Interspeech 2022
18-22 September 2022, Incheon, Korea

Federated Domain Adaptation for ASR with Full Self-Supervision

Junteng Jia, Jay Mahadeokar, Weiyi Zheng, Yuan Shangguan, Ozlem Kalinli, Frank Seide

Meta AI

{juntengjia,jaym,wyz,yuansg,okalinli,seide}@fb.com

Abstract

Cross-device federated learning (FL) protects user privacy by collaboratively training a model on user devices, therefore eliminating the need for collecting, storing, and manually labeling user data. While important topics such as the FL training algorithm, non-IID-ness, and Differential Privacy have been well studied in the literature, this paper focuses on two challenges of practical importance for improving on-device ASR: the lack of ground-truth transcriptions and the scarcity of compute resource and network bandwidth on edge devices. First, we propose a FL system for on-device ASR domain adaptation with full self-supervision, which uses self-labeling together with data augmentation and filtering techniques. The system can improve a strong Emformer-Transducer based ASR model pretrained on out-of-domain data, using in-domain audio without any ground-truth transcriptions. Second, to reduce the training cost, we propose a self-restricted RNN Transducer (SR-RNN-T) loss, a variant of alignment-restricted RNN-T that uses Viterbi alignments from self-supervision. To further reduce the compute and network cost, we systematically explore adapting only a subset of weights in the Emformer-Transducer. Our best training recipe achieves a 12.9% relative WER reduction over the strong out-of-domain baseline, which equals 70% of the reduction achievable with full human supervision and centralized training.

Index Terms: ASR, Adaptation, Federated Learning

1. Introduction

With the increasing adoption of deep learning in artificial intelligence applications, data privacy is of growing concern [1, 2]. Automatic speech recognition (ASR) is a particularly sensitive use case due to the personalized nature of speech [3, 4]. Aiming to protect user privacy, cross-device federated learning (FL) has been proposed and applied to a variety of tasks [5, 6, 7, 8]. FL is a distributed training paradigm that allows a loose federation of trainers to collectively improve a shared model [9, 10, 11, 12].

Cross-device FL for ASR faces several unique challenges. These include the lack of ground truth transcriptions; the high costs of computation and network communication; the non independent and identical distribution of training data (non-IID-ness); and the difficulty of providing privacy guarantees. Several recent works have considered cross-device FL for ASR applications [13, 14, 15, 16, 17, 18]. In particular, weighted model averaging [13, 14] and federated variation noise [16] have been proposed to address the challenge of training on non-IID data, while differential privacy has been proposed to provide formal privacy guarantees [19, 20]. In this paper, however, we address the other two practical challenges for on-device FL: the lack of ground truth transcriptions, and the scarcity of compute resources and network bandwidth.

A common assumption in existing literature on FL for ASR is the availability of ground-truth transcriptions [13, 14, 15, 16, 17, 18]. However, this assumption is not practical. Users of on-device ASR have neither the incentive to manually transcribe their audio, nor the desire to frequently edit inaccurate automatic transcriptions. Assuming no labeled audio data, [11] proposes to use a large pretrained teacher model to label target domain audio on cloud servers; other works, not focused on ASR, have studied semi-supervised FL for keyword spotting, audio classification [21], and image classification [22, 23]. However, to the best of our knowledge, there has been limited literature that explored on-device adaptation for ASR with self-supervision. To this end, we propose an FL system for on-device model domain adaptation with self-supervision that runs alongside an ASR application.

