

Decoupled Federated Learning for ASR with Non-IID Data

Han Zhu^{1,2}, Jindong Wang³, Gaofeng Cheng^{†1}, Pengyuan Zhang^{1,2}, Yonghong Yan^{1,2}

¹Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics CAS, China

²University of Chinese Academy of Sciences, China

³Microsoft Research Asia, China

{zhuhan, chenggaofeng, zhangpengyuan, yanyonghong}@hcccl.ioa.ac.cn, jindong.wang@microsoft.com

Abstract

Automatic speech recognition (ASR) with federated learning (FL) makes it possible to leverage data from multiple clients without compromising privacy. The quality of FL-based ASR could be measured by recognition performance, communication and computation costs. When data among different clients are not independently and identically distributed (non-IID), the performance could degrade significantly. In this work, we tackle the non-IID issue in FL-based ASR with *personalized FL*, which learns personalized models for each client. Concretely, we propose two types of personalized FL approaches for ASR. Firstly, we adapt the *personalization layer based FL* for ASR, which keeps some layers locally to learn personalization models. Secondly, to reduce the communication and computation costs, we propose *decoupled federated learning (DecoupleFL)*. On one hand, DecoupleFL moves the computation burden to the server, thus decreasing the computation on clients. On the other hand, DecoupleFL communicates secure high-level features instead of model parameters, thus reducing communication cost when models are large. Experiments demonstrate two proposed personalized FL-based ASR approaches could reduce WER by 2.3% - 3.4% compared with FedAvg. Among them, DecoupleFL has only 11.4% communication and 75% computation cost compared with FedAvg, which is also significantly less than the personalization layer based FL.

Index Terms: federated learning, speech recognition, personalization, pseudo-labeling, semi-supervised learning

1. Introduction

ASR relies on massive training data for decent performance and conventionally uses centralized training as in Fig. 1a, where raw data of all clients are aggregated to the server. However, due to concerns and regulations [1] of data privacy, the client's data is in the form of isolated islands and is not allowed to be shared. Therefore, federated learning (FL) [2, 3] is proposed to collaboratively train the model for many clients without compromising privacy under the coordination of a server. Existing literature on FL-based ASR [4, 5, 6, 7, 8, 9, 10, 11] mostly follows the paradigm of FedAvg [2], where model parameters are exchanged instead of raw data. As shown in Fig. 1b, client models are trained locally for some epochs (local epoch) and then aggregated globally in the server. The local training and global aggregation are performed multiple times (global epoch).

FedAvg trains a global model for all clients and ignores the personalization of each client. Therefore, its performance could

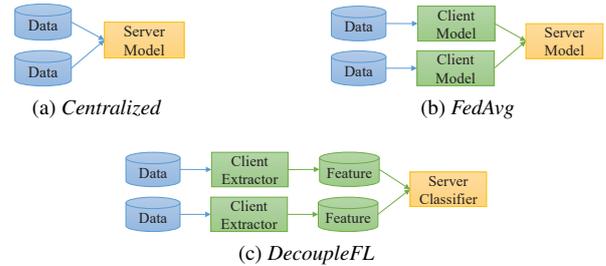


Figure 1: Illustration of different training methods.

degrade when data among clients are non-IID [12]. Personalized FL [13] can be used to alleviate this issue by learning personalized models for individual clients. Local fine-tuning based approaches study how to obtain a better initial global model [14, 15] or perform elaborate local optimization [16]. Model mixture approaches [17, 18] propose to interpolate the local and the global models. Personalization layer based approaches [19, 20, 21, 22] keep some layers locally for personalized models while aggregating the rest with FedAvg. In this work, we adapt this idea and design two variants for ASR.

