FEDRPO: FEDERATED RELAXED PARETO OPTIMIZATION FOR ACOUSTIC EVENT CLASSIFICATION

Meng Feng[1*], *Chieh-Chi Kao*[2], *Qingming Tang*[2], *Amit Solomon*[2], *Viktor Rozgic*[2], *Chao Wang*[2]

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

ABSTRACT

Performance and robustness of real-world Acoustic Event Classification (AEC) solutions depend on ability to train on diverse data from wide range of end-point devices and acoustic environments. Federated Learning (FL) provides a framework to leverage annotated and non-annotated AEC data from servers and client devices in a privacy preserving manner. In this work we propose a novel Federated Relaxed Pareto Optimization (FedRPO) method for semi-supervised FL with heterogeneous client data. In contrast to federated averaging class of FL algorithms (fedAvg) that perform unconstrained weighted aggregation across all data sources, FedRPO enables special treatment of data with high quality annotations vs. data with pseudo-labels of unknown, varying qualities. In particular, FedRPO computes the updates to the global model solving a constrained linear program, with explicit Pareto constraints to prevent performance degradation on annotated data, and controlled relaxation of the Pareto constraints on pseudo-labeled data to prevent learning of patterns in conflict with the annotated data. We show FedRPO significantly outperforms FedAvg on Amazon internal de-identified dataset on AEC tasks. On supervised learning, FedRPO improved precision by 32.5% over FedAvg when maintaining recall at 90%. Combined with FixMatch [1] for semi-supervised learning, FedRPO outperformed FedAvg on precision by 50.5% at 90% recall.

***Index Terms*—** Acoustic Event Classification, Federated Learning, Pareto Optimization, Semi-supervised Learning

1. INTRODUCTION

Deep-learning solutions for Acoustic Event Classification (AEC) [2–4] often require large amount of audio (unannotated or annotated) for training. However, in many real world applications, audio are processed locally on the device and unavailable for improving the AEC solution post deployment. In order to improve the model by learning from real data, it is of critical importance to move towards distributed on-device training that allows the model to improve in a way that preserves user privacy. Consequently, the obtained models might not generalize to all devices and acoustic environments. It is of critical importance to move towards distributed training that allows use of unlabeled data from all devices while following the data-handling policies for privacy protection.

Federated Learning is the learning task solved by a loose federation of participating devices, commonly referred to as clients, which are coordinated by a central server [5]. Each client has a training dataset which is stored only locally and is never uploaded to the central server. Instead, each client computes an update to the current global model maintained by the server, and only this update is shared with the server. Federated Averaging (FedAvg) [5] is perhaps the most commonly used FL algorithm due to its simplicity and have shown impressive results on a number of supervised-learning tasks. It aggregates the client updates by taking the weighted average of client model weights. Intuitively it can be viewed as a naive extension of the stochastic gradient descent (SGD) to distributed learning. Ideally to handle partially labeled dataset for AEC tasks, one would simply replace SGD components from the existing semi-supervised learning algorithms with FedAvg. Unfortunately, recent work on semi-supervised federated learning (SSFL) noted that doing so often led to degraded performance [6–8]. A number of approaches have been proposed to mitigate the issue of performance degradation from the learning perspective. While the common goal is to control drift of client models [9] from the initial model trained on labeled data, many of these approaches resort to introducing heuristic designs based on the specific learning tasks [6–8].

Inspired by the recent work [10, 11] that systematically frames FL as multi-objective optimization, we propose **Fed**erated **R**elaxed **P**areto **O**ptimization (FedRPO) for both FL and SSFL. As its name suggests, FedRPO is based on Pareto optimization [12], a classic optimization method of ensuring no degradation of performance among all users. We introduce a simple relaxation to the Pareto constraint and apply it to both labeled and partially labeled dataset. We compare FedRPO vis-à-vis FedAvg and show that FedRPO achieves superior performance. These improvements are observed both in the FL framework, as well as the SSFL framework, when our method is combined with an extended version of FixMatch [1], a novel algorithm for semi-supervised learning. Compared to [10, 11] that works on FL only, our method has a stronger focus on model utility and is applicable to both FL and SSFL thanks to our proposed relaxation technique.

