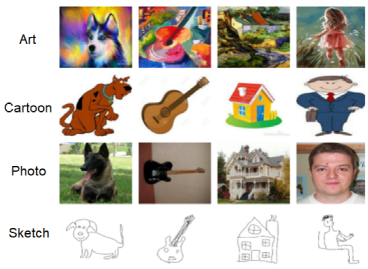
A. Experimental Details

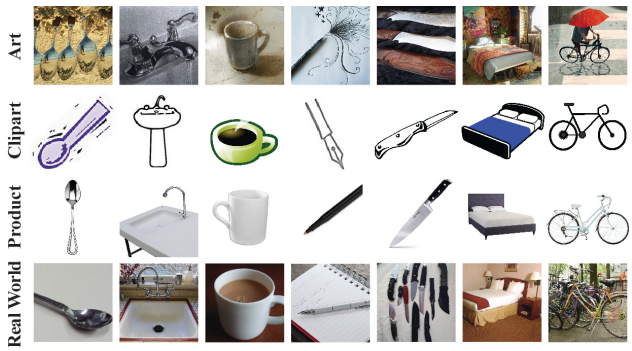
A.1. Visualization of Benchmark Datasets

In this section, we show example images in different domains from the adopted benchmark datasets, *i.e.*, PACS (Fig. 1a), Digits (Fig. 1b), and OfficeHome (Fig. 1c). We can see that there exists strong appearance variation and distribution shifts across different domains, e.g., in PACS we have both photo-like realistic pictures (*Photo*) and highly abstract human sketches (*Sketch*). Therefore, by assigning data from one of the domains to each client, we are able to simulate the experimental setting with non-IID features in FL.



(a) PACS [5]

(b) Digits



(c) OfficeHome [9]

Figure 1: Example images from the selected benchmark datasets with non-IID features. *Best viewed in color.*

A.2. Hyperparameter Settings

In this section, we provide more details about the training hyperparameters, as well as their search space in Tab. 2. We use 1 NVIDIA GeForce GTX TITAN X with 12GB RAM to run the experiments. We use PyTorch [8] to implement our algorithm. For all baselines and the algorithms proposed in the previous work, we apply the same hyperparameter search as FRAug and report the best performance. For the PACS, OfficeHome, and Digits benchmarks, we apply data augmentation given in Tab. 1 during the training following the previous work [2]. For the medical dataset RxRx1, we follow the setting used in the WILDS benchmark [4] and do not apply any data augmentation. We also display the detailed hyperparameter selection for the proposed method on different benchmarks for possible future reproduction.

Augmentation	Parameters
RandomResizedCrop	portion: $[0.6, 1.0]$
RandomHorizontalFlip	probability: 0.5 (0.0 for Digits)
ColorJitter	jitter degree: 0.3
RandomGrayscale	probability: 0.1
Normalize	ImageNet statistics [1]

Table 1: Data augmentations for OfficeHome, Digits and PACS benchmarks.

In FRAug, the ratio λ_c used for computing the class-wise average embedding $\overline{u}_c^k$ is computed with an exponential ramp-up schedule. Specifically,

$$\lambda_c = \begin{cases} \lambda_0 \cdot exp(-5(1-\frac{r}{r_0})), & r < r_0 \\ \lambda_0, & r \geq r_0 \end{cases} \quad (1)$$

where λ_0 is set to 0.3, and r_0 is set to 5% of the total communication rounds $(0.05 \cdot R)$.

For the weighting coefficient λ_{syn} that controls the impact of the generated residuals during the training, we use an exponential schedule, *i.e.*, $\lambda_{syn} = e^{0.01(r-R)}$.

A.3. Model Architecture

Following [6], we use a 6-layer CNN with its details listed in Tab. 3 for the Digits dataset: For the convolutional layer (Conv2D), we list parameters with the sequence of input and output dimensions, kernel size, stride, and padding. For the max-pooling layer (MaxPool2D), we list kernel and stride. For the fully-connected layer (FC), we list input and output dimensions. For the Batch Normalization layer (BN), we list the channel dimension. We adopt the last FC layer as the prediction head, which defines the feature dimension with 512.

For the classification models on OfficeHome and PACS datasets, we use the widely adopted backbone ResNet18 [3] and change the output dimension of the last fully-connected layer (FC) to match the class number C of the dataset. We adopt the last FC layer as the prediction head, which defines the feature dimension with 512.

