation function curves.

Table 1: A study on the effect of various activation functions.

Activation	Central	FL	Mean Act
ReLU	95.59	72.92	0.28
LReLU	95.54	73.41	0.26
PReLU	95.41	74.24	0.20
SoftPlus	95.28	73.86	0.70
GELU	95.82	77.44	0.07
SiLU	95.81	79.52	0.04
ELU	95.59	78.25	-0.07

values within specific input ranges. This characteristic suggests a potential advantage in facilitating outputs that are more centered around zero.

To provide a quantitative assessment of this distinction, we calculate the mean activation values of ResNet-M using respective activation functions, across all layers when presented with COVID-FL dataset. The results, outlined in the "Mean Act" column of Table 1, underscore a consistent pattern: GELU, SiLU and ELU with better model performance maintain mean activation values close to zero. In contrast, other activation functions showcase more significant deviations from zero, implying a more skewed distribution of activation values. These empirical insights lead us to a compelling conclusion: the choice of a **smooth and near-zero-centered** activation function proves to be advantageous in heterogeneous FL.

3.3 Reducing activation and normalization layers

Transformer blocks, in contrast to traditional CNN blocks, generally incorporate fewer activation and normalization layers (Dosovitskiy et al., 2020; Touvron et al., 2021b; Liu et al., 2021). Prior works show that CNNs can remain stable or even attain higher performance with fewer activation layers (Liu et al., 2022) or without normalization layers (Brock et al., 2021b). In this section, we delve into this design choice in the context of heterogeneous FL.

Drawing inspiration from ConvNeXt, we evaluate three different block instantiations: 1) *Normal Block*, which serves as the basic building block of our ResNet-M baseline; 2) *Invert Block*, originally proposed in (Sandler et al., 2018), which has a hidden dimension that is four times the size of the input dimension; and 3) *InvertUp Block*, which modifies Invert Block by repositioning the depth-wise convolution to the top of the block. The detailed block configurations is illustrated in Figure 3. Notably, regardless of the specific block configuration employed, we always ensure similar FLOPs across models by adjusting the number of blocks, maintaining the fair comparison.

Reducing activation layers. We begin our experiments by aggressively reducing the number of activation layers. Specifically, we retain only one activation layer in each block to ensure the preservation of non-linearity. As highlighted in Table 2a, all three block designs showcase at least one configuration that delivers substantial performance gains in heterogeneous FL. For example, the best configurations yield an improvement of 6.89% for Normal Block (from 77.44% to 84.33%), 3.77% for Invert Block (from 80.35% to 84.12%), and 1.48% for InvertUp Block (from 80.96% to 82.44%). More intriguingly, a simple rule-of-thumb design principle emerges for these optimal configurations: the activation function is most effective when placed subsequent to the channel-expanding convolution layer, whose output channel dimension is larger than its input dimension.

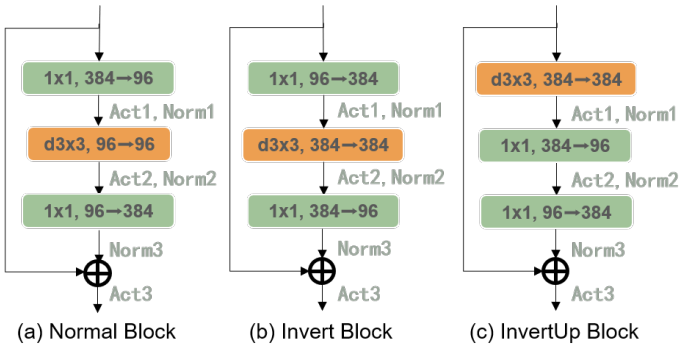

Figure 3: Block architecture details.

Table 2: (a) Analysis of the effect of reducing activation functions. "ActX" refers to the block that only keeps the activation layer after the Xth convolution layer. "All" refers to keeping all activation layers within the block. (b) Analysis of the effect of reducing normalization functions. "NormY" refers to the block that only keeps the normalization layer after the Yth convolution layer. "No Norm" refers to removing all normalization layers within the block.

