PARTIAL FEDERATED LEARNING: UNLOCKING NON-BIOMETRIC TEXT INFORMATION SHARING FOR FEDERATED LEARNING

Tiantian Feng[1]*, Anil Ramakrishna*[2]*, Jimit Majmudar*[2]*, Charith Peris*[2]*, Jixuan Wang*[2]*, Clement Chung*[2]*, Richard Zemel*[2]*, Morteza Ziyadi*[2]*, Rahul Gupta*[2]

[1]University of Southern California, [2]Amazon Alexa AI

ABSTRACT

Federated Learning (FL) is a popular algorithm to train machine learning models on user data constrained to edge devices (for example, mobile phones) due to privacy concerns. Typically, FL is trained with the assumption that no part of the user data can be egressed from the edge. However, in many production settings, specific data-modalities/meta-data are limited to be on device while others are not. For example, in commercial SLU systems, it is typically desired to prevent transmission of biometric signals (such as audio recordings of the input prompt) to the cloud, but egress of locally (i.e. on the edge device) transcribed text to the cloud may be possible. In this work, we propose a new algorithm called Partial Federated Learning (PartialFL), where a machine learning model is trained using data where a subset of data modalities or their intermediate representations can be made available to the server. We further restrict our model training by preventing the egress of data labels to the cloud for better privacy, and instead use a contrastive learning based model objective. We evaluate our approach on two different multi-modal datasets and show promising results with our proposed approach.

1. INTRODUCTION

Existing FL paradigms typically assume a uniform set of restrictions applied to all available data modalities. With this framework of uniform restrictions, depending on available permissions on the data, we maybe allowed to move all of them to a server for *central* model training or we maybe required to retain all data on the user devices (hereafter referred to as edge devices) for a *federated* model training. However, in many real world settings, a subset of data modalities may carry looser constraints allowing easy egress of such data to a server. As an example, consider a machine learning model built using medical data. In this case, certain biometric modalities such as the patient's speech recordings, electrocardiogram (ECG), electroencephalogram (EEG), continuous heart rate (HR) measurements are likely to be protected more strictly compared to anonymized doctor's notes and reports. This is because biometric data are associated with Personal Identifiable Information (P.I.I) [1], and sharing these can raise privacy concerns leading to a number of legislations to protect them, including the recently introduced California Biometric privacy law [2] and Illinois Biometric Information Privacy Act [3].

Federated learning has been studied extensively in unimodal settings while multi-modal federated learning has recently gained attention from several works [4, 5]. However, existing learning paradigms do not support efficient and large scale training with distributed data modalities as described above, leading ML practitioners to choose the more restrictive setting of keeping all data on edge devices for federated model training. This brings with it all the challenges typical for federated learning: i) restricted model sizes due to small computational power on edge and ii) gradient drift issues due to heterogeneity in data, thus leading to lower performance compared to centralized training.

In this work, we propose to address this gap by building a "Partial" Federated Learning model (PartialFL), where a model is trained using distributed data for which some modalities are shared centrally while other modalities and the class labels are only available on the edge device. In addition to the distributed training, a closely related objective is to utilize the shared modalities to improve on FL model performance by addressing the data heterogeneity challenge. Our approach is related to the existing paradigm of vertical federated learning [6], where features for same set of samples maybe separated among different edge devices; in our case, we similarly have portions of data distributed between the central server and the edge devices, but the key difference in our work is that here we assume entirely different modalities of data exist on the edge and hence we can train a new artifact, such as an edge specific text encoder or acoustic signal encoder. Our main contributions in this work are:

- We present a new Federated Learning algorithm to train a machine learning model when data modalities are split among different devices
- We evaluate our algorithm on two different data sets present key results showing improvements in model performance

We also present additional results on a new dataset in the supplemental material along with various ablation results, high-

lighting the key areas of improvement introduced by our approach.