The network architecture of the generator and the Representation Transformation Network (RTNet) are given in Tab. 4 and Tab. 5, respectively. For the generator, we adopt a two-layer MLP, which takes a noise vector $\boldsymbol{z}$ with dimension d_z and a one-hot encoded label y as the input, and outputs a client-agnostic feature representation $\hat{\boldsymbol{v}}$. For RT-

	Hyperparameter	OfficeHome	PACS	Digits	RxRx1
Shared Parameters	Learning rate	0.01	0.01	0.01	0.01
	Image size	224x224	224x224	32x32	256x256
	Optimizer	SGD	SGD	SGD	SGD
	Optimizer momentum	0.5	0.5	0.5	0.5
	Communication rounds (R)	200	200	200	200
Shared Search Space	Local update steps (T)	{5, 10, 20}	{5, 10, 20}	{5, 10, 20}	{5, 10, 20}
	Batch size B	{16, 32, 64}	{16, 32, 64}	{64, 128, 256}	{16, 32}
FRAug	$\boldsymbol{\omega}$ and $\boldsymbol{\phi}$ optimizer	SGD	SGD	SGD	SGD
	Synthetic batch size B_{syn}	16	32	64	32
	η_g	0.05	0.05	0.005	0.01
	η_m	0.025	0.05	0.005	0.01
	α	1.5	1.25	1	1
	β	1.25	1.25	1.5	1
	d_z	128	256	256	128

Table 2: Hyperparameter configurations for different datasets

Layer	Details
1	Conv2D(3, 64, 5, 1, 2) BN(64), ReLU(), MaxPool2D(2, 2)
2	Conv2D(64, 64, 5, 1, 2) BN(64), ReLU(), MaxPool2D(2, 2)
3	Conv2D(64, 128, 5, 1, 2) BN(128), ReLU(), Flatten()
4	FC(6272, 2048) BN(2048), ReLU()
5	FC(2048, 512) BN(512), ReLU()
6	FC(512, 10)

Table 3: Classification model architecture for the Digits benchmark.

Layer	Details
1	FC($d_z + C$, d_u) BN(d_u), ReLU()
2	FC(d_u, d_u) BN(d_u), ReLU()

Table 4: Generator architecture.

Layer	Details
1	FC(d_u, d_z) BN(d_z), ReLU()
2	FC(d_z, d_u) BN(d_u)

Table 5: Representation Transformation Network (*RTNet*) architecture.

Net, we adopt a two-layer MLP, which takes the output of the generator $\hat{\boldsymbol{v}}$ and outputs a client-specific feature residual with dimension d_u.

B. Additional Results and Analyses

B.1. Ablation Study

To illustrate the importance of different FRAug components, we conducted ablation studies on three benchmark datasets, *i.e.*, OfficeHome, Digit and PACS, where the results are shown in Tab. 6, Tab. 7, and Tab. 8, respectively. First, we find that solely applying the RTNet based on the real feature embeddings, *i.e.*, training without a shared generator barely brings performance gain. We assume that RTNet is restricted to the client local distribution and is only helpful when it accesses the client-agnostic. Moreover, using the client-agnostic synthetic embeddings $\hat{\boldsymbol{v}}^k$ leads to only minimal performance gain, highlighting the importance of the proposed representation transformation schema, *i.e.*, RTNet. Moreover, the results demonstrate that using both types of synthetic embeddings, *i.e.*, $\hat{\boldsymbol{u}}_c^k$ and $\hat{\boldsymbol{u}}^k$, yields the largest performance boosts for both benchmarks.

G ($\hat{\boldsymbol{v}}$)	RTNet ($\hat{\boldsymbol{u}}$)	EMA ($\hat{\boldsymbol{u}}_c$)	OfficeHome A	C	P	R	avg
	✓		56.61±0.3	57.08±0.5	73.14±0.6	71.49±0.4	64.58±0.5
	✓	✓	57.51±0.2	56.95±0.3	73.58±0.5	72.28±0.5	65.08±0.4
✓			56.24±0.5	59.65±0.3	72.90±0.3	73.09±0.5	65.47±0.8
✓	✓		57.34±0.9	59.10±0.4	73.65±0.7	74.25±0.9	66.09±0.2
✓		✓	56.65±0.2	59.50±0.7	73.50±0.5	74.31±0.9	65.99±0.3
✓	✓	✓	**57.61**±0.9	**60.03**±0.8	**74.03**±0.3	**74.69**±0.2	**66.60**±0.4

Table 6: Ablation study for different components of FRAug on OfficeHome benchmark. The average evaluation accuracy of all clients are reported.