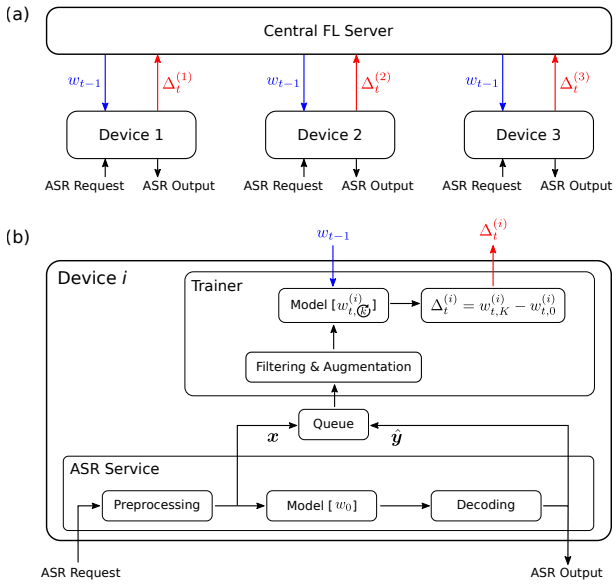

Figure 1: *A FL system for on-device ASR domain adaptation with self-supervision. Each device runs a local trainer alongside the ASR service.*

On a high level, user audio and respective automatic transcriptions (*pseudo labels*) from an on-device ASR model are stored temporarily on the device in an encrypted queue. Then, a trainer that runs alongside the ASR process consumes mini-batches of examples from the queue and performs local model updates. Model updates are periodically sent to a central FL server for aggregation. We develop data augmentation and filtering techniques to improve robustness of the self-supervised learning. Our offline experiments on two diverse FL adaptation scenarios confirm the effectiveness of the proposed system.

Another important challenge in cross-device FL is the cost of training and synchronization with the central FL server [10], since edge devices typically have limited compute, battery, and network resources [9]. In this work, we present a combination of techniques to improve an Emformer-Transducer based ASR model [24, 25], which is trained with the RNN-T loss [26]. First, to reduce the compute cost during on-device adaptation,

10.21437/Interspeech.2022-803

Algorithm 1: FL system for ASR domain adaptation.

Input: pretrained model w_0, # of rounds T, # of devices N, # of updates per round K
Output: finetuned model w_T

Procedure `CentralFLServer()`:
  for $t \in 1 \dots T$ **do**
    for $i \in 1 \dots N$ **do**
      `SendModel`(w_{t-1}, i, t)
      $\Delta_t^{(i)} \leftarrow$ `ReceiveUpdate`(i, t)
    $\Delta_t \leftarrow \frac{1}{N} \sum_{i=1}^{N} \Delta_t^{(i)}$
    $w_t \leftarrow w_{t-1} + \beta(w_{t-1} - w_{t-2}) + \Delta_t$

Procedure `Device`(i):
  $Q_i \leftarrow \{\}$
  `ASRService`(Q_i)
  `Trainer`(i, Q_i)

Procedure `ASRService`(Q_i):
  loop
    $\boldsymbol{x} \leftarrow$ `GetRequest()`
    $\hat{\boldsymbol{y}} \leftarrow$ `EncodeDecode`$(w_0, \boldsymbol{x})$
    `PushToQueue`$(Q_i, (\boldsymbol{x}, \hat{\boldsymbol{y}}))$

Procedure `Trainer`(i, Q_i):
  for $t \in 1 \dots T$ **do**
    $w_{t-1} \leftarrow$ `ReceiveModel`(i, t)
    $w_{t,0}^{(i)} \leftarrow w_{t-1}$
    for $k \in 1 \dots K$ **do**
      $B_k \leftarrow$ `PopBatchFromQueue`(Q_i)
      $\bar{B}_k \leftarrow$ `FilterAndAugment`(B_k)
      $w_{t,k}^{(i)} \leftarrow$ `ModelUpdate`$(w_{t,k-1}^{(i)}, \bar{B}_k)$
    $\Delta_t^{(i)} \leftarrow w_{t,K}^{(i)} - w_{t,0}^{(i)}$
    `SendUpdate`$(\Delta_t^{(i)}, i, t)$

we propose a self-restricted (SR)-RNN-T loss, which extends the alignment-restricted (AR)-RNN-T [27] by replacing the external alignment with a Viterbi alignment (similar to [28]) from the model itself. Secondly, to further reduce the network cost during synchronization, we consider only adapting a subset of model weights. Inspired by a recent work on speaker adaptation [29], we study which subset of weights in the ASR model would give the best domain-adaptation performance. We find that adapting the key/value matrices in the self-attention module can match (or surpass) full model adaptation, even though those matrices only account for 5.7% model weights. A related approach called partial variation training adapts different subset of weights on different devices [30].

