A Appendix

A.1 Experiment settings

Comparing with baselines We choose MobileNetV3-small ((Howard et al., 2019)) for Cifar-10, base model of NASBench201 ((Dong & Yang, 2019)) for FEMNIST, and modified smaller ResNet18 ((He et al., 2016)) for Speech Command and OpenImage as initial models. The detailed architecture of the base model of NasBench201 is shown in Figure 14. The detailed architecture of the modified small ResNet18 is shown in Figure 15.

To fairly evaluate the performance across different methods, HeteroFL, SplitMix, and FLuID should use the same architecture as FedTrans. However, HeteroFL, Splitmix, and FLuID shrink models, which means they take a large model and adopt some algorithm to reduce, compress, or prune it to form multiple small models. Therefore, we give the largest model transformed by FedTrans as the input large model to HeteroFL, SplitMix, and FLuID. Since HeteroFL and SplitMix do not support convolutional layer with groups, we convert the grouped convolution layer to non-grouped one, which potentially increases the complexity of the layer.

The hyperparameter setting for FedTrans is shown in Table 7. The training is considered complete when either the maximum number of training rounds is reached or the validation accuracy converges, which is defined as the accuracy not improving by more than 1% over 10 consecutive rounds. The hyperparameter settings for HeteroFL, SplitMix, and FLuID are the same as those in their paper.

Quality of transformed models To evaluate the quality of transformed models (Fig. 9), we fine-tune each transformed model on all the clients. We use the default FedAvg (McMahan et al., 2017) setting for this evaluation part, which means we remove the hardware capacity constraints and disable the transformation, adaptive model assignment, and soft aggregation.

B Computation and communication overheads analysis

Due to the challenge of data heterogeneity and the nature of distributed computing, FL training itself is expensive. Therefore, FedTrans introduces minimal computation and communication overhead compared with standard FedAvg.

Clients The local training on the client is the same as FedAvg, with no computation overhead. After the local training, clients are required to upload the model weights, model gradient, and training loss back to the coordinator. However, the updated model weights can be easily derived from the model gradient and the model weights of the last round. Therefore, only the training loss is considered as communication overhead for clients. Overall, on the side of clients, there is no computational overhead and negligible (i.e.. a floating number) communication overhead.

Overhead	Estimated value
client's computation	0
client's communication	rpc
coordinator's computation	$r(mn+1)c+\|W\|c$
coordinator's communication	0

Table 5. Computation and communication overheads analysis for m registered clients, p participated clients, n models, r rounds, where c is a small constant and $|W|$ is the average size of the model weights.

Method	Avg. (s)	Std. (s)
FedTrans + FedAvg	134.5	237.1
FedAvg	226.3	325.6

Table 6. Round completion time comparison.

Coordinator After receiving the updates from clients, the coordinator is scheduled to do four steps of computation, which are (1) updating utilities, (2) updating local weights, (3) updating the degree of convergence (DoC), and (4) model transformation. Among these steps, updating utilities, updating the degree of convergence, and model transformation are computational overhead. Given m clients and n models, the coordinator needs to do $m \times n$ times of utility updating operations. For each utility update, the coordinator needs to calculate the standardized loss and the subtraction, which are considered to have constant complexity. Updating DoC calculates the average of loss slopes, which is considered to have constant complexity. We consider the model transformation happens at constant times. For each model transformation, the coordinator calculates the layer activeness and applies the widening and/or deepening operations, whose complexity is considered to be proportional to the size of model weights. As for communication, FedTrans does not introduce any overhead on the side of the coordinator. Overall, the computational and communication overhead analysis is summarized in Table 5.

C FedTrans mitigates the straggler issue.

In synchronous federated learning, slow clients could slow down the training process if clients are given the same workload, which is referred to as the straggler issue. FedTrans can mitigate the straggler issue as we assume each client has a hard requirement for the model complexity (MACs). As shown in Table 6, FedTrans improves FedAvg both in the average and the std of the round completion time among clients on FEMNIST dataset compared with FedAvg.

Hyperparameters	Cifar-10	FEMNIST	Speech Command	OpenImage
# of participants per round	10	100	100	100
maximum number of training rounds	1000	2000	1500	2000
step size to calculate the loss slope (δ)	20	30	100	50
local training steps		20		
batch size		10		
learning rate		0.05		
decay factor		0.98		
# of consecutive gradient to calculate activeness (T)		5		

Table 7. Hyperparameters

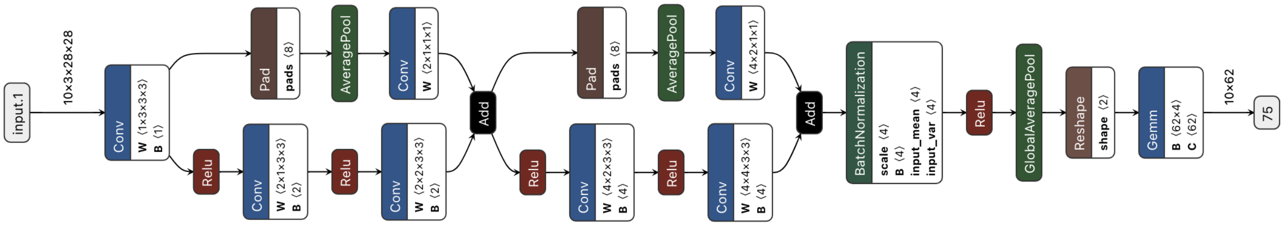

Figure 14. Base model of NASBench201

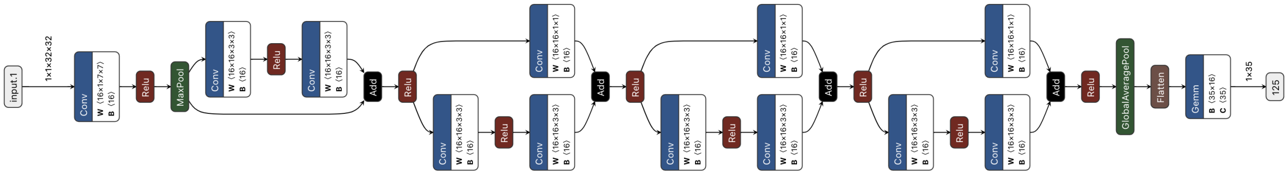

Figure 15. Modified smaller ResNet18