G ($\hat{\boldsymbol{v}}$)	RTNet ($\hat{\boldsymbol{u}}$)	EMA ($\hat{\boldsymbol{u}}_c$)	Digits				
			MT	MM	SV	UP	avg
	✓		97.26±0.2	74.25±0.1	75.42±0.3	**97.98**±0.1	86.23±0.2
	✓	✓	96.97±0.1	75.75±0.2	75.90±0.1	97.79±0.3	86.60±0.1
✓			97.48±0.0	75.98±0.5	77.90±0.3	97.63±0.2	87.25±0.1
✓		✓	97.49±0.1	79.66±0.4	78.80±0.6	96.99±0.4	88.24±0.2
✓	✓		**97.95**±0.1	81.40±0.1	80.78±0.2	97.92±0.1	89.51±0.1
✓	✓	✓	97.81±0.1	**81.65**±0.9	**81.24**±0.3	97.67±0.4	**89.59**±0.4

Table 7: Ablation study for different components of FRAug on Digits benchmark. The average evaluation accu- racy of all clients are reported.

G ($\hat{\boldsymbol{v}}$)	RTNet ($\hat{\boldsymbol{u}}$)	EMA ($\hat{\boldsymbol{u}}_c$)	PACS				
			A	C	P	S	avg
	✓		83.80±0.5	83.95±0.3	96.64±0.2	89.12±0.4	88.38±0.5
	✓	✓	83.43±0.3	84.51±0.2	97.19±0.1	88.86±0.2	88.50±0.2
✓			86.06±0.7	**88.61**±0.9	98.24±0.1	90.37±0.4	90.82±0.5
✓	✓		86.54±0.8	88.19±0.3	98.44±0.3	89.78±1.0	90.74±0.4
✓		✓	87.34±0.5	88.47±0.9	**98.64**±0.6	**90.95**±0.4	**91.35**±0.1
✓	✓	✓	**87.50**±0.9	88.33±0.9	97.66±0.5	90.70±0.6	91.05±0.3

Table 8: Ablation study for different components of FRAug on PACS benchmark. The average evaluation accu- racy of all clients are reported.

B.2. Analysis of Local Dataset Size

In this section, we investigate the effectiveness of our representation augmentation technique for different sizes of client-specific local datasets. Hereby, we vary the number of datapoints available on each client from 100% to 10% of its original local dataset. Tab. 10 depicts the results of this experiment. We compare FRAug with two baseline methods, *i.e.*, FedAvg and Single, as well as FedBN on OfficeHome, and conduct the experiment with 3 different seeds. Compared to FedAvg and FedBN, the improvement achieved by FRAug is stable across different dataset sizes, highlighting the suitability of representation augmentation for scenarios involving non-IID features with scarce and large amounts of data. Compared to local training (*Single*) without collaboration, we observe that the performance improvement yielded by federated learning methods increases as the dataset size decreases. Note that we do not highlight the result of oracle baseline *All* when it achieves the best results, since it does not fulfill the requirement of FL, *i.e.*, datasets from different clients should be decentralized and private.

B.3. Hyperparameter Sensitivity

In this section, we further demonstrate the low sensitivity of the proposed method to the selection of different hyperparameters and present the results of the experiments.

B.3.1 Effects of α, β and d_z

In this section, we show the performance of local classification models in FRAug trained with different combinations of loss ratio in generator optimization, *i.e.*, α, loss ratio in RTNet optimization, *i.e.*, β and dimension of the random noise input of both, *i.e.*, d_z on the OfficeHome and Digits benchmark. We select α and β from $\{0.5, 0.75, 1.0, 1.25, 1.5\}$ and select d_z from $\{64, 128, 256, 512\}$. We display the results in the format of box-plots in Fig. 2 and Fig. 3. From the results, we conclude that FRAug is not sensitive to the selection of these hyperparameters.

B.3.2 Effects of η_g and η_m

In this section, we show the performance of local classification models trained with different combinations of learning rate for the generator, *i.e.*, η_g and learning rate for the RTNet, *i.e.*, η_m. Here, we select $\eta_g, \eta_m \in \{0.05, 0.025, 0.01\}$ and the results is given in Tab. 9. The results show that FRAug is robust to the selection of the learning rate of the generator and the RTNet.