However, existing personalized FL approaches have several limitations: (1) model parameters are communicated between server and clients, leading to high communication costs when models are large; (2) the low-resource client are required to do most computation while the computation on the high-resource server could be neglected; (3) clients data are required to be labeled despite it is troublesome. To tackle these challenges, we propose *decoupled federated learning (DecoupleFL)* for personalized FL in ASR. As shown in Fig. 1c, DecoupleFL decouples the training of the ASR model: the extractor that has contact with raw data is trained on clients (stage 1), and the classifier is trained on the server with the secure features from clients (stage 2). In this way, DecoupleFL communicates secure features instead of model parameters, thus reducing the communication cost. Then, some training burdens are moved to the server, thus reducing computation on clients. Additionally, DecoupleFL adopts pseudo-labeling (PL) approaches [23, 24] for unsupervised learning, avoiding the unrealistic labeled data assumption. Moreover, one potential concern is communicating features might lead to privacy leakage. To address such concern, we remove the speaker information from features with speaker-invariant training (SIT) [25, 26], which could protect speaker information while does not hamper performance.

Experiments show that compared with FedAvg, all personalized FL approaches (personalization layer based approaches and DecoupleFL) can reduce WER by 2.3% - 3.4%. Among them, DecoupleFL achieves the lowest communication and computation costs, which are 11.4% and 75% of FedAvg.

[†] Corresponding author.

This work is partially supported by the Youth Innovation Promotion Association, Chinese Academy of Sciences and the Frontier Exploration Project Independently Deployed by Institute of Acoustics, Chinese Academy of Sciences under Grant QYTS202011.

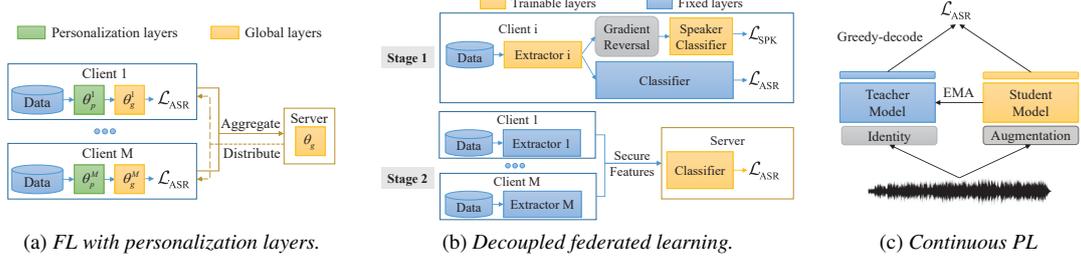


Figure 2: (a) FL with personalized layers. Personalization layers are kept locally while global layers are aggregated in the server. (b) DecoupleFL. In stage 1, extractors are trained locally with client’s data. In stage 2, the classifier is trained globally with secure features from clients. (c) Continuous pseudo-labeling. Pseudo labels are generated from teacher model, which is the EMA of the student model.

2. Proposed Approach

In this section, we first introduce FL with personalization layers approach and two proposed variants for ASR. Then, we describe the proposed DecoupleFL approach.

2.1. FL with Personalization Layers

A straightforward way for personalized FL is to keep some personalization layers locally and only aggregate other global layers as in Fig. 2a. Therefore, the personalization layers can learn the personalized perspective of clients, while global layers learn the common knowledge shared among clients.

We denote the parameters of personalization and global layers as θ_p and θ_g , respectively. Client’s index is k , training update is t , and local update number is E . Same with FedAvg, in each client, all layers are trained using stochastic gradient descent (SGD):

$$\left(\theta_{p,t+1}^k, \theta_{g,t+1}^k\right) \leftarrow \text{SGD}\left(\theta_{p,t}^k, \theta_{g,t}^k\right). \quad (1)$$

Then, when $\text{mod}(t, E) = 0$, the server aggregates global layers from all K clients and send them back to each client as:

$$\theta_{g,t+1}^k \leftarrow \frac{1}{K} \sum_{k=1}^K \theta_{g,t+1}^k. \quad (2)$$

It is intuitive to design two variants for ASR: FedNorm and FedExtract. FedNorm uses all normalization layers as personalization layers to reduce the distribution shift before each layer; FedExtract uses the bottom few layers (extractor) as personalization layers to reduce the distribution shift before the global classifier. However, they require communicating parameters, leading to high communication costs for large models. Since SGD is performed only on clients while parameter aggregation is performed on the server, the computation burden for low-resource clients is much larger than the high-resource server.

