FEDERATED SELF-SUPERVISED LEARNING FOR ACOUSTIC EVENT CLASSIFICATION

Meng Feng[1*], *Chieh-Chi Kao*[2], *Qingming Tang*[2], *Ming Sun*[2], *Viktor Rozgic*[2], *Spyros Matsoukas*[2], *Chao Wang*[2]

Massachusetts Institute of Technology[1] Amazon.com Inc[2]

ABSTRACT

Standard acoustic event classification (AEC) solutions require large-scale collection of customer data from client devices for model optimization. However, they inevitably suffer from the risks of compromising customer privacy. Federated learning (FL) is a compelling framework that decouples data collection and model training to protect customer privacy. In this work, we investigate the feasibility of applying FL to improve AEC performance under a strict constraint that no customer data can be directly uploaded to the server. We assume no pseudo labels can be inferred from on-device user inputs, aligning the typical use cases of AEC. We adapt self-supervised learning to the FL framework for on-device continual learning of representations. By training representation encoders on a growing and increasingly diverse pool of local customer data, we demonstrate that it results in improved performance of the downstream AEC classifiers without labeled/pseudo-labeled data available. Compared to the baseline w/o FL, the proposed method improves precision up to 20.3% relatively while maintaining the recall. Our work differs from prior work in FL that our approach does not require user-generated learning targets, and we use internal data from Amazon Alexa to maximally simulate the production settings.

***Index Terms*—** Federated learning, representation learning, self-supervised learning, acoustic event classification

1. INTRODUCTION

Acoustic event classification (AEC) is a task of automatically detecting the occurrence of a set of events within the sound clips recorded from the target environments. The target events can range from a pre-engineered list such as baby crying and dog barking to event types specified by the customers themselves. AEC has played an important role in a wide range of applications in the domain of surveillance [1] [2] and recommendation systems [3]. It has been conventionally studied with classical speech recognition techniques [4] [5] [6] and more recently overtaken by deep learning algorithms [7] [8] [9] [10] thanks to the advancements in machine learning. Recent state-of-the-art works on AEC commonly need to collect a large set of data to on the server to support complex model optimization routines. The strong coupling between data and model under the centralized optimization framework exposes significant privacy risks. For applications that involve highly sensitive data (e.g. smart speakers in a household), compromising on customer data privacy can be a deal-breaker.

Federated learning (FL) provides a compelling alternative framework to achieve this goal. FL is a distributed learning framework that exploits distributed resources to collaboratively train a machine learning model [11]. Thanks to the decoupling of data and model, it is able to keep the sensitive training data locally at the participating devices and never collect them centrally. Numerous recent successes [12] [13] have shown the viability of applying FL to boost privacy preservation as well as offer competitive model performance. However, these work assumes access to data annotations directly from user inputs. Unfortunately, users rarely have any interaction with the client devices in a typical AEC setting. Consequently, it is difficult to obtain data annotations directly from customers. In this work, we assume no annotated data are available besides a small annotated dataset from internal users. Since the classifier models naïvely trained from this dataset are likely not able to generalize well for general public users, it is necessary to take advantage of the customer data locally stored on client devices for the learned models to generalize to more client users.

In this paper, we apply federated learning to improve the performance of realistic AEC tasks under the privacy constraint. The goal is to improve AEC model precision and generalization to unseen client users after the deployment of the initial model trained on the small annotated dataset. We propose a self-supervised federated learning framework that learns improved representations from the un-labeled audio data stored on local client devices. Our empirical findings show that improvement of learned representations after federated learning can lead to improvement of classification performance even the classifiers are trained on the same annotated dataset. Unlike prior work done on public datasets [14], we conduct our experiments with internal data collected from Amazon employees only. Our dataset closely resembles the non-independent and identically distributed (IID) distribution of data and devices from highly realistic production settings, which no public datasets for AEC [15] [16] can simulate.