	$\eta_m = 0.05$	$\eta_m = 0.025$	$\eta_m = 0.01$
$\eta_g = 0.05$	66.37	67.00	66.33
$\eta_g = 0.025$	66.69	65.98	66.71
$\eta_g = 0.01$	66.03	66.47	66.19

Table 9: Average test accuracy using different combinations of learning rate η_g and η_m on OfficeHome benchmark.

B.3.3 Effects of T and B_{real}

In this section, we further display the performance of local classification models in FRAug trained with different combinations of local update steps $T \in \{5, 10, 15, 20\}$ and the batch size for the real training samples $B_{real} \in \{16, 32, 64\}$ in Tab. 11. The results show that FRAug can consistently outperform FedAvg and is robust to the selection of the batch size of real samples as well as the local update steps.

B.4. UMAP Visualizations

In Fig. 4, we provide the UMAP [7] visualization of the feature embeddings, extracted by the models optimized by FedAvg and FRAug in PACS benchmark. From the results, we observe that the features extracted by FRAug show better separability, indicating the better robustness of FRAug against the feature distribution shift.

Client	Method	100%	80%	60%	40%	20%	10%
Art	Single	73.06±1.0	69.96±1.2	67.35±1.3	62.28±1.3	50.21±1.8	35.80±0.2
	FedAvg	72.43±0.9	71.06±1.6	68.48±1.4	65.48±1.5	62.28±1.4	56.38±1.1
	FedBN	72.55±0.7	71.78±0.6	68.98±0.5	65.22±0.9	63.58±0.7	57.59±0.8
	FRAug	**73.11**±0.7	**72.99**±1.0	**69.07**±1.4	**66.53**±1.7	**64.20**±0.6	**57.61**±0.6
	All (Orcale)	67.76±1.8	66.12±0.9	67.63±1.9	63.79±0.9	62.41±1.9	56.65±0.7
Clipart	Single	**80.09**±1.6	77.66±0.4	74.29±1.2	68.65±0.3	53.24±1.4	45.54±0.8
	FedAvg	77.57±0.5	77.50±1.2	75.74±1.0	73.03±0.2	67.05±1.3	57.21±0.9
	FedBN	77.96±0.5	77.40±0.8	76.25±0.4	73.46±0.6	67.75±0.9	56.52±0.3
	FRAug	78.92±1.3	77.65±1.0	**76.85**±0.6	**73.53**±1.4	67.66±1.1	**60.03**±0.5
	All (Orcale)	78.71±1.2	78.64±1.3	76.28±1.0	73.30±0.2	68.57±2.0	58.81±1.6
Product	Single	86.51±0.9	85.56±1.8	84.08±1.5	83.03±0.9	73.95±1.8	67.04±0.8
	FedAvg	85.21±1.0	85.14±1.2	83.26±1.0	82.73±1.5	76.35±1.1	73.87±0.8
	FedBN	85.52±0.7	84.46±0.9	84.06±1.0	81.95±0.8	77.95±0.3	73.55±1.0
	FRAug	**86.94**±0.5	**85.91**±1.3	84.42±0.9	**83.55**±1.3	**78.38**±0.4	**74.03**±0.8
	All (Orcale)	85.81±0.3	84.98±1.3	84.68±1.6	83.10±0.9	76.57±1.8	71.39±0.3
Real World	Single	81.57±1.5	79.74±1.3	75.92±1.3	71.94±0.8	65.83±1.5	61.16±0.7
	FedAvg	82.07±0.5	81.65±0.9	80.14±0.9	78.40±1.1	75.61±1.3	70.64±0.3
	FedBN	82.75±0.4	81.73±0.5	80.74±1.0	79.92±0.9	76.61±0.8	72.40±0.9
	FRAug	**84.14**±0.5	**82.95**±0.4	**82.26**±0.2	**81.23**±1.2	**78.06**±1.1	**74.58**±0.4
	All (Orcale)	80.31±0.8	80.28±2.0	78.90±1.5	77.29±1.0	74.62±1.7	72.63±1.3

Table 10: Model performance over different portion of the datasets, *i.e.*, using $\{100\%, 80\%, 60\%, 40\%, 20\%, 10\%\}$ of the original datasets in OfficeHome benchmark. The average accurcay of all clients are reported.

