OFASD: One-Shot Federated Anisotropic Scaling Distillation

Shunsei Koshibuchi Hiroshi Kera Kazuhiko Kawamoto
Chiba University, Chiba, Japan
{shunsei.koshibuchi, kera}@chiba-u.jp, kawa@faculty.chiba-u.jp

Abstract

One-shot federated learning often suffers from degraded performance under heterogeneous client data. To address this, we propose a task arithmetic-based knowledge integration method that applies anisotropic scaling. We optimize the scaling coefficients via knowledge distillation leveraging training data and ensemble outputs of local models. Using a ResNet-18 pre-trained on ImageNet-1K, we evaluate our method on CIFAR-10, CIFAR-100, and SVHN, each split according to a Dirichlet distribution. Our method outperforms the baseline across varying heterogeneity levels and achieves high accuracy with minimal training time.

1. Introduction

In many real-world scenarios, data remain distributed across multiple organizations and cannot be centralized due to privacy regulations and data-transfer costs. This situation has motivated the development of federated learning (FL) [20], which enables collaborative training without sharing raw data. In FL, clients train local models on private data and transmit only model updates to a central server, which aggregates them into a global model.

However, FL faces a challenge when client data are highly non-i.i.d., known as *data heterogeneity*, where local updates diverge and degrade global performance. To address this, various methods have been proposed. For example, FedProx [17] and SCAFFOLD [12] mitigate parameter drift during local training, and MOON [15] introduces a representation-alignment loss that encourages consistency between local and global feature space. Although effective, these methods require multiple rounds of communication to correct client-side biases, thereby increasing both communication overhead and local computation costs.

Therefore, correcting data heterogeneity in a single communication round, known as one-shot federated learning [7], remains challenging. One prominent line of work is knowledge distillation-based approaches [3, 6, 14, 29], which compress models while minimizing client-side computation. These methods train the global model on the server by using the ensemble of local outputs as teacher signals to integrate client knowledge. Distilling the ensemble into a global model with the same architecture should, in theory, match its accuracy, but a clear gap remains.

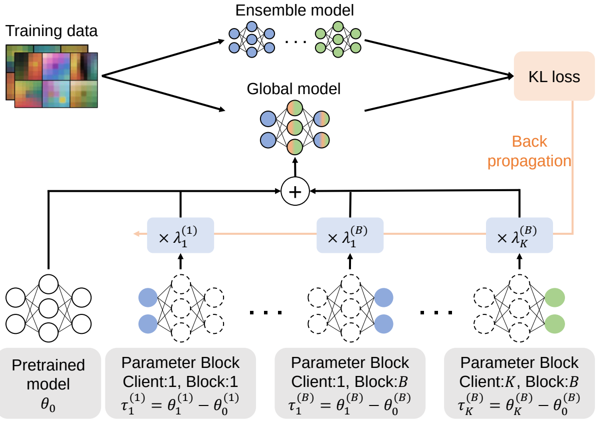
Training data
Ensemble model
. . .
Global model
KL loss
Back propagation
+
$\times \lambda_1^{(1)}$
$\times \lambda_1^{(B)}$
$\times \lambda_K^{(B)}$
. . .
. . .
Pretrained model θ_0
Parameter Block Client:1, Block:1 $\tau_1^{(1)} = \theta_1^{(1)} - \theta_0^{(1)}$
Parameter Block Client:1, Block:B $\tau_1^{(B)} = \theta_1^{(B)} - \theta_0^{(B)}$
Parameter Block Client:K, Block:B $\tau_K^{(B)} = \theta_K^{(B)} - \theta_0^{(B)}$

Figure 1. In parameter block-wise OFASD, task vectors τ are computed for each parameter block (highlighted layers), their scaling factors λ are independently optimized via knowledge distillation, and the global model is composed as a weighted average of the task vectors using these optimized scaling factors.