[*]The work was done during Meng's internship at Amazon.

2. RELATED WORK

There has been a great volume of work on learning when labeled data is scarce. A common class of solutions focuses on learning condensed and generalizable high-level representations from the surface features such as raw audio waveforms or spectrograms. For example, representations can be learned from autoregressive predictive coding (APC) [17] [18] [19], PASE [20] [21], or triplet loss [22] [23]. These self-supervised learning techniques may benefit downstream tasks such as AEC. Small encoder models [24] applicable for mobile devices also share similar findings. Our work builds on top of [18] to extract high-level features via federated learning.

Federated learning is able to indirectly learn from an increasing amount of data under the privacy constraints as opposed to being limited to a fixed dataset. Recent work [12] [13] [25] [26] [27] shows that FL can output competitive models when the learning targets are given via supervised learning methods. However, these works either assume available training targets from user inputs or perform study in a different field. Perhaps the closest approach to ours is [14], in which federated self-supervised learning is applied to learn representations from multi-sensor data. Since its users are simulated by randomly dividing the training set, its experiment cannot simulate the non-IID distribution of data and devices. In addition, its downstream task uses a linear classifier which severely lacks expressiveness. In comparison, we conduct our experiment on internal datasets, where the partitioning of users and devices is real rather than simulated. We simulate the assumptions from realistic production settings that none of our customer data are uploaded or accessible. We use federated self-supervised learning to improve learned representations, and we show that the improvement in representation learning translates to improvement in event classification performance with no addition of labeled data. To the best of our knowledge, this work is the first to apply self-supervised FL to AEC problems on an industrial-level dataset under realistic assumptions. Our work provides clear evidence of the viability of FL when data labels are unavailable.

3. METHODS

Given an audio signal $\boldsymbol{x} = (x_1, x_2, ..., x_N)$, where N is the length of a full utterance, we consider the task to train a predictor $\boldsymbol{f}$ make a binary prediction $\boldsymbol{z} \in \{0, 1\}$ on whether a certain event presents in $\boldsymbol{x}$. We denote $D_{Server} = \{(\boldsymbol{x}, \boldsymbol{z})\}$ as the fully annotated dataset on the server and $D_{Client} = \{\boldsymbol{x}\}$ as the unlabeled client dataset stored in the client devices from customer client devices $S = (S_1, ..., S_K)$.

As shown in Fig 1, we first train an APC model that consists of an encoder $g_{enc} : x \rightarrow h, x \in \mathbb{R}^n, h \in \mathbb{R}^m$ and a decoder $g_{dec} : h \rightarrow x, h \in \mathbb{R}^m, x \in \mathbb{R}^n$, where m, n are the dimension of the latent feature vector and post-processed

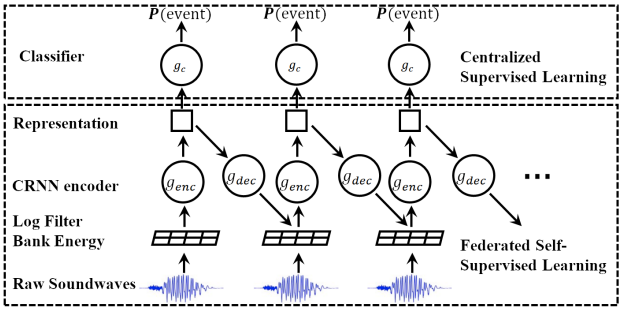


Fig. 1: Our proposed model architecture. We decouple representation learning and downstream training of classifiers. We first train a feature encoder g_{enc} to encode input signals (e.g. LFBE) to latent feature space. We apply federated learning to expose models to an increasing pool of users for benefits in the generalization of the learned representations. We then apply conventional supervised learning on the server using server data to fine-tune the classifier.

input (e.g., LFBE) respectively. The decoder then uses the feature vector as the input to predict a future frame x_{i+n}, where n is the number of steps the prediction is ahead of x_i. We optimize the L1 reconstruction loss between the predicted sequence $\boldsymbol{y} = (y_1, y_2, ..., y_N)$ and the target sequence $\boldsymbol{t} = (x_{1+n}, x_2, ..., x_{N+n})$. w_{enc} and w_{dec} are the parameters of the encoder and decoder respectively.