2.2. Decoupled Federated Learning

This paper proposes *DecoupleFL* to address the above challenges in personalized FL for ASR. Specifically, DecoupleFL decouples the model training into two stages: training of extractor on clients (stage 1) and classifier on the server (stage 2). Thus, it reduces the computation cost on clients and reduces the communication cost when ASR models are large. The procedure of DecoupleFL is shown in Fig. 2b. Note that DecoupleFL requires a pre-trained ASR model as the initialization.

2.2.1. Stage 1: Training of Extractor

In stage 1, the classifier θ_c is fixed and the extractors θ_e are optimized locally on client’s data. Therefore, similar with FedExtract, extractors are also used as personalization layers to tackle the non-IID distributions among clients. To remove speaker information from the features, we applied speaker-invariant training on the extractor. Specifically, we add a speaker classifier θ_s and train the entire model with the min-max objective:

$$\min_{\theta_e} \max_{\theta_s} \mathcal{L}_{\text{ASR}}(\theta_e, \theta_c) - \lambda \mathcal{L}_{\text{SPK}}(\theta_e, \theta_s) \quad (3)$$

Where \mathcal{L}_{ASR} is the ASR loss (CTC) and \mathcal{L}_{SPK} is the speaker classification loss (cross-entropy).

With this objective, the speaker classifier θ_s is optimized to minimize the speaker classification loss, while the extractor θ_e maximizes it. In this way, the extractor is trained to remove speaker information from features so that the speaker classifier cannot accurately classify speakers. Simultaneously, the extractor is optimized to minimize the ASR loss \mathcal{L}_{ASR} , which ensures the extractor could maintain decent ASR performance. This min-max optimization is implemented by inserting a gradient reversal layer between the extractor and speaker classifier [27].

2.2.2. Stage 2: Training of Classifier

In stage 2, each client’s features extracted with the personalized extractor from stage 1 are aggregated in the server to refine the common classifier. Since the personalized extractors are constrained by a fixed common classifier θ_c in stage 1, these features are well aligned. Thus it is reasonable to use these features to refine the common classifier. The optimization objective is:

$$\min_{\theta_c} \mathcal{L}_{\text{ASR}}(\theta_c) \quad (4)$$

where the classifier is optimized and extractors are not used.

2.2.3. Unsupervised Training with Continuous PL

We adopt continuous PL (shown in Fig. 2c) [23, 24] to compute the \mathcal{L}_{ASR} in both stage 1 and stage 2. Note that the continuous PL approach could also be used in other FL approaches like FedNorm and FedExtract. Given an unlabeled sample x , the ASR loss in the t -th round of update is computed as:

$$\mathcal{L}_{\text{ASR}}(\theta) = -\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \log p_{\theta_t}(\hat{\mathbf{y}} | a(\mathbf{x})), \quad (5)$$

where θ_t is the t -th student model, $a(\cdot)$ is the data augmentation function and $\hat{\mathbf{y}}$ denotes the pseudo label which is generated as:

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\text{argmax}} \log p_{\xi_t}(\mathbf{y} | \mathbf{x}), \quad (6)$$

where argmax denotes the greedy decoding. ξ_t is the teacher model in the t -th update, which is the exponential moving average of the student model θ_t :

$$\xi_t = \alpha \xi_{t-1} + (1 - \alpha) \theta_t \quad (7)$$

where α is the decay factor.

3. Experiments

3.1. Experimental Setup

We evaluate all approaches by adapting a baseline ASR model to an unseen target domain. The baseline ASR model consists of two layers of CNN and 14 layers of transformer. We use CNN and the bottom 7 transformer layers as extractors, while other layers are the classifier. The baseline ASR model is centralized trained in the server with CTC criterion on datasets in Table 1.

Table 1: Structure of labeled datasets for baseline model.

Dataset	Duration (Hours)
AMI	100
Fisher	1,761
SwitchBoard	317
LibriSpeech	960
Wall Street Journal (WSJ)	81
TED-LIUM v3	452
Internal Dataset	3,248
Total	6,919

The target domain dataset is constructed from the unseen Common Voice corpus [28]. Specifically, we select three accents from Common Voice: Australia (AU), England (EN), and India (IN), with a training/validation/test ratio of $100h : 10h : 10h$. Each accent accounts for 1/3 in each set. We take each accent as a local client to simulate the non-IID scenario.