2. RELATED WORK

FedAvg has been successfully applied to a number of acoustic and visual tasks [5, 13–17] for fully supervised learning. However, theoretical analyses of FedAvg are mostly pessimistic [18–20] Different variants of FedAvg have been proposed to address problems that can impair the learning performance, but most of them resort to tweaking the loss functions [9, 21–24]. FedAvg has been used as a direct replacement of SGD when adapting centralized learning algorithms to the federated learning framework with minimal changes. For example, [25] extends self-supervised learning to FL via FedAvg in sound and vision domains. [26] adds additional neural network layers to client models to allow more personalized training on the client dataset. Domain-specific implementations based on FedAvg are also abundant, e.g. [27, 28]. Semi-supervised learning offers an effective means to improve the model performance by leveraging unlabeled data. A well-known class of SSL methods involves producing artificial labels, widely referred to as pseudo-labels, for unlabeled data. [29] incorporates pseudo-labels derived from model's predictions for self-training. Similarly, consistency regularization [30–32] generates pseudo-labels by randomly perturbing the inputs or model functions. Recent state-of-the-art SSL algorithms such as FixMatch

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

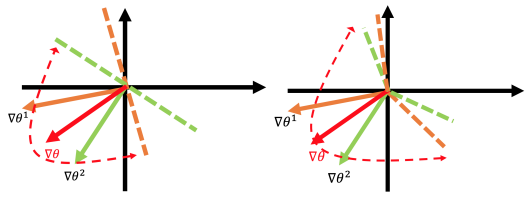
[1] and Remixmatch [33] regulate the prediction consistency from more advanced data augmentation functions [34–36]. In this work, we extend FixMatch to our FedRPO framework.

Semi-supervised learning algorithms have been naively adapted to the federated learning framework by replacing SGD with FedAvg but resulted in worse model performance [6, 7]. Recently multiple methods were proposed to alleviate the issue. These methods differ in the execution details, but share the same intention–to keep the client model weights close to the initial global model, i.e. to avoid the client drift. For example, [6] uses alternate training to control the client drift. After each global model update, it trains the updated global model on the server dataset again to prevent its accuracy from dropping on the server dataset. [7] trains two separate copies of model parameters on server and client data respectively. When training on server data, it freezes the client model parameters and vice versa. At test time, it adds the two sets of model parameters to yield the final model. Similar to FedRPOx [21], a proximity loss term is included to discourage potential client drifts. Evidently, these methods include a number of heuristic choices in the algorithmic designs and, by deliberately weakening the influence of client model updates, reveal a general lack of confidence on the client models. Our work introduces a much more principled optimization approach that offers theoretical guarantee based on Pareto Optimization to prevent degradation on server data and safely exposes the model to unlabeled data to learn patterns which do *not* conflict with the server data.

Pareto Optimization is a classical method rooted in economy theories [12] that solves the problem of optimizing individual client utilities while not sacrificing a subset of clients for the rest in the optimization process. Recently, Pareto Optimization has been applied in Federated Learning to improve the model fairness [10, 11]. Inspired by this work, we propose a simple relaxation of Pareto Optimization with the focus of obtaining better performance. We demonstrate that our proposed relaxation is effective on both labeled dataset and partially labeled dataset. Our work significantly differs from [11] in that our proposed algorithm does not concentrate on fairness but instead on global performance. In addition, [11] is limited to fully labeled dataset whereas we show that our work has a strong focus on partially labeled dataset for semi-supervised learning.