2. APPROACH

In a multi-modal FL setting, we can categorize each data modality into either the protected **non-shareable group** or the less restricted **shareable group** based on whether they are permitted to be egressed to the central server. In this work, without loss of generalization, we assume that anonymized text data are shareable with the remote server (but note that our treatment holds for any other modality that is deemed shareable), while holding other modalities in the non-shareable group on the edge devices. For further protection against information leakage, we map the raw text data into latent representations using a pre-trained language models such as DistilBERT [7], and only share these representations with the server in all our experiments. The benefit of training with shareable data in the server is that we can now train a larger model due to increased availability of compute in the server, and also extract a more robust representation when compared to the model trained on an unbalanced local data set. We assume availability of labels on edge for local modal training, but in their absence we can leverage user feedback similar to [8], which we defer for future work.

2.1. Outline

We consider a decentralized setup with K edge devices and a given multi-modal data set $\mathcal{D}$. For ease of exposition, we describe our approach using text and audio data, but note that it can be applied to any multi-modal setting. As noted in the previous section, latent representations of text data is assumed shareable but no form of audio data is shareable. We denote $\mathcal{D}^k : \{\mathbf{X}^k_{A,i}, \mathbf{X}^k_{T,i}, \mathbf{y}^k_i\}_{i=1}^{n_k}$ as the multi-modal dataset from k^{th} edge device containing audio, text and labels respectively. n_k is the size of the data set in $\mathcal{D}^k$, with $\mathcal{D} = \{\mathcal{D}^1, \mathcal{D}^2, ..., \mathcal{D}^K\}$ and the total number of utterances in $\mathcal{D}$ is $N = \sum_{k=1}^{K} n_k$.

PartialFL follows a structure similar to regular FL but with one key modification: we maintain additional models on both the server and edge devices trained entirely on the shareable modality. We further augment model training by using cross device and cross-modality contrastive loss objectives. These are described in more detail below.

2.2. Learning Components

Given the multi-modal setting described in the last section, the PartialFL framework consists of three model components as shown in Figure 1: the server model $\mathcal{F}_s(\cdot)$, the global model $\mathcal{F}_g(\cdot)$, and the local models $\mathcal{F}_k(\cdot)$ where $k \in \{1, 2, ..., K\}$.

The server model $\mathcal{F}_s(\cdot)$ exists in the server as an encoder which is trained on the shareable data ($\mathbf{X_T}$) to generate embeddings. In our example, we define the server-side textual embedding as $\mathbf{z_T}$. Since we do not have training labels at the server to train a classifier, the learning objective of $\mathcal{F}_s(\cdot)$ is to reduce the distance between the server generated textual embedding $\mathbf{z_T}$ with the local textual embedding $\mathbf{z'_T}$.

The global model $\mathcal{F}_g(\cdot)$ is a typical FL model and is trained in a distributed manner on the non-shareable data on edge, followed by a global aggregation in the server using typical FL algorithms, like FedAvg [9] or FedProx [10]. The global model can be either audio only or a multi-modal global model. The objective is to learn $\mathcal{F}_g(\cdot)$ parameterized by θ_g over the data set $\mathcal{D}$ without accessing $\mathbf{X_A}$ from edge devices. Since not every edge device may have every modality, training the global model can further help those devices with missing modalities.

Further, since the edge-side training of this model can suffer from gradient drifts [11], we add a cross-modal alignment objective to decrease the distance between the audio embedding $\mathbf{z_A}$ (or multi-modal embedding $\mathbf{z_M}$) and $\mathbf{z_T}$. This idea is similar to the model contrastive loss presented in [12].

The local model $\mathcal{F}_k(\cdot)$ is only available in the k^{th} edge device. Unlike the server model $\mathcal{F}_s(\cdot)$, here we have access to data labels so the local model includes a classifier trained using cross-entropy loss over the text modality. Note that the full $\mathcal{F}_k(\cdot)$ is not shareable since the server would then be able to infer local labels using $\mathcal{F}_k(\cdot)$ as $\mathbf{X_T}$ is already uploaded to the server. Instead, the edge device only sends the local textual embeddings $\mathbf{z'_T}$ generated from the encoder layer of $\mathcal{F}_k(\cdot)$. Similar to training the global model, over-fitting can occur easily while training this model, so we add an embedding alignment loss to minimize the distance between the local embeddings $\mathbf{z'_T}$ from the server generated embeddings $\mathbf{z_T}$.

