Taming Heterogeneity to Deal with Test-Time Shift in Federated Learning

Anonymous Author(s)*

ABSTRACT

Federated learning (FL) is an effective machine learning paradigm where multiple clients can train models based on heterogeneous data in a decentralized manner without accessing their private data. However, existing FL systems undergo performance deterioration due to feature-level test-time shifts, which are well investigated in centralized settings but rarely studied in FL. The common non-IID issue in FL usually refers to inter-client heterogeneity during training phase, while the test-time shift refers to the intra-client heterogeneity during test phase. To explore the possibility of using inter-client heterogeneity in handling intra-client heterogeneity, we firstly propose a contrastive learning-based FL framework, namely FedICON, to capture invariant knowledge among heterogeneous clients and consistently tune the model to adapt to test data. Extensive experiments validate the effectiveness of the proposed FedICON in taming heterogeneity to handle test-time shift problems.

ACM Reference Format:

Anonymous Author(s). 2018. Taming Heterogeneity to Deal with Test-Time Shift in Federated Learning. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Federated learning (FL) is a promising machine learning paradigm that enables multiple clients or devices to collaboratively train models without sacrificing their data privacy [19, 42]. Practical FL frameworks usually suffer from notorious non-IID (independent and identically distributed) local data distributions, which may lead to severe performance degradation during the test phase [25, 48]. To improve the performance of the trained model on local data distribution, a variety of personalized federated learning (PFL) methods are proposed to help the model adapt to local datasets [37]. In existing PFL systems, it is commonly assumed that the training and test datasets within a client share the same or similar distribution. However, this assumption often does not hold in real-world applications where test samples may encounter natural variations or corruptions. Take federated chest X-ray classification [31] among multiple hospitals as an example, while most existing studies acknowledge that different hospitals can have varying data distributions due to diverse patient populations (e.g., races, ages, etc.), they overlook the possibility of variation within a single hospital. For instance, there may be multiple X-ray machines from different vendors, each with its own unique internal parameters (e.g., exposure time, spectrum, etc.). Similar issues can be found in various other domains, such as autonomous vehicles and video surveillance. Consequently, an efficient and effective solution to handle test-time shifts is necessary to achieve satisfactory performance.

The extensively studied heterogeneous distribution in FL, also known as non-IID issue, mainly refers to the *inter-client heterogeneity* where clients own heterogeneous training data during their federated training phase [19, 26, 48]. Differently, the aforementioned test-time distribution shift during the inference stage of PFL refers to the heterogeneity within a client, which can be defined as *intra-client heterogeneity*. As illustrated in Fig. 1(a), the test data of a client can differ from its training data in various aspects. In this case, a natural question that arises is: *what is the difference between the test-time shift in centralized settings and that in FL?* In this paper, we pinpoint that heterogeneous sources can also convey a wealth of useful information that potentially helps alleviate test-time shift problems in FL systems. A benefit of this heterogeneity is that it enables the global model to capture robust features that are less sensitive to both inter-client and intra-client shifts. We firstly discover that this information has the potential to help alleviate test-time shift problems in FL systems, as it enables the global model to capture robust features that are less sensitive to both inter-client and intra-client shifts. Through our motivated experiments (Fig. 1(b)), we observe that FL under *inter-client heterogeneity* can improve the ability of clients to handle test-time shift problems compared to training on homogeneous data. In this way, we turn the problem of inter-client heterogeneity as a blessing in solving test-time shift.

Motivated by the above findings, we introduce **Fed**erated Learning via **I**nvarian**C**e Extracti**O**n and Shari**N**g (FedICON), a contrastive learning-based FL approach that enables clients to learn the robust features on heterogeneous data. Our theme is to fully take advantage of the benefit brought by *inter-client heterogeneity* during training time, and then, at the test phase, tune the local model of each client on its private test data to conquer the *intra-client heterogeneity*. During the training phase, clients first carry out local representation learning under the guidance of label information so that invariant information underlying each training sample, as well as the class it belongs to, is extracted. At the global aggregation stage of federated training, the parameters of feature encoders in all clients are aggregated to further boost the ability of clients to learn invariant information among inter-client heterogeneity. Then, during the test phase, we introduce self-supervision signals to the unlabeled test samples and smoothly tune the feature encoder based on the acquired knowledge in identifying invariant properties between diverse training data and shifted test data. Notably, we leverage contrastive learning paradigms for both federated training and local test-time

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

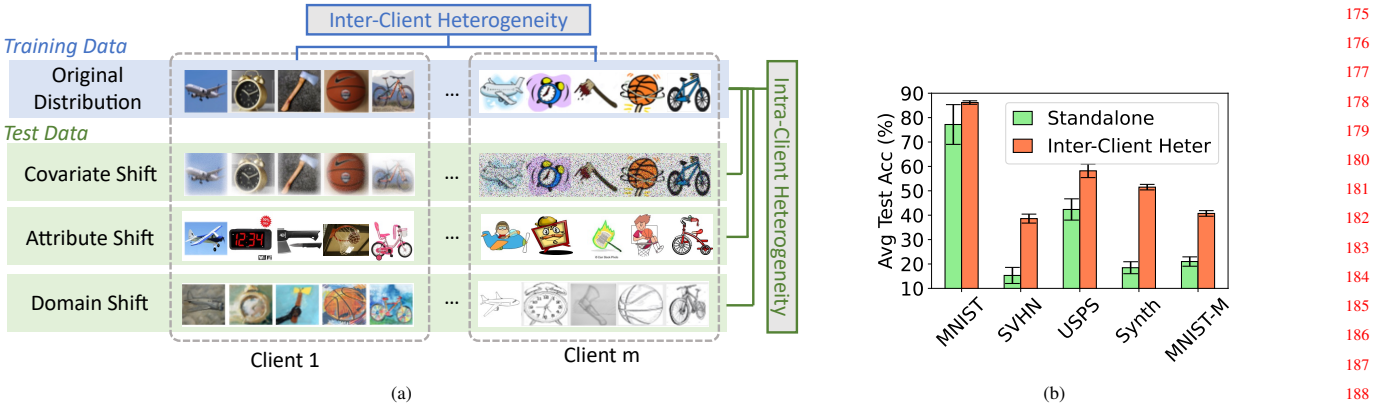


Figure 1: (a) Illustration of our setting where both *inter-client heterogeneity* and *intra-client heterogeneity* exist in an FL framework. (b) Compared with training on data from a single domain, the improved robustness to test-time shift is achieved when training on heterogeneous data sources, indicating that the inherent *inter-client heterogeneity* in FL has a positive effect in handling test-time shift problems. Detailed settings can be found in Appendix A.1.

adaptation, enabling the trained model to well adapt to test data distribution.