To take advantage of locally stored data D_{client}, we apply the *FederatedAveraging* algorithm [28]. We first train an initial APC model $\mathcal{M}_0$ from an annotated dataset. This dataset is assumed to be collected from internal testing participants who explicitly agree to upload their data to the server. $\mathcal{M}_0$ is sent to all client devices and serves as the global starting point for federated learning at $t = 0$. Each of the participating client devices in a given round of communication accumulates a local dataset $\mathcal{D}_k$, where k is the index of the client device. The size of dataset D_k may vary from device to device. Each client device optimizes its local model on its local dataset by running stochastic gradient descent (SGD) on L1 loss. The model weights of a selected set of client devices are uploaded back to the server at the end of the communication. The server aggregates the weights of the models to obtain a new global model $\mathcal{M}_1$. $\mathcal{M}_1$ is sent and optimized on the participating devices in the next round of communication. This process repeats to incorporate an increasing amount of decentralized data in training the global model $\mathcal{M}$. In essence, after adopting the federated learning framework the loss function can be written as Eq 1, where $n_k = |\mathcal{D}_k|$ and $n = \sum_{k=1}^{K} n_k$.

$$\begin{aligned} &\min_{w_{enc}, w_{dec}} L(w_{enc}, w_{dec}) \\ \text{s.t.,} \quad &L(w_{enc}, w_{dec}) = \sum_{k=1}^{K} \frac{n_k}{n} l(w_{enc}, w_{dec}) \\ &l_{w_{enc}, w_{dec}} = \sum_{i=1}^{N-n} |x_{i+n} - y_i| \end{aligned} \tag{1}$$

Server executes
initialize APC model weights $w = (w_{enc}, w_{dec})$
initialize the classification-layer parameter w_c
// stage I: pre-train
train w_0 on $\mathcal{D}_{server}$
// stage II: federated self-supervised learning
for each round $t = 1, 2, ...$ **do**
  $S_t \leftarrow$ (random set of m clients)
  for each client $k \in S_t$ **in parallel do**
    $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
  end for
  $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$
end for
// stage III: train classifier on $\mathcal{D}_{server}$
Fix w_{enc}
for $\mathcal{B}$ in $\mathcal{D}_{server}$ **do**
  $h \leftarrow g_{enc}(\mathcal{B})$
  $p' \leftarrow g_c(h)$
  $w_c \leftarrow w_c - \eta \nabla l_c(w_c, p; \mathcal{B})$
end for
ClientUpdate(k**,** w**)**:
$\mathcal{B} \leftarrow$ (split $\mathcal{D}_k$ into batches of size B)
for each local epoch i from 1 to E **do**
  for batch $b \in \mathcal{B}$ **do**
    $w \leftarrow w - \eta \nabla l(w; b)$
  end for
end for

Algorithm 1: `Federated Self-Supervised Federated Learning (FSSL)`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Applying federated learning to train $\mathcal{M}$ on client data can improve the generalization of the learned feature encoders to customers who are not in the internal testing program. Once $\mathcal{M}$ converges, its parameters are frozen.

To predict event occurrences, we train a classifier $g_c : h \rightarrow p, h \in \mathbb{R}^m, p \in \mathbb{R}$ which takes the input of the encoded feature vectors and outputs the binary prediction $\boldsymbol{z} \in \{0, 1\}$. The classifiers are trained with a modified binary cross-entropy (BCE) loss seen in Eq (2),

$$l_c = -\left[c \cdot z_n \cdot \ln p_n + (1 - z_n) \ln(1 - p_n)\right] \qquad (2)$$

where c is a positive scaling factor to adjust the loss for positive samples. In this paper, we conduct our experiment on single event classification. However, the same procedures can be easily extended to multi-class classification problems by adding an additional binary classifier for each new class of event following a one-vs-all paradigm.