Block	Act	Central	FL
Normal	All	95.82	77.44
	Act1	95.41	79.24
	Act2	95.41	80.12
	Act3	95.89	84.33
Invert	All	95.64	80.35
	Act1	96.19	84.12
	Act2	95.24	82.06
	Act3	95.46	78.60
InvertUp	All	95.76	80.96
	Act1	95.21	76.97
	Act2	95.71	82.44
	Act3	95.56	77.46

(a)

Block	Act	Norm	Central	FL
Normal	Act3	All	95.89	84.33
	Act3	Norm1	95.84	82.06
	Act3	Norm2	95.36	82.33
	Act3	Norm3	95.34	81.36
	Act3	No Norm	95.29	82.50
Invert	Act1	All	96.19	84.12
	Act1	Norm1	95.76	82.48
	Act1	Norm2	96.04	83.02
	Act1	Norm3	95.59	86.19
	Act1	No Norm	95.94	84.63
InvertUp	Act2	All	95.71	82.44
	Act2	Norm1	95.64	82.45
	Act2	Norm2	95.64	83.46
	Act2	Norm3	95.71	85.70
	Act2	No Norm	95.74	85.65

(b)

Reducing normalization layers. Building upon our experiments above with only one best-positioned activation layer, we further investigate the impact of aggressively removing normalization layers. Similarly, we hereby are interested in keeping only one normalization layer in each block. As presented in Table 2b, we observe that the effects of removing normalization layers are highly block-dependent in heterogeneous FL. Notably, the removal of normalization layers consistently hampers the performance of the Normal Block, while the Invert Block and InvertUp block experience improvements with enhancements of up to 2.07% and 3.26%, respectively. This block-dependent variability underscores the importance of considering the tailored normalization choice for each block when optimizing for heterogeneous FL scenarios. Despite this, in out next step, we aspire to explore further for a more generalized paradigm for normalization layer design.

Normalization-free setup. Another interesting direction to explore is by removing all normalization layers from our models. This is motivated by recent studies that demonstrate high-performance visual recognition with normalization-free CNNs (Brock et al., 2021a;b). To achieve this, we train normalization-free networks using the Adaptive Gradient Clipping (AGC) technique, following (Brock et al., 2021b). The results are presented in Table 2b. Surprisingly, we observe that these normalization-free variants are able

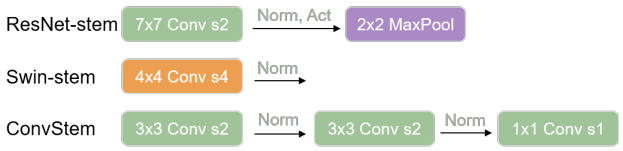

Figure 4: Illustration of different stem setups, including ResNet-stem, Swin-stem and ConvStem. 's' denotes the stride of convolution.

Table 3: Analysis of the effect of various stem layers. "(Kernel Size 5)" denotes using a kernel size of 5 in convolution. "(No MaxPool)" denotes removing the max-pooling layer and increasing the stride of the first convolution layer accordingly.

Stem	Central	FL
ResNet-stem	95.82	77.44
Swin-stem	95.61	79.44
ConvStem	95.26	**83.01**
Swin-stem (Kernel Size 5)	95.64	82.76
ResNet-stem (No MaxPool)	95.76	82.59

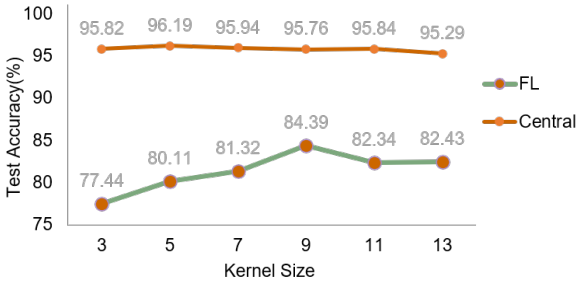

Figure 5: The study on the effect of different kernel sizes.

to achieve competitive performance compared to their best counterparts with only one activation layer and one normalization layer, *e.g.*, 82.50% *vs.* 82.33% for Normal Block, 84.63% *vs.* 86.19% for Invert Block, and 85.65% *vs.* 85.70% for InvertUP Block. Additionally, a normalization-free setup offers practical advantages such as faster training speed (Singh & Shrivastava, 2019) and reduced GPU memory overhead (Bulo et al., 2018). In our context, compared to the vanilla block instantiations, removing normalization layers leads to a 28.5% acceleration in training, a 38.8% cut in GPU memory, and conveniently bypasses the need to determine the best strategy for reducing normalization layers.