The key intuition behind PartialFL is that by iteratively aligning server side representations of the shared text modality ($\mathbf{z_T}$) with different client side representations $\mathbf{z'_T}$ on the server, and by aligning the edge side text and audio representations with the realigned $\mathbf{z_T}$, we are improving model robustness to extreme data heterogeneity.

2.3. Cross-modal Alignment

Cross-modal alignment is a popular learning task when working with multiple modalities [13, 14]. This learning task focuses on aligning embeddings of the same instance across the different modalities. More concretely, we compute modality specific representations denoted as $\mathbf{z_A}$, $\mathbf{z_T}$ and $\mathbf{z_M}$ for the audio embedding, text embedding and the multi-modal embedding respectively. Cross-modal alignment in PartialFL aims to push these embeddings close to each other if they belong to the same data sample, and increase the distance between them otherwise.

Similar to previous work [15], a **positive pair** is defined as $\mathbf{z_A}$ (or $\mathbf{z_M}$) and $\mathbf{z_T}$ from the same data sample. On the other hand, **negative pairs** are constructed from different data samples. More precisely, we define **intra-modal negative**

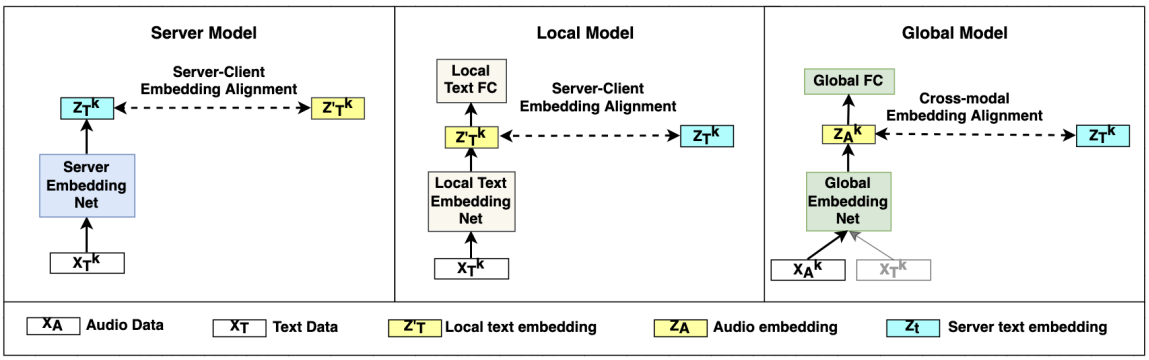

Fig. 1. Different models in the PartialFL learning architecture.

pairs as the embeddings from the same modality but different data samples. For instance, $(\mathbf{z}^k_{A,i}, \mathbf{z}^k_{A,j})$ is a negative pair if $i \neq j$. Further, we define **inter-modality negative pairs** as embeddings from different modalities *and* data samples. An example of the inter-modality negative pair is $(\mathbf{z}^k_{A,i}, \mathbf{z}^k_{T,j})$ where $i \neq j$. With the audio global model, we can define the loss term $\mathcal{L}_i^{T\to A}$ for data $\mathbf{X}^k_i$ in a training batch of size B with a temperature parameter τ as:

$$\mathcal{L}_i^{T\to A} = -\log \frac{e^{(\mathbf{z}^k_{A,i})^T \mathbf{z}^k_{T,i}/\tau}}{\sum_{j=1,i\neq j}^{B} e^{(\mathbf{z}^k_{A,i})^T \mathbf{z}^k_{A,j}/\tau} + e^{(\mathbf{z}^k_{A,i})^T \mathbf{z}^k_{T,i}/\tau}} \quad (1)$$