To close this gap, we focus on interference during the integration of heterogeneous local models and reduce it by proposing anisotropic scaling of task vectors. This approach is illustrated in Fig. 1. Each task vector is defined as the difference between model parameters before and after local training. Subsequently, we introduce scalar coefficients that are optimized anisotropically, either per task vector or per parameter block. By directly aggregating parameter differences, our method enables more detailed knowledge transfer while mitigating interference among heterogeneous clients. Moreover, since only the fine-tuned local parameters are required, our approach adds no extra communication overhead.

To assess the impact of our approach on global model accuracy, we integrate it into a one-shot federated learning method based on knowledge distillation. We use a ResNet-18 pre-trained on ImageNet as the backbone and evaluate classification accuracy across multiple datasets under varying degrees of data heterogeneity. Experimental results demonstrate that incorporating task arithmetic consistently enhances accuracy and produces a global model with greater stability and precision.

2. Related Work

2.1. One-Shot Federated Learning

In one-shot federated learning, client–server communication is limited to a single round to reduce overhead. Existing methods under this setting are commonly categorized into four main types: (i) knowledge distillation approaches [6, 7, 14], (ii) generative model-based approaches [2, 9, 27], (iii) ensemble learning approaches [1, 5], (iv) hybrid methods that combine these techniques [3, 19, 29].

Knowledge Distillation-Based Approaches: Knowledge distillation-based approaches have each client run a shared unlabeled dataset through its locally trained model and use the ensemble outputs as soft labels for global model training [6, 7, 14]. These approaches incur minimal computational overhead on clients and enable server-side model compression, unlike generative model-based techniques [2, 9, 27]. However, their effectiveness depends on the public dataset's relevance: low-quality or domain-mismatched data can degrade predictive performance [18, 23].

Generative Model–Based Approaches: Generative model-based approaches either train generators locally on each client [9] or rely on large pretrained generative models [2, 27]. These models generate pseudo-data that are transmitted to the server for global model training, avoiding the need for any public dataset. This advantage comes at a cost. Training a local generator requires more memory and computation than a standard classification model. Moreover, since raw local data are used to train these generators, the resulting models or their outputs may leak sensitive information.

Ensemble Learning Approaches: Ensemble learning approaches [1, 5] perform inference on the server by aggregating logits from all client models, using either equal or weighted averaging to produce the final prediction. Although this strategy fully exploits each model's knowledge, its memory consumption and inference latency on the server grow linearly with the number of clients. As discussed above, each approach has its own limitations. To overcome the limitations, recent work has explored hybrid strategies that combine multiple techniques [3, 19, 29].

2.2. Federated Learning and Task Arithmetic

Tao *et al.* [24] reformulate a single communication round of Federated Averaging (FedAvg) as task arithmetic, demonstrating equivalence in the update equations. They then provide a theoretical analysis that quantifies the error induced by data heterogeneity and proposes mitigation strategies by integrating existing federated learning algorithms. This work establishes a formal connection between task arithmetic and federated learning, laying a theoretical foundation for subsequent research.

Morafah *et al.* [21] observe that in federated learning environments with heterogeneous model capacity and data volume, standard knowledge distillation tends to suppress the logits of stronger models when aggregated with weaker ones. To address this, they introduce the TAKFL framework, which treats each device's distillation output as a separate task. These tasks are then adaptively fused via task arithmetic to achieve optimal knowledge integration. While TAKFL addresses cross-device heterogeneity, our study focuses on one-shot federated learning among clients with identical architectures, targeting a complementary problem setting.

3. Proposed Method

We propose One-Shot Federated Anisotropic Scaling Distillation (OFASD), a task arithmetic–based method designed for one-shot federated learning under data heterogeneity. In this study, data heterogeneity refers to the variation in class distribution across clients [16, 26]. While the data aggregated over all clients covers all classes sufficiently, individual clients may lack examples of certain classes due to skewed label proportions. This imbalance can degrade global model accuracy and slow convergence.