2 NOTATIONS AND PRELIMINARIES

2.1 Notations

Considering m clients in an FL system, P_i and Q_i are training and test data distributions of the i-th client ($i \in \{1, \cdots, m\}$) respectively and $\mathcal{D}_i^P$ and $\mathcal{D}_i^Q$ are the corresponding training dataset and test dataset. A sample-label pair is denoted as $(\mathbf{x}, y)$ where $\mathbf{x}$ is the raw input data sample and y is its corresponding label.

2.2 Inter-Client Heterogeneity

For arbitrary client i and client j in an FL system, the *inter-client heterogeneity* refers to varying P_i and P_j with respect to their raw data sample $\mathbf{x}$ and/or label y. For example, in the cases where $P_i(y) \neq P_j(y)$ while $P_i(\mathbf{x}|y) = P_j(\mathbf{x}|y)$, there exists an inter-client label distribution shift. The case where $P_i(\mathbf{x}) \neq P_j(\mathbf{x})$ while $P_i(y|\mathbf{x}) = P_j(y|\mathbf{x})$ can lead to inter-client feature distribution shift, which is common in real-world FL systems.

2.3 Intra-Client Heterogeneity

Most existing works on FL consider the heterogeneity of training datasets $\mathcal{D}_i^P$ across clients while overlooking the intra-client heterogeneity $P_i \neq Q_i$. [17] examines test distribution in robust FL benchmarks, combining label-level and feature-level shifts. To better understand their distinct impacts, we propose to study these shifts separately. Here, we focus on varying P_i and Q_i with respect to $\mathbf{x}$ rather than y, i.e., addressing the feature-level test-time shift problem in FL. Incorporating real-world scenarios, we thoroughly examine inter-client feature distribution shifts, including covariate shift, attribute shift, and domain shift. We assume that all clients have access to their own *labelled* training data when participating in the federated training process and test the performance of their personally tuned model on their own test set.

Challenges. It is a challenging task to deal with both inter-client and intra-client heterogeneity in an FL system. Although there is useful information underlying heterogeneous clients, how to effectively extract it and how to further leverage it to benefit the test procedure remain unexplored. Hence, there is a need for feasible solutions to tame inherent inter-heterogeneity to handle the test-time intra-client heterogeneity in FL.

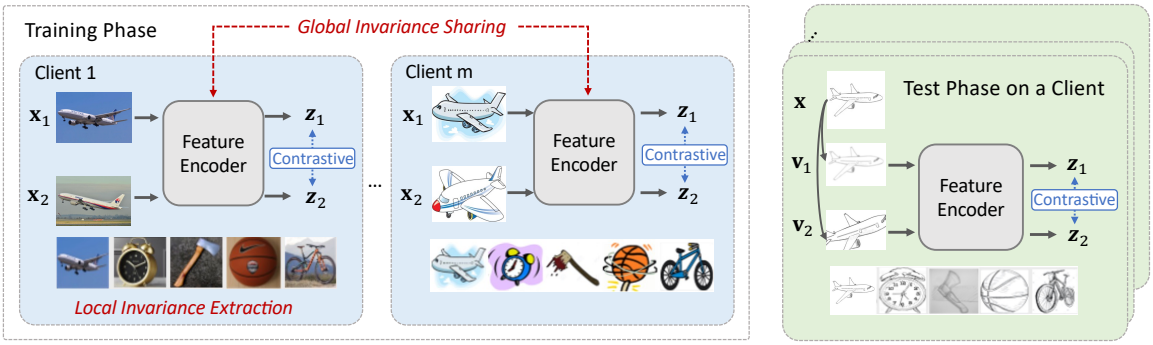
3 FEDERATED LEARNING VIA INVARIANCE EXTRACTION AND SHARING

3.1 Training under Inter-Client Heterogeneity.

Compared with the test-time shift problem in centralized settings, there are inherent heterogeneous source data in the federated learning framework, which is of great potential to utilize to deal with the shifted test data. Although the inter-client heterogeneous data are not directly accessible due to privacy concerns, their underlying invariant information can be captured by each client during local invariance extraction procedure and then globally shared across all clients to further enhance the invariance extraction ability.

Local Invariance Extraction. To capture invariant information from samples within the same class, we design a local invariance extraction procedure based on supervised contrastive learning. As shown in Fig. 2(a), samples are first fed into a feature encoder to generate the representation for each sample. Then, a contrastive learning scheme is applied to the representations under the guidance of their corresponding labels so that the class-level invariant information can be extracted during the local training phase.

To make the representation ability of the feature encoder more robust, we add an augmentation module during the data pre-processing stage that incorporates some perturbations to raw samples. In this way, there are various views of the same sample fed into the feature encoder to strengthen its sample-based discrimination ability. Concretely, for the i-th client, given a batch of input samples, we generate two random augmented views for each sample. The total size of the augmented local dataset is $2|\mathcal{D}_i|$. For a specific sample $\mathbf{x}$, we use $A(\mathbf{x})$ to denote the local dataset without $\mathbf{x}$.



(a) Federated Training under Inter-Client Heterogeneity (b) Test-Time Adaptation

Figure 2: An overview of the proposed FedICON. (a) Federated training phase under inter-client heterogeneity. Each client conducts local invariance extraction during local training and performs parameter-wise global invariance sharing during global aggregation. (b) Test phase on a single client. Clients carry out independent test-time adaptation to generalize to their local test sets.

