Table A1. Architecture of Vallina AlexNet

Layer	Details
1	Conv2d(3, 64, 11, 4, 2), BN(64), ReLU, MaxPool2D(3,2)
2	Conv2d(64, 192, 5, 1, 2), BN(192), ReLU, MaxPool2D(3,2)
3	Conv2d(192, 384, 3, 1, 1), BN(384), ReLU
4	Conv2d(384, 256, 3, 1, 1), BN(256), ReLU
5	Conv2d(256, 256, 3, 1, 1), BN(256), ReLU, MaxPool2D(3,2)
6	AdaptiveAvgPool2D(6, 6)
7	FC(9216, 1024), BN(1024), ReLU
8	FC(1024, 1024), BN(1024), ReLU
9	FC(1024, num_classes)

Table A2. Architecture of FDSE's AlexNet

Layer	Details
1	DSEBlock(3, 64, 11, 4, 2, G=2, dw=3), MaxPool2D(3, 2)
2	DSEBlock(64, 192, 5, 1, 2, G=2, dw=3), MaxPool2D(3, 2)
3	DSEBlock(192, 384, 3, 1, 1, G=2, dw=3)
4	DSEBlock(384, 256, 3, 1, 1, G=2, dw=3)
5	DSEBlock(256, 256, 3, 1, 1, G=2, dw=3), MaxPool2D(3, 2)
6	AdaptiveAvgPool2D(6, 6)
7	DSEBlock(9216, 1024, 1, 1, 1, G=2, dw=1)
8	DSEBlock(1024, 1024, 1, 1, 1, G=2, dw=1)
9	FC(1024, num_classes)

Table A3. Architecture of DSEBlock(S,T,kernel_size, stride, padding, G, dw)

Layer	Details
1	Conv2d(S, $\lceil$T/G$\rceil$, kernel_size, stride, padding),BN_{DSE}($\lceil$T/G$\rceil$),ReLU
2	Conv2d($\lceil$T/G$\rceil$, T-$\lceil$T/G$\rceil$, dw, 1, dw//2)
3	Concat(out_{layer1},out_{layer2})
4	BN_{DFE}(T), ReLU

A1. Experimental Details

A1.1. Datasets

We use three popular datasets of multi-domain image classification tasks: Office-Caltech10 [17], DomainNet [6], and PACS [19]. The details of the three datasets are as below

Office-Caltech10. Office-Caltech10 is constructed by selecting the 10 overlapping categories (e.g., backpack, bike, calculator, headphones, keyboard, laptop, monitor, mouse, mug and projector) between the Office dataset [16] and Caltech256 dataset [3]. It contains four different domains: amazon, caltech10, dslr and webcam. These domains contain respectively 958, 1123, 295, and 157 images.

DomainNet. We follow [20] to select 10 categories from the 345 categories of objects of the original dataset. The domains of this dataset include clipart, real, sketch, infograph, painting, and quickdraw.

PACS. PACS [7] consists of four domains, namely Photo (1,670 images), Art Painting (2,048 images), Cartoon (2,344 images) and Sketch (3,929 images). Each domain contains seven categories.

We follow [20] to allocate each single domain's data to a client in our experiments. The visualized examples of the three datasets are respectively shown in Figure A1 (a), (b), and (c). We resize each sample into the size of 224×224 before feeding them into the model. We split each client's local data into training/validation/testing datasets by the ratios 0.8/0.1/0.1. The model is trained on training datasets and is selected according to its optimal performance on validation datasets. We finally report the metrics of the selected optimal model on each client's testing data.

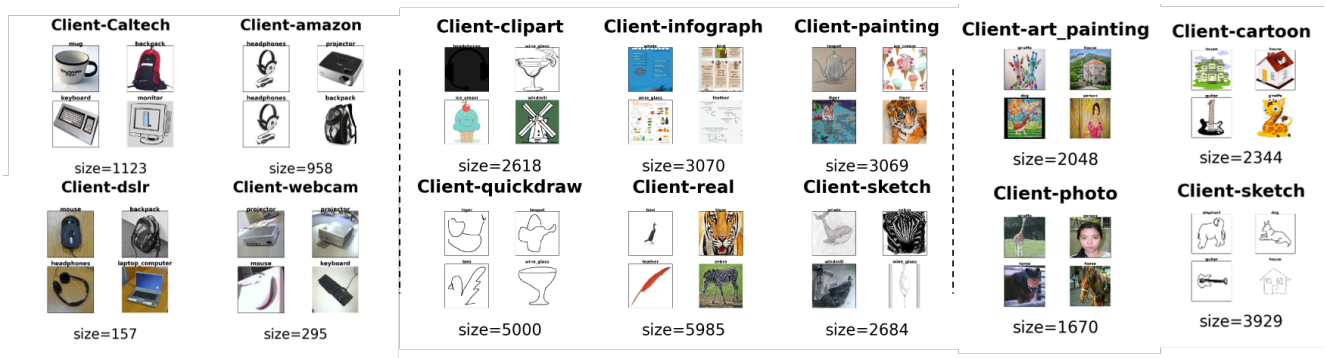
A1.2. Model Architecture

Backbone. We follow [20] to use AlexNet across our experiments. The architecture of the model is as shown in Table A1. The model used by FedFA has a similar architecture with Vallina AlexNet where the first five layers are respectively attached with an additional FFALayer. FDSE replaces each layer in the Vallina AlexNet with a DSEBlock as is shown in Table A1, and the details of each DSEBlock are listed in Table A3. Particularly, we follow [4] to preserve one identity mapping in the DSE convolution (e.g., layer 2).