The next subsection reviews task arithmetic and introduces OFASD, which combines anisotropic scaling of task vectors with one-shot distillation to address these issues.

3.1. Task Arithmetic

Task arithmetic begins with a pre-trained model and computes, for each task, a task vector as the difference between the pre-trained and fine-tuned parameters. These task vectors are then added back to the pre-trained parameters to adjust the model.

Let the model $f(\boldsymbol{x}\,; \boldsymbol{\theta})$ map an input $\boldsymbol{x} \in \mathcal{X}$ to an output $y \in \mathcal{Y}$, given parameters $\boldsymbol{\theta} \in \Theta$. Let $\boldsymbol{\theta}_0$ denote pre-trained parameters, and let $\boldsymbol{\theta}_t, t = 1, 2, \ldots, T$, be the parameters after fine-tuning on task t. We then define the t-th task vector as

$$\boldsymbol{\tau}_t = \boldsymbol{\theta}_t - \boldsymbol{\theta}_0. \tag{1}$$

By combining these vectors linearly with scalar coefficients $\lambda_t \in \mathbb{R}$, we obtain the model $f(\cdot\,; \boldsymbol{\theta}_0 + \sum_{t=1}^{T} \lambda_t \boldsymbol{\tau}_t)$, which achieves strong performance on each task. The coefficients λ_t are either selected by grid search [11] or learned directly as model parameters [25, 28], enabling effective interference-aware knowledge integration.

3.2. One-Shot Federated Anisotropic Scaling Distillation (OFASD)

In the one-shot federated setting, we propose OFASD, which applies anisotropic scaling via task arithmetic to fuse knowledge from local models more effectively. OFASD achieves more precise knowledge integration than standard global aggregation. This improvement is achieved by optimizing scalar weights for each task vector or parameter block, guided by the ensemble prediction.

A key observation underlying OFASD is that the output of the model with the average task vector is approximately equal to the ensemble prediction to first order:

$$f\left(\boldsymbol{x};\,\boldsymbol{\theta}_0+\frac{1}{K}\sum_{k=1}^{K}\boldsymbol{\tau}_k\right)\approx\frac{1}{K}\sum_{k=1}^{K}f\left(\boldsymbol{x};\,\boldsymbol{\theta}_0+\boldsymbol{\tau}_k\right). \quad (2)$$

Both models share the same first-order expansion at $\boldsymbol{\theta}_0$:

$$f(\boldsymbol{x};\,\boldsymbol{\theta}_0)+\frac{1}{K}\sum_{k=1}^{K}\boldsymbol{\tau}_k^{\top}\nabla_{\boldsymbol{\theta}}f(\boldsymbol{x};\,\boldsymbol{\theta}_0). \quad (3)$$

This implies that averaging task vectors approximates the ensemble prediction to first order. However, due to higher-order effects, the actual outputs can differ significantly. OFASD mitigates this discrepancy by optimizing task vector coefficients to better align with the ensemble prediction beyond the linear approximation (see Appendix A for the derivation and Appendix B for optimization details).

To enable finer-grained knowledge integration, we split each task vector $\boldsymbol{\tau}_k$ into B disjoint parameter blocks and assign an independent scaling coefficient to each block. We refer to the case without partitioning as *client-wise*, and the case with partitioning as *parameter block-wise*. We adopt the same parameter partitioning strategy as aTLAS [28], grouping parameters by weight and bias within convolutional, fully connected, and batch normalization layers using functorch's `make_functional_with_buffers`. The global model is then constructed by adding as scaled task vectors or parameter blocks to the pre-trained parameters. To stabilize training, each vector or block is normalized to unit norm before scaling. Algorithm 1 summarizes the optimization procedure, where $\|\cdot\|$ denotes the ℓ_2 norm and $\mathrm{KL}(\cdot\|\cdot)$ is the Kullback–Leibler divergence.