All methods are trained for 100 epochs using Adam optimizer with the learning rate 10^{-4} and decay factor $\alpha = 0.9998$. For evaluation, we average the 10 best checkpoints and apply beam-search decoding with LM, where the 4-gram LM is trained with transcripts in Table 1.

3.2. Unsupervised Training Approach

Table 2: Comparison of PL approaches.

Method	AU	EN	IN	AVG
Baseline	19.4	15.7	22.6	19.2
Vanilla PL	17.6	15.2	20.6	17.8
Continuous PL	16.9	14.6	19.7	17.1

We compare continuous PL with vanilla PL [29, 30], where pseudo labels are generated with a fixed ASR model through beam-search with LM. As shown in Table 2, continuous PL consistently outperforms vanilla PL in centralized training. Thus, we use continuous PL for unsupervised learning as follows.

3.3. Main Results

We compare FedNorm, FedExtract and DecoupleFL with:

- *Baseline*: the unadapted baseline model.
- *Client*: separately adapts baseline model for each client.
- *Centralized*: centralized training.
- *FedAvg*: the standard FL, where the local epoch is 1.

As shown in Table 3, centralized outperforms client, indicating more training data helps even when non-IID. FedAvg

Table 3: Comparison of non-federated and federated learning approaches in terms of communication (comm.), computation cost, and word error rate (WER).

Method	Comm.	Computation		WER (%)			
		client	server	Australia	England	Indian	AVG
Baseline	-	-	-	19.4	15.7	22.6	19.2
Client	-	100	-	17.4	14.9	20.3	17.5
Centralized	-	-	100	16.9	14.6	19.7	17.1
FedAvg	20.70GB	100	-	17.3	14.9	20.4	17.5
FedNorm	20.69GB	100	-	16.9	14.6	19.7	17.1
FedExtract	10.71GB	100	-	16.8	14.5	19.8	17.0
DecoupleFL	2.35GB	50	0.5 * 50	16.5	14.5	19.7	16.9

only performs similarly with client due to the non-IID issue. FedNorm, FedExtract and DecoupleFL outperform FedAvg by alleviating the non-IID issue with personalized models. Among them, DecoupleFL requires much less communication cost (i.e., communicated data size) and decreases computation cost (i.e., training epochs) by 50% for clients and by 25% in total, which will be further discussed in Section 4.

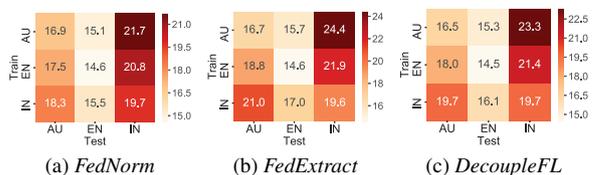


Figure 3: Illustration of personalization effect.

We illustrate the personalization effects of three personalized FL approaches in Fig. 3. The y-axis and x-axis denote the trained personalized client models and the client-specific testing set, respectively. All approaches perform best when we use the client model on this own testing set and degrades otherwise.

3.4. Ablation Study of DecoupleFL

In this section, we perform the ablation study to show the importance of each component in DecoupleFL. Note that each approach in the ablation study is trained for the same epochs.

Table 4: Ablation study of DecoupleFL

Method	AU	EN	IN	AVG
Baseline	19.4	15.7	22.6	19.2
DecoupleFL	16.5	14.5	19.7	16.9
- stage 1	17.3	14.7	20.3	17.4
- stage 2	16.7	14.9	20.3	17.3
- decouple	16.8	14.5	19.6	17.0

As shown in Table 4, when stage 1 is discarded, the classifier is only trained with features extracted by a shared extractor. The performance degrades due to the lack of personalization. On the other hand, when stage 2 is removed, each client optimizes a client-specific extractor while keeping the classifier fixed. The performance is also significantly degraded. Finally, we discard decoupled training and perform end-to-end optimization for the entire model, which is equivalent to centralized training with personalized extractors for each client. The results show that the decoupled optimization can achieve similar performance to the end-to-end optimization.