Taking advantage of the powerful representation ability brought by contrastive learning frameworks, we force the feature encoder to discriminate the augmented samples from different classes. Given an input sample $\mathbf{x}$, the output of the feature encoder is

$$z(\mathbf{x}) = h\left(\mathbf{x}; \theta_i\right). \tag{1}$$

where θ_i is the learnable parameter of the feature encoder. We denote $z(\mathbf{x})$ as $z_{\mathbf{x}}$ for short. Samples from the same class are forced to be closer, and samples from different classes are forced to be farther. We perform local training with the loss function as below,

$$L_{i,\text{sup}}\left(w_i; \mathcal{D}_i^P\right) = \sum_{(\mathbf{x},y)\in\mathcal{D}_i^P} \frac{-1}{|P(\mathbf{x})|} \sum_{\mathbf{p}\in P(\mathbf{x})} \log \frac{\exp\left(z_{\mathbf{x}} \cdot z_{\mathbf{p}}/\tau\right)}{\sum_{a\in A(\mathbf{x})} \exp\left(z_{\mathbf{x}} \cdot z_{\mathbf{a}}/\tau\right)}, \tag{2}$$

where $z_{\mathbf{a}}$ is the representation of sample $\mathbf{a} \in A(\mathbf{x})$, and τ is the temperature adjusting the tolerance for representation difference. $P(\mathbf{x})$ is the set of samples that have the same labels with $\mathbf{x}$ and can be represented as

$$P(\mathbf{x}) = \{\mathbf{p} \in A(\mathbf{x}) : y_{\mathbf{p}} = y_{\mathbf{x}}\}. \tag{3}$$

By training the local feature encoder in a contrastive manner, the local invariances can be directly learned from the augmented data and be encoded to the feature encoder. Even if there is inter-client heterogeneity among different clients, their invariance encoding abilities over heterogeneous data are universal, which makes it possible for clients to mutually boost their performance on invariance extraction in a federated framework.

Global Invariance Sharing. To tame the inter-client heterogeneity during the federated training process, there should be a uniform information channel that is globally shared across clients and continuously provides meaningful knowledge to boost local invariance extraction ability. Considering the fact that the feature encoder introduced above is trained to capture and encode the invariance, we can employ the parameters in the feature encoder as the information carrier and perform global invariance sharing at the server side. These invariant robust properties captured by the feature encoder are shared within different augmented views of a specific sample and different samples from the same class.

Similar to the standard federated aggregation process, for each client $i \in \{1, 2, \cdots, m\}$, the learnable parameter θ_i of its feature encoder is sent to the server. After receiving parameters from all the clients, the server aggregates them as

$$\bar{\theta} = \frac{1}{m} \sum_{i=1}^{m} \frac{|\mathcal{D}_i^P|}{N^P} \theta_i, \tag{4}$$

and sends $\bar{\theta}$ back to each client as their initialized parameter in a new round. Since the feature encoder of each client is trained to extract invariance efficiently rather than classify the samples correctly, the parameter aggregation is less likely to be affected by inter-client heterogeneity. Global invariance sharing with respect to the feature encoder incorporates the invariance encoding abilities learned upon heterogeneous source data, which reversely provides sufficient homogeneous information for each client and increases the robustness of local representation learning. To employ the model trained alternately via local invariance extraction and global invariance sharing, we further train a linear classifier on top of the fixed feature encoder [6, 20]. The linear classifier can be trained in a pure local manner or trained in a federated learning manner. In this paper, we conduct multiple federated training steps at first and then personalize the classifier for each client.

3.2 Test-Time Adaptation to Handle Intra-Client Heterogeneity.

To keep the losses of the training and test phase unified and make the whole process consistent, we adopt a self-supervised contrastive learning scheme to carry out test-time adaptation, which is also a natural and efficient way to generalize the trained model to the unlabeled and shifted test set.

Concretely, for client i, the trained feature encoder is further tuned by optimizing the following unsupervised loss function on its local test set $\mathcal{D}_i^Q$.

$$L_{i,\text{us}}\left(\theta_i; \mathcal{D}_i^Q\right) = \sum_{\mathbf{x}\in\mathcal{D}_i^Q} -\log \frac{\exp\left(z_{\mathbf{x}} \cdot z'_{\mathbf{x}}/\tau\right)}{\sum_{a\in A(\mathbf{x})} \exp\left(z_{\mathbf{x}} \cdot z_{\mathbf{a}}/\tau\right)}. \tag{5}$$

where $z'_{\mathbf{x}}$ is the representation of the augmentated sample of $\mathbf{x}$. Eq. 2 and 5 are consistently formulated and can share the same optimization strategy.

Table 1: Test accuracy on Digit-5 under *covariate* test-time shift.

Method	MNIST	SVHN	USPS	Synth	MNIST-M
Local	60.65(4.79)	34.43(1.87)	16.91(4.55)	39.93(3.25)	18.07(3.26)
FedAvg	86.18(0.72)	38.60(1.84)	58.19(2.78)	51.50(1.13)	40.72(1.15)
FedAvg-FT	85.92(1.14)	40.25(1.47)	54.28(1.90)	51.23(0.72)	41.08(1.21)
FedProx	86.45(0.83)	38.95(2.29)	57.80(2.52)	52.40(1.32)	40.55(0.48)
FedRep	66.78(6.40)	44.17(3.29)	15.81(5.84)	44.60(2.40)	19.33(3.08)
FedBN	61.42(4.50)	41.38(0.77)	41.73(7.44)	42.77(4.44)	42.35(1.13)
FedTHE	84.52(0.21)	39.00(2.21)	58.03(2.96)	51.92(1.27)	40.55(1.23)
Tent	86.13(0.56)	43.87(0.81)	67.51(2.40)	50.47(0.12)	47.73(0.94)
EATA	85.37(0.08)	43.65(1.55)	67.12(1.48)	50.70(0.22)	47.58(0.77)
T3A	72.58(2.50)	44.23(1.77)	49.44(5.63)	45.62(0.95)	46.13(1.53)
FedICON	**89.67**(1.48)	**45.23**(0.35)	**72.13**(2.22)	**56.13**(0.63)	**47.98**(0.50)

Table 2: Test accuracy on Office-10 under *domain* test-time shift.