2.4. Embedding Alignment at Server and Local models

As noted before, to prevent over-fitting, we aim to decrease the distance between the local and server side textual embeddings $\mathbf{z}'_\mathbf{T}$ and $\mathbf{z}_\mathbf{T}$ using the contrastive loss. We define a positive pair as $\mathbf{z}'_\mathbf{T}$ and $\mathbf{z}_\mathbf{T}$ from the same data sample, and a negative pair from different samples. We can then write the server side contrastive loss $\mathcal{L}_i^{L\to S}$ as:

$$\mathcal{L}_i^{L\to S} = -\log \frac{e^{(\mathbf{z}^k_{T,i})^T \mathbf{z'}^k_{T,i}/\tau}}{\sum_{j=1,i\neq j}^{B} e^{(\mathbf{z}^k_{T,i})^T \mathbf{z}^k_{T,j}/\tau} + e^{(\mathbf{z}^k_{T,i})^T \mathbf{z'}^k_{T,i}/\tau}} \quad (2)$$

We define the edge side loss $\mathcal{L}_i^{S\to L}$ similarly.

2.5. Learning Algorithm

Unlike training centralized contrastive models, PartialFL needs to optimize the server model and the edge models asynchronously. To optimize the edge model and the server model in such manner, we use alternating minimization (AM) similar to FedGKT [16], where we alternatively fix one model and optimize the other. The full algorithm is presented in Appendix C.

2.6. Implementation

We use PyTorch to implement the PartialFL and the other baselines. In this study, we experiment with two popular FL algorithms: FedAvg and FedProx. We fix the weight of the proximal loss in FedProx as 0.01. We fix the number of edge devices as 200. When experimenting with the Food-101 data set, we choose the edge sample rate in each training round as 10%. On the other hand, we regard each speaker as a separate client in the emotion recognition task as it provides a natural data split in the FL. Since there are fewer clients in emotion recognition data sets, we decide to use an edge sample rate of 50% in each global training round.

2.7. Hyper-parameters

We set the local training batch size as 16 and the local training epoch as 1 in all FL algorithms. We set the learning rate as 0.0001 and 0.0005 in training the emotion recognition task and Food-101 classification tasks, respectively. We apply the Adam optimizer in all experiments. The global training round is 150 in the emotion recognition task and 200 in the Food-101 data set. In PartialFL, we explore the temperature parameter $\tau \in \{0.05, 0.1, 0.2\}$. We tune the weight β in both $\mathcal{L}_{glob}$ and $\mathcal{L}_{loc}$ in $\{0.001, 0.01\}$ in training emotion recognition models.

3. EXPERIMENTS

We evaluate the PartialFL algorithm on two datasets from the Speech Emotion Recognition task (SER): IEMOCAP [17] which contains utterances from ten subjects expressing various categorical emotions, and MSP-Improv [18] which contains multi-modal emotion recognition data set collected from 12 speakers. Additional details on all our datasets and models are provided in Appendix D. We also present results on image classification task from the Food101 dataset along with detailed ablation studies in Appendix E.

3.1. Baselines

3.1.1. Centralized

Here we assume that all data modalities are available in one central server for training in both uni-modal and multi-modal