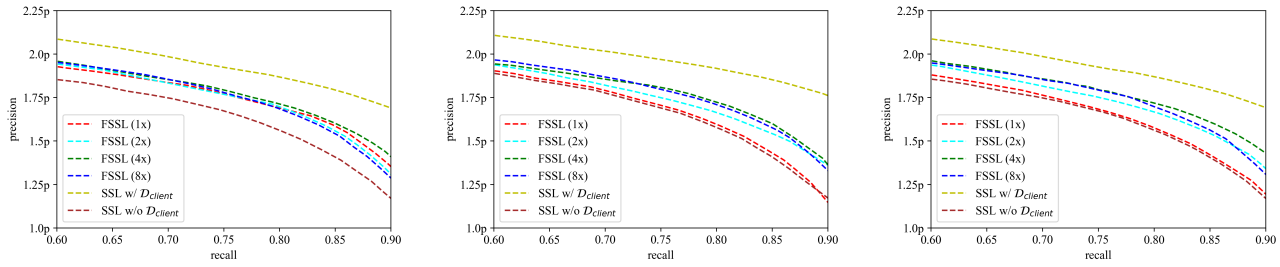
4. EXPERIMENTS

Data The data we use is collected from our internal Beta program, which consists of only Amazon employees. In our primary study, we use fully annotated data from March to July 2021. It consists of 28,069 unique device numbers (DSNs) and 330,412 audio clips. Each audio clip is 10-second long and contains information on the timestamp and the DSN of the source device. The annotated data from March 2021 are used to simulate the server data $\mathcal{D}_{server}$. The remaining data from April to July 2021 are used to simulate the client data $\mathcal{D}_{client}$. Under this simulation setting, $\mathcal{D}_{server}$ is analogous to participants in the internal testing program, and $\mathcal{D}_{client}$ is analogous to production users with privacy constraints after product launch. Let $\Theta(\mathcal{D})$ be the set of unique DSNs in a dataset $\mathcal{D}$, and we define three subsets of users $\mathcal{I} = \Theta(\mathcal{D}_{server}) \cap \Theta(\mathcal{D}_{client})$, $\mathcal{U} = \Theta(\mathcal{D}_{server})^C \cap \Theta(\mathcal{D}_{client})$, and $\mathcal{T} = \Theta(\mathcal{D}_{server})^C \cap \Theta(\mathcal{D}_{client})^C$ Intuitively, $\mathcal{I}$ corresponds to the users who participated in both the internal testing program and post-deployment product improvement program, $\mathcal{U}$ points to the users who only participated in the post-deployment product improvement program, and $\mathcal{T}$ represents the users who were in neither of the programs. In the result section, we report our model performances tested on these three partitions $\mathcal{D}_{\mathcal{I}}$, $\mathcal{D}_{\mathcal{U}}$, and $\mathcal{D}_{\mathcal{T}}$ for the "dog barking" event respectively to analyze the impact of applying federated learning on in-distribution and out-of-distribution data samples. We assume the device is communicated with the server every 24 hours after running the **ClientUpdate** routine. After each communication, the data stored on the client devices is cleared due to memory constraints of the client device. In our ablative study, we augment the $\mathcal{D}_{client}$ with unlabeled data uniformly subsampled from the same time period. We use the labeled data in $\mathcal{D}_{client}$ to estimate the associated model performances.