Method	Test Domain			
	Amazon	Caltech	DSLR	Webcam
Local	24.07(0.61)	18.62(0.31)	36.46(2.76)	28.81(2.04)
FedAvg	44.91(2.99)	26.27(1.63)	61.46(3.12)	48.21(3.76)
FedAvg-FT	41.78(1.23)	24.94(1.07)	55.56(2.41)	46.52(5.72)
FedProx	42.53(1.76)	24.44(0.93)	56.60(1.20)	42.94(3.53)
FedRep	20.02(1.16)	17.68(1.34)	34.03(5.92)	24.86(1.49)
FedBN	42.53(2.18)	28.64(0.62)	51.04(5.41)	43.69(1.63)
FedTHE	41.61(1.61)	24.79(0.95)	56.94(2.17)	44.63(1.79)
Tent	38.25(1.92)	25.93(2.57)	54.17(1.04)	42.00(1.98)
EATA	38.25(2.03)	25.73(2.59)	54.86(0.60)	42.75(1.18)
T3A	39.76(0.52)	22.77(1.20)	53.82(0.60)	45.76(2.87)
FedICON	**50.46**(0.78)	**33.28**(0.67)	**67.01**(1.59)	**48.21**(2.67)

4 EXPERIMENTS

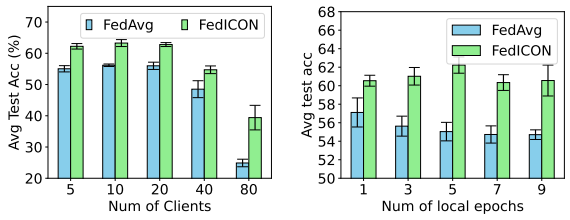
4.1 Experimental Setup

We evaluate our proposed method on the following four benchmark datasets: Digit-5 [49], Office-10 [11], DomainNet dataset [33], and CIFAR10 [21]. Digit-5, Office-10, and DomainNet dataset are three benchmark dataset consisting of multiple sub-datasets from different domains. We use them to simulate the *covariate shift* and *domain shift* during the test phase. We use CIFAR10 and its variant CIFAR10.1 [34, 39] to simulate the *attribute shift*. Details about the test-time shift settings can be found in Appendix A.2.

We first compare our proposed method with the following baselines: 1) Local, 2) FedAvg, 3) FedAvg-FT, 4) FedProx, 5) FedRep, 6) FedBN, 7) FedTHE. We also compare with three test-time adaptation (TTA) methods, including 1) Tent, 2) EATA, and 3) T3A. In practice, we implement these TTA methods with the FedAvg framework. Details about the baseline methods can be found in Appendix A.3.

4.2 Performance Comparison

Performance on various test-time shifts. Table 1 reports the test accuracy of FedICON and baselines on Digit-5 dataset under *covariate* test-time shift. The results are reported in mean(standard error) format over three independent runs. The results suggest that our proposed FedICON outperforms existing FL/PFL methods and test-time adaptation methods and shows a strong ability to deal with covariate test-time shift problem. It should be noticed that although FedRep and FedBN are two state-of-the-art solutions in personalized federated learning, they usually fail to perform well on the shifted test sets. The main reason for that is both of them have a special



(a) (b)

Figure 3: Test accuracy on Digit-5 under varying numbers of clients (*left*) and varying numbers of local epochs (*right*).

focus on local training data distribution and accordingly adjust part of their model to better fit local training data distribution, leading to a more biased model than the generic model during test phase.

Table 2 reports the test accuracy of FedICON and baselines on Office-10 dataset under *domain* test-time shift, respectively. The results are reported in mean(standard error) format over three independent runs. The results suggest that our proposed FedICON outperforms existing FL/PFL methods and test-time adaptation methods and shows a strong ability to deal with test-time shift problem. All results in Table 1 and 2 are reported in mean(std error) format over three independent runs.

Effect of varying numbers of participating clients. To test the performance of FedICON in a larger FL system, we increase the number of clients from 5 to 80 on Digit-5 dataset under *covariate* test-time shift setting. The total number of training samples is kept constant. As the number of clients becomes larger, the number of samples in each client becomes fewer. As shown in Fig. 3(a), both FedAvg and FedICON have a decreased performance when there are more clients as a result of the further divergence among participating clients. Although performance degradation is inevitable, FedICON still stably outperforms FedAvg in FL systems at different scales, demonstrating the scalability of FedICON and its ability to adapt to a large-scale FL system.

Effect of varying numbers of local epochs. To test how local epoch number affects the test performance, we vary the number of local epochs and report the results of FedAvg and FedICON on Digit-5 dataset under *covariate* test-time shift setting. As shown in Fig. 3(b), the performance of FedAvg keeps dropping as the number of local epochs increases, while FedICON partially enjoys the benefit of more local training, thus more communication-efficient than FedAvg.

5 CONCLUSION

In this paper, we aim to address the challenging but unexplored feature-level test-time shift problems in FL. We take the first initiative to thoroughly investigate the inherent *inter-client heterogeneity* in FL and propose to use it to deal with the test-time *intra-client heterogeneity*. We propose a novel algorithm, namely FedICON, where invariant information underlying heterogeneous data is extracted in a contrastive learning manner during the federated training phase and further used to deal with test-time shifts in a client. On our proposed benchmarks tailored for this problem, extensive experiments are conducted to show the superiority of FedICON compared with baseline methods.

REFERENCES

[1] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. 2019. Federated Learning with Personalization Layers. *arXiv:1912.00818* (2019).

[2] Alexander Bartler, Andre Bühler, Felix Wiewel, Mario Döbler, and Bin Yang. 2022. Mt3: Meta test-time training for self-supervised test-time adaption. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 3080–3090.

[3] Duc Bui, Kshitiz Malik, Jack Goetz, Honglei Liu, Seungwhan Moon, Anuj Kumar, and Kang G Shin. 2019. Federated user representation learning. *arXiv preprint arXiv:1909.12535* (2019).

[4] Hong-You Chen and Wei-Lun Chao. 2022. On Bridging Generic and Personalized Federated Learning for Image Classification. In *ICLR*.