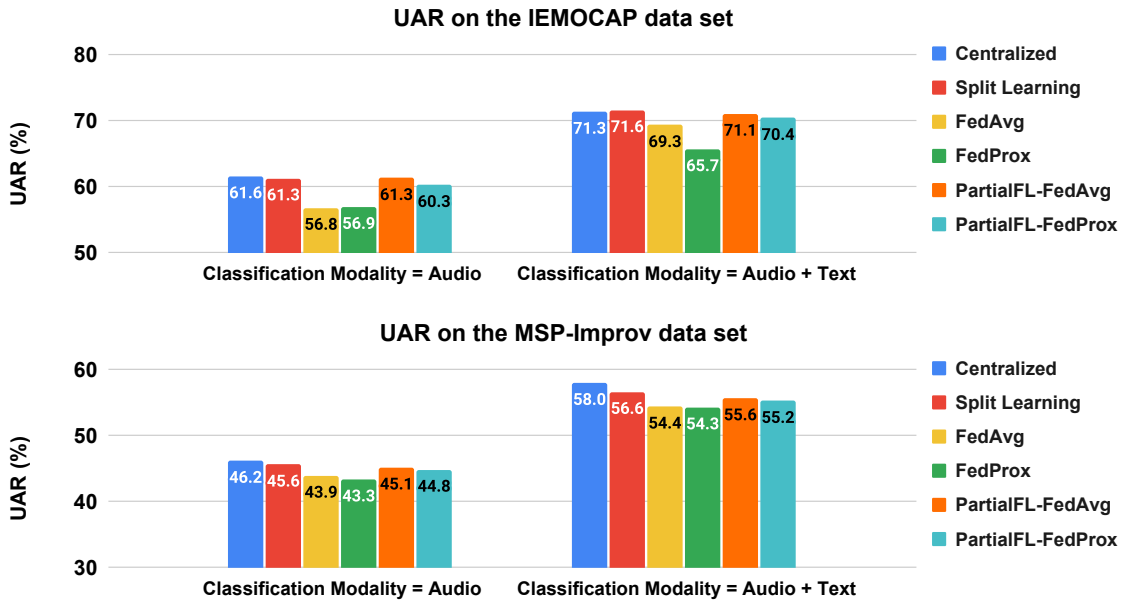

Fig. 2. Model performance for the IEMOCAP and MSP-Improv data set. PartialFL considerably outperforms FL and SL and has performance close to the Centralized upper bound.

settings. Since we are no longer limited by edge size computational power, this acts as an upper limit for the model performance.

3.1.2. Split Learning

We also implemented split learning to compare against our model. We set model size to be same as in the centralized baseline, and it serves as an upper bound for decentralized training. However, split learning incurs significant communication costs in practice compared to typical federated learning (refer to Section 3.3 in [19] for a comparison of training times for both algorithms).

3.1.3. Federated Learning

Finally, we also compare against a typical FL baseline. We experiment with both FedAvg and FedProx [10] variants of FL, and set the global model size to be same as PartialFL for a fair comparison.

4. RESULTS

We report unweighted average recall (UAR) score to measure the model performance. We use each recording session as a separate test fold, and repeat the training 5 and 6 times on the IEMOCAP and the MSP-Improv data sets respectively.

4.0.1. Uni-model global model (audio only)

Full results from our experiments are shown in Figure 2; PartialFL showed stronger results than both FedProx and FedAvg in both datasets, but especially so in IEMOCAP, where we observed nearly 4.00% improvement. Further, PartialFL approaches the performance of the centralized and split learning baselines by leveraging the additional data modality leading to improved robustness, while still retaining the benefits of federated learning.

4.0.2. Multi-modal global model (audio+text)

In this setting, we observed stronger performance in the centralized and SL baselines, with SL surprisingly outperforming centralized training in both data sets. However, similar to the previous experiments, PartialFL consistently outperforms both FL baselines, with overall improvement in the range of 1.0-2.0%, highlighting the robustness of our proposed approach.

5. CONCLUSION

We propose a novel multi-modal Federated Learning framework called PartialFL with a goal of addressing the heterogeneity challenge in FL and improve the final model performance. Unlike traditional FL, PartialFL allows some modalities to be shared with the server which enables us to train a robust embedding network over the shared modality in the server. We experiment with two multi-modal data sets and report strong performance against three baselines. We observe that PartialFL consistently outperforms traditional FL in all tasks, and approaches the performance of centralized models, as well as split learning without any of its communication overhead or straggler problems. Future work includes deploying PartialFL in real world applications to further evaluate its efficacy.

