Making Local Models Learn Autonomously With Global Feature Tracking and Client Drift Releasing for Federated Learning

Silong Chen[1], Yuchuan Luo[1(✉)], Liang Gao[1], Shaojing Fu[1], Ming Xu[1], and Yawei Zhao[2]

[1] College of Computer Science and Technology, National University of Defense Technology
{chensilong, luoyuchuan09, gaoliang13}@nudt.edu.cn
[2] Chinese PLA General Hospital

Abstract. Data heterogeneity presents a significant challenge in federated learning, leading to inconsistent optimization of local models. Specifically, the client drift resulting from this heterogeneity significantly undermines model performance. To mitigate client model drift, existing approaches either **align client and server models** or **utilize partial variance reduction**. However, these methods compel the convergence direction on the client side to align with the global model, thereby **restricting local model training from fully capturing the knowledge of the local dataset.** Additionally, **they underutilize global information, potentially causing local overfitting**. To address this problem, we propose FedTR, a novel algorithm incorporating global feature tracking and client-released strategies to **empower local models to learn autonomously without constraints**. Meanwhile, **shared global feature centroids effectively prevent local models from overfitting**. Experimental results on five real medical datasets demonstrate the significant advantages of our algorithm over existing methods in non-IID data settings, improving both local model accuracy and convergence rate. Specifically, the FedTR algorithm enhances average accuracy by **3%** ∼ **5%**. We provide evidence of the convergence rate of our algorithm.

Keywords: Federated Learning · Data heterogeneity · Client Drift.

1 Introduction

Federated learning (FL) has emerged as a promising approach to training machine learning models across decentralized devices while preserving data privacy. Clients can reap optimization benefits through extensive data without compromising their local private information. The popular FL algorithm, FedAvg [16], has demonstrated remarkable performance in many situations where data distributions across clients are independent and identically distributed (IID). However, for instance, personalized healthcare and medical expertise of different individuals,

hospitals, and countries contribute to heterogeneous (non-IID) data, in which FedAvg converges slowly or not at all [14]. As it is proved, the poor performance of FedAvg results from the parameter drift caused by data heterogeneity [7]. Due to the difference in data distribution, there is a fundamental contradiction between minimizing the local and global empirical loss. Hence, in a highly heterogeneous environment, FedAvg lacks a convergence guarantee [6].

To remedy this, various methods have been proposed to improve the performance of federated learning in the case of non-IID settings. The majority of them focus on mitigating the client drift by penalizing the distance between local and global models [11,3], or by performing variance reduction techniques while updating client models [1,7,18]. For example, FedProx [13] adds a proximal regularization term to control divergence during local updates. The endeavors mentioned above restrict the updating direction of local models, necessitating their alignment with the global model to cater to all clients. Nevertheless, local users who emerged as healthcare stakeholders increasingly desire to obtain tailored models that meet their specific needs. **A practical FL framework should consider the client drift and flexibly accommodate local objectives during joint training.** Furthermore, [3] note that in the non-IID setting, the deeper layers, especially the final classification layer, exhibit more significant variability, which is highly task-correlated across clients compared to IID scenarios. The success of deep learning in centralized systems also demonstrates that the feature extractor plays the role of a typical structure. At the same time, the classifier tends to be highly task-correlated [2,4,21]. Hence, combining the tracking of global feature representation with client drift release is crucial to harness the benefits of federated learning fully.

Given the abovementioned results, we proposed FedTR, a novel federated learning algorithm that empowers local models to learn autonomously with global features tracking and client drift releasing. Specifically, we decompose the neural network model into the feature-extractor and classification layers. By utilizing global feature representation derived from the aggregation of each local model, we can effectively regularize the local training and mitigate the overfitting issue caused by global information underutilization. Meanwhile, by releasing client drift, we empower the local models to learn the parameter drift from the global instead of restricting the updating direction, which achieves fast convergence and personalization simultaneously. We conduct experiments on five real-world healthcare datasets (OrganAMNIST, OrganCMINIST, OrganSMINIST [22], PAMAP2 [17], COVIDx [20]). The evaluation results indicate that our proposed algorithm can enhance the accuracy of local models by **3%** $\sim$ **5%** with fast convergence. In summary, our contributions are as follows:

- We propose a novel multi-institutional collaborative federated learning algorithm (FedTR) to address the non-IID data problem and learn adequately without overfitting, facilitating accuracy and rapid convergence of the global model.
- FedTR possesses a comprehensive feature extraction capability and local adaptive classification. The former enhances the capacity of local models

by leveraging global representations, to aggregate a comprehensive feature extractor. Simultaneously, the latter guarantees adequate local learning without overfitting.
– Our experimental findings conclusively demonstrate that our algorithm outperforms state-of-the-art methods on one widely-used benchmark dataset and five real-world healthcare datasets under various non-IID settings. Moreover, the results indicate that FedTR improves accuracy and accelerates convergence, reducing communication costs.

2 Related Work

Federated learning is experiencing rapid growth. This section introduces algorithms designed for non-IID settings, where data distributions are heterogeneous between clients. FedAvg [16], among various techniques, stands out as a practical method that successes under IID data conditions while exhibiting a significant decrease in both accuracy and efficiency in non-IID scenarios [3].