Algorithm 1 OFASD: One-shot Federated Anisotropic Scaling Distillation

```
Require: Pre-trained parameter $\boldsymbol{\theta}_0$, Local model parameters $\{\boldsymbol{\theta}_k\}_{k=1}^{K}$, coefficients $\{\lambda_k^{(b)}\}_{b=1,\dots,B,\;k=1,\dots,K}$, Training dataset $\mathcal{D}$, Number of clients $K$, Number of parameter blocks $B$, Learning rate $\eta$
Ensure: Updated coefficients $\{\lambda_k^{(b)}\}_{b=1,\dots,B,\;k=1,\dots,K}$
  for $k$ in $1,\dots,K$ do
    Compute task vector: $\boldsymbol{\tau}_k \leftarrow \boldsymbol{\theta}_k - \boldsymbol{\theta}_0$
    Decompose into parameter blocks: $\left\{\boldsymbol{\tau}_k^{(b)}\right\} \leftarrow \boldsymbol{\tau}_k$
  end for
  for sampling batch $\mathcal{D}_{\mathbf{b}}$ in $\mathcal{D}$ do
    // Construct global model from task vectors
    Initialize: $\boldsymbol{\theta}_{\text{g}} \leftarrow \boldsymbol{\theta}_0$
    for each $(k,b)$ with $k=1,\dots,K$, $b=1,\dots,B$ do
      $\boldsymbol{\theta}_{\text{g}} \leftarrow \boldsymbol{\theta}_{\text{g}} + \lambda_k^{(b)}\frac{\boldsymbol{\tau}_k^{(b)}}{\|\boldsymbol{\tau}_k^{(b)}\|}$
    end for
    $L \leftarrow \frac{1}{|\mathcal{D}_{\mathbf{b}}|}\sum_{\boldsymbol{x}\in\mathcal{D}_{\mathbf{b}}}\mathrm{KL}(F_{\text{ens}}\left(\boldsymbol{x};\{\boldsymbol{\theta}_k\}_{k=1}^{K}\right)||f\left(\boldsymbol{x};\boldsymbol{\theta}_{\text{g}}\right))$
    // Update anisotropic scaling coefficients
    for each $(k,b)$ with $k=1,\dots,K$, $b=1,\dots,B$ do
      $\lambda_k^{(b)} \leftarrow \lambda_k^{(b)} - \eta\frac{\partial L}{\partial\lambda_k^{(b)}}$
    end for
  end for
  return $\{\lambda_k^{(b)}\}_{b=1,\dots,B,\;k=1,\dots,K}$
```

4. Experiment

We conducted comparative experiments in a one-shot federated learning setup to evaluate the effectiveness of OFASD. To this end, we integrated OFASD into DENSE [29], a representative ensemble-distillation framework for one-shot federated learning. In DENSE, a generator on the server feeds synthetic data into the global model, which is trained using the ensemble of local model outputs as teacher signals. The generator is optimized using three loss terms: (i) ensemble mimicry (similarity), (ii) real-data statistics matching (stability), (iii) global-model differentiation (transferability) (see [29] for details). Detailed experimental settings are described below.

4.1. Experimental Setup

Datasets: We evaluated on three widely used real-image datasets for federated learning: CIFAR-10 [13], CIFAR-100 [13], and SVHN [22]. To simulate data heterogeneity across clients, we partitioned each dataset using a Dirichlet distribution [16, 26, 29]. Letting C denote the number of classes, for each class $c \in 1,\dots,C$ we sampled a K-dimensional probability vector $\boldsymbol{p}_c \sim \mathrm{Dir}(\alpha)$, and assigned a fraction $p_{c,k}$ of class–c samples to client k. Lower α yields stronger label skew across clients.

Evaluation Metric: We report the test accuracy of the global model. For each method and heterogeneity level, we performed three trials with different random seeds and report the mean ± standard deviation.