Combining all techniques, we achieve 12.9% relative WER reduction for on-device adaptation — 70% of the reduction achievable with human supervision and centralized training.

2. Methodology

2.1. Federated Learning Alongside the ASR Process

This section presents the design of the proposed FL system. An initial ASR model is pretrained on large amounts of supervised data from a broad range of sources, using ground truth transcriptions or high-quality machine generated transcriptions from a large offline teacher model. Then, the FL system adapts the pretrained ASR model to target domain data using pseudo labels

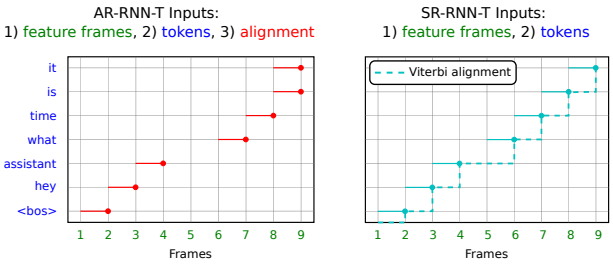

Figure 2: *Illustration of the proposed SR-RNN-T loss. In contrast with AR-RNN-T, which relies on an external alignment (red), SR-RNN-T uses the Viterbi forced-alignment (cyan).*

generated by the pretrained model itself, via a hierarchical SGD method known in the FL community as *Federated Averaging with Momentum*, abbreviated as FedAvgM [31], although the same method was first developed in the speech community under the name *blockwise model-update filtering*, or BMUF [32].

Figure 1 illustrates the cross-device FL system, comprising a central FL server and many ASR devices, implementing Algorithm 1. Each device i executes an ASR process that runs the pretrained model w_0, which is fixed throughout the FL process. A device-specific trainer manages a second copy of the ASR model $w_{t,k}^{(i)}$ that is updated via FL. Each time the device receives an ASR request, the pretrained model is used to transcribe audio $\boldsymbol{x}$ to generate transcription $\hat{\boldsymbol{y}}$, which we refer as *pseudo labels* (this differs from [11] which uses a large offline teacher model for this purpose). The $(\boldsymbol{x}, \hat{\boldsymbol{y}})$ pairs are added to a queue for later consumption by the trainer. In each local model update, the trainer samples a minibatch from the queue and computes model gradients with the SR-RNN-T loss. We find that it is important for adaptation performance to use confidence filtering and data augmentation. Finally, after K updates, the difference $\Delta^{(i)}$ is sent back to the central FL server for aggregation.

2.2. Efficient RNN-T Adaptation with Self-Alignment

We use an Emformer-Transducer [26, 33] as our ASR model, pretrained with AR-RNN-T loss and centralized distributed data-parallel (DDP) training. Let $\boldsymbol{x}$ denote the audio features over T time steps and $\boldsymbol{y}$ denote the reference text over U target tokens. First, the encoder maps the audio features $\boldsymbol{x}$ into input embeddings $\boldsymbol{h}$, and the predictor maps the reference tokens $\boldsymbol{y}$ into output embeddings $\boldsymbol{g}$. Then, a joiner network combines every pair of $h_t, g_u \in \boldsymbol{h}, \boldsymbol{g}$ to compute the probability distribution $\Pr(\cdot|h_t, g_u)$ over all possible output tokens including blank. During training, the RNN-T loss optimizes the sum of the probabilities over all possible audio-text alignments. During inference, a beam search algorithm is used to output the most probable sequence, which we refer to as $\hat{\boldsymbol{y}}$.

In FL scenario, $\boldsymbol{y}$ is not available. Instead, we use the most probable sequence $\hat{\boldsymbol{y}}$ from inference as target labels. A straightforward implementation of the RNN-T loss would directly use $\boldsymbol{x}$ and $\hat{\boldsymbol{y}}$ and optimize over all possible alignments. However, this approach is computationally intensive. As a potential alternative, [27] showed that the AR-RNN-T loss can greatly reduce memory and compute by restricting the forward-backward algorithm to a band around a ground truth alignment. However, AR-RNN-T relies on external ground-truth alignments generated offline with a separate hybrid model. Motivated by some recent studies [28], we propose to use Viterbi forced-alignment from the model itself in place of an external alignment. Particularly, the forced-alignments are computed on the fly over the RNN-T grid using the backtracing (see Fig. 2). We refer

the resulting loss function as self-restricted (SR)-RNN-T loss. In our design of the FL system, the Viterbi forced-alignment is further approximated with the most probable alignment encountered during beam search decoding after pruning.