Implementation details We first post-process the raw audio signals by computing their Log Filter Bank Energy (LFBE) features with window size 25 ms and hop size of 10 ms. The number of mel coefficients is 20, which results in a leg-mel spectrogram feature of size 998×20. Features are further normalized by global cepstral mean and variance normalization (CMVN). Our encoder consists of 5 layers of convolutional layers followed by an LSTM layer with 64 units, where the kernels and strides are $[(3,3),(3,3),(3,3),(3,1),(3,1)]$ and $[(2,2),(2,2),(2,1),(2,1),(2,1)]$ respectively. Our choice of decoder is a Conv1D layer that reconstructs the LFBE signals. The AEC classifier is made by an additional LSTM layer with hidden size of 96 followed by a dense layer on top of the encoder. A sigmoid function then maps the dense layer output to $p \in [0, 1]$.

Evaluation Metric We evaluate the performance of models based on the area under the curve (AUC) and the precision-recall curves on $\mathcal{D}_{\mathcal{I}}$, $\mathcal{D}_{\mathcal{U}}$, and $\mathcal{D}_{\mathcal{T}}$. We compare the precisions at the recall values ranging from 0.6 to 0.9 as these regions are of practical interest for real use cases.

Baseline We compare our model performance with two baselines: (1)*SSL w/o* $\mathcal{D}_{client}$: classifier trained from pre-trained APC model $\mathcal{M}_0$. This corresponds to the method of directly deploying models trained from server data without FL, and



(a) $\mathcal{D}_{\mathcal{I}}$ (b) $\mathcal{D}_{\mathcal{U}}$ (c) $\mathcal{D}_{\mathcal{T}}$

Fig. 2: The effect of the size of the client data $\mathcal{D}_{client}$ on precision-recall curves. For reference, FSSL (1x) model is trained on $\mathcal{D}_{client}$ containing 287,302 utterances. FSSL(2x), FSSL(4x), and FSSL(8x) are trained on augmented $\mathcal{D}_{client}$ by incorporating extra un-labeled data.

Methods	$\mathcal{D}_{\mathcal{I}}$				$\mathcal{D}_{\mathcal{U}}$				$\mathcal{D}_{\mathcal{T}}$			
	AUC (%)	**Precision (%) at recall r**			**AUC (%)**	**Precision (%) at recall r**			**AUC (%)**	**Precision (%) at recall r**		
		$r = 0.7$	$r = 0.8$	$r = 0.9$		$r = 0.7$	$r = 0.8$	$r = 0.9$		$r = 0.7$	$r = 0.8$	$r = 0.9$
SSL w/o $\mathcal{D}_{client}$	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
FSSL	104.55	107.41	110.51	118.56	103.15	102.94	107.56	120.30	104.97	105.72	108.59	118.80
SSL w/ $\mathcal{D}_{client}$	113.17	114.01	121.60	139.57	116.96	117.71	122.75	152.36	114.87	113.15	123.39	150.83

Table 1: Relative benchmark results for the proposed method and baseline models. Note that we set "SSL w/o $\mathcal{D}_{client}$" as the baseline (100%) to calculate the relative performance for other two methods in each column. FSSL is equivalent to the FSSL (4x) in Fig. 2. FSSL consistently outperforms SSL w/o $\mathcal{D}_{client}$ baseline on all data partitions. The improvement, however, is less compared to SSL w/ $\mathcal{D}_{client}$ when the $\mathcal{D}_{client}$ is directly used to train the classifier.

(2)*SSL w/ $\mathcal{D}_{client}$*: classifier trained with all annotated data $\mathcal{D} = \mathcal{D}_{server} \cup \mathcal{D}_{client}$. This corresponds to the scenario where all customer data are directly accessible under the centralized training framework, where SSL stands for self-supervised learning. Note that (2) is unrealistic for real world applications but it can be treated as an upper bound for classifiers at here. Both models (1) and (2) consume only the server data for learning the representations, whereas (2) uses all annotated data and (1) only uses the server data to train the classifiers.