Model Architecture: Each client uses a ResNet-18 [8] from PyTorch's torchvision, initialized with pretrained weights from ImageNet-1K [4].

Compared Methods: We compare OFASD against the following baselines. First, we consider two non-distillation

Table 1. Comparison of predictive accuracy across methods. Mean accuracy ± standard deviation is reported; within each row, the best result is highlighted in bold and the second-best is underlined.

	α	Accuracy (%)					
		Weight Average	Ensemble	DENSE		OFASD	
				w/o pretrain	w/ pretrain	client-wise	parameter block-wise
CIFAR10	0.01	10.00 ± 0.01	$\underline{16.46 \pm 0.73}$	13.06 ± 2.45	14.49 ± 2.59	11.76 ± 0.52	$\mathbf{17.01 \pm 0.80}$
	0.05	10.79 ± 0.62	$\underline{32.73 \pm 3.47}$	28.44 ± 1.47	30.56 ± 2.15	23.71 ± 4.31	$\mathbf{36.54 \pm 6.11}$
	0.1	17.86 ± 6.82	$\underline{48.98 \pm 1.14}$	36.91 ± 4.02	42.43 ± 1.31	40.15 ± 6.67	$\mathbf{55.21 \pm 1.75}$
	0.5	38.78 ± 20.72	$\mathbf{75.78 \pm 1.04}$	46.39 ± 2.25	53.65 ± 2.33	67.40 ± 1.96	$\underline{73.91 \pm 1.18}$
CIFAR100	0.01	2.48 ± 0.59	$\mathbf{28.57 \pm 0.75}$	14.46 ± 0.90	$\underline{18.29 \pm 2.04}$	12.43 ± 1.02	17.31 ± 1.02
	0.05	3.13 ± 0.49	$\mathbf{30.01 \pm 1.81}$	14.30 ± 2.16	18.20 ± 2.14	17.58 ± 1.76	$\underline{21.57 \pm 0.67}$
	0.1	2.26 ± 1.78	$\mathbf{32.51 \pm 0.87}$	15.90 ± 1.20	19.43 ± 0.29	18.25 ± 2.29	$\underline{24.52 \pm 0.03}$
	0.5	5.49 ± 2.53	$\mathbf{43.43 \pm 0.22}$	16.54 ± 1.22	20.85 ± 0.59	31.47 ± 1.57	$\underline{35.55 \pm 0.25}$
SVHN	0.01	15.96 ± 4.65	$\underline{20.73 \pm 1.26}$	16.02 ± 1.03	18.89 ± 3.55	16.66 ± 2.85	$\mathbf{22.11 \pm 5.33}$
	0.05	15.95 ± 3.99	$\underline{30.04 \pm 4.81}$	29.25 ± 3.51	$\mathbf{34.63 \pm 3.39}$	23.30 ± 5.16	28.03 ± 7.44
	0.1	14.25 ± 3.30	$\underline{51.74 \pm 7.69}$	43.07 ± 5.66	47.91 ± 8.57	35.79 ± 8.99	$\mathbf{52.38 \pm 11.23}$
	0.5	31.60 ± 9.89	$\underline{83.18 \pm 1.39}$	56.97 ± 3.15	65.98 ± 2.15	74.30 ± 1.22	$\mathbf{83.33 \pm 0.75}$

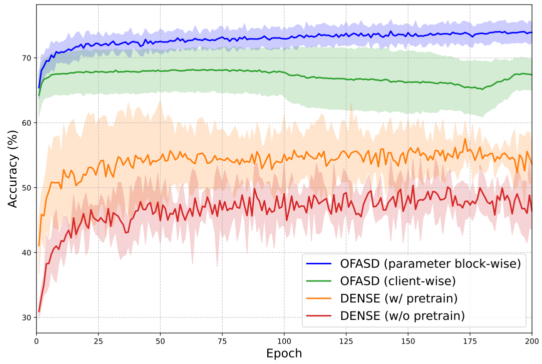

Figure 2. Accuracy progression during global model training on CIFAR-10 ($\alpha = 0.5$) for each method. Solid lines indicate the mean accuracy, and shaded bands show mean ± one standard deviation.