The study by [14] highlights a significant challenge in achieving convergence stemming from the drift between servers and clients. Various attempts have been made to reduce the variance of client updates and address this issue. However, employing a unified model across clients containing non-IID distributed data often minimizes the empirical risk function, hindering convergence towards a satisfactory global model. FedProx [13] introduces an approximation term to the local training target to maintain updated parameters closely aligned with the original downloaded model. Nonetheless, this approximation fails to align global and local optimal solutions. To further explore the relationship between client drift and data heterogeneity, some studies employ statistical variables without considering local target personalization. For instance, SCAFFOLD [7] introduces control variables to correct for drift in local updates. Based on it, FedPVR [3] also makes further efforts to explore the effectiveness of partial variance reduction. FedDyn [1] proposes a dynamic regularizer for each device, making global and local solutions consistent and saving transmission costs. FedDC [6] builds on a SCAFFOLD for more granular tracking by employing local drift decoupling as well as the technique of correction to address the challenges posed by non-IID data, improving the robustness and generalization performance of federated learning models. Other approaches focus on reducing communication overhead by compressing the gradient of transmission.

Overall, these improved methods, which are superior to FedAvg in convergence speed and performance, contribute to advancing the field of federated learning by proposing innovative techniques to optimize model performance, communication efficiency, and privacy preservation in decentralized learning scenarios. However, a practical FL framework should consider the client drift and flexibly accommodate local objectives during joint training. The actual needs of local users like healthcare stakeholders, who are increasingly desirous of obtaining tailored models that meet their specific needs, should be considered. Moreover, these methods may fail to leverage the full benefits of the global information brought about by federated learning.

This paper introduces FedTR, a method that separates local and global models through the tracking of feature representations and addressing local drift. The methods mentioned earlier are compatible with our approach and can be easily integrated.

3 Overview of Proposed Framework

The proposed federated learning algorithm, FedTR, is specifically designed for multi-institutional collaborations within highly regulated data domains. In this section, we present the objectives and provide an overview of our approach to training models by tracking the representation of global features and releasing client drift. The detailed design of the algorithm will be presented in the next section.

3.1 Problem Formulation

We contemplate a scenario with a central server and N clients actively participating in federated learning. Each client collaboratively trains a local model without divulging local data. The data distribution $X \times Y$ in client i is denoted as D_i, where X and Y represent the domains of inputs and labels, encompassing M categories. Consequently, the global dataset $D = \bigcup_{i \in [N]} D_i$ and each pair of D_i and D_j with $i \neq j$ is distinct. The model parameter for client i is w_i, and $f_{w_i}(x)$ signifies the predicted result using the model with input x sampled from D_i. $L(f_w(x))$ denotes the loss function. Hence, we formula the optimization problem for federated learning as follows:

$$\min_{W} \{L(W) = \sum_{i=1}^{N} \frac{|D_i|}{|D|} E_{x,y \sim D_i}[L_i(f_{w_i}(x), y)]\} \tag{1}$$

Here, $W = (w_1, w_2, ..., w_N)$ comprises all local models, and we can get the optimal global model $w^* = \arg\min_{W} L(W)$. For instance, in FedAvg, the server takes the expectation of local models' parameters that are updated by clients after each training round as follws:

$$w = \sum_{i=1}^{N} \frac{|D_i|}{|D|} w_i \tag{2}$$

In fact, every client can not share its raw data with others, and the optimization is usually achieved by empirical risk minimization(ERM). This means we can only calculate the best local model, but the combination of local optimality does not represent the global, generally speaking.

3.2 Releasing Client Drift

As discussed earlier, achieving local optimality does not necessarily ensure global optimality. A discrepancy, called the client drift, exists between each client's model trained on its local dataset, and the global model trained on the entire dataset. The slow and occasionally unstable convergence observed in FedAvg can be attributed to client drift [7,15]. Previous studies have attempted to mitigate client drift by penalizing the disparity between a client model and the server model [18], thereby ensuring consistency between local and global models. However, this approach often results in a model that overlooks inconsistencies between local and global objectives. While it aids in reducing gradient drift, it fails to address the gradual amplification of parameter deviation. Based on this observation, we propose releasing client drift, making local models learn the parameter drift from the global instead of restrict the updating direction. The challenge is how to evaluate the similarity between local models and the global.

To this end, we contend that releasing client drift in local models for improved performance. We define $d_i = w - w_i$ as the client drift, where w_i is the parameter of client i's local model and w is the parameter of the global model. We need to prevent this drift from getting out of our control. For client i, we further convert this restriction as a regularization term as

$$G_i(w_i; d_i) = \left\| d_i + w_i^{(k)} - w \right\|^2 \tag{3}$$

where $w_i^{(k)}$ is the model parameter at local update step k and $w_i^+ = w^{(local-epoch)}$. The role of term G_i is to enable clients to learn as freely as possible without experiencing gradient runaway. Each client incorporates this regularization term along with its empirical loss term on the respective dataset to optimize both the model parameters and local drift variables.

3.3 Tracking of Global Features

For making local models learn autonomously, in the realm of federated learning with non-IID (non-independent and identically distributed) data, the risk of overfitting arises when the local datasets on each device deviate significantly from one another or from the global dataset. To mitigate this, one of the reasonable approaches is to optimize the utilization of the shared model parameters.

Federated learning can be seen as indirectly sharing local knowledge by sharing model parameters. In this paper, we follow FedRep[4], dividing a standard Convolutional Neural Network (CNN) model W into two segments: the feature extraction layer W^s containing general feature domain information and the classification layer W^p containing client-specific task information. The output of W^s that is the last fully connected layer in the backbone serves as the input of W^p. We propose the tracking of global features to prevent local model w_i from overfitting. We achieve this by integrating the global expectation feature representation into the penalty term, allowing for indirect utilization of the global

data. The penalty term is given by

$$R_i(w_i^s; c) = \sum_{j=1}^{|D_i|} ||f_{w_i^s}(x_j) - c_{y_j}||^2 \tag{4}$$

where c_{y_j} is the global feature centroid of class y_j. This penalty term is advantageous for incentivizing each client to fully leverage general feature domain information bought by the shared model parameters and constraint local feature representation divergence while minimizing local classification loss.