6. REFERENCES

[1] Umut Uludag, Sharath Pankanti, Salil Prabhakar, and Anil K Jain, "Biometric cryptosystems: issues and challenges," *Proceedings of the IEEE*, vol. 92, no. 6, pp. 948–960, 2004.

[2] SB-1189, "California bio-metric privacy law," `https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=202120220SB1189`, 2022.

[3] BIPA, "Illinois biometric information privacy act," `https://www.ilga.gov/legislation/ilcs/ilcs3.asp?ActID=3004&ChapterID=57`, 2008.

[4] Jiayi Chen and Aidong Zhang, "Fedmsplit: Correlation-adaptive federated multi-task learning across multimodal split networks," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 87–96.

[5] Tiantian Feng, Digbalay Bose, Tuo Zhang, Rajat Hebbar, Anil Ramakrishna, Rahul Gupta, Mi Zhang, Salman Avestimehr, and Shrikanth Narayanan, "Fedmultimodal: A benchmark for multimodal federated learning," *arXiv preprint arXiv:2306.09486*, 2023.

[6] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[7] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[8] Rahul Sharma, Anil Ramakrishna, Ansel MacLaughlin, Anna Rumshisky, Jimit Majmudar, Clement Chung, Salman Avestimehr, and Rahul Gupta, "Federated learning with noisy user feedback," *arXiv preprint arXiv:2205.03092*, 2022.

[9] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[10] 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.

[11] 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.

[12] 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.

[13] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid, "Multi-modal transformer for video retrieval," in *European Conference on Computer Vision*. Springer, 2020, pp. 214–229.

[14] Simon Ging, Mohammadreza Zolfaghari, Hamed Pirsiavash, and Thomas Brox, "Coot: Cooperative hierarchical transformer for video-text representation learning," *Advances in neural information processing systems*, vol. 33, pp. 22605–22618, 2020.

[15] Mohammadreza Zolfaghari, Yi Zhu, Peter Gehler, and Thomas Brox, "Crossclr: Cross-modal contrastive learning for multi-modal video representations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1450–1459.

[16] Chaoyang He, Murali Annavaram, and Salman Avestimehr, "Group knowledge transfer: Federated learning of large cnns at the edge," *Advances in Neural Information Processing Systems*, vol. 33, pp. 14068–14080, 2020.

[17] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan, "IEMOCAP: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, no. 4, pp. 335–359, 2008.

[18] Carlos Busso, Srinivas Parthasarathy, Alec Burmania, Mohammed AbdelWahab, Najmeh Sadoughi, and Emily Mower Provost, "Msp-improv: An acted corpus of dyadic interactions to study emotion perception," *IEEE Transactions on Affective Computing*, vol. 8, no. 1, pp. 67–80, 2016.

[19] Chandra Thapa, Pathum Chamikara Mahawaga Arachchige, Seyit Camtepe, and Lichao Sun, "Splitfed: When federated learning meets split learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, vol. 36, pp. 8485–8493.

[20] Ligeng Zhu, Zhijian Liu, and Song Han, "Deep leakage from gradients," *Advances in neural information processing systems*, vol. 32, 2019.

[21] Chong Fu, Xuhong Zhang, Shouling Ji, Jinyin Chen, Jingzheng Wu, Shanqing Guo, Jun Zhou, Alex X Liu, and Ting Wang, "Label inference attacks against vertical federated learning," in *31st USENIX Security Symposium (USENIX Security 22), Boston, MA*, 2022.

[22] Tiantian Feng, Hanieh Hashemi, Rajat Hebbar, Murali Annavaram, and Shrikanth S Narayanan, "Attribute inference attack of speech emotion recognition in federated learning settings," *arXiv preprint arXiv:2112.13416*, 2021.

[23] Otkrist Gupta and Ramesh Raskar, "Distributed learning of deep neural network over multiple agents," *Journal of Network and Computer Applications*, vol. 116, pp. 1–8, 2018.