methods: *Weight Average*, which simply averages client parameters, and *Ensemble*, which averages predictions from local models. As a distillation-based baseline, we adopt DENSE [29], evaluated with two different global model initialization: random parameters (w/o pretrain) and ImageNet-pretrained parameters (w/ pretrain).

Training Details: We simulate $K = 10$ clients. Each local model trains for 50 epochs using SGD with a learning rate of 0.01, a momentum of 0.9, and a batch size of 128. The synthetic-data generator is trained following the procedure described in [29].

4.2. Results

Tab. 1 reports mean accuracy ± standard deviation for each dataset and heterogeneity level. As described in Sec. 3, Weight Average is defined as the simple average of local model weights, which corresponds to combining the pretrained model with the mean task vector. Weight Average and Ensemble share the same linear term. However, their predictive accuracies diverge markedly, indicating that matching linear components alone is insufficient to capture the full benefit of ensembling.

OFASD variants consistently achieve higher accuracy than the DENSE baseline, except for two cases: CIFAR-100 with $\alpha = 0.01$ and SVHN with $\alpha = 0.05$. The largest gains occur under relatively mild heterogeneity. Moreover, OFASD often surpasses the output-ensemble method (Ensemble), potentially due to the same phenomenon reported in LoRA research [10], where restricting parameter updates to a low-dimensional subspace helps mitigate overfitting.

Fig. 2 (CIFAR-10, $\alpha = 0.5$) plots the accuracy over training epochs for DENSE (w/ and w/o pretrain) and OFASD (client-wise and parameter block-wise). OFASD exceeds 60% accuracy after just one epoch, already outperforming DENSE even after 200 epochs. In addition, the parameter block-wise variant shows the lowest epoch-to-epoch variance, indicating the most stable global model convergence.

5. Conclusion

We propose OFASD, a one-shot federated learning method that integrates task arithmetic with anisotropic scaling. OFASD achieves higher accuracy than both simple averaging and prior distillation methods, without incurring any additional training cost or longer global training time. These results highlight the effectiveness of fine-grained task vector aggregation in improving knowledge integration under data heterogeneity. OFASD's anisotropic scaling coefficients are optimized for a fixed client set and must be recalibrated when the number of participants changes. Future work will explore zero-shot task arithmetic to build high-accuracy global models regardless of client count.

Acknowledgements. This work was supported by JSPS KAKENHI Grant Number JP23K24914

References

[1] Youssef Allouah, Akash Dhasade, Rachid Guerraoui, Nirupam Gupta, Anne-Marie Kermarrec, Rafael Pinot, Rafael Pires, and Rishi Sharma. Revisiting ensembling in one-shot federated learning. In *Advances in Neural Information Processing Systems*, pages 68500–68527, 2024. 2

[2] Haokun Chen, Hang Li, Yao Zhang, Jinhe Bi, Gengyuan Zhang, Yueqi Zhang, Philip Torr, Jindong Gu, Denis Krompass, and Volker Tresp. Fedbip: Heterogeneous one-shot federated learning with personalized latent diffusion models. In *the IEEE / CVF Computer Vision and Pattern Recognition Conference*, pages 30440–30450, 2025. 2

[3] Rong Dai, Yonggang Zhang, Ang Li, Tongliang Liu, Xun Yang, and Bo Han. Enhancing one-shot federated learning through data and ensemble co-boosting. In *International Conference on Learning Representations*, 2024. 1, 2

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *the IEEE / CVF Computer Vision and Pattern Recognition Conference*, pages 248–255, 2009. 3