5. RESULTS

In order to study the effect of the size of the client dataset $\mathcal{D}_{client}$ used for FL training, we further incorporate extra unlabeled data (~2M) collected in the same time period as $\mathcal{D}_{client}$ to find the best performing setup. We vary the size of $\mathcal{D}_{client}$ from 1x to 2x, 4x, and 8x. Fig. 2 shows that further model improvement can be achieved by increasing the size of client dataset. However, such improvement diminishes when the client dataset is expanded to 8x. Therefore, we discuss the performance of the proposed method using FSSL(4x) in the following section.

The results of different models benchmarked on $\mathcal{D}_{\mathcal{I}}$, $\mathcal{D}_{\mathcal{U}}$, and $\mathcal{D}_{\mathcal{T}}$ are shown in Table 1. The proposed method consistently outperforms the baseline model SSL w/o $\mathcal{D}_{client}$ in all benchmarks, whereas SSL w/ $\mathcal{D}_{client}$ yields the best model performance among all three models. Although the improvement in the overall AUC brought by FL of representations is relatively small, the improvements at high recall regions (e.g., 0.6-0.9) are significant. Since high recall regions are of practical production interest, our results show clear evidence that fine-tuning acoustic event classification model in the post-deployment stage through continual learning of representations is feasible. The vastly superior performance from SSL w/ $\mathcal{D}_{client}$ model indicates that when well-annotated data is accessible to centralized computing resources, conventional supervised learning is still more effective in optimizing model weights with respect to fixed learning targets. However, for particular use cases where sensitive data is involved and learning algorithm is not allowed to directly interacting with large datasets on the cloud, centralized learning algorithms is consequently infeasible. In turn, federated learning may be one of the few solutions to the task.

6. CONCLUSIONS

We show that leveraging self-supervised federated learning to train AEC models on local client data leads to improvement in model performance with no extra cost of adding labeled data. Although training representation encoders is relatively less effective than directly training the classifiers using centralized training methods , we note that federated learning can become the deciding factors for certain applications given growing concerns around consumer data privacy.

7. REFERENCES

[1] Marco Cristani, Manuele Bicego, and Vittorio Murino, "Audio-visual event recognition in surveillance video sequences," *IEEE Transactions on Multimedia*, vol. 9, no. 2, pp. 257–267, 2007.

[2] Giuseppe Valenzise, Luigi Gerosa, Marco Tagliasacchi, Fabio Antonacci, and Augusto Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," in *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*. IEEE, 2007, pp. 21–26.

[3] Pedro Cano, Markus Koppenberger, and Nicolas Wack, "Content-based music audio recommendation," in *Proceedings of the 13th annual ACM international conference on Multimedia*, 2005, pp. 211–212.

[4] Annamaria Mesaros, Toni Heittola, Antti Eronen, and Tuomas Virtanen, "Acoustic event detection in real life recordings," in *2010 18th European Signal Processing Conference*. IEEE, 2010, pp. 1267–1271.

[5] Jort F Gemmeke, Lode Vuegen, Peter Karsmakers, Bart Vanrumste, et al., "An exemplar-based nmf approach to audio event detection," in *2013 IEEE workshop on applications of signal processing to audio and acoustics*. IEEE, 2013, pp. 1–4.

[6] Annamaria Mesaros, Toni Heittola, Onur Dikmen, and Tuomas Virtanen, "Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations," in *2015*. IEEE, 2015, pp. 151–155.

[7] Emre Cakır, Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[8] Chieh-Chi Kao, Weiran Wang, Ming Sun, and Chao Wang, "R-crnn: Region-based convolutional recurrent neural network for audio event detection," *arXiv preprint arXiv:1808.06627*, 2018.

[9] Il-Young Jeong, Subin Lee, Yoonchang Han, and Kyogu Lee, "Audio event detection using multiple-input convolutional neural network," *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2017.

[10] Yun Wang, Juncheng Li, and Florian Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," in *ICASSP 2019-2019*. IEEE, 2019, pp. 31–35.