2.3. Data Filtering and Augmentation

One main concern with self-supervision is the quality of the pseudo labels $\hat{\boldsymbol{y}}$. Incorrect pseudo labels carry the risk of reinforcing errors. We find an effective mitigation to be data filtering, where all utterances with $\log \Pr(\hat{\boldsymbol{y}}) < \theta$ are discarded. The threshold θ is a tunable hyper-parameter, which is empirically determined on a small development dataset. We find that data filtering is especially important if adaptation domain data is noisy (see section 3.4).

On the other hand, when the target domain audios are clean, we find the semi-supervised self-training can be further boosted with data augmentation (speed perturbation and additive noise) while keeping the pseudo labels unchanged (see section 3.3).

2.4. Finetuning Only a Subset of Model Weights

To reduce the cost of computation, memory, and communication, we consider adapting and synchronizing only a subset of model weights. Inspired by a recent study [29], we systemically studied which subset of weights in the Emformer-Transducer model provides the best trade-off between domain-adaptation performance vs. communication cost. We explore adaptation of encoder, predictor and joiner separately. Within the encoder, we carefully analyze adapting only attention or attention key/value matrices. We find adapting the key/value matrices is very effective for the target domain, and freezing the rest of weights serves as a regularization technique, which prevents catastrophic forgetting on the source domain.

3. Experiments

Table 1: *Different experiment setups for simulating diverse set of FL experiments. The 1st row in each setup denotes the source domain data used for model pretraining, and 2nd row denotes the target domain data used for FL adaptation.*

Setup	Dataset	Train (hours)	Eval (utterances)
Device	Mixture	1.5M	3.6K
	Edge	9.7K	28.6K
Video	Clean	17.4K	1.2K
	Noisy	1.2K	1.5K

3.1. Data Setup

To simulate realistic federated learning in practical scenarios, we consider two different experiment setups. Each setup takes a converged model pretrained on a source domain and performs FL adaptation to a target domain. The adapted model is tested on the evaluation data from both the source and target domains. Table 1 gives more information on the data used in each setup.

Device: This setup closely resembles a cross-device FL use case. For pretraining, we use a mixture of data sources, including public videos, audios collected by paid third-party vendors on portal, and synthetic audios generated by text-to-speech. This dataset contains 1.5 Million hours of audios, which are transcribed either through human annotators or a large teacher model (see [34] for details). For adaptation, we use assistant and dictation audios recorded on edge devices, which are collected by external vendors under clean acoustic conditions. The source domain evaluation data is a separately prepared conversation dataset, while the target domain evaluation data is a held out subset of the on-device recordings.

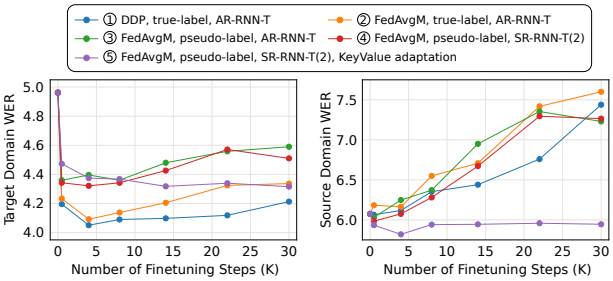

Figure 3: *Convergence of the Device adaptation experiments. The combination of FedAvgM/BMUF, pseudo-labels, SR-RNN-T and KeyValue adaptation not only achieves a low WER on the target domain, but also prevents catastrophic forgetting on the source domain.*

Video: For pretraining, we use 17.4K hours of long form videos (averaged 45 sec) with relative clean audios. For adaptation, we use short form videos (averaged 12 sec) that might be recorded in noisy environments. A subset of the audios from both the clean and noisy videos are held out for evaluation.