A1.3. Baselines

We consider the following baselines in this work

- **Local** is a non-federated method where each client independently trains its local model;
- **FedAvg** [14] is the classical FL method that iteratively averages the locally trained models to update the global model;
- **LG-FedAvg**[12] is a method that jointly learns compact local representations on each device and a global model across all devices.
- **FedProx**[9] restricts the model parameters to be close to the global ones during clients' local training to alleviate the negative impact of data heterogeneity.
- **Scaffold**[5] corrects the model updating directions during model training to mitigate client drift's effects.
- **FedDyn**[1] maintains consistent local and global objectives during model training to avoid model overfitting on local objectives.
- **MOON**[8] restricts the model's representation space to be close to the global ones during clients' local training.
- **Ditto** [10] personalizes the local model by limiting its distance to the global model for each client with a proximal



(a) Office-Caltech10 (b) DomainNet (c) PACS

Figure A1. The visualization of each client's local data.

Algorithm 1 FedBN-Adaption

Input:The trained model $\mathcal{M}$, the target domain's testing data $\mathcal{D}_{target}$

for batch data $(\mathbf{X}, y) \in \mathcal{D}_{target}$ **do**
  the target client collects local statistics by computing $\mathcal{M}(\mathbf{X})$
end for
return $\mathcal{M}$

term.

- **PartialFed**[17] personalizes partial model parameters to suit the global model to local distributions.
- **FedBN**[11] lets BN layers be locally kept by each client without aggregation to adapt the global model to their local datasets.
- **FedFA**[20] augments features in the intermediate layers of the model to enhance clients' consensus from the feature level.
- **FedHeal**[2] mitigates gradient conflicts of important model parameters to enhance clients' consensus from the model parameter level.

A1.4. Hyper-parameters

Common parameters. We respectively tune the learning rate $\eta \in \{0.001, 0.01, 0.05, 0.1, 0.5\}$ by grid search for each method. We clip the gradient's norm to be no larger than 10. We run each trial for 500 communication rounds. The batch size is fixed to 50 and the local epochs for Domainnet, Office-Caltech10, and PACS are respectively 5, 1, and 5. We decay the learning rate by the ratio 0.998 per round. We select all the clients at each communication round like other works in cross-silo FL [13].

Algorithmic parameters. For Ditto [10] and FedProx [9], we tune the regularization coefficient $\mu \in [0.0001, 0.001, 0.01, 0.1, 1.0]$. For MOON [8], we follow its setting to set the range of the coefficient μ as $[0.1, 1.0, 5.0, 10.0]$ and fix the value of $\tau = 0.5$. For FedDyn [1], we tune the regularization coefficient $alpha \in [0.001, 0.01, 0.03, 0.1]$. For FedHeal, we tune the $\tau \in [0.1, 0.2, 0.3, 0.4, 0.5]$. For FDSE, we fix $\beta = 0.001$ and only tune $\lambda \in [0.01, 0.1, 1.0], \tau \in [0.001, 0.01, 0.1, 0.5]$.

Algorithm 2 FDSE-Adaption

Input:The trained model $\mathcal{M}$, the target domain's testing data $\mathcal{D}_{target}$, the number of epochs E, the learning rate η

the target client freezes the gradient of trainable parameters θ_u in $\mathcal{M}$ if θ_u does not belong to any DSE modules and fixes all the statistical parameters of BN_{DFE}.
for epoch $i = 1, ..., E$ **do**
  for batch data $(\mathbf{X}, y) \in \mathcal{D}_{target}$ **do**
    the target client computes model forward $\mathcal{M}(\mathbf{X})$
    the target client hook DSE module's outputs $\{\mathbf{X}_k^{(l)}\}$
    the target client compute regularization term in Sec. 4.2
    the target client optimizes the non-frozen parameters to minimize the regularization term via gradient descent with step size η.
  end for
end for
return $\mathcal{M}$

A1.5. Adapation Details

We illustrate the details of model adaptation for each method in Sec. 5.4. For FedAvg, we directly use the global model to make predictions on the target domain. For FedBN, we first collect local statistics for 1 epoch on the target domain's testing dataset and then evaluate the adapted model, as is shown in Algo. 1. For FDSE, we fine-tune

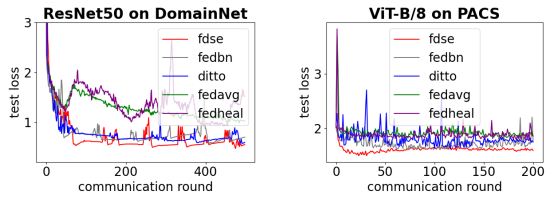


Figure A2. Testing loss curves on other model architectures.