Moreover, our proposed normalization-free approach is particularly noteworthy for its superior performance compared to FedBN (Li et al., 2021) (see appendix for detailed comparisons), a widely acknowledged normalization technique in FL. While FedBN decentralizes normalization across multiple clients, ours outperforms it by completely eliminating normalization layers. These notable findings prompt a reevaluation of the role of "already extensively discussed" normalization layers in heterogeneous FL, highlighting the potential benefits of adopting normalization-free models.

3.4 Stem layer

The initial processing of input data, referred to as the *stem*, differs between CNNs and Transformers. Previous studies Xiao et al. (2021) have underscored the substantial impact of early-stage processing on overall model performance. Typically, CNNs employ a stack of convolutions to downsample images into desired-sized feature maps, while Transformers use patchify layers to directly divide images into a set of tokens. To better understand the impact of the stem layer in heterogeneous FL, we comparatively study diverse stem designs, including the default ResNet-stem, Swin-stem, and ConvStem inspired by (Xiao et al., 2021). A visualization of these stem designs is provided in Figure 4, and the empirical results are reported in Table 3. We note that 1) both Swin-stem and ConvStem outperform the vanilla ResNet-stem baseline, and 2) ConvStem attains the best performance. Next, we probe potential enhancements to ResNet-stem and Swin-stem by leveraging the "advanced" designs in ConvStem.

Overlapping convolution. We first investigate the performance gap between Swin-stem and ConvStem. We posit that the crux of this gap might be attributed to the variation in patch overlapping. Specifically, Swin-stem employs a convolution layer with a stride of 4 and a kernel size of 4, thereby extracting features from non-overlapping patches; while ConvStem resorts to overlapping convolutions, which inherently bring in

adjacent gradient consistency and spatial smoothness (Graham et al., 2021). To validate our hypothesis, we modify the Swin-stem by increasing the kernel size to 5 while retaining a stride of 4. This seemingly modest alteration yielded a marked performance enhancement of +3.32% (from 79.44% to 82.76%), confirming the pivotal role of overlapping convolutions within stem layers in heterogeneous FL.

Convlutions-only downsampling. ResNet-stem, despite its employment of a 7×7 convolution layer with a stride of 2 — thereby extracting features from overlapped patches — remarkably lags behind Swin-stem in performance. A noteworthy distinction lies in the ResNet-stem's integration of an additional max-pooling layer to facilitate part of its downsampling; while both Swin-stem and ConvStem exclusively rely on convolution layers for this purpose. To understand the role of the max-pooling layer within ResNet-stem, we remove it and adjust the stride of the initial convolution layer from 2 to 4. As shown in Table 3, this modification, dubbed "ResNet-stem (No MaxPool)", registers an impressive 5.15% absolute accuracy improvement over the vanilla ResNet-stem. This observation suggests that employing convolutions alone (hence no pooling layers) for downsampling is important in heterogeneous FL.

In summary, our analysis highlights the significance of the stem layer's design in securing model performance in heterogeneous FL. Specifically, two key factors are identified, *i.e.*, the stem layer needs to extract features from overlapping patches and employs convolutions only for downsampling.

3.5 Kernel size

Global self-attention is generally recognized as a critical factor that contributes to the robustness of ViT across diverse data distributions in FL (Qu et al., 2022b). Motivated by this, we explore whether augmenting the receptive field of a CNN– by increasing its kernel size — can enhance this robustness. As depicted in Figure 5, increasing the kernel size directly corresponds to significant accuracy improvements in heterogeneous FL. The largest improvement is achieved with a kernel size of 9, elevating accuracy by 6.95% over the baseline model with a kernel size of 3 (*i.e.*, 84.39% *vs.* 77.44%). It is worth noting, however, that pushing the kernel size beyond 9 ceases to yield further performance enhancements and might, in fact, detract from accuracy.