3. METHODS

We assume *label-at-server* [7] setting, where labeled data only exists on the server, and the client data is completely unlabeled. Given a labeled dataset on the server $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{S}|}$ and a set of unlabeled datasets from the clients $\mathcal{U} = \{\mathcal{U}_1, ..., \mathcal{U}_K\}$, $\mathcal{U}_k = \{u_i\}_{i=1}^{|\mathcal{U}_k|}$. The goal is to train a global model $\mathcal{M}(\theta, \theta \in \Theta) : x_i \mapsto y_i'$ that maps from the sound input x_i to the event prediction y_i', where x_i is the input sound clip, y_i is the event label, and y_i' is the predicted event classification. We assume that the amount of server data is significantly smaller than the amount of client data $|\mathcal{S}| << |\mathcal{U}|$. In addition, the client data $\mathcal{U}$ cannot be shared with the server. The mainstream federated learning algorithms [5] leverage the client data by optimizing the model parameter θ with respect to the weighted average of the server and client losses $\min_{\theta \in \mathbb{R}^d} \sum_{k=1}^{K} \frac{n_k}{n} F_k(\theta)$ where $F_k(\theta) = \frac{1}{n_k} \sum_{i \in n_k} f_i(\theta)$, f is the loss function on the edge device, $n_k \in \mathbb{Z}$ is the local data size on the edge device k, and $n = \sum_{k=1}^{K} n_k$. There exists a wide variety of implementations of solving this problem, but most of them are based on FedSGD and FedAvg which update the global model by taking the weighted average of model gradients (Eq.(1)) or weights (Eq. (2)) respectively from the client device.



(a) Pareto (b) Relaxed Pareto

Fig. 1: Graphical visualization of (relaxed) pareto optimization.

$$\theta_{t+1} \leftarrow \theta_t - \eta \sum_{k=1}^{K} \frac{n_k}{n} \nabla \theta_t^k \tag{1}$$

$$\theta_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \theta_{t+1}^k, \ \forall k, \theta_{t+1}^k \leftarrow \theta_t^k - \eta \nabla \theta_t^k \tag{2}$$

Notably, the weight aggregation step is unconstrained and may cause an increase in client losses in contrast to the centralized SGD.

Pareto Optimization

In contrast of FedAvg based methods, Pareto Optimization [12] optimizes the client losses *directly*,

$$\min_{\theta} [f_1(\theta), ..., f_K(\theta)] \tag{3}$$

It treats each client loss as an objective and finds the *Pareto solution* θ^* defined as follows.

$$\nexists \theta' \in \Theta \ s.t. \ \forall i : f_i(\theta') \leq f_i(\theta^*) \text{ and } \exists j : f_j(\theta') \leq f_j(\theta^*) \tag{4}$$

In words, it means that θ^* is the solution that cannot be further optimized on a subset of losses without harming the rest. With gradient descent, at each optimization step the loss gradients $[\nabla_\theta f_1(\theta), ..., \nabla_\theta f_K(\theta)]$ can be obtained from the clients. To calculate the aggregated gradient without harming any client losses, one can search for a vector $d^* \in \mathbb{R}^K$ so that d^* points towards a direction not conflicting with any of the loss gradients. As illustrated in Fig. 1a, intuitively it means the angle between the desired aggregated gradient d^* and the individual loss gradients are less or equal to $\pi/2$. This can be mathematically expressed as the dot products between d^* and individual gradient losses.

$$d^{*T} \nabla_\theta f_i(\theta) \geq 0 \, \forall i \in \{1, ..., K\} \tag{5}$$

With sufficiently small gradient step η, by applying d^* to the global model, the individual client losses either decrease or stay unchanged [11]. In practice, it is difficult to directly search for d^*. Instead, it is common to find d^* in the convex hull of the loss gradients $G = [\nabla_\theta f_1(\theta), ..., \nabla_\theta f_K(\theta)]$ [11]. We can then re-write d^* as $d^* = \alpha^* G$, where $G \in \bar{G} = \{\sum_{i=1}^{K} \{\alpha_i \nabla_\theta f_i(\theta)\} \mid \alpha_j \geq 0 \, \forall j \text{ and } \sum_{j=1}^{K} \alpha_j = 1\}$. Now the goal is to find $\alpha^* \in \mathbb{R}^K$, and it can be formulated as the linear program as follows.