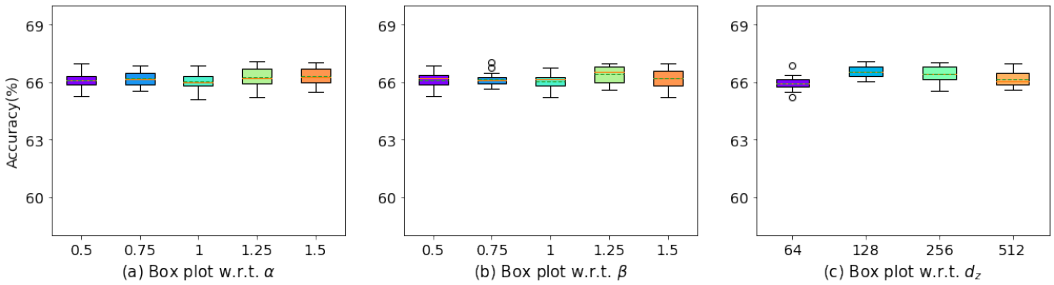


Figure 2: Evaluation results of FRAug with different hyperparameter combinations, *i.e.*, α (loss ratio in generator optimization) and β (loss ratio in RTNet optimization) and d_z (dimension of the random noise input), on OfficeHome benchmark. *Best viewed in color.*

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[2] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2020. 1

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[4] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021. 1

[5] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017. 1

[6] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*, 2021. 1

[7] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimen-

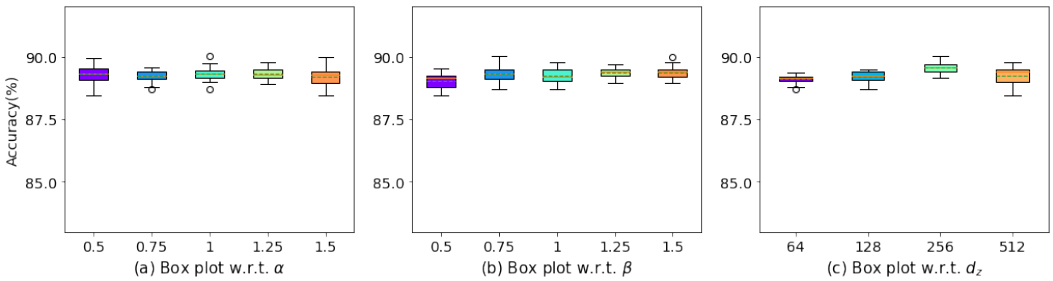


Figure 3: Evaluation results of FRAug with different hyperparameter combinations, *i.e.*, α (loss ratio in generator optimization) and β (loss ratio in RTNet optimization) and d_z (dimension of the random noise input), on Digits benchmark. *Best viewed in color.*