Table A4. Model performance (↑) on other model architectures.

Method	DomainNet-ResNet50		PACS-ViT-B/8	
	ALL	AVG	ALL	AVG
FedAvg	$59.90_{\pm 0.96}$	$58.71_{\pm 1.06}$	$26.44_{\pm 4.16}$	$25.94_{\pm 4.46}$
FedHeal	$66.16_{\pm 0.62}$	$64.52_{\pm 0.55}$	$30.05_{\pm 3.44}$	$29.51_{\pm 2.98}$
FedBN	$69.36_{\pm 0.29}$	$66.99_{\pm 0.53}$	$36.67_{\pm 2.01}$	$36.22_{\pm 2.86}$
Ditto	$67.70_{\pm 0.36}$	$64.99_{\pm 0.49}$	$31.96_{\pm 4.32}$	$31.75_{\pm 4.13}$
FDSE	$\mathbf{72.98}_{\pm \mathbf{0.39}}$	$\mathbf{70.44}_{\pm \mathbf{0.32}}$	$\mathbf{38.24}_{\pm \mathbf{1.69}}$	$\mathbf{38.41}_{\pm \mathbf{1.90}}$

Table A5. Model performance (↑) on unseen clients.

Dataset		FedAvg	FedBN	FedDG-GA	FedSR	FDSE
Office	**C**	51.78	**60.71**	55.35	56.25	57.14
	A	70.52	70.52	72.63	**75.78**	**75.78**
	D	80.00	80.00	86.66	86.66	**93.33**
	W	65.61	55.17	68.96	69.32	**75.86**
	avg	66.95	66.60	70.90	72.00	**75.52**
DomainNet	**C**	62.81	62.56	62.43	60.75	**65.22**
	I	30.15	31.14	30.70	31.81	**32.34**
	P	55.53	57.26	57.04	56.18	**59.32**
	O	48.86	53.06	48.26	52.13	**55.00**
	R	59.74	63.17	59.85	64.15	**64.28**
	S	58.92	62.46	58.92	58.55	**65.28**
	avg	52.66	54.94	52.87	53.92	**56.91**

the DSE modules and fix other parameters to minimize the consistency regularization (e.g., Sec. 4.2) for several epochs before evaluation as shown in Algo. 2.

A2. Additional Experiments

A2.1. Other Model Architecture

We have studied the effectiveness on relatively large models in Table A4. We replace the last operator of each layer (i.e., ResNet50's block and Vit-B/8's feedforward layer) with DSE module. FDSE consistently outperforms baselines (e.g., Table A4) and exhibits faster convergence speed (e.g., Figure A2).

A2.2. Additional Baselines of generalizabilityy

We compare FDSE with the additional baselines [15, 18] for unseen clients in Table A5. FDSE outperforms all baselines, which we attribute to the additional adaptation steps.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*, 2021. 1, 2

[2] Yuhang Chen, Wenke Huang, and Mang Ye. Fair federated learning under domain skew with local consistency and domain diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12077–12086, 2024. 2

[3] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech 256, 2022. 1

[4] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1580–1589, 2020. 1

[5] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020. 1

[6] Aristotelis Leventidis, Laura Di Rocco, Wolfgang Gatterbauer, Renée J Miller, and Mirek Riedewald. Domainnet: Homograph detection and understanding in data lake disambiguation. *ACM Transactions on Database Systems*, 48(3): 1–40, 2023. 1

[7] 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

[8] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10713–10722, 2021. 1, 2

[9] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020. 1, 2

[10] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021. 1, 2

[11] 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. 2

[12] Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020. 1

[13] Jun Luo and Shandong Wu. Adapt to adaptation: Learning personalization for cross-silo federated learning. In *IJCAI: proceedings of the conference*, page 2166. NIH Public Access, 2022. 2

[14] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 1

[15] A. Tuan Nguyen, Philip Torr, and Ser Nam Lim. Fedsr: A simple and effective domain generalization method for federated learning. In *Advances in Neural Information Processing Systems*, pages 38831–38843. Curran Associates, Inc., 2022. 3

[16] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pages 213–226. Springer, 2010. 1

[17] Benyuan Sun, Hongxing Huo, Yi Yang, and Bo Bai. Partialfed: Cross-domain personalized federated learning via partial initialization. *Advances in Neural Information Processing Systems*, 34:23309–23320, 2021. 1, 2

[18] Ruipeng Zhang, Qinwei Xu, Jiangchao Yao, Ya Zhang, Qi Tian, and Yanfeng Wang. Federated domain generalization with generalization adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3954–3963, 2023. 3

[19] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *Proceedings of the AAAI conference on artificial intelligence*, pages 13025–13032, 2020. 1

[20] Tailin Zhou, Jun Zhang, and Danny HK Tsang. Fedfa: Federated learning with feature anchors to align features and classifiers for heterogeneous data. *IEEE Transactions on Mobile Computing*, 2023. 1, 2