$$\begin{aligned} & \min_{\alpha} \alpha^T G^T G w \\ \text{such that} \quad & \alpha^T G^T G \geq 0 \\ & \sum_{i=1}^{K} \alpha_i = 1, \ \alpha_i \geq 0 \, \forall i \in \{1, ..., K\} \\ & \sum_{i=1}^{K} w_i = 1, w_i \geq 0 \, \forall i \in \{1, ..., K\} \end{aligned} \tag{6}$$

where $w \in \mathbb{R}^K$ is the client prioritization vector. By iteratively updating the global model with d^* in a gradient descent loop, the global model can gradually converge to θ^* with an important guarantee that every update is constrained to avoid degradation in losses on any of the clients when the gradient step is sufficiently small [11].

Federated Relaxed Pareto Optimization (FedRPO)

The vanilla Pareto Optimization can easily get stuck at a local optima due to the constraints on the gradient update [11]. To jump out of the local optima, it may be necessary to sacrifice a subset of clients for higher global gain. Inspired by this idea, we introduce a simple relaxation to the Pareto Optimization. As shown in Inequality (7), we allow the angles between d^* and a subset of loss gradients to be slightly greater than $\pi/2$. Importantly, the exceeded angle is controllable by the choice of c.

$$\alpha^T G^T G \geq c,\ c \in [-1, 0] \tag{7}$$

The idea of the relaxation in Inequality (7) is especially powerful for partially labeled dataset. Pseudo-labeling is central in many recent state-of-the-art semi-supervised learning algorithms [1,33]. Regardless how the pseudo-labels are generated, the pseudo-labels are generally less trustworthy than the ground-truth annotations. Therefore, it is reasonable to apply the relaxed Pareto constraint on the pseudo-labels. As for the labeled data, we apply the non-relaxed Pareto constraints to prevent performance degradation. We split the loss gradients into two groups: G_s denotes the loss gradients from the labeled data, and G_u denotes the loss gradients from the unlabeled data. We can formulate the relaxed Pareto constraints for partially labeled as follows.

$$\begin{aligned} \alpha^T G^T G_s &\geq 0 \quad \text{(for labeled data)} \\ \alpha^T G^T G_u &\geq c,\ c \in [-1, 0] \quad \text{(for unlabeled data)} \end{aligned} \tag{8}$$

With the above constraints, the global model can be safely exposed to pseudo-labels without concern of performance degradation on the labeled data. The strength of relaxation on unlabeled data is controlled by c.

FedRPO with FixMatch

We implement naive extensions of FixMatch [1] with both FedAvg and FedRPO. FixMatch generates pseudo-labels by a pre-trained intermediate teacher model for unlabeled data when the predicted confidence is greater than a preset threshold τ. Both the teacher model and the global model under training are iteratively updated on the ground-truth annotations and pseudo-labels. The training objective is shown below.

$$\begin{aligned} & f(\theta) = f_s(\theta) + \lambda f_u(\theta) \\ \text{where} \quad & f_s(\theta) = \sum_{i=1}^{|\mathcal{S}|} H(p_b, p_m(y \mid \alpha(x_i)) \\ & f_u(\theta) = \sum_{i=1}^{|\mathcal{U}|} \delta(\max(q_b) \geq \tau) H(\text{argmax}(q_b), p_m(y \mid \mathcal{A}(u_i)) \end{aligned}$$

where p_b is the event label, f_s is the loss function on labeled data, and f_u is the loss function on unlabeled data. H is the cross-entropy loss. $p_m(y \mid x)$, $q_b = p_m(y \mid \alpha(x))$, $p_m(y \mid \mathcal{A}(u))$ are the event distributions predicted by the model from the vanilla input x, the weakly augmented input $\alpha(x)$ and the strongly augmented input $\mathcal{A}(u)$ respectively. $\max(q_b)$ is the prediction confidence.

4. EXPERIMENTS AND RESULTS

Data We conduct all of our experiments on Amazon internal de-indentified dataset drawn from June to September 2021.