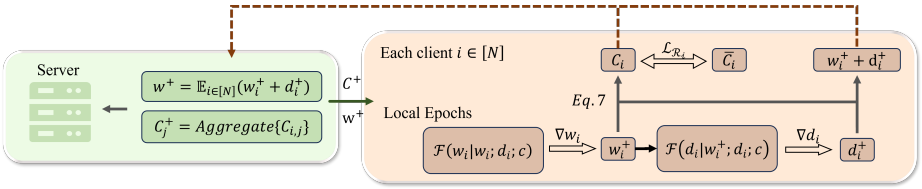


Fig. 1: An illustration of the training procedure. In each round, the local model parameters w_i and the variable d_i are iteratively updated on the client side for each client (Local Epochs) first. Then, the clients update their local centroids and send them to the central server. The central server generates global centroids and aggregates the models, returning them to all clients.

4 Method

In this section, we propose an optimization approach aimed at iteratively learning the global feature extractor by tracking features and expediting convergence through the release of client drift. Besides, in Section 4.3, we introduce the theoretical analyses of our algorithm. FedTR mainly contains two steps: the local training procedure and global aggregation. The entire learning progress is shown in Algorithm 1, and figure 1 illustrates a brief of the learning procedure. Locally updated model parameters and feature statistics will be transmitted to the central server for aggregation.

4.1 Local Training Procedure

In FedTR, the loss function of each client comprises three components: the local empirical loss, the penalty term, and a regularization term. Specifically, for client i, the learning objective is to minimize the following loss function:

$$\mathcal{F}(w_i; d_i; c) = \mathcal{L}(w_i) + R_i(w_i^s; c) + G_i(w_i; d_i) \tag{5}$$

where $\mathcal{L}(w_i)$, $R_i(w_i^s; c)$ and $G_i(w_i; d_i)$ respectively denote as the local empirical loss function, the penalty term and the regularization term.

Algorithm 1 The Learning Progress in FedTR

Input: N clients with their local data; initial model f with parameter w; local learning rate η_f and η_g;
Output: Client models $\{f_i\}_{i=1}^{N}$ with parameters w_i

for each round $r = 1, ..., R$ **do**
  sample clients $S \subseteq \{1, ..., N\}$
  communicate $\{w, C\}$ to all client $i \in S$
  on client $i \in S$ **in parallel do**
    initialize local model $w_i \leftarrow w$
    initialize local centroids $c_i \leftarrow C_i$
    for $t = 1, ..., Epoch$ **do**
      ξ_i denotes the mini-batch of data
      $g_i \leftarrow \nabla_w[\mathcal{L}(w_i^t; \xi_i) + R_i(w_i^{s,t}; c) + G_i(w_i; d_i)]$
      $w_i^t \leftarrow w_i^t - \eta_f g_i$
      $d_i \leftarrow d_i - \eta_g \nabla_d[G_i(w_i; d_i)]$
    end for
    update c_i^+ with equation 7
    communicate $\{w_i^+ + d_i^+, c_i^+\}$ to the server
  end on client
  Server aggregate $\{w^+, c^+\}$ as equation 8, 9
end for
return $\{w_1, ..., w_N\}$

Update the Local Model Parameters. At each round, we first replace the local feature extractor w_i^s with the received new global model parameter w^s and update the private classifier analogously. Then, we conduct stochastic gradient decent steps to train the model parameters as follows:

$$w_i \leftarrow w_i - \eta \nabla_w \mathcal{F}(w_i; d_i; c) \tag{6}$$

where η is the learning rate. This procedure is executed *Epoch* times in each round.

Update the Local Feature Centroids. After updating the local model parameters, each client should compute the local feature centroid for each class as fellow:

$$c_{i,j}^+ = \frac{\sum_{k=1}^{|D_i|}(y_k == j) f_{w_i^s}(x_k)}{\sum_{k=1}^{|D_i|}(y_k == j)}, \quad \forall j \in [M] \tag{7}$$

Correct the Local Model Parameters. Before communicating parameters, clients should correct their local model parameters using the local drifts as $(w_i^+ + d_i^+)$.

4.2 Global Aggregation

Aggregate Global Feature Extractor. FedTR, akin to prevalent algorithms like FedAvg as shown in Equation 2, employs weighted averaging of local repre-

sentation layers, with each coefficient determined by the local data size.

$$w^+ = \sum_{i=1}^{N} \lambda_i (w_i^+ + d_i^+), \quad \lambda_i = \frac{|D_i|}{|D|} \tag{8}$$

Aggregate Global Feature Centroids. Following receiving local feature representation centroids, the subsequent progress computes a new estimated global feature centroid C for all classes.

$$C_j^+ = \frac{1}{\sum_{i=1}^{N} n_{i,j}} \sum_{i=1}^{N} n_{i,j} c_{i,j}^+, \quad n_{i,j} = \sum_{k=1}^{|D_i|} (y_k == j) \tag{9}$$

4.3 Convergence Analysis

We provide insights into the theoretical analysis for improving federated learning from both the feature extractor and classifier perspectives. The local objective function is defined in Eq. 5 as $\mathcal{F}(w_i; d_i; c)$. We utilize a subscript to indicate the number of iterations as $\mathcal{F}_i$ and make the assumptions similar to existing general frameworks[19]. The detailed proofs are provided in Appendix A.

Convergence of Releasing Client Drift.