[5] Yiqun Diao, Qinbin Li, and Bingsheng He. Towards addressing label skews in one-shot federated learning. In *International Conference on Learning Representations*, 2023. 2

[6] Xuan Gong, Abhishek Sharma, Srikrishna Karanam, Ziyan Wu, Terrence Chen, David Doermann, and Arun Innanje. Preserving privacy in federated learning with ensemble cross-domain knowledge distillation. *the AAAI Conference on Artificial Intelligence*, 36(11):11891–11899, 2022. 1, 2

[7] Neel Guha, Ameet Talwalkar, and Virginia Smith. One-shot federated learning. *arXiv:1902.11175*, 2019. 1, 2

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *the IEEE / CVF Computer Vision and Pattern Recognition Conference*, pages 770–778, 2016. 3

[9] Clare Elizabeth Heinbaugh, Emilio Luz-Ricca, and Huajie Shao. Data-free one-shot federated learning under very high statistical heterogeneity. In *International Conference on Learning Representations*, 2023. 2

[10] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 4

[11] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023. 2

[12] 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, 2020. 1

[13] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 2009. 3

[14] Qinbin Li, Bingsheng He, and Dawn Song. Practical one-shot federated learning for cross-silo setting. In *International Joint Conferences on Artificial Intelligence*, pages 1484–1490, 2021. 1, 2

[15] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *the IEEE / CVF Computer Vision and Pattern Recognition Conference*, pages 10713–10722, 2021. 1

[16] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *International Conference on Data Engineering*, pages 965–978, 2022. 2, 3

[17] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Machine Learning and Systems*, pages 429–450, 2020. 1

[18] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. In *Advances in Neural Information Processing Systems*, pages 2351–2363, 2020. 2

[19] Kangyang Luo, Shuai Wang, Yexuan Fu, Renrong Shao, Xiang Li, Yunshi Lan, Ming Gao, and Jinlong Shu. Dfdg: Data-free dual-generator adversarial distillation for one-shot federated learning. In *The IEEE International Conference on Data Mining*, pages 281–290, 2024. 2

[20] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, pages 1273–1282, 2017. 1

[21] Mahdi Morafah, Vyacheslav Kungurtsev, Hojin Chang, Chen Chen, and Bill Lin. Towards diverse device heterogeneous federated learning via task arithmetic knowledge integration. In *Advances in Neural Information Processing Systems*, pages 127834–127877, 2024. 2

[22] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *Advances in Neural Information Processing Systems Workshop*, 2011. 3

[23] Shangchao Su, Bin Li, and Xiangyang Xue. Domain discrepancy aware distillation for model aggregation in federated learning. *arXiv:2210.02190*, 2022. 2

[24] Zhixu Tao, Ian Mason, Sanjeev Kulkarni, and Xavier Boix. Task arithmetic through the lens of one-shot federated learning. *Transactions on Machine Learning Research*, 2025. Featured Certification. 2

[25] Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations*, 2024. 2

[26] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International Conference on Machine Learning*, pages 7252–7261, 2019. 2, 3

[27] Obaidullah Zaland, Shutong Jin, Florian T. Pokorny, and Monowar Bhuyan. One-shot federated learning with classifier-free diffusion models. *arXiv:2502.08488*, 2025. 2

[28] Frederic Z. Zhang, Paul Albert, Cristian Rodriguez-Opazo, Anton van den Hengel, and Ehsan Abbasnejad. Knowledge

composition using task vectors with learned anisotropic scaling. In *Advances in Neural Information Processing Systems*, pages 67319–67354, 2024. 2, 3

[29] Jie Zhang, Chen Chen, Bo Li, Lingjuan Lyu, Shuang Wu, Shouhong Ding, Chunhua Shen, and Chao Wu. Dense: Data-free one-shot federated learning. In *Advances in Neural Information Processing Systems*, pages 21414–21428, 2022. 1, 2, 3, 4