For supervised learning experiment, we construct a fully labeled dataset consists of 100 clients, each client contains 100 utterances. 80% data on each client is used for training, the rest 20% is used for testing. Each utterance stores a 10-second audio clip and the de-identified device serial number (DSN) of the source device. The label of an utterance describes the presence of 6 event candidates.

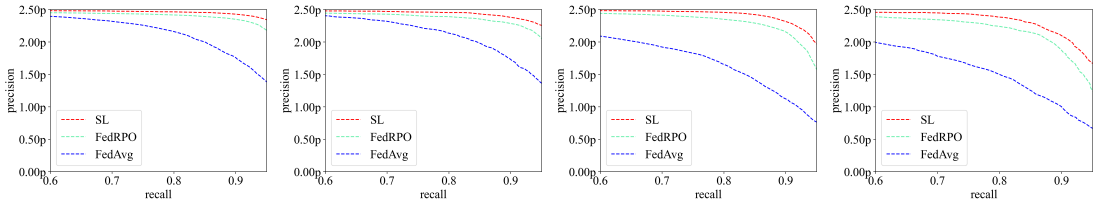
For semi-supervised learning experiment, we construct another dataset $\mathcal{D}_{server}$ to simulate the labeled data stored on the cloud server. This dataset consists of 400 utterances for each of the 6 events. We also use the same dataset in the supervised learning experiment to simulate the client dataset $\mathcal{D}_{client}$. 80% data on each client is used for training, and the remaining 20% is used for in-distribution testing. We denote the in-distribution test dataset as $\mathcal{D}^{in}_{client}$. To test our model on unseen data, we construct a out-of-distribution test dataset, $\mathcal{D}^{out}_{client}$, consisting of 200 utterances per event for 6 events. The included data is drawn from devices whose DSNs are unused in neither training nor fine-tuning.

Implementation details We use the same neural network model architecture for both supervised and semi-supervised experiments. 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 log-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. 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 softmax function then maps the dense layer output to a categorical distribution. The presented results are trained with a learning rate of 0.05, and we obtain consistent results from other learning rates as well (e.g. 0.1, 0.001).

Evaluation Metric We evaluate the performance of models based on the precision-recall curves. 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.

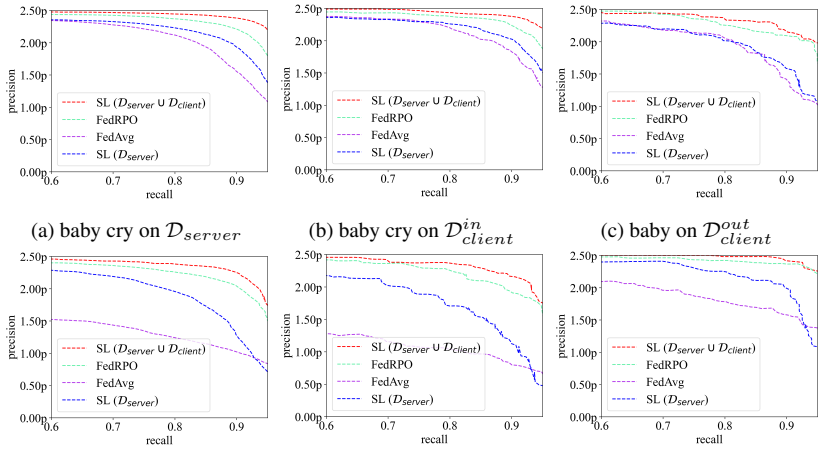
Baseline For the supervised learning experiment, we compare our model performance with FedAvg and centralized supervised learning (denoted as SL in figures). For semi-supervised learning experiment, we compare our model performance with FedAvg as well as SL trained with only $\mathcal{D}_{server}$ and SL trained with $\mathcal{D}_{server} \cup \mathcal{D}_{client}$. The former is often used as the empirical lower bound since it is trained with only server data, and the latter is often used as the empirical upper bound since it is trained with all available data, ignoring the fact that client data is often not directly accessible and unlabeled. To study how FedRPO performs on supervised learning, we run FedRPO and FedAvg without pre-training or data augmentation. As shown in Fig. 2, FedRPO consistently outperforms FedAvg on both train and test dataset by a large margin. We posit that the weaker results of FedAvg may be caused by the stronger statistical heterogeneity presented in the real-world data.