[11] Ji Liu, Jizhou Huang, Yang Zhou, Xuhong Li, Shilei Ji, Haoyi Xiong, and Dejing Dou, "From distributed machine learning to federated learning: A survey," *arXiv preprint arXiv:2104.14362*, 2021.

[12] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays, "Applied federated learning: Improving google keyboard query suggestions," *arXiv preprint arXiv:1812.02903*, 2018.

[13] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[14] Aaqib Saeed, Flora D Salim, Tanir Ozcelebi, and Johan Lukkien, "Federated self-supervised learning of multisensor representations for embedded intelligence," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 1030–1040, 2020.

[15] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017*. IEEE, 2017, pp. 776–780.

[16] Annamaria Mesaros, Toni Heittola, Aleksandr Diment, Benjamin Elizalde, Ankit Shah, Emmanuel Vincent, Bhiksha Raj, and Tuomas Virtanen, "Dcase 2017 challenge setup: Tasks, datasets and baseline system," in *DCASE 2017-Workshop on Detection and Classification of Acoustic Scenes and Events*, 2017.

[17] Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James Glass, "An unsupervised autoregressive model for speech representation learning," *arXiv preprint arXiv:1904.03240*, 2019.

[18] Yu-An Chung and James Glass, "Generative pre-training for speech with autoregressive predictive coding," in *ICASSP 2020-2020*. IEEE, 2020, pp. 3497–3501.

[19] Yu-An Chung and James Glass, "Improved speech representations with multi-target autoregressive predictive coding," *arXiv preprint arXiv:2004.05274*, 2020.

[20] Santiago Pascual, Mirco Ravanelli, Joan Serra, Antonio Bonafonte, and Yoshua Bengio, "Appliedlearning problem-agnostic speech representations from multiple self-supervised tasks," *arXiv preprint arXiv arXiv:1904.03416*, 2019.

[21] Ho-Hsiang Wu, Chieh-Chi Kao, Qingming Tang, Ming Sun, Brian McFee, Juan Pablo Bello, and Chao Wang, "Multi-task self-supervised pre-training for music classification," in *ICASSP 2021*. IEEE, 2021, pp. 556–560.

[22] Aren Jansen, Manoj Plakal, Ratheet Pandya, Daniel PW Ellis, Shawn Hershey, Jiayang Liu, R Channing Moore, and Rif A Saurous, "Unsupervised learning of semantic audio representations," in *2018*. IEEE, 2018, pp. 126–130.

[23] Joel Shor, Aren Jansen, Ronnie Maor, Oran Lang, Omry Tuval, Felix de Chaumont Quitry, Marco Tagliasacchi, Ira Shavitt, Dotan Emanuel, and Yinnon Haviv, "Towards learning a universal non-semantic representation of speech," *arXiv preprint arXiv:2002.12764*, 2020.

[24] Marco Tagliasacchi, Beat Gfeller, Félix de Chaumont Quitry, and Dominik Roblek, "Self-supervised audio representation learning for mobile devices," *arXiv preprint arXiv:1905.11796*, 2019.

[25] Dhruv Guliani, Françoise Beaufays, and Giovanni Motta, "Training speech recognition models with federated learning: A quality/cost framework," in *ICASSP 2021-2021*. IEEE, 2021, pp. 3080–3084.

[26] Xiaodong Cui, Songtao Lu, and Brian Kingsbury, "Federated acoustic modeling for automatic speech recognition," in *ICASSP 2021-2021*. IEEE, 2021, pp. 6748–6752.

[27] Yan Gao, Titouan Parcollet, Javier Fernandez-Marques, Pedro PB de Gusmao, Daniel J Beutel, and Nicholas D Lane, "End-to-end speech recognition from federated acoustic models," *arXiv preprint arXiv:2104.14297*, 2021.

[28] 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*. PMLR, 2017, pp. 1273–1282.