[5] Junming Chen, Meirui Jiang, Qi Dou, and Qifeng Chen. 2023. Federated Domain Generalization for Image Recognition via Cross-Client Style Transfer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 361–370.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*. PMLR, 1597–1607.

[7] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. 2021. Exploiting Shared Representations for Personalized Federated Learning. In *ICML*.

[8] Xin Dong, Sai Qian Zhang, Ang Li, and HT Kung. 2022. Spherefed: Hyperspherical federated learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*. Springer, 165–184.

[9] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized federated learning: A meta-learning approach. In *NeruIPS*.

[10] Haozhe Feng, Zhaoyang You, Minghao Chen, Tianye Zhang, Minfeng Zhu, Fei Wu, Chao Wu, and Wei Chen. 2021. KD3A: Unsupervised Multi-Source Decentralized Domain Adaptation via Knowledge Distillation. In *ICML*.

[11] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*.

[12] Filip Hanzely, Slavomír Hanzely, Samuel Horváth, and Peter Richtárik. 2020. Lower bounds and optimal algorithms for personalized federated learning. *NeruIPS* 33 (2020), 2304–2315.

[13] Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*.

[14] Wenke Huang, Mang Ye, and Bo Du. 2022. Learn from others and be yourself in heterogeneous federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10143–10153.

[15] Yusuke Iwasawa and Yutaka Matsuo. 2021. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems* 34 (2021), 2427–2440.

[16] Jaehee Jang, Heoneok Ha, Dahuin Jung, and Sungroh Yoon. 2022. Fedclassavg: Local representation learning for personalized federated learning on heterogeneous neural networks. In *Proceedings of the 51st International Conference on Parallel Processing*. 1–10.

[17] Liangze Jiang and Tao Lin. 2023. Test-Time Robust Personalization for Federated Learning. (2023).

[18] Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. 2019. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488* (2019).

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

[20] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised Contrastive Learning. *NeruIPS* 33 (2020).

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

[22] Yoonho Lee, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. 2023. Surgical fine-tuning improves adaptation to distribution shifts. In *International Conference on Learning Representations*.

[23] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. 2021. Ditto: Fair and robust federated learning through personalization. In *ICML*. PMLR, 6357–6368.

[24] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *MLSys* (2020).

[25] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2019. On the Convergence of FedAvg on Non-IID Data. In *ICLR*.

[26] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. 2020. FedBN: Federated Learning on Non-IID Features via Local Batch Normalization. In *ICLR*.

[27] Paul Pu Liang, Terrance Liu, Liu Ziyin, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2020. Think Locally, Act Globally: Federated Learning with Local and Global Representations. *NeurIPS* (2020).

[28] Yuejiang Liu, Parth Kothari, Bastien Van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. 2021. TTT++: When does self-supervised test-time training fail or thrive? *Advances in Neural Information Processing Systems* 34 (2021), 21808–21820.

[29] Zhengquan Luo, Yunlong Wang, Zilei Wang, Zhenan Sun, and Tieniu Tan. 2022. Disentangled Federated Learning for Tackling Attributes Skew via Invariant Aggregation and Diversity Transferring. In *ICML*.

[30] H Brendan McMahan, Eider Moore, Daniel Ramage, et al. 2017. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*.

[31] Dinh C Nguyen, Quoc-Viet Pham, Pubudu N Pathirana, Ming Ding, Aruna Seneviratne, Zihuai Lin, Octavia Dobre, and Won-Joo Hwang. 2022. Federated learning for smart healthcare: A survey. *ACM Computing Surveys (CSUR)* 55, 3 (2022), 1–37.

[32] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. 2022. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*. PMLR, 16888–16905.

[33] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. 2019. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1406–1415.

[34] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. 2018. Do CIFAR-10 Classifiers Generalize to CIFAR-10? (2018). https://arxiv.org/abs/1806.00451.

[35] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. 2020. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*. PMLR, 9229–9248.

[36] Canh T Dinh, Nguyen Tran, and Tuan Dung Nguyen. 2020. Personalized Federated Learning with Moreau Envelopes. In *NeruIPS*.

[37] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. 2022. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[38] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. 2022. FedProto: Federated Prototype Learning across Heterogeneous Clients. In *AAAI*.

[39] Antonio Torralba, Rob Fergus, and William T. Freeman. 2008. 80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 11 (2008), 1958–1970.

[40] Bram van Berlo, Aaqib Saeed, and Tanir Ozcelebi. 2020. Towards federated unsupervised representation learning. In *Proceedings of the third ACM international workshop on edge systems, analytics and networking*. 31–36.

[41] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. 2021. Tent: Fully Test-Time Adaptation by Entropy Minimization. In *International Conference on Learning Representations*.

[42] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM TIST* (2019).

[43] Honglin Yuan, Warren Richard Morningstar, Lin Ning, and Karan Singhal. 2022. What Do We Mean by Generalization in Federated Learning?. In *International Conference on Learning Representations*.

[44] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. 2021. A survey on federated learning. *Knowledge-Based Systems* 216 (2021), 106775.

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

[46] Jie Zhang, Song Guo, Xiaosong Ma, Haozhao Wang, Wenchao Xu, and Feijie Wu. 2021. Parameterized Knowledge Transfer for Personalized Federated Learning. *NeruIPS* 34 (2021).

[47] Marvin Zhang, Sergey Levine, and Chelsea Finn. 2022. MEMO: Test time robustness via adaptation and augmentation. *Advances in Neural Information Processing Systems* 35 (2022), 38629–38642.

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

[49] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. 2020. Learning to generate novel domains for domain generalization. In *ECCV*.

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

[51] Weiming Zhuang, Yonggang Wen, and Shuai Zhang. 2020. Divergence-aware Federated Self-Supervised Learning. In *International Conference on Learning Representations*.

A EXPERIMENTAL DETAILS

A.1 Detailed Setting of Figure 1(b)

Experiments are implemented with FedAvg [30] on Digit-5 dataset [49] under feature shift non-IID setting [26]. The number of communication rounds is 100. The total number of clients is 4.