3.6 Component combination

We now introduce FedConv, a novel CNN architecture designed to robustly handle heterogeneous clients in FL. Originating from our ResNet-M baseline model, FedConv incorporates five pivotal design elements, including *SiLU activation function*, *retraining only one activation function per block*, *normalization-free setup*, *ConvStem*, and *a large kernel size of 9*. By building upon three distinct instantiations of CNN blocks (as illustrated in Figure 3), we term the resulting models as FedConv-Normal, FedConv-Invert, and FedConv-InvertUp.

Our empirical results demonstrate that these seemingly simple architectural designs collectively lead to a significant performance improvement in heterogeneous FL. As shown in Table 4, FedConv models achieve the best performance, surpassing strong competitors such as ViT, Swin-Transformer, and ConvNeXt. The standout performer, FedConv-InvertUp, records the highest accuracy of 92.21%, outperforming the prior art, ConvNeXt, by 2.64%. These outcomes compellingly contest the assertions in (Qu et al., 2022b), highlighting that a pure CNN architecture can be a competitive alternative to ViT in heterogeneous FL scenarios.

4 Generalization and Practical Implications

In this section, we assess the generalization ability and communication costs of FedConv, both of which are critical metrics for practical applications in real-world scenarios. To facilitate our analysis, we choose FedConv-InvertUp, our top-performing variant, to serve as the default model for our ablation study.

Table 4: Performance comparison on COVID-FL. By incorporating archiectural elements such as SiLU activation function, retaining only one activation function, the normalization-free setup, ConvStem, and a large kernel size of 9, our FedConv models consistently outperform other advanced solutions in heterogeneous FL.

Model	FLOPs	Central	FL
ResNet50	4.1G	95.66	73.61
ResNet-M	4.6G	95.82	77.44
Swin-Tiny	4.5G	95.74	88.38
ViT-Small	4.6G	95.86	84.89
ConvNeXt-Tiny	4.5G	96.01	89.57
FedConv-Normal	4.6G	95.84	90.61
FedConv-Invert	4.6G	96.19	91.68
FedConv-InvertUp	4.6G	96.04	92.21

Table 5: Performance comparison on CIFAR-10 and iNaturalist. Our FedConv model consistently outperforms other models. Notably, as data heterogeneity increases, FedConv's strong generalization becomes more evident.

Model	CIFAR-10				iNaturalist
	Central	Split-1	Split-2	Split-3	FL
ResNet50	97.47	96.69	95.56	87.43	12.61
Swin-Tiny	98.31	98.36	97.83	95.22	24.57
ViT-Small	97.99	98.24	97.84	95.64	40.30
ConvNeXt-Tiny	98.31	98.20	97.67	95.85	22.53
FedConv-InvertUp	98.42	98.11	97.74	96.26	54.19

4.1 Generalizing to other Datasets

To assess the generalizability of our model, we evaluate its performance on two additional heterogeneous FL datasets: CIFAR-10 (Krizhevsky et al., 2009) and iNaturalist (Van Horn et al., 2018).

CIFAR-10. Following (Qu et al., 2022b), we use the original test set as our validation set, and the training set is divided into five parts, with each part representing one client. Leveraging the mean Kolmogorov-Smirnov (KS) statistic to measure distribution variations between pairs of clients, we create three partitions, each representing different levels of label distribution skewness: split-1 (KS=0, representing an IID set), split-2 (KS=0.49), and split-3 (KS=0.57).

iNaturalist. iNaturalist is a large-scale fine-grained visual classification dataset, containing natural images taken by citizen scientists (Hsu et al., 2020). For our analysis, we use a federated version, iNature, sourced from FedScale (Lai et al., 2021). This version includes 193K images from 2295 clients, of which 1901 are in the training set and the remaining 394 are in the validation set.

Table 5 reports the performance on CIFAR-10 and iNaturalist datasets. As data heterogeneity increases from split1 to split3 on CIFAR-10, while FedConv only experiences a modest accuracy drop of 1.85%, other models drop the accuracy by at least 2.35%. On iNaturalist, FedConv impressively achieves an accuracy of 54.19%, surpassing the runner-up, ViT-Small, by more than 10%. These results confirm the strong generalization ability of FedConv in highly heterogeneous FL settings.