For semi-supervised learning, we pre-train a starting model on $\mathcal{D}_{server}$. We then run FedAvg and FedRPO on the same dataset as in the supervised learning experiments but without using the labels. We evaluate the model performance on $\mathcal{D}_{server}$, $\mathcal{D}^{in}_{client}$, and $\mathcal{D}^{out}_{client}$. As seen in Fig. 3, FedRPO is consistently better than FedAvg and occasionally reaches comparable precision as centralized



(a) baby cry, train (b) baby cry, test (c) snore, train (d) snore, test

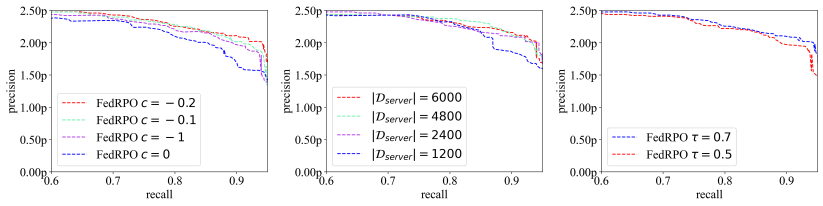
Fig. 2: Precision-recall Curves on fully labeled datasets. Precision on y-axis is relative. In the event of baby crying, FedRPO is 32.5% relatively better than FedAvg on out-of-distribution test dataset $\mathcal{D}_{client}^{out}$ at recall value of 0.9. Due to the page limit, we only include 2 events here. See the link to full results in the footnote.[1]



(a) baby cry on $\mathcal{D}_{server}$ (b) baby cry on $\mathcal{D}_{client}^{in}$ (c) baby on $\mathcal{D}_{client}^{out}$

(d) snore on $\mathcal{D}_{server}$ (e) snore on $\mathcal{D}_{client}^{in}$ (f) snore on $\mathcal{D}_{client}^{out}$

Fig. 3: PR curves for the semi-supervised experiments on Amazon internal de-identified dataset (labeled server data, unlabeled client data). Precision on y-axis is relative. In the event of baby crying, FedRPO is 50.5% relatively better than FedAvg on out-of-distribution test dataset $\mathcal{D}_{client}^{out}$ at recall value of 0.9. See the link to full results in the footnote. [1]



(a) vary c, SSL (b) vary dataset size (c) vary τ

Fig. 4: Ablation PR curves for the semi-supervised learning experiments on the event of baby crying. Precision on y-axis is relative.

supervised learning. We run ablation studies on the choice of c, the server dataset with fixed server to client ratio, and the value of tau. According to our results in Fig. 4, c has clear impact on the model precision. In this case, $c = -0.1$ and $c = -0.2$ produce the best results, whereas $c = -1$ (full relaxation) is slightly weaker. $c = 0$ (no relaxation) delivers the weakest result among all tested values of c. As we vary the dataset size, we found that our results are close when the size of the server dataset is from 2400 to 6000. When the server dataset is reduced to the size of 1200 (200 utterances per event for 6 events), the resulting model performance drops noticeably. The value of τ is roughly correlated with the quality of pseudo-labels. Shown in Fig. 4c, FedRPO is resilient against noisy pseudo-labels. Even at $\tau = 0.5$, the resultant model precision is still reasonably close to the chosen one at $\tau = 0.7$.

5. CONCLUSION

In this work, we presented FedRPO inspired by Pareto Optimization. We showed that FedRPO significantly outperforms FedAvg on supervised and semi-supervised learning tasks for AEC tasks on an Amazon internal de-identified dataset. In particular, FedRPO performed better than vanilla Pareto Optimization by exhibiting clear resilience against noisy pseudo-labels for unlabeled data. Though we focused on AEC, but our approach can also be applied to other audio tagging tasks.