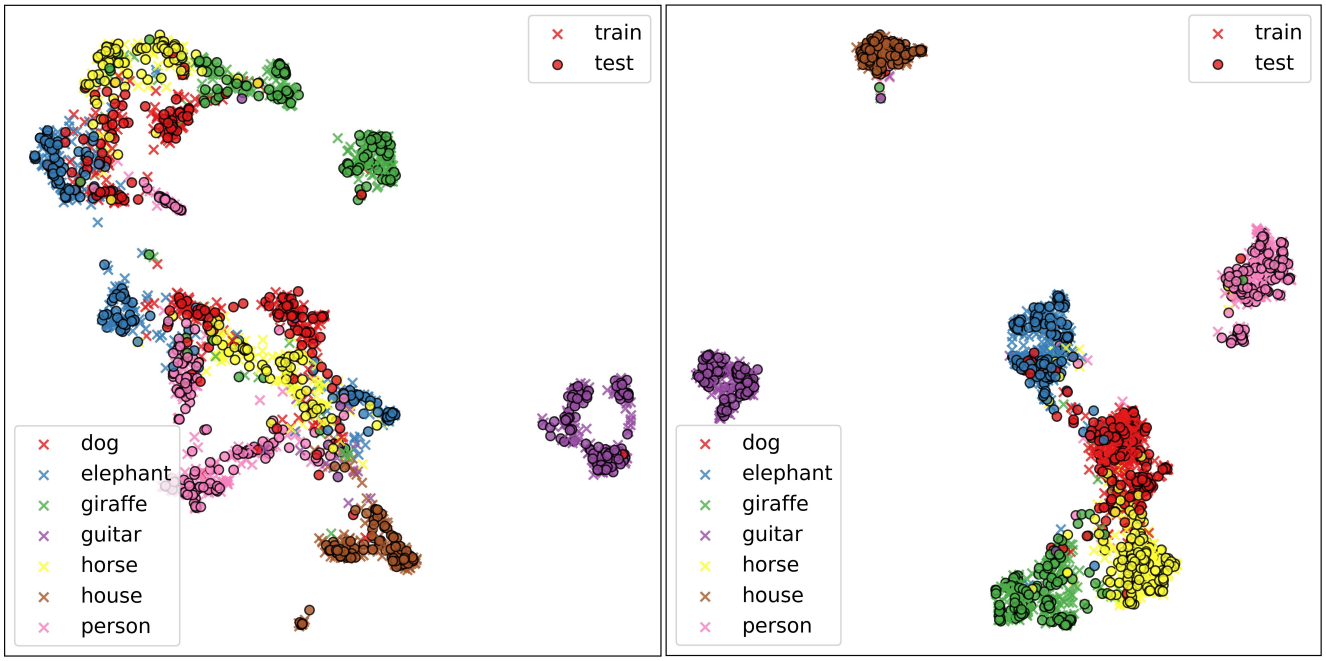
Setting	Method	Art	Clipart	Product	Real-World	Average
B_{real}=16, T=5	FedAvg	59.05±1.2	57.67±0.2	71.17±0.9	72.47±0.3	65.10±0.9
	FRAug	58.23±0.2	57.89±1.0	73.42±0.9	74.89±0.1	**66.11**±0.6
B_{real}=16, T=10	FedAvg	55.97±1.1	58.12±0.1	72.64±0.6	72.71±0.2	64.85±0.4
	FRAug	58.23±0.6	59.04±0.5	73.54±0.9	74.20±0.1	**66.25**±0.6
B_{real}=16, T=15	FedAvg	57.41±1.1	58.35±0.7	72.64±0.8	71.33±0.9	64.93±0.3
	FRAug	58.64±0.6	59.38±0.6	72.64±0.8	74.77±0.5	**66.36**±0.3
B_{real}=16, T=20	FedAvg	56.99±1.2	58.23±0.8	73.42±0.2	72.25±0.9	65.23±0.9
	FRAug	57.61±0.6	60.03±0.5	74.03±0.8	74.58±0.4	**66.60**±0.3
B_{real}=32, T=5	FedAvg	56.17±0.9	55.72±0.3	72.30±0.2	72.48±0.7	64.17±0.5
	FRAug	58.85±1.0	57.44±0.5	74.44±0.8	74.77±0.9	**66.37**±0.7
B_{real}=32, T=10	FedAvg	57.41±1.2	56.29±0.1	72.18±0.1	72.59±0.3	64.62±0.6
	FRAug	57.61±1.2	58.81±0.4	74.55±0.5	75.34±0.1	**66.58**±0.5
B_{real}=32, T=15	FedAvg	59.25±0.8	57.32±0.6	72.41±0.8	71.67±0.8	65.16±0.3
	FRAug	57.61±0.4	58.58±0.2	73.76±0.9	74.77±0.5	**66.18**±0.3
B_{real}=32, T=20	FedAvg	56.38±0.1	56.75±0.7	72.41±0.1	72.59±0.9	64.54±0.5
	FRAug	59.29±0.2	58.58±0.5	73.87±1.2	74.19±0.6	**66.58**±0.4
B_{real}=64, T=5	FedAvg	55.35±0.9	54.92±0.9	72.97±0.3	73.40±0.2	64.16±0.7
	FRAug	58.44±0.9	56.86±0.6	72.97±0.9	74.31±0.5	**65.65**±0.6
B_{real}=64, T=10	FedAvg	55.97±1.3	54.81±0.9	71.62±0.2	72.25±0.7	63.67±0.9
	FRAug	58.44±1.2	56.18±0.9	74.21±0.9	75.11±0.3	**65.99**±0.3
B_{real}=64, T=15	FedAvg	58.23±0.2	53.78±0.9	72.64±0.1	72.48±0.5	64.28±0.5
	FRAug	58.44±1.0	57.78±0.1	73.65±0.9	73.85±0.3	**65.93**±0.6
B_{real}=64, T=20	FedAvg	55.97±1.0	56.18±0.1	72.52±0.5	72.36±0.1	64.25±0.4
	FRAug	58.02±0.1	56.52±0.2	73.42±0.9	73.97±0.1	**65.48**±0.4

Table 11: Test accuracy using different combinations of batch size of real samples B_{real} and local update steps T on Office-Home benchmark.

sion reduction. *arXiv preprint arXiv:1802.03426*, 2018. 3

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 1

[9] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017. 1



(a) FedAvg (b) FRAug

Figure 4: UMAP visualization of the training and testing samples using the model optimized with FedAvg (left) and FRAug (right) on PACS benchmark. *Best viewed in color.*