- For non-convex and β_1-Lipschitz smooth local empirical loss function $\mathcal{F}_i, \forall i \in [N]$, there exists a $\beta_d > 0$, where $\nabla^2 \mathcal{F}_i \geq -\beta_d I$.
- For non-convex and B-dissimilarity local empirical loss function $\mathcal{F}_i$, where $B(\theta^t) \leq B$, the global empirical loss of FedTR decreases as fellow:

$$\mathbb{E}[\mathcal{F}_i(w^t)] \leq \mathcal{F}_i(w^{t-1}) - 2p||\nabla \mathcal{F}_i(w^{t-1})||^2 \tag{10}$$

where $p = (\frac{\gamma}{\alpha} - \frac{B(1+\gamma)\sqrt{2}}{\bar{\alpha}\sqrt{N}} - \frac{\beta B(1+\gamma)}{\alpha\bar{\alpha}} - \frac{\beta(1+\gamma)^2 B^2}{2\bar{\alpha}^2} - \frac{\beta B^2(1+\gamma)^2(2\sqrt{2C}+2)}{\bar{\alpha}^2 N}) > 0$. Appendix A.1 provides more details of the convergence guarantee.

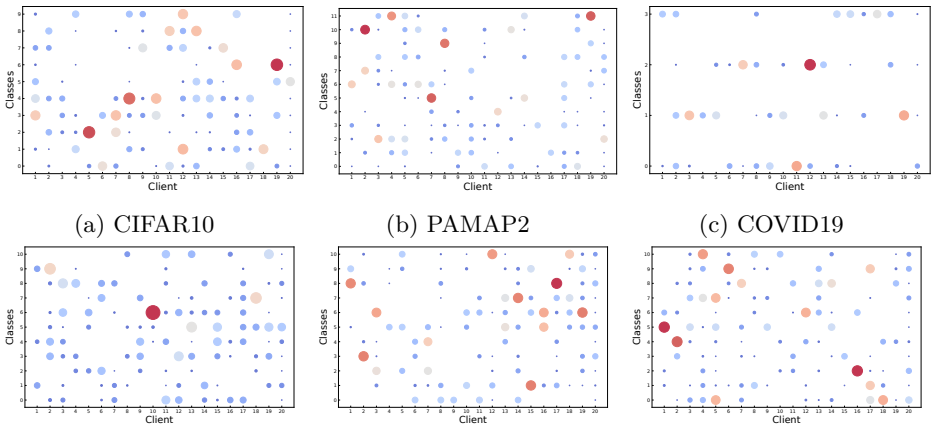
Reduce Testing Loss by Feature Alignment. Inspired by [21], benefiting from FL, the penalty term 4 can be further estimated by local empirical data D_i and global feature representations c as follows,

$$\Delta R_i = \frac{2}{|D_i|} \sum_{j=1}^{|D_i|} [c_{y_j} - f_{w^s}(x_j)]^T f_{w^p}(y_j)/M \tag{11}$$

where c_{y_j} is the global feature representation of class y_j estimated by Eq. 9. The inconsistency of local-global feature representations causes this term. Therefore, we denote it as an extra penalty term during the local learning phase.

5 Experiments

This section focuses on image classification tasks and assesses our method's performance on one widely used simulated dataset, CIFAR-10 [8], and five



(a) CIFAR10 (b) PAMAP2 (c) COVID19

(d) OrganAMNIST (e) OrganCMNIST (f) OrganSMNIST

Fig. 2: The distribution of five datasets. The size of the scatters denotes the number of samples allocated to each client.

healthcare datasets regarding convergence speed and model accuracy. Additionally, we compare FedTR with several state-of-the-art (SOTA) methods across different settings and datasets mentioned earlier. The evaluation primarily focuses on model accuracy and convergence speed.

5.1 Datasets and Models

We conducted experiments on real-world datasets for image classification tasks, including CIFAR-10, PAMAP2, COVID-19, OrganAMNIST, OrganCMNIST, and OrganSMNIST. We explored various model types: COVID-Net for COVID-19, a CNN architecture for CIFAR-10, PAMAP2-Net for PAMAP2, and LeNet-5 for OrganXMNIST. To establish the problem scenario in FedTR, we partition the dataset using a Dirichlet Distribution, as described in [23]. Figures 2a-2f visually depict the distribution of samples for CIFAR-10, COVID-19, PAMAP2, OrganAMNIST, OrganCMNIST, and OrganSMNIST, respectively.

5.2 Compared Methods

We compare our method with the following six methods, including standard FL methods and FL methods designed for non-IID data sectors:

FedAvg [16] learns a single global model by aggregating clients' models with no extra optimization. **FedProx** [13] is the first variant of FedAvg. The main difference from FedAvg is that FedProx adds a quadratic proximal term to limit the local model updates explicitly. We set the coefficient of the proximal term to 0.01. **SCAFFOLD** [9] is another important variant of FedAvg. The key difference from FedAvg is that each client keeps a variate to control the local

Table 1: The comparison of top-1 test accuracy (%) on six datasets with 20 clients training for 200 rounds on the 0.1-Dirichlet (D1) non-IID setting, and the result on 0.2-Dirichlet (D2) (**the lower alpha is, the more heterogeneous data is**) non-IID setting is presented in the appendix A.2. **The arrow** (↑, ↓) besides shows the comparison to the FedAvg.