There are five subdatasets in Digit-5, *i.e.*, MNIST, SVHN, USPS, SynthDigits, and MNIST-M. For the Standalone case, clients own training samples from the same dataset of Digit-5, while for the Inter-Client Heterogeneity case, each client owns a specific dataset of Digit-5 as its training set. As for the test set, both of these two cases have to deal with test-time *domain* shifts where the test samples are from an unseen dataset. We follow the leave-one-out principle to select one of the five datasets as the test set in turn. The results report the mean accuracy across all experiments, and each experiment has three independent runs.

A two-layer CNN architecture is used as the backbone model here. The local training batch size is 32. An SGD optimizer with a weight decay rate 5e-4 and a learning rate 0.01 is used. The number of local epochs is 5.

A.2 Detailed Setting of Test-Time Shifts

Covariate shift. The number of clients is 5, 4, and 6 for Digit-5, Office-10, and DomainNet, respectively. Compared with the feature-level distribution of the training set, we add covariate shift to the test set following [13, 17]. There is a hyperparameter ranging within $\{1, 2, 3, 4, 5\}$ that controls the severity level of corruption. We set the hyperparameter as the highest 5 for all covariate shift experiments.

Attribute shift. We followed the natural shift setting in [17]. There are 20 clients in total. Clients own the training set sampled from CIFAR10 and the test set sampled from CIFAR10.1 which is an attribute-shifted dataset of CIFAR10. The training set and test set of a specific client have the same label distribution, while the datasets across different clients are of various label distributions simulated by Dirichlet distribution. Similar to [17], we consider the hyperparameter α of Dirichlet distribution as 0.1 and 1, respectively.

Domain shift. The number of clients is 4, 3, and 5 for Digit-5, Office-10, and DomainNet, respectively. We follow the leave-one-out principle to select one domain as the test domain for all clients and the other domains as the training domains. Each client owns training data from a unique domain that is different from others.

A.3 Detailed Setting of Baselines

The detailed setting and the hyperparameters of all the baselines are listed below.

- For **FedAvg** [30], the learning rate is 0.01, and the weight decay rate is 5e-4. The number of local epochs is 5. The number of communication rounds is 100 for Digit-5, Office-10, CIFAR10, and 20 for DomainNet.
- For **Local**, the hyperparameters are the same as **FedAvg** except that each client only performs local training without any communication.
- For **FedAvg-FT**, there is an additional local training round to update the global model based on the local training dataset.
- For **FedProx** [24], we tune the penalty constant μ from the limited candidate set $\{0.001, 0.01, 0.1, 1\}$ given in the original paper and select 0.01 for all settings.
- For **FedRep** [7], similar to the original paper, we use the last fully-connected layer as the local head and the other part of the model as the global representation layer.
- For **FedBN** [26], the batch normalization layers are kept locally trained while the other part of the model is globally shared. For simple CNN model architecture, we add batch normalization layers after the convolutional layers.
- For **FedTHE** [17], since there is no publicly available code at present, we follow the algorithm procedure claimed in the original paper and reproduce its framework using its default hyperparameters.
- For **Tent** [41], the step of test adaptation is set to be 1.
- For **EATA** [32], the number of samples used to compute the fisher information matrix and the trade-off between entropy and regularization loss are all set as the default value 2000. The entropy margin is set to be $0.4\log(1000)$ and ϵ is set to be 0.05.
- For **T3A** [15], the size of support set is selected from $\{1, 5, 20, 50, 100\}$ and 50 is selected as the value for all settings.

A.4 Detailed Setting of Model Architectures

For Digit-5, Office-10, and CIFAR10, we use a simple two-layer CNN as the backbone model, while for DomainNet dataset, we use ResNet18 pretrained on ImageNet1K as the backbone model.

The detailed architecture of the simple CNN is shown in Table 3.

Table 3: The two-layer simple CNN model architecture. FC refers to the fully connected layer.

Layer	Details
1	Conv2D(3, 32, 5, 1), ReLU, MaxPool2D(2, 2)
2	Conv2D(32, 64, 5, 1), ReLU, MaxPool2D(2, 2)
3	FC(64 * 5 * 5, 64), ReLU
4	FC(64, 10)

B ADDITIONAL EXPERIMENTS

Table 4 summarizes the results of test accuracy on CIFAR10 dataset under *attribute* test-time shift. We follow the setting of the naturally shifted test in [17]. The results in Table 4 indicate that our proposed FedICON is an effective solution to extract key invariant implicit information underlying both the training and test set, even though there are explicitly different properties conveyed by the training and test set. We also report how the performance varies on the original test set and the attribute test-time shift scenarios. The results show that FedICON has the smallest performance difference among all federated methods, which reflects the robustness of FedICON against attribute shift during test time.

We also carry out experiments on Office-10 and DomainNet datasets to verify the effectiveness of FedICON on handling *covariate* test-time shift problem. The results are provided in Table 5 and Table 6, respectively. The results demonstrate that FedICON

consistently outperforms existing baselines during the local test of all clients. We additionally show the experimental results of Digit-5 under *domain* test-time shift. The results are provided in Table 7. The results demonstrate that FedICON outperforms the baselines in most clients.

Ablation Studies. We carry out ablation studies to see the effect of each key technical component in FedICON. We alternatively remove the local invariance extraction, global invariance sharing and test-time adaptation procedure to see how these components affect the performance in handling test-time shift issues in FL. It is worth noting that when the local invariance extraction component is removed, the local training objective changes from the supervised contrastive loss to the conventional classification loss (*i.e.*, cross-entropy), and the global invariance sharing component is in charge of sharing the learnable parameters of the local model. In Table 8, we can observe that both global invariance sharing and test-time adaptation play an important role in improving the ability to deal with test-time shift problems. The effect of global invariance sharing is more notable because it is the core step to explore the inherent inter-client heterogeneity and broadcast the invariant encoding knowledge to benefit their local learning. The test-time adaptation brings about 1% to 5% improvement compared with the variants without this component, verifying the effectiveness of consistent representation alignment via a self-supervised learning scheme during the test phase.