3.5. Privacy Protection in DecoupleFL

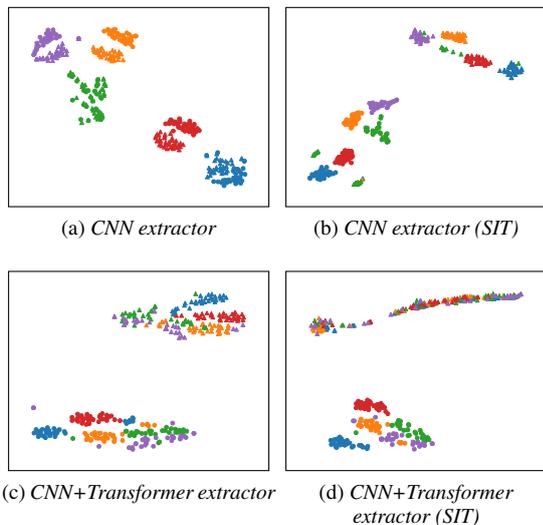


Figure 4: *t*-SNE Visualization of 5 speakers features from a client (EN). Unadapted and personalized features are denoted by circles and triangles, respectively. Each color represents a speaker. Best viewed in color and zoom in.

We illustrate the privacy protection properties for DecoupleFL by discussing attacks against it. The inversion attack [31] that aims to reconstruct the raw data could be avoided since the personalized extractors are only stored locally. However, it is unclear whether the attribute inference attack [32, 33] that tries to infer private attributes (like speaker identity) could be defended. We denote features extracted from unadapted and personalized extractors as *unadapted* and *personalized* features, respectively. There are two potential attacking strategies to infer speaker information from personalized features: (I) an attacker who has audios of a speaker tries to retrieve this speaker’s personalized features. Since the personalized extractors are invisible to the attacker, the attacker could only utilize the unadapted feature of this raw audio to find similar personalized features. To avoid this attack, the unadapted and personalized features of the same speaker should be sufficiently different. (II) the attacker tries to cluster the personalized features by speakers. To avoid this attack, the personalized features of the same speaker should not in the same neighborhood.

We illustrate the risks by visualizing unadapted and personalized features. We consider two types of extractors: CNN or CNN + bottom 7 transformer layers (CNN+Transformer).

As shown in Fig. 4, with CNN extractor, unadapted and personalized features of the same speaker are close, indicating attack I is easy. SIT can help separate the unadapted and personalized features. However, the CNN extractor is not powerful enough to learn speaker-invariant representation. Thus the attack II is still possible. With CNN+transformer extractor, the unadapted and personalized features are already separated without SIT, avoiding the attack I. And the personalized features become speaker-invariant with SIT, which avoids the attack II. Note that each client owns multiple speakers. Otherwise, the attack II could be done simply by tracing who sends the feature.

In conclusion, DecoupleFL can protect data privacy since (1) Raw data does not leave clients; (2) Personalized extractors

are stored in clients, thus avoiding the inversion attack; (3) SIT is used to protect the speaker identity in the features; (4) We could further improve the security of the features with homomorphic encryption [34] and differential privacy [35].

3.6. Selection of Extractor

We have illustrated that the CNN+Transformer extractor is better than CNN regarding privacy protection. Now we discuss the performance and computation of different extractors.

Table 5: The performance of DecoupleFL w. and w/o SIT exploiting different extractors

Extractor	SIT	AU	EN	IN	AVG
CNN	×	16.7	14.6	19.7	17
	✓	16.7	14.8	19.7	17.1
CNN+Transformer	×	16.5	14.5	19.7	16.9
	✓	16.4	14.7	19.8	17.0

For performance, as shown in Table 5, SIT does not hamper the performance clearly. And the performance with CNN+Transformer extractor slightly outperforms the one with CNN extractor. In terms of computation, no matter which extractor we select, the computation is the same in stage 1 since the trainable extractor is in the bottom of the model, which requires the forward and backpropagation for the entire model. Although the training memory in stage 1 could be decreased with a smaller extractor since only gradients of the extractor are kept in memory, the computation will correspondingly increase in stage 2 since a larger classifier is trained. Therefore, it is reasonable to use CNN+Transformer rather than CNN as the extractor for better privacy, performance and computation.