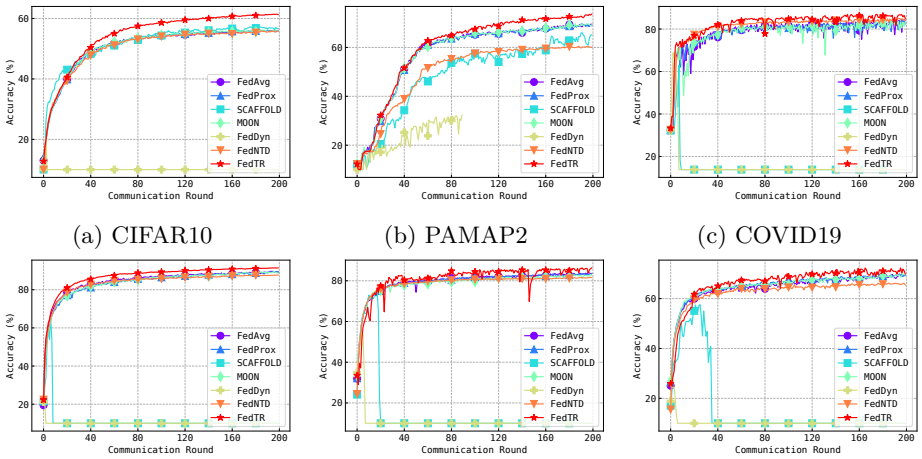
Algorithm	CIFAR10 $\alpha = 0.1$	PAMAP2 $\alpha = 0.1$	COVID-19 $\alpha = 0.1$	OrganA. $\alpha = 0.1$	OrganC. $\alpha = 0.1$	OrganS. $\alpha = 0.1$
FedAvg	56.10 -	69.47 -	84.24 -	89.63 -	84.04 -	69.68 -
FedDyn	20.23 ↓	33.32 ↓	69.00 ↓	21.22 ↓	54.68 ↓	25.02 ↓
FedProx	56.04 ↓	69.63 ↑	84.50 ↑	89.55 ↑	83.47 ↓	70.00 ↑
MOON	56.23 ↑	69.71 ↑	83.77 ↓	89.44 ↓	83.15 ↓	69.97 ↑
SCAFFOLD	57.28 ↑	65.98 ↓	63.03 ↓	65.58 ↓	74.25 ↓	57.57 ↓
FedNTD	56.93 ↑	60.51 ↓	84.94 ↑	87.68 ↓	81.66 ↓	66.25 ↓
FedTR	**61.41** ↑	**73.59** ↑	**86.98** ↑	**91.44** ↑	**86.31** ↑	**73.32** ↑

model updates in Scaffold. However, the control variable's size is equal to that of the full model, which is prohibitively inefficient for the learning tasks with large-scale full models [5]. **FedDyn** [1] align the client models using a dynamic regularizer. **MOON** [12] is a simple and effective federated learning framework. Its key idea is to utilize the similarity between model representations to correct the local training of individual parties, i.e., conduct contrastive learning at the model level. **FedNTD** [10] preserves the global perspective on locally available data only for the not-true classes.

5.3 Results and Analysis

We conducted extensive experiments on real healthcare datasets, with Table 1 presenting the primary classification results, including the robustness across varying levels of data heterogeneity. Highlighted numbers denote the advantages of FedTR over existing Federated Learning (FL) optimization baselines. Our main findings are as follows: 1) FedTR achieves superior accuracy compared to other methods, 2) empirical results confirm the robustness of FedTR across different levels of data heterogeneity, and 3) The FedAvg algorithm remains competitive on diverse datasets without the requirement for fine-tuning any additional hyper-parameters, demonstrating its adaptability. All results are obtained by running the experiments on the NVIDIA 4090 five times and averaging the outcomes.

Better accuracy of FedTR. Table 1 summarizes the best accuracy achieved by FedTR and the baseline methods across evaluation datasets with various settings. The red and blue arrows indicate the better and worse performance on accuracy. Meanwhile, Figure 3 presents the evolution of average test accuracy over global communication rounds for selected experiments on 0.1-Dirichlet non-IID setting detailed in Table 1. It is evident from these results that our method consistently outperforms other baselines throughout the entire training process.



(a) CIFAR10 (b) PAMAP2 (c) COVID19

(d) OrganAMNIST (e) OrganCMNIST (f) OrganSMNIST

Fig. 3: The test accuracy of FedTR and the other baselines on five datasets with 0.1-Dirichlet non-IID setting. The truncated portion in the middle suggests that the algorithm experiences gradient explosion or vanishing during this communication.

Notably, FedAvg's accuracy curves closely align with others, demonstrating competitive performance comparable to state-of-the-art methods with a high level of data heterogeneity on various datasets. This underscores FedAvg's simplicity and effectiveness as a formidable approach.

Effects of Data Heterogeneity. We vary the values of α to simulate different levels of data heterogeneity, while $\alpha = 0.1$ indicates a highly heterogeneous case (pathological non-IID) and $\alpha = 0.5$ means data across clients are more homogeneous. The effect of our evaluation on the PAMAP2 dataset and the results of different methods are also reported in Appendix B. It can be found that our method consistently outperforms other baselines, which demonstrates its adaptability and robustness in a variety of heterogeneous data scenarios.

Fast Convergence of FedTR. We first report the number of rounds required to achieve a certain level of Top 1% accuracy (55% for CIFAR-10) in Appendix B. An algorithm is more efficient in communication if it requires fewer rounds to achieve the same accuracy and transmits fewer parameters between the clients and server. Compared to the baseline approaches, we require much fewer rounds for almost all types of data heterogeneity and models.

6 Conclusion

This paper proposes a novel Federated Learning (FL) algorithm named FedTR, which introduces global feature tracking to address the overfitting issue caused by limited local data and mitigate client drift resulting from heterogeneous data,

dynamically bridging the gap between local and global models. By leveraging this approach, we maximize the benefits of federated learning to develop both personalized and universal models with enhanced performance. Additionally, we provide theoretical and empirical justifications for their effectiveness in heterogeneous settings.

In future research, we aim to explore more optimal aggregation for aligning local feature extractors, such as utilizing adaptive procedures. Moreover, investigating the application of these optimizations to overly parameterized models like YOLOvX may also prove significant for the effective application of such optimizations.

Acknowledgements. This work was partially supported by the National Natural Science Foundation of China No.62472431 and No.62302522. and partially by the Natural Science Foundation of Hunan Province, China under Grant No.2021JJ40688.