3.2. Modeling

Our ASR model is an Emformer-Transducer [33]. Features are 80 mel-spaced log filter-bank channels with a sliding window of 25ms at 10ms frame shift, regularized with SpecAugment [35]. A time reduction layer stacks features with a stride of 6, and the resulting 480-dimensional vectors form the input to the encoder. The predictor is a 2-layer LSTM model with a hidden dimension of 256. Both the encoder and the predictor outputs are projected to 768-dimension tensors and combined in the RNN-T joiner, which consists of a ReLU followed by a softmax layer with a sentence-piece vocabulary size of 4096.

For each task, we first pretrain the ASR model on the source domain until convergence, using an AR-RNN-T loss with buffer size $b_l = 5, b_r = 25$ and an Adam optimizer with learning rate 10^{-3} and tri-stage learning rate scheduling. We then finetune the ASR model to the target domain using 32 parallel workers. Each worker uses an Adam optimizer with a fixed learning rate of 10^{-4} for local model updates. The local models on different workers are synchronized every 20 local updates using FedAvgM/BMUF with a block momentum of 0.8.

3.3. Device Adaptation

The Device adaptation setup adapts from the source domain "Mixture" to the target domain "Edge". The baseline pretrained model achieves a lower WER on the target domain (4.96) than on the source domain (6.07), since the latter captures diverse acoustic conditions while the former is a commissioned data collection that is mostly clean.

Lower bound: We first consider supervised adaptation with centralized DDP training, using the ground-truth labels and the AR-RNN-T loss. The adapted model's WER on the source and target domains across different checkpoints are plotted in figure 3 ①. In particular, the best checkpoint (see table 2 ①) reduces the WER $4.96 \rightarrow 4.05$ on the target domain, which serves as a lower bound for semi-supervised FL adaptations. We also observe a slight WER increase $6.07 \rightarrow 6.12$ on the

Table 2: *Summary of Device adaptation experiments.*

Model	Target	Source
Pretrained	4.96	6.07
① (DDP, True Labels, AR-RNN-T)	4.05	6.12
② (① + FedAvgM/BMUF)	4.09	6.16
③ (② + Pseudo Labels)	4.35	6.17
④ (③ + SR-RNN-T)	4.32	6.08
⑤ (④ + KeyValue Adaptation)	4.32	5.95

source domain.

FedAvgM/BMUF: Next, we look at the effect of using the FL training algorithm, by replacing DDP with FedAvgM/BMUF while keeping the labels and the loss function the same (②). We observe a marginal WER increase on both the target domain (to 4.09) and the source domain (to 6.16).

Table 3: *The effect of data augmentation. When target domain is clean, adaptation is less effective without data augmentation.*

Model	Target	Source
③	4.35	6.17
③ - Data Augment	4.48	6.44
③ - Data/Spec Augment	4.89	6.67

Pseudo labels: Now, we introduce pseudo labels to determine the effects of self-supervision (③). Although not as effective as the true labels, self-supervision still reduces the pretrained model's WER on the target domain $4.96 \rightarrow 4.35$. We note the data filtering techniques discussed in section 2.3 greatly contributes to the success of self-supervision. Without filtering, adaptation with the pseudo labels becomes much less effective, causing a WER increase on the target domain $4.35 \rightarrow 4.48$ (see table 3). If we also remove SpecAugment, the target domain WER further increases to 4.89. On the other hand, confidence filtering makes little difference since the target domain WER of the pretrained model is already low.

SR-RNN-T: To reduce on-device computation cost, the proposed SR-RNN-T loss restricts the forward/backward to a limited window of 2 frames around the most probable alignment ($b_l = b_r = 2$) according to the respective current model. The SR-RNN-T trained model is marginally more accurate than the AR-RNN-T on both target (4.32) and source domain (6.08).