[1] https://drive.google.com/file/d/1qBkgzjSawiW-rvVIu30PcglUGoguulhW/view?usp=sharing

6. REFERENCES

[1] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Advances in neural information processing systems*, vol. 33, pp. 596–608, 2020.

[2] 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.

[3] 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.

[4] Qin Zhang, Qingming Tang, Chieh-Chi Kao, Ming Sun, Yang Liu, and Chao Wang, "Wikitag: Wikipedia-based knowledge embeddings towards improved acoustic event classification," in *ICASSP 2022*, 2022.

[5] 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.

[6] Enmao Diao, Jie Ding, and Vahid Tarokh, "Semifl: Communication efficient semi-supervised federated learning with unlabeled clients," *arXiv preprint arXiv:2106.01432*, 2021.

[7] Wonyong Jeong, Jaehong Yoon, Eunho Yang, and Sung Ju Hwang, "Federated semi-supervised learning with inter-client consistency & disjoint learning," *arXiv preprint arXiv:2006.12097*, 2020.

[8] Antti Tarvainen and Harri Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *arXiv preprint arXiv:1703.01780*, 2017.

[9] 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*. PMLR, 2020, pp. 5132–5143.

[10] Zeou Hu, Kiarash Shaloudegi, Guojun Zhang, and Yaoliang Yu, "Federated learning meets multi-objective optimization," *IEEE Transactions on Network Science and Engineering*, 2022.

[11] Sen Cui, Weishen Pan, Jian Liang, Changshui Zhang, and Fei Wang, "Addressing algorithmic disparity and performance inconsistency in federated learning," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[12] Harold M Hochman and James D Rodgers, "Pareto optimal redistribution," *The American economic review*, vol. 59, no. 4, pp. 542–557, 1969.

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

[18] Margalit R Glasgow, Honglin Yuan, and Tengyu Ma, "Sharp bounds for federated averaging (local sgd) and continuous perspective," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 9050–9090.

[19] Blake E Woodworth, Kumar Kshitij Patel, and Nati Srebro, "Minibatch vs local sgd for heterogeneous distributed learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6281–6292, 2020.

[20] Jianyu Wang, Rudrajit Das, Gauri Joshi, Satyen Kale, Zheng Xu, and Tong Zhang, "On the unreasonable effectiveness of federated averaging with heterogeneous data," *arXiv preprint arXiv:2206.04723*, 2022.

[21] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[22] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith, "Fair resource allocation in federated learning," *arXiv preprint arXiv:1905.10497*, 2019.

[23] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh, "Agnostic federated learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 4615–4625.

[24] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in neural information processing systems*, vol. 33, pp. 7611–7623, 2020.

[25] Qinbin Li, Bingsheng He, and Dawn Song, "Model-contrastive federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10713–10722.

[26] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi, "Adaptive personalized federated learning," *arXiv preprint arXiv:2003.13461*, 2020.

[27] Peihua Yu and Yunfeng Liu, "Federated object detection: Optimizing object detection model with federated learning," in *Proceedings of the 3rd International Conference on Vision, Image and Signal Processing*, 2019, pp. 1–6.

[28] Jed Mills, Jia Hu, and Geyong Min, "Communication-efficient federated learning for wireless edge intelligence in iot," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5986–5994, 2019.

[29] Dong-Hyun Lee et al., "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, 2013, vol. 3, p. 896.

[30] Philip Bachman, Ouais Alsharif, and Doina Precup, "Learning with pseudo-ensembles," *Advances in neural information processing systems*, vol. 27, 2014.

[31] Samuli Laine and Timo Aila, "Temporal ensembling for semi-supervised learning," *arXiv preprint arXiv:1610.02242*, 2016.

[32] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," *Advances in neural information processing systems*, vol. 29, 2016.

[33] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel, "Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring," *arXiv preprint arXiv:1911.09785*, 2019.

[34] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le, "Unsupervised data augmentation for consistency training," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6256–6268, 2020.

[35] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 702–703.

[36] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.