References

1. Acar, D.A.E., Zhao, Y., Navarro, R.M., Mattina, M., Whatmough, P.N., Saligrama, V.: Federated learning based on dynamic regularization. arXiv preprint arXiv:2111.04263 (2021)
2. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence **35**(8), 1798–1828 (2013)
3. Bo, L., Schmidt, M.N., Alstrøm, T.S., Stich, S.U.: On the effectiveness of partial variance reduction in federated learning with heterogeneous data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3964–3973 (2023)
4. Collins, L., Hassani, H., Mokhtari, A., Shakkottai, S.: Exploiting shared representations for personalized federated learning. In: International conference on machine learning. pp. 2089–2099. PMLR (2021)
5. Ding, Y., Niu, C., Wu, F., Tang, S., Lyu, C., Chen, G., et al.: Federated submodel optimization for hot and cold data features. Advances in Neural Information Processing Systems **35**, 1–13 (2022)
6. Gao, L., Fu, H., Li, L., Chen, Y., Xu, M., Xu, C.Z.: Feddc: Federated learning with non-iid data via local drift decoupling and correction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10112–10121 (2022)
7. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: Scaffold: Stochastic controlled averaging for federated learning. In: International conference on machine learning. pp. 5132–5143. PMLR (2020)
8. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Handbook of Systemic Autoimmune Diseases **1**(4) (2009)
9. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
10. Lee, G., Jeong, M., Shin, Y., Bae, S., Yun, S.Y.: Preservation of the global knowledge by not-true distillation in federated learning. Advances in Neural Information Processing Systems **35**, 38461–38474 (2022)

11. Li, Q., He, B., Song, D.: Model-contrastive federated learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10713–10722 (2021)
12. Li, Q., He, B., Song, D.: Model-contrastive federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10713–10722 (June 2021)
13. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Proceedings of Machine learning and systems **2**, 429–450 (2020)
14. Li, X., Huang, K., Yang, W., Wang, S., Zhang, Z.: On the convergence of fedavg on non-iid data. arXiv preprint arXiv:1907.02189 (2019)
15. Liang, P.P., Liu, T., Ziyin, L., Allen, N.B., Auerbach, R.P., Brent, D., Salakhutdinov, R., Morency, L.P.: Think locally, act globally: Federated learning with local and global representations. arXiv preprint arXiv:2001.01523 (2020)
16. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017)
17. Reiss, A., Stricker, D.: Introducing a new benchmarked dataset for activity monitoring. In: 2012 16th international symposium on wearable computers. pp. 108–109. IEEE (2012)
18. Shamir, O., Srebro, N., Zhang, T.: Communication-efficient distributed optimization using an approximate newton-type method. In: International conference on machine learning. pp. 1000–1008. PMLR (2014)
19. Tan, Y., Long, G., Liu, L., Zhou, T., Lu, Q., Jiang, J., Zhang, C.: Fedproto: Federated prototype learning across heterogeneous clients. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 8432–8440 (2022)
20. Wang, L., Lin, Z.Q., Wong, A.: Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. Scientific reports **10**(1), 19549 (2020)
21. Xu, J., Tong, X., Huang, S.L.: Personalized federated learning with feature alignment and classifier collaboration. arXiv preprint arXiv:2306.11867 (2023)
22. Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., Ni, B.: Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. Scientific Data **10**(1), 41 (2023)
23. Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K., Hoang, N., Khazaeni, Y.: Bayesian nonparametric federated learning of neural networks. In: International conference on machine learning. pp. 7252–7261. PMLR (2019)

A Convergence Proof of FedTR

In this section, we present detailed proof of convergence for FedTR. The local objective function is defined in Eq. 5 as $\mathcal{F}(w_i; d_i; c)$. We utilize a subscript to indicate the number of iterations as $\mathcal{F}_i$ and make the following assumptions similar to existing general frameworks [19].

Assumption 1. (Lipschitz Smooth). *Each local objective function is* β_1*-Lipschitz smooth, which means that the gradient of the local objective function is* β_1*-Lipschitz smooth continuous*,

$$||\nabla \mathcal{F}_{t_1} - \nabla \mathcal{F}_{t_2}||_2 \leq \beta_1 ||w_{i,t_1} - w_{i,t_2}||_2, \forall t_1, t_2 > 0, i \in \{1, 2, ..., N\}. \tag{12}$$

that also implies a quadratic upper bound,

$$\mathcal{F}_{t_2} \leq \mathcal{F}_{t_1} + \langle \nabla \mathcal{F}_{t_1}, w_{i,t_2} - w_{i,t_1} \rangle + \frac{\beta_1}{2} + ||w_{i,t_2} - w_{i,t_1}||^2 \tag{13}$$

Assumption 2. (μ-convex function). *Each local objective function is a μ-convex function, which means*,

$$\langle \nabla \mathcal{F}_{t_1}, w_{i,t_2} - w_{i,t_1} \rangle \leq \mathcal{F}_{t_2} - \mathcal{F}_{t_1} - \frac{\mu}{2} + ||w_{i,t_2} - w_{i,t_1}||^2, \forall \mu > 0, i \in \{1, 2, ..., N\}. \tag{14}$$

Assumption 3. (γ-inexact solution). *We define function $F_i(w_i, \hat{w}_i)$ as $F_i(w_i, \hat{w}_i) = \mathcal{F}_{t_1} + \frac{\alpha}{2}||w_i - \hat{w}_i||^2$, where $\alpha \in [0, 1]$ and $\hat{w}_i = w - d_i$, we get the gradient of F_i: $\nabla F_i(w_i, \hat{w}_i) = \nabla L_i(w_i) + \alpha(w_i - \hat{w}_i)$. If w_i^* is a γ-inexact point of $\min F_i(w_i, \hat{w}_i)$, it satisfies $||\nabla F_i(w_i^*, \hat{w}_i)|| \leq \gamma ||\nabla F_i(\hat{w}_i, \hat{w}_i)||$.*