KeyValue adaptation: Lastly, we experiment with finetuning only subsets of the model weights (see table 4). Finetuning only the key/value matrices of the self-attention module not only achieves a low target domain WER (4.32), but also prevents catastrophic forgetting on the source domain as the number of updates increases (see figure 3). Most importantly, communication is greatly reduced as key/value matrices only account for

Table 4: *Adapting a subset of model weights. Finetuning only the key/value weights gives the best adaptation performance.*

Adaptation	Params (M)	Target	Source
All (④)	72.4 (100%)	4.32	6.08
Encoder	67.4 (93%)	4.31	6.15
Attention	8.2 (11%)	4.35	6.16
KeyValue (⑤)	4.1 (5.7%)	4.32	5.95
Predictor	1.7 (2.5%)	4.68	6.19
Joiner	3.1 (4.4%)	4.70	6.20
Bias	0.2 (0.3%)	4.35	5.75

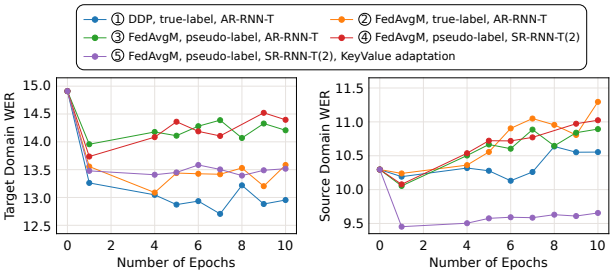

Figure 4: *Convergence of Video adaptation experiments.*

Table 5: *Summary of Video adaptation experiments.*

Model	Target	Source
Pretrained	14.91	10.30
① (DDP, True Labels, AR-RNN-T)	12.71	10.26
② (① + FedAvgM/BMUF)	13.09	10.36
③ (② + Pseudo Labels)	13.89	10.33
④ (③ + SR-RNN-T)	13.74	10.08
⑤ (④ + KeyValue Adaptation)	13.36	9.43
⑥ (③ - Filtering)	14.72	11.09

5.7% of all weights. Similarly, finetuning only the bias also has the same benefits, however, it takes more updates to converge.

3.4. Video Adaptation

The Video adaptation setup adapts from the source domain "Clean" to the target domain "Noisy". The adaptation results are summarized in figure 4 and table 5, and they confirms most of our conclusions in section 3.3. In particular, FL adaptation of key/value matrices with pseudo labels (⑤) effectively reduces the WER on the target domain ($14.91 \rightarrow 13.36$). Moreover, we have two additional learnings.

Filtering: Since the audios in the target domain is noisy, the pseudo labels from the pretrained model have a significant error rate. We find it necessary to apply confidence filtering using the decoding scores to remove low-confidence examples. The optimal threshold parameter $\theta = -0.2$ removes 27% of training examples. As shown in table 5 row ⑥, semi-supervised adaptation is much less effective when confidence filtering is removed, which negates most of the gains ($13.89 \rightarrow 14.72$).

KeyValue adaptation: Adapting only the key/value weights outperforms full model adaptation by large margins on both target (13.36 vs. 13.74) and source domain (9.43 vs. 10.08). Moreover, the adapted model even outperforms the pretrained model on the source domain (9.43 vs. 10.30). This is likely because the target domain contains more diverse audio.

4. Conclusions

This paper addressed two practical challenges for domain adaptation of an Emformer-Transducer ASR model on edge devices using Federated Learning. First, we show that the combination of self-restricted RNN-T loss with data augmentation and filtering techniques can improve an on-device ASR model on target domain data with full self-supervision. Second, we explored techniques reducing the FL compute and network communication costs by adapting only a subset of weights in the model. We will further explore effects of non-IID-ness and differential privacy on our proposed techniques as part of future work.

5. References

[1] F. Mireshghallah, M. Taram, P. Vepakomma, A. Singh, R. Raskar, and H. Esmaeilzadeh, "Privacy in deep learning: A survey," *arXiv preprint arXiv:2004.12254*, 2020.

[2] T. Ha, T. K. Dang, H. Le, and T. A. Truong, "Security and privacy issues in deep learning: a brief review," *SN Computer Science*, vol. 1, no. 5, pp. 1–15, 2020.

[3] B. Schuller and A. Batliner, *Computational paralinguistics: emotion, affect and personality in speech and language processing*. John Wiley & Sons, 2013.