4. Discussions

We discuss the proposed approaches as follows.

Data privacy. Personalization layer based FL, i.e., FedNorm and FedExtract, follow the FedAvg paradigm, whose security property is thoroughly discussed in previous work [33, 36]. For DecoupleFL, we illustrate that it could protect privacy in subsection 3.5 and more discussions are required in future.

Communication cost. DecoupleFL transfers features once, while FedNorm and FedExtract transfer model parameters multiple times. Therefore, when data size per client is relatively small or the global epochs are large, DecoupleFL will have less communication cost and vice versa.

Computation cost. FedNorm and FedExtract perform training on clients and the computation on the server could be neglected. But DecoupleFL trains on clients for half epochs (stage 1) and transfers the other half to the server (stage 2). In the server, since we only train the classifier, computation is reduced by 50%, which results in the 25% reduction in total.

5. Conclusions

In this work, we tackle FL-based ASR in the non-IID scenario with personalized FL. Firstly, We adapt the personalization layer based personalized FL approach for ASR. Secondly, we propose DecoupleFL to reduce communication and computation costs. Experiments show that the two types of personalized FL approaches achieve lower WER than FedAvg. Among them, DecoupleFL has the lowest communication and computation cost. Furthermore, we show DecoupleFL can protect data privacy with SIT without performance compromise.