Assumption 4. (Bounded dissimilarity assumption for $\mathcal{F}(w_i; d_i; c)$). *There exists a B_ϵ while $\epsilon > 0$, for any w, that satisfies $||\nabla \mathcal{F}(w_i; d_i; c)||^2 > \epsilon$, and $B(w) > B_\epsilon$.*

A.1 Convergence of FedTR in non-convex case

In the proof, we follow the techniques of [13], assume the local empirical loss $\mathcal{F}(w_i; d_i; c)$ is γ-inexactness solver. We define e_i^t as

$$\nabla \mathcal{F}_i(w_i^t) + \alpha(w_i^t - \hat{w}_i^{t-1}) - e_i^t = 0. \tag{15}$$

In addition, we have $\nabla G_i(\hat{w}_i^{t-1}, \hat{w}_i^{t-1}) = \nabla \mathcal{F}_i(\hat{w}_i^{t-1})$, so with the B-local dissimilarity bounded assumption we can get: $||\nabla G_i(w_i^t, \hat{w}_i^{t-1})|| \leq ||\nabla G_i(w_i^*, \hat{w}_i^{t-1})|| \leq \gamma ||\nabla G_i(\hat{w}_i^{t-1}, \hat{w}_i^{t-1})||$, that implies

$$||e_i^t|| \leq \gamma ||\nabla \mathcal{F}_i(\hat{w}_i^{t-1})||. \tag{16}$$

As $\bar{w}^t = \mathbb{E}_i w_i^t = \mathbb{E}_i \hat{w}_i^t$, so that we get the following equation

$$\bar{w}^t - \bar{w}^{t-1} = \mathbb{E}_i[w_i^t - \hat{w}_i^{t-1}] = \frac{1}{\alpha} \mathbb{E}_i(-\nabla \mathcal{F}_i(w_i^t) + e_i^t). \tag{17}$$

Let $\bar{\alpha} = \alpha - \mathcal{F}_d > 0$ and $\ddot{w}_i^t = \arg\min_w G_i(w, \hat{w}_i^{t-1})$. Due to that G_i is $\bar{\alpha}$ strong convex function, we get

$$||\ddot{w}_i^t - w_i^t|| \leq \frac{\gamma}{\bar{\alpha}} ||\nabla \mathcal{F}_i(\hat{w}_i^{t-1})||. \tag{18}$$

With the strong convex nature of G_i again, we get

$$||\ddot{w}_i^t - \hat{w}_i^{t-1}|| \leq \frac{1}{\bar{\alpha}} ||\nabla \mathcal{F}_i(\hat{w}_i^{t-1})||. \tag{19}$$

Using triangle inequality for 18 and 19, we get:

$$||w_i^t - \hat{w}_i^{t-1}|| \leq \frac{1 + \gamma}{\bar{\alpha}} ||\nabla \mathcal{F}_i(\hat{w}_i^{t-1})||. \tag{20}$$

With the bounded dissimilarity assumption and $\mathbb{E}[\hat{w}_i^{t-1}] = \mathbb{E}[\hat{w}^{t-1}] = \bar{w}_i^{t-1}$, we get

$$||\bar{w}^t - \bar{w}^{t-1}|| \leq \mathbb{E}_i||w_i^t - \hat{w}_i^{t-1}|| \tag{21}$$

$$\leq \frac{1+\gamma}{\bar{\alpha}}\mathbb{E}_i||\nabla\mathcal{F}_i(\hat{w}_i^{t-1})|| \tag{22}$$

$$\leq \frac{1+\gamma}{\bar{\alpha}}\sqrt{\mathbb{E}_i||\nabla\mathcal{F}_i(\hat{w}_i^{t-1})||^2} \tag{23}$$

$$\leq B\frac{1+\gamma}{\bar{\alpha}}||\nabla\mathcal{F}(\bar{w}^{t-1})||, \tag{24}$$

where the last inequality is due to the bounded dissimilarity assumption and $G(w) = \mathcal{F}(w)$, $\nabla\nabla_w G(w) = \nabla_w\mathcal{F}_i(w_i)$.

We define M_t as $\bar{w}^t - \hat{w}^{t-1} = -\frac{1}{\alpha}(\nabla\mathcal{F}(w^{t-1}) + M_t))$. Taking Eq. 17 into it, we get $M_t = \mathbb{E}_i[(\nabla\mathcal{F}_i(w_i^t) - \nabla\mathcal{F}_i(\hat{w}_i^{t-1}) - e_i^t]$. M_t is bounded with

$$||M_t|| \leq \mathbb{E}_i[(\beta||w^t - \hat{w}^{t-1}|| + ||e_i^t||] \tag{25}$$

$$\leq (\frac{\beta(1+\gamma)}{\bar{\alpha}} + \gamma)\mathbb{E}_i||\nabla\mathcal{F}_i(\hat{w}^{t-1})|| \tag{26}$$

$$\leq (\frac{\beta(1+\gamma)}{\bar{\alpha}} + \gamma)B||\nabla\mathcal{F}(w^{t-1})||, \tag{27}$$

The last is due to $\nabla_w\mathcal{F}(w) = \nabla_w G(w) = \nabla_w\mathcal{F}_i(w)$ and the bounded dissimilarity assumption.

Because $h_i^t = h_i^{t-1} + \Delta w_i^t$, we get $\mathbb{E}(w_i^t - w_i^{t-1}) = \mathbb{E}(h_i^t - h_i^{t-1}) = \mathbb{E}(\bar{w}^t - \ddot{w}^{t-1})$, and $w^t - w^{t-1} = \mathbb{E}(w_i^t + h_i^t) - \mathbb{E}(w_i^{t-1} + h_i^{t-1}) = 2\mathbb{E}(w_i^t - w_i^{t-1})$.