[4] D. Cherkassky, "The voice privacy problem," Kardome, Blog, 2021. [Online]. Available: https://www.kardome.com/blog-posts/voice-privacy-concerns

[5] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, and F. Beaufays, "Applied federated learning: Improving google keyboard query suggestions," *arXiv preprint arXiv:1812.02903*, 2018.

[6] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[7] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, and J. Dureau, "Federated learning for keyword spotting," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6341–6345.

[8] S. Ramaswamy, R. Mathews, K. Rao, and F. Beaufays, "Federated learning for emoji prediction in a mobile keyboard," *arXiv preprint arXiv:1906.04329*, 2019.

[9] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[10] 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.

[11] 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.

[12] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[13] D. Dimitriadis, K. Kumatani, R. Gmyr, Y. Gaur, and S. E. Eskimez, "A federated approach in training acoustic models." in *Interspeech*, 2020, pp. 981–985.

[14] 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," *arXiv preprint arXiv:2104.14297*, 2021.

[15] W. Yu, J. Freiwald, S. Tewes, F. Huennemeyer, and D. Kolossa, "Federated learning in asr: Not as easy as you think," in *Speech Communication; 14th ITG Conference*. VDE, 2021, pp. 1–5.

[16] 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.

[17] 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.

[18] M. Paulik, M. Seigel, H. Mason, D. Telaar, J. Kluivers, R. van Dalen, C. W. Lau, L. Carlson, F. Granqvist, C. Vandevelde *et al.*, "Federated evaluation and tuning for on-device personalization: System design & applications," *arXiv preprint arXiv:2102.08503*, 2021.

[19] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy." *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014.

[20] 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.

[21] V. Tsouvalas, A. Saeed, and T. Ozcelebi, "Federated self-training for semi-supervised audio recognition," *arXiv preprint arXiv:2107.06877*, 2021.

[22] W. Jeong, J. Yoon, E. Yang, and S. J. Hwang, "Federated semi-supervised learning with inter-client consistency," *arXiv preprint arXiv:2006.12097*, 2020.

[23] C. He, Z. Yang, E. Mushtaq, S. Lee, M. Soltanolkotabi, and S. Avestimehr, "Ssfl: Tackling label deficiency in federated learning via personalized self-supervision," *arXiv preprint arXiv:2110.02470*, 2021.

[24] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6381–6385.

[25] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 193–199.

[26] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[27] J. Mahadeokar, Y. Shangguan, D. Le, G. Keren, H. Su, T. Le, C.-F. Yeh, C. Fuegen, and M. L. Seltzer, "Alignment restricted streaming recurrent neural network transducer," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 52–59.

[28] J. Kim, H. Lu, A. Tripathi, Q. Zhang, and H. Sak, "Reducing streaming asr model delay with self alignment," *arXiv preprint arXiv:2105.05005*, 2021.

[29] Y. Huang, G. Ye, J. Li, and Y. Gong, "Rapid speaker adaptation for conformer transducer: Attention and bias are all you need," *Proc. Interspeech 2021*, pp. 1309–1313, 2021.

[30] T.-J. Yang, D. Guliani, F. Beaufays, and G. Motta, "Partial variable training for efficient on-device federated learning," *arXiv preprint arXiv:2110.05607*, 2021.

[31] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv preprint arXiv:1909.06335*, 2019.

[32] K. Chen and Q. Huo, "Scalable training of deep learning machines by incremental block training with intra-block parallel optimization and blockwise model-update filtering," in *2016 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2016, pp. 5880–5884.

[33] Y. Shi, Y. Wang, C. Wu, C.-F. Yeh, J. Chan, F. Zhang, D. Le, and M. Seltzer, "Emformer: Efficient memory transformer based acoustic model for low latency streaming speech recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6783–6787.

[34] A. Xiao, W. Zheng, G. Keren, D. Le, F. Zhang, C. Fuegen, O. Kalinli, Y. Saraf, and A. Mohamed, "Scaling asr improves zero and few shot learning," *arXiv preprint arXiv:2111.05948*, 2021.

[35] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.