Extending to unsupervised scenarios. The main framework of our proposed method is based on representation learning, which makes it flexible to migrate the whole learning paradigm into the unsupervised representation learning manner. During the local invariance extraction and global invariance sharing stages, we remove the label information. The feature encoder is purely trained in a self-supervised manner and the label information is accessible only when training the classifier. As is shown in Table 9, the extended version of our method FedICON, termed as FedICON-*us*, achieves the runner-up performance on Digit-5 and Office-10 under covariate test-time shift.

Table 4: Test accuracy on CIFAR10 dataset under *attribute* test-time shift. All results are reported in mean(std error) format over three independent runs. Δ represents the difference between the accuracy on *original* test set and that on *attribute* test-time shift case.

Method	Non-IID Dir(0.1)			Non-IID Dir(1)		
	Original	**Attribute Shift**	Δ	Original	**Attribute Shift**	Δ
Local	89.15(1.13)	83.54(0.89)	-5.61	67.60(0.65)	61.61(1.99)	**-5.99**
FedAvg	64.25(1.42)	50.09(1.36)	-14.16	73.23(0.38)	57.17(0.34)	-16.06
FedAvg-FT	87.82(0.59)	80.95(2.55)	-6.87	78.21(0.68)	65.59(1.46)	-12.62
FedProx	59.79(5.91)	52.25(2.53)	-7.54	72.02(0.37)	61.72(1.29)	-10.3
FedRep	88.73(0.26)	81.72(0.63)	-7.01	79.25(0.59)	66.69(1.22)	-12.56
FedBN	83.09(4.56)	78.14(0.57)	-4.95	77.67(1.07)	66.90(1.31)	-10.77
FedTHE	90.55(0.41)	81.91(0.54)	-8.64	**80.49**(0.89)	66.25(0.96)	-14.24
Tent	**91.39**(0.77)	85.72(1.24)	-5.67	78.61(0.63)	71.50(1.29)	-7.11
EATA	91.37(0.78)	85.60(1.16)	-5.77	79.72(0.62)	70.44(1.67)	-7.28
T3A	90.79(0.73)	85.91(1.68)	-4.88	78.80(0.95)	70.93(0.68)	-7.87
FedICON	90.24(0.52)	**86.47**(1.27)	**-3.77**	79.43(0.78)	**72.41**(1.29)	-7.02

Table 5: Test accuracy on Office-10 dataset under *covariate* test-time shift. All results are reported in mean(std error) format over three independent runs.

Method	Amazon	Caltech	DSLR	Webcam
Local	60.94(1.56)	39.70(2.45)	66.67(4.77)	64.97(2.59)
FedAvg	61.63(0.80)	35.41(1.43)	67.71(1.80)	65.45(1.81)
FedAvg-FT	64.93(0.80)	41.04(0.93)	67.79(2.43)	68.04(1.45)
FedProx	61.81(1.83)	36.15(1.36)	64.58(1.80)	67.23(3.53)
FedRep	55.90(2.57)	32.15(3.59)	44.79(9.02)	54.07(4.17)
FedBN	49.83(3.01)	38.37(1.12)	64.58(3.61)	56.50(5.95)
FedTHE	62.85(0.80)	37.33(2.35)	63.54(3.61)	69.49(3.39)
Tent	58.51(3.94)	44.44(1.33)	61.46(7.22)	67.23(4.27)
EATA	58.16(4.24)	44.30(1.43)	61.46(7.22)	65.54(2.59)
T3A	58.68(2.67)	42.67(2.31)	63.54(3.61)	66.10(1.69)
FedICON	**71.53**(0.30)	**48.44**(0.89)	**69.79**(6.51)	**70.06**(3.53)

Table 6: Test accuracy on DomainNet dataset under *covariate* test-time shift. All results are reported in mean(std error) format over three independent runs.

Method	Clipart	Infograph	Painting	Quickdraw	Real	Sketch
Local	72.05(1.88)	36.53(1.72)	74.31(0.23)	50.20(0.57)	79.25(1.34)	67.78(0.89)
FedAvg	78.94(0.94)	37.32(0.54)	70.60(0.23)	50.05(1.48)	77.90(0.23)	70.13(1.15)
FedAvg-FT	77.23(1.75)	37.91(0.11)	75.27(0.34)	46.80(1.84)	80.44(0.46)	70.76(0.51)
FedProx	79.13(0.67)	37.95(0.32)	69.47(0.23)	49.95(2.62)	77.77(0.64)	69.86(0.26)
FedRep	73.67(1.48)	36.30(2.26)	74.47(1.60)	53.70(0.99)	79.05(2.21)	67.78(0.64)
FedBN	78.32(0.94)	37.15(0.75)	74.82(2.40)	44.75(4.60)	80.32(0.41)	70.04(1.02)
FedTHE	66.63(0.13)	35.01(0.43)	60.58(0.23)	43.70(1.13)	56.74(0.64)	62.18(0.64)
Tent	49.24(0.54)	27.47(0.11)	43.62(1.37)	30.30(0.42)	31.18(0.17)	46.21(0.51)
EATA	49.43(0.54)	27.09(0.22)	42.89(1.71)	29.35(0.07)	30.90(0.23)	46.21(0.51)
T3A	78.71(1.61)	36.77(0.65)	74.76(1.94)	53.10(7.21)	80.32(0.64)	72.65(1.40)
FedICON	**79.66**(0.27)	**38.43**(0.75)	**76.01**(0.11)	**57.40**(0.71)	**84.72**(0.23)	**76.44**(2.17)

Table 7: Test accuracy on Digit-5 dataset under *domain* test-time shift. All results are reported in mean(std error) format over three independent runs.