With β-Lipschitz smoothness assumption of $\mathcal{F}$ and Taylor expansion, we get

$$\mathcal{F}(w^t) \leq \mathcal{F}(w^{t-1}) + < \nabla\mathcal{F}(w^{t-1}), w^t - w^{t-1} > + \frac{\beta}{2}||w^t - w^{t-1}||^2 \tag{28}$$

$$\leq_1 \mathcal{F}(w^{t-1}) + < \nabla\mathcal{F}(w^{t-1}), 2\mathbb{E}_i(w_i^t - w_i^{t-1}) > + \frac{\beta}{2}||2\mathbb{E}(w_i^t - w_i^{t-1})||^2 \tag{29}$$

$$\leq_2 \mathcal{F}(w^{t-1}) - \frac{2}{\alpha}||\nabla\mathcal{F}(w^{t-1})||^2 - \frac{2}{\alpha} < \nabla\mathcal{F}(w^{t-1}), M_t > + \frac{2\beta B^2(1+\gamma)^2}{\hat{\alpha}^2}||\nabla\mathcal{F}(w^{t-1})||^2 \tag{30}$$

$$\leq \mathcal{F}(w^{t-1}) - (\frac{2 - 2\gamma B}{\alpha} - \frac{2\beta B(1+\gamma)}{\alpha\bar{\alpha}} - 2\beta\frac{B^2(1+\gamma)^2}{\hat{\alpha}^2})||\nabla\mathcal{F}(w^{t-1})||^2, \tag{31}$$

where $(\leq_1)$ is due to $w^t - w^{t-1} = 2\mathbb{E}(w_i^t - w_i^{t-1})$, $(\leq_2)$ is due to the definition of M. Set a proper α for the above inequality, $\mathcal{F}(w^t) - \mathcal{F}(w^{t-1})$ is decrease proportional to $||\nabla\mathcal{F}(w^{t-1})||^2$. The above inequality demonstrates that the works would be decreased if the hyper-parameter α of the penalized term is large enough.

In summary, For non-convex and β_1-Lipschitz smooth function $\mathcal{F}_i$, $\forall i \in [N]$, there exist a $\beta_d > 0$, where $\bar{\alpha} = \alpha - \beta_d > 0$ and $\nabla^2\mathcal{F}_i \geq -\beta_d I$. We assume the local empirical loss $\mathcal{F}_i$ is non-convex and B-dissimilarity, in which $B(w^t) \leq B$. The global objective of FedTR decreases as follows:

$$\mathbb{E}[\mathcal{F}_i(w^t)] \leq \mathcal{F}_i(w^{t-1}) - 2p||\nabla\mathcal{F}_i(w^{t-1})||^2, \tag{32}$$

where $p = (\frac{\gamma}{\alpha} - \frac{B(1+\gamma)\sqrt{2}}{\tilde{\alpha}\sqrt{N}} - \frac{\beta B(1+\gamma)}{\alpha\tilde{\alpha}} - \frac{\beta(1+\gamma)^2B^2}{2\tilde{\alpha}^2} - \frac{\beta B^2(1+\gamma)^2(2\sqrt{2C}+2)}{\tilde{\alpha}^2 N}) > 0$.

B The Additional Experimental Results

Table 2: The first table shows the comparison of top-1 test accuracy (%) on six datasets with 20 clients training for 200 rounds on the 0.5-Dirichlet (D2) non-IID setting. The last two tables present the number of communication rounds in different methods to achieve a target accuracy on **D1** and **D2** non-IID settings, respectively.

Algorithm	CIFAR10 $\alpha = 0.5$	PAMAP2 $\alpha = 0.5$	COVID-19 $\alpha = 0.5$	OrganA. $\alpha = 0.5$	OrganC. $\alpha = 0.5$	OrganS. $\alpha = 0.5$
FedAvg	79.78 -	78.54 -	93.15 -	95.05 -	95.08 -	89.61 -
FedDyn	23.58 ↓	82.23 ↑	**97.55** ↑	34.27 ↓	74.48 ↓	69.55 ↓
FedProx	79.75 ↓	78.76 ↑	94.75 ↑	98.06 ↑	96.16 ↑	89.69 ↑
MOON	79.71 ↓	78.73 ↑	94.75 ↑	98.18 ↑	96.17 ↑	89.62 ↑
SCAFFOLD	87.10 ↑	78.09 ↓	79.59 ↓	99.24 ↑	97.16 ↑	94.73 ↑
FedNTD	79.85 ↑	78.16 ↓	95.21 ↑	98.09 ↑	95.64 ↑	89.69 ↑
FedTR	**88.42** ↑	**88.52** ↑	96.10 ↑	**98.61** ↑	**98.18** ↑	**95.99** ↑

Algorithm	CIFAR10 55%	PAMAP2 60%	COVID-19 80%	OrganA. 85%	OrganC. 80%	OrganS. 65%
FedAvg	133	54	54	60	55	71
FedDyn	>200	>200	>200	>200	>200	>200
FedProx	115	55	54	71	58	59
MOON	114	59	54	78	74	58
SCAFFOLD	96	143	>200	>200	>200	>200
FedNTD	131	176	32	68	69	100
FedTR	59	40	30	36	28	36

Algorithm	CIFAR10 78%	PAMAP2 78%	COVID-19 94%	OrganA. 98%	OrganC. 95%	OrganS. 89%
FedAvg	111	183	>200	157	138	161
FedDyn	>200	159	63	>200	>200	>200
FedProx	108	183	164	184	104	163
MOON	109	182	164	159	112	162
SCAFFOLD	54	197	>200	45	44	48
FedNTD	101	198	142	176	137	165
FedTR	25	150	63	37	34	40