4.2 Generalizing to other FL Methods

Our proposed FedConv architecture is designed to be largely independent of existing optimization-based FL methods, enabling seamless integration and the potential for further performance improvements.

Table 6: Performance comparison with different FL methods on COVID-FL. 'Share' denotes 'FedAVG-Share'. We note our FedConv consistently shows superior performance.

Model	FedProx	Share	FedYogi
ResNet50	72.92	92.43	66.01
ViT-Small	87.07	93.89	87.69
Swin-Tiny	87.74	94.02	91.86
ConvNeXt-Tiny	89.35	95.11	92.46
FedConv-InvertUp	92.11	95.23	93.10

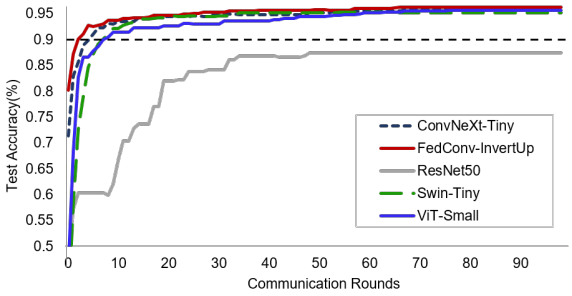

Figure 6: Test accuracy versus communication rounds conducted on the split-3 of CIFAR-10. The black dashed line is the target test accuracy. Our model shows the fastest convergence speed.

To demonstrate the effectiveness of our model, we evaluated it using three different FL methods, namely FedProx (Li et al., 2020c), FedAVG-Share (Zhao et al., 2018), and FedYogi (Reddi et al., 2020). FedProx introduces a proximal term to estimate and restrict the impact of the local model on the global model; FedAVG-Share utilizes a globally shared dataset to collect data from each client for local model updating; FedYogi incorporates the adaptive optimization technique Yogi (Zaheer et al., 2018) into the FL context.

The evaluation results, as reported in Table 6, consistently highlight the superior performance of our FedConv model across these diverse FL methods. This observation underscores FedConv's potential to enhance a wide range of heterogeneous FL methods, enabling seamless integration and suggesting its promise for further performance improvements.

4.3 Communication Cost

In FL, communication can be a major bottleneck due to the inherent complications of coordinating numerous devices. The process of client communication is often more time-consuming than local model updates, thereby emerging as a significant challenge in FL (Van Berkel, 2009). The total number of communication rounds and the size of the messages transmitted during each round are key factors in determining the communication efficiency (Li et al., 2020b). To comprehensively evaluate these aspects, we follow the methodology proposed in (Qu et al., 2022b). Specifically, we record the number of communication rounds required for different models to achieve a preset accuracy threshold. Additionally, we use Transmitted Message Size (TMS), which is calculated by multiplying the number of model parameters with the associated communication rounds, to quantify communication costs.

As shown in Figure 6, our FedConv-InvertUp achieves the fastest convergence speed among all models. In CIFAR-10 split3, where high data heterogeneity exists, FedConv-InvertUp only needs 4 communication rounds to achieve the target accuracy of 90%, while ConvNeXt necessitates 7 rounds. This efficiency also translates to a marked reduction in TMS in FL, as reported in Table 7. In contrast, ResNet struggles to converge to the 90% accuracy threshold in the CIFAR-10 split3 setting. These results demonstrate the effectiveness of our proposed FedConv architecture in reducing communication costs and improving the overall FL performance.

Table 7: Comparison based on TMS. TMS is calculated by multiplying the number of model parameters with the communication rounds needed to attain the target accuracy. We note our FedConv requires the lowest TMS to reach the target accuracy.