6. References

- [1] P. Voigt and A. Von dem Bussche, “The eu general data protection regulation (gdpr):” *A Practical Guide, 1st Ed.*, Cham: Springer International Publishing, vol. 10, p. 3152676, 2017.
- [2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.
- [3] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, “Fedhealth: A federated transfer learning framework for wearable healthcare,” *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 83–93, 2020.
- [4] D. Dimitriadis, K. Kumtani, R. Gmyr, Y. Gaur, and S. E. Eskimez, “A federated approach in training acoustic models,” in *Interspeech*, 2020, pp. 981–985.
- [5] C. Tan, D. Jiang, J. Peng, X. Wu, Q. Xu, and Q. Yang, “A de novo divide-and-merge paradigm for acoustic model optimization in automatic speech recognition,” in *IJCAI*, 2020, pp. 3709–3715.
- [6] X. Cui, S. Lu, and B. Kingsbury, “Federated acoustic modeling for automatic speech recognition,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6748–6752.
- [7] D. Guliani, F. Beaufays, and G. Motta, “Training speech recognition models with federated learning: A quality/cost framework,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3080–3084.
- [8] Y. Gao, T. Parcollet, S. Zaiem, J. Fernandez-Marques, P. P. de Gusmao, D. J. Beutel, and N. D. Lane, “End-to-end speech recognition from federated acoustic models,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7227–7231.
- [9] K. Nandury, A. Mohan, and F. Weber, “Cross-silo federated training in the cloud with diversity scaling and semi-supervised learning,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3085–3089.
- [10] T.-J. Yang, D. Guliani, F. Beaufays, and G. Motta, “Partial variable training for efficient on-device federated learning,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 4348–4352.
- [11] D. Guliani, L. Zhou, C. Ryu, T.-J. Yang, H. Zhang, Y. Xiao, F. Beaufays, and G. Motta, “Enabling on-device training of speech recognition models with federated dropout,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8757–8761.
- [12] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, “Federated learning with non-iid data,” *arXiv preprint arXiv:1806.00582*, 2018.
- [13] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh, “Three approaches for personalization with applications to federated learning,” *arXiv preprint arXiv:2002.10619*, 2020.
- [14] A. Fallah, A. Mokhtari, and A. Ozdaglar, “Personalized federated learning: A meta-learning approach,” *arXiv preprint arXiv:2002.07948*, 2020.
- [15] Y. Jiang, J. Konečný, K. Rush, and S. Kannan, “Improving federated learning personalization via model agnostic meta learning,” *arXiv preprint arXiv:1909.12488*, 2019.
- [16] T. Yu, E. Bagdasaryan, and V. Shmatikov, “Salvaging federated learning by local adaptation,” *arXiv preprint arXiv:2002.04758*, 2020.
- [17] Y. Deng, M. M. Kamani, and M. Mahdavi, “Adaptive personalized federated learning,” *arXiv preprint arXiv:2003.13461*, 2020.
- [18] S. Mdhaffar, M. Tommasi, and Y. Estève, “Study on acoustic model personalization in a context of collaborative learning constrained by privacy preservation,” in *International Conference on Speech and Computer*. Springer, 2021, pp. 426–436.
- [19] X. Li, M. JIANG, X. Zhang, M. Kamp, and Q. Dou, “Fedbn: Federated learning on non-iid features via local batch normalization,” in *International Conference on Learning Representations*, 2020.
- [20] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, “Federated learning with personalization layers,” *arXiv preprint arXiv:1912.00818*, 2019.
- [21] P. P. Liang, T. Liu, L. Ziyin, N. B. Allen, R. P. Auerbach, D. Brent, R. Salakhutdinov, and L.-P. Morency, “Think locally, act globally: Federated learning with local and global representations,” *arXiv preprint arXiv:2001.01523*, 2020.
- [22] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, “Exploiting shared representations for personalized federated learning,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 2089–2099.
- [23] Y. Higuchi, N. Moritz, J. L. Roux, and T. Hori, “Momentum Pseudo-Labeling for Semi-Supervised Speech Recognition,” in *Proc. Interspeech 2021*, 2021, pp. 726–730.
- [24] V. Manohar, T. Likhomanenko, Q. Xu, W.-N. Hsu, R. Collobert, Y. Saraf, G. Zweig, and A. Mohamed, “Kaizen: Continuously improving teacher using exponential moving average for semi-supervised speech recognition,” in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 518–525.
- [25] Z. Meng, J. Li, Z. Chen, Y. Zhao, V. Mazalov, Y. Gong, and B.-H. Juang, “Speaker-invariant training via adversarial learning,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5969–5973.
- [26] B. M. L. Srivastava, A. Bellet, M. Tommasi, and E. Vincent, “Privacy-preserving adversarial representation learning in asr: Reality or illusion?” in *Proc. Interspeech 2019*, 2019.
- [27] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” in *International conference on machine learning*. PMLR, 2015, pp. 1180–1189.
- [28] R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. Tyers, and G. Weber, “Common voice: A massively-multilingual speech corpus,” in *Proceedings of the 12th Language Resources and Evaluation Conference*, 2020, pp. 4218–4222.
- [29] D. S. Park, Y. Zhang, Y. Jia, W. Han, C.-C. Chiu, B. Li, Y. Wu, and Q. V. Le, “Improved noisy student training for automatic speech recognition,” *Proc. Interspeech 2020*, pp. 2817–2821, 2020.
- [30] Q. Xu, T. Likhomanenko, J. Kahn, A. Hannun, G. Synnaeve, and R. Collobert, “Iterative pseudo-labeling for speech recognition,” *Proc. Interspeech 2020*, pp. 1006–1010, 2020.
- [31] M. Fredrikson, S. Jha, and T. Ristenpart, “Model inversion attacks that exploit confidence information and basic countermeasures,” in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333.
- [32] T. Feng, H. Hashemi, R. Hebbur, M. Annavam, and S. S. Narayanan, “Attribute inference attack of speech emotion recognition in federated learning settings,” *arXiv preprint arXiv:2112.13416*, 2021.
- [33] N. Tomashenko, S. Mdhaffar, M. Tommasi, Y. Estève, and J.-F. Bonastre, “Privacy attacks for automatic speech recognition acoustic models in a federated learning framework,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6972–6976.
- [34] R. L. Rivest, L. Adleman, M. L. Dertouzos *et al.*, “On data banks and privacy homomorphisms,” *Foundations of secure computation*, vol. 4, no. 11, pp. 169–180, 1978.
- [35] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, “Deep learning with differential privacy,” in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.
- [36] L. Fowl, J. Geiping, W. Czaja, M. Goldblum, and T. Goldstein, “Robbing the fed: Directly obtaining private data in federated learning with modified models,” in *International Conference on Learning Representations*, 2022.