Method	Test Domain				
	MNIST	**SVHN**	**USPS**	**Synth**	**MNIST-M**
Local	41.35(2.60)	14.98(0.48)	32.67(2.37)	18.68(0.53)	18.22(0.94)
FedAvg	81.30(1.81)	21.04(1.21)	54.32(0.67)	42.04(1.65)	31.92(0.67)
FedAvg-FT	79.63(1.95)	20.86(0.93)	53.79(1.07)	40.98(1.91)	31.44(0.67)
FedProx	83.28(1.02)	23.26(1.49)	53.87(1.16)	41.75(1.02)	31.96(0.89)
FedRep	57.39(3.88)	17.72(1.03)	42.72(1.56)	24.08(1.23)	20.88(0.20)
FedBN	78.65(1.63)	20.74(1.86)	63.41(3.67)	37.04(1.75)	23.20(0.43)
FedTHE	79.09(1.30)	20.78(1.55)	55.62(0.47)	41.98(1.10)	30.74(0.95)
Tent	74.65(2.41)	32.89(0.45)	65.61(0.16)	45.62(1.19)	28.36(0.87)
EATA	74.91(1.77)	**33.02**(0.40)	65.99(0.16)	45.71(1.08)	28.29(0.56)
T3A	79.30(0.80)	19.93(0.82)	53.86(1.57)	41.10(1.12)	31.22(0.37)
FedICON	**89.75**(0.31)	30.22(2.56)	**74.14**(0.40)	**51.73**(1.60)	**36.03**(1.01)

Table 8: Comparison between the FedICON and its variants. Experiments are conducted on Digit-5 dataset under *covariate* test-time shift. The number of clients is 5. Each of them owns data samples from a specific domain. Corruptions are applied to their local test set to establish the *covariate* test-time shift setting. The number of communication rounds is 100.

Design Component	**Variants**							
Local Invariance Extraction		✓			✓		✓	✓
Global Invariance Sharing			✓		✓	✓		✓
Test-Time Adaptation				✓		✓	✓	✓
Test Accuracy	34.00	48.60	55.04	39.92	59.39	59.54	49.67	62.23

Table 9: Test accuracy on Digit-5 and Office-10 dataset under *covariate* test-time shift. The best and runner-up results are highlighted with bold and underline, respectively.

Method	Digit-5	Office-10
Local	34.00(3.12)	58.07(2.84)
FedAvg	55.04(1.00)	57.55(1.46)
FedAvg-FT	54.55(0.83)	60.45(1.40)
FedICON	**62.23**(0.88)	**64.96**(1.43)
FedICON-*us*	<u>61.60</u>(1.45)	<u>63.79</u>(1.17)

C RELATED WORK

Personalized Federated Learning. Personalized federated learning (PFL) aims to enable each client to learn a personalized model that performs well on its local dataset [37, 44]. Existing PFL methods are developed on the basis of various techniques. A branch of work utilizes model decoupling to separate the local model into the globally shared part and personally trained part to realize different levels of personalization [1, 4, 7, 8, 27]. [12, 23, 36, 38, 46] augment the original objective function with additional regularization terms to induce personalization via representation alignment, knowledge transfer, bi-level optimization, etc. [9, 18] incorporate meta-learning to generate the generalized model which can be further optimized as a personalized model. Existing personalized federated learning methods can only personalize the model to local training data, lacking the ability to generalize to the test dataset.

Federated Learning with Feature-Level Distribution Shifts. Non-IID issue is the core challenge in both FL and PFL. A common type of non-IID data in real-world scenarios is known as feature-level distribution shifts, including attribute shift, covariate shift, and domain shift [17]. For instance, in FL, the training and test images can be collected in different time period, differnt locations, and even from different photographic equipment. [29] leverages mutual information theory and diversity augmentation to mitigate the negative effects brought by attribute shift and domain shift. [26] proposes to use local batch normalization to alleviate the feature-level distribution shift problems. Enlightened by the idea of domain generalization methods, [5, 10] design different knowledge transfer strategies to learn high-quality domain consensus knowledge across shifted clients. Based on feature-level distribution shift setting, [14] and [43] aim to alleviate the inter-domain knowledge forgetting problem and performance degradation problem on unseen clients/domains, respectively.

Test-Time Adaptation. Test-time adaptation methods aim to improve the model performance on out-of-distribution data by adapting the learned model to test data. Several methods propose to adjust the model by minimizing the pre-defined test-time entropy in various forms [32, 41, 47]. [15] presents a classifier-only adjustment module for test domain adaptation, and [22] further shows that selectively fine-tuning part of the model can address different types of distribution shifts. [2, 35] tune the model on unlabelled test data via self-supervised learning objectives. [28] further proposes to use more informative self-supervised learning to solve the test-time performance degradation in the presence of severe distribution shifts. The performance of existing centralized test-time adaptation methods can be hindered due to the inherent heterogeneity in FL during the training phase, which further affects the test adaptation process.

Federated Representation Learning. Representation learning focuses on extracting higher-level representations from raw data using deep neural networks [40]. Combining it with the federated learning framework enables distributed clients to learn powerful representation and thus benefit their downstream tasks [3]. [51] introduces a generalized federated self-supervised learning framework and provides in-depth empirical studies on the effect of key components in federated representation learning. [16] incorporates model decoupling and representation learning for better PFL. [45, 50] aim to learn a common representation model without supervision under the FL framework. Both of them utilize Siamese networks as their local representation module.

D ALGORITHM

The algorithm of FedICON is presented in Algorithm 1.

Algorithm 1 FedICON

Training Phase with **Input:** Training dataset $\mathcal{D}_i^P$, parameter of feature encoder θ_i, $i = 1, \cdots, m$.

Server executes:

Initialize feature encoder with parameter $\bar{\theta}$.
for each round $T = 1, 2, \ldots$ **do**
  for each client i **in parallel do**
    $\theta_i \leftarrow \text{LocalUpdate}(i, \bar{\theta})$
  end for
  Update global feature encoder by aggregating all the parameters from clients.
end for

LocalUpdate$(i, \bar{\theta})$:

for each local epoch **do**
  for each batch in $\mathcal{D}_i$ **do**
    Compute $L_{i,\text{sup}}$ and update θ_i by Eq. (2).
  end for
end for
return θ_i

Test Phase at the i-th Client with **Input:** Test dataset $\mathcal{D}_i^Q$, parameter of feature encoder θ_i.

LocalTest(i, θ_i):

for each local epoch **do**
  for each batch in $\mathcal{D}_i^Q$ **do**
    Compute $L_{i,\text{us}}$ and fine-tune θ_i by Eq. (5).
  end for
end for
return θ_i

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009