Model	ResNet50	Swin-Tiny	ViT-Small	ConvNext-Tiny	FedConv-InvertUp
TMS	∞	27.5M×10	21.7M×11	27.8M×8	25.6M×5

5 Conclusion

In this paper, we conduct a comprehensive investigation of several architectural components in ViT and integrated beneficial elements into CNNs to enhance their performance in heterogeneous FL. Our findings show that these straightforward modifications can substantially improve model robustness to Non-IID data. Moreover, by integrating these modifications together, we are able to build a pure CNN architecture that can outperform ViT in heterogeneous FL scenarios. Our extensive experimentation across diverse FL benchmarks verifies the effectiveness of our approach. We hope that our proposed FedConv can provide a stronger baseline for future research in the field of FL.

References

Yutong Bai, Jieru Mei, Alan Yuille, and Cihang Xie. Are transformers more robust than CNNs? In *NeurIPS*, 2021.

Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. *arXiv preprint arXiv:2103.14586*, 2021.

Christopher Briggs, Zhong Fan, and Peter Andras. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In *IJCNN*, 2020.

Andrew Brock, Soham De, and Samuel L Smith. Characterizing signal propagation to close the performance gap in unnormalized resnets. In *ICLR*, 2021a.

Andy Brock, Soham De, Samuel L Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. In *ICML*, 2021b.

Samuel Rota Bulo, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. In *CVPR*, 2018.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). In *ICLR*, 2016.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical data augmentation with no separate search. In *NeurIPS*, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.

Zhixu Du, Jingwei Sun, Ang Li, Pin-Yu Chen, Jianyi Zhang, Hai Li, Yiran Chen, et al. Rethinking normalization methods in federated learning. *arXiv preprint arXiv:2210.03277*, 2022.

Claude Dugas, Yoshua Bengio, François B'elisle, Claude Nadeau, and Ren'e Garcia. Incorporating second-order functional knowledge for better option pricing. In *NeurIPS*, 2000.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.

Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *ICCV*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *ICCV*, 2019.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861*, 2017.

Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. The non-iid data quagmire of decentralized machine learning. In *ICML*, 2020.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated visual classification with real-world data distribution. In *ECCV*, 2020.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *ICML*, 2020.

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

Fan Lai, Yinwei Dai, Xiangfeng Zhu, Harsha V Madhyastha, and Mosharaf Chowdhury. Fedscale: Benchmarking model and system performance of federated learning. In *ResilientFL*, 2021.

Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

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

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020b.

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

Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*, 2021.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017.

Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. In *AAAI*, 2022.

Liangqiong Qu, Niranjan Balachandar, Miao Zhang, and Daniel Rubin. Handling data heterogeneity with generative replay in collaborative learning for medical imaging. *Medical Image Analysis*, 78:102424, 2022a.

Liangqiong Qu, Yuyin Zhou, Paul Pu Liang, Yingda Xia, Feifei Wang, Ehsan Adeli, Li Fei-Fei, and Daniel Rubin. Rethinking architecture design for tackling data heterogeneity in federated learning. In *CVPR*, 2022b.

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.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Saurabh Singh and Abhinav Shrivastava. Evalnorm: Estimating batch normalization statistics for evaluation. In *ICCV*, 2019.

Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. In *ICLR Workshop*, 2016a.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016b.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021a.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *ICCV*, 2021b.

Asher Trockman and J Zico Kolter. Patches are all you need? *arXiv preprint arXiv:2201.09792*, 2022.

CH Van Berkel. Multi-core for mobile phones. In *DATE*, 2009.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *CVPR*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li. Optimizing federated learning on non-iid data with reinforcement learning. In *INFOCOM*, 2020a.

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *ICLR*, 2020b.

Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021.

Tete Xiao, Piotr Dollar, Mannat Singh, Eric Mintun, Trevor Darrell, and Ross Girshick. Early convolutions help transformers see better. *NeurIPS*, 2021.

Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V Le. Smooth adversarial training. *arXiv preprint arXiv:2006.14536*, 2020.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

Rui Yan, Liangqiong Qu, Qingyue Wei, Shih-Cheng Huang, Liyue Shen, Daniel Rubin, Lei Xing, and Yuyin Zhou. Label-efficient self-supervised federated learning for tackling data heterogeneity in medical imaging. *IEEE Transactions on Medical Imaging*, 2023.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.

Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. In *NeurIPS*, 2018.

Chongzhi Zhang, Mingyuan Zhang, Shanghang Zhang, Daisheng Jin, Qiang Zhou, Zhongang Cai, Haiyu Zhao, Xianglong Liu, and Ziwei Liu. Delving deep into the generalization of vision transformers under distribution shifts. In *CVPR*, 2022.

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

Wei Zhang, Xiang Li, Hui Ma, Zhong Luo, and Xu Li. Federated learning for machinery fault diagnosis with dynamic validation and self-supervision. *Knowledge-Based Systems*, 213:106679, 2021.

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.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, 2020.

Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *ICML*, 2021.

A Analysis of FedBN

We experiment with FedBN(Li et al., 2021), a popular algorithm that also tackles data heterogeneity from the perspective of model architecture. Its key idea is to not average BN layers in FedAVG. We apply this method on the regular ResNet-50, and a ConvNeXt-Tiny model with its normalization layer changed from LN-C to BN. As shown in Table 8, in split 1 and 2, where data heterogeneity is not so severe, FedBN achieves close performance compared to FedAVG. However, in a more extreme heterogeneous scenario like CIFAR-10 split3, the performance of both ResNet and ConvNeXt-BN trained by FedBN drops sharply. Specifically, from split1 to split 3, ResNet and ConvNeXt-BN shows a drop of 14.60% (96.42% to 81.82%), and 24.12% (97.99% to 73.87%), respectively. By contrast, the original ConvNeXt that chooses LN-C as its normalization layer, shows only a performance drop of 2.35% (98.20% to 95.85%). Also, our proposed FedConv achieves the best accuracy of 96.26% in split3, demonstrating the effectiveness of the normalization-free design.

Table 8: Performance comparison on CIFAR-10 dataset. '±' indicates the range of accuracy between clients.

Method	Model	Split1	Split2	Split3
FedBN	ResNet50	96.42±0.18	93.15±1.35	81.82±1.52
	ConvNeXt-BN	97.99±0.05	95.76±0.82	73.87±12.57
FedAVG	ResNet50	96.69±0.00	95.56±0.00	87.43 ±0.00
	ConvNeXt	98.20±0.00	97.67±0.00	95.85±0.00
	FedConv-InvertUp	98.11±0.00	97.74±0.00	96.26±0.00

B Implementation Details

B.1 Pre-training Recipe

Following Liu et al. (2022), we pre-train our model for 300 epochs. The learning rate is set to 4e-3 with linear warmup for 20 epochs and cosine decay schedule in subsequent epochs. AdamW (Loshchilov & Hutter, 2019) is adopted with weight decay set to 0.05. For data augmentation, we adopt Mixup (Zhang et al., 2018), CutMix (Yun et al., 2019), RandAugment (Cubuk et al., 2020) and Random Erasing (Zhong et al., 2020). For regularization, we adopt Stochastic Depth(Huang et al., 2016) and Label Smoothing (Szegedy et al., 2016b). Layer Scale (Touvron et al., 2021b) of initial value 1e-6 is applied. All layer weights are initialized using truncated normal distribution.

B.2 Fine-tuning Recipe

CIFAR-10. Following Qu et al. (2022b), for CNN structures and ViT, a SGD optimizer without weight decay is adopted. For Swin-Transformer, a AdamW with a weight decay of 0.05 is adopted. The learning rate is set to 0.03 for CNNs and ViT, and 3.125e-5 for Swin-Transformer. We use 5-epoch linear warmup and cosine learning rate decay. Stochastic Depth rate is applied with value set to 0.1, and gradient clipping is set to 1 for all models. For FL settings, we train models for 100 communication rounds, and we choose all 5 clients in each round.

iNaturalist. We follow the training settings in CIFAR-10. For FL settings, we train models for 2000 communication rounds, and choose 25 clients in each round.

B.3 FL methods

In FedProx, μ is set to 5e-5 for Swin, and 5e-4 for other models. We share 5% images from each client in FedAVG-Share. In FedYogi, we set β_1, β_2 to 0.9 and 0.99 following Reddi et al. (2020), client learning rate η to 0.01, and adaptivity τ to 5e-2 for ResNet50, 4e-3 for other models.