[24] Baochen Xiong, Xiaoshan Yang, Fan Qi, and Changsheng Xu, "A unified framework for multi-modal federated learning," *Neurocomputing*, vol. 480, pp. 110–118, 2022.

[25] Qiying Yu, Yang Liu, Yimu Wang, Ke Xu, and Jingjing Liu, "Multimodal federated learning via contrastive representation ensemble," *arXiv preprint arXiv:2302.08888*, 2023.

[26] Yuanyuan Zhang, Jun Du, Zirui Wang, Jianshu Zhang, and Yanhui Tu, "Attention based fully convolutional network for speech emotion recognition," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2018, pp. 1771–1775.

[27] Xin Wang, Devinder Kumar, Nicolas Thome, Matthieu Cord, and Frederic Precioso, "Recipe recognition with large multimodal food dataset," in *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2015, pp. 1–6.

[28] Mimansa Jaiswal and Emily Mower Provost, "Privacy enhanced multimodal neural representations for emotion recognition," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, pp. 7985–7993, Apr. 2020.

[29] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al., "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[30] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[31] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[32] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He, "Federated learning on non-iid data silos: An experimental study," *arXiv preprint arXiv:2102.02079*, 2021.

[33] Feng Wang and Huaping Liu, "Understanding the behaviour of contrastive loss," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2495–2504.

A. LIMITATIONS AND FUTURE WORK

The primarily limitation of our proposed approach is the added computation needed to train local and server models when compared to typical FL model training. Further, since we use pretrained models to extract sentence and image representations, we maybe inadvertently exposed to any underlying dataset biases in these models. We encourage downstream applications of this approach to suitably evaluate the model performance against different demographic cohorts before deployment.

Another area of concern is the privacy risk associated with sharing the representation of a modality to the remote server. Many recent studies have demonstrated that FL is vulnerable to privacy attacks such as data reconstruction attacks [20], label inference attack [21], and property inference attack [22]. In our experiments, we employ preliminary strategies to protect privacy by sharing representations of the shareable modality instead of raw data as well as only sharing representations from modalities without any biometric information (text). Further, we also place a strong constraint in our model training whereby we do not share class labels with the server. Nevertheless, it is still possible that some of aforementioned privacy attacks can achieve high success rates using just the shareable representations. For instance, the privacy attacker may formulate contrastive learning objectives between the model updates and shared representations to implement more effective inference attacks. Our future works plan to extensively investigate the privacy risks associated with sharing modalities in FL.

B. RELATED WORK

Several algorithms have been developed to train with distributed data with disjoint modalities stored on different devices, but each of these have specific drawbacks which we highlight here.

While we can assume that all modalities are restricted to the edge devices and use traditional FL, this poses challenges for training on low power devices and in cases of high data heterogeneity. *Split learning* (SL) [23] is one way to address these concerns by splitting the entire ML model into multiple smaller parts and distributing them on both the central server and edge devices. Each model part can further be constrained to a specific data modality. In a typical SL setting, the first edge device initializes the training by performing forward propagation on local data, and then uploads the *smashed* data (outermost activations of the local model) and ground-truth labels to the central server. In the next step, the central server continues forward propagation from the *smashed* data using a server-side model. Then, the server starts back-propagation on its model and sends the gradients at the cut layer back to the edge for local back-propagation. After finishing back-propagation at the first edge device, it shares the trained model to the next edge device, and the training process continues. While this theoretically allows us to train large multi-modal models, the main drawback here is the sequential training process which makes it considerably slower than FL. Further, there is a significant communication overhead for each model update, since we need to propagate the intermediate activations from edge to server in the forward pass and then back in the backward pass, *for each data sample and on all the edge devices*. In addition, SL can also suffer from server straggler problem as the training process requires frequent communications between the server and the edge devices. In our work, we avoid all of these problems by training device specific encoders until convergence which are then shared with other devices/the centralized server more efficiently (in terms of both communication cost and time). We assume availability of labels on edge for local modal training, but in their absence we can leverage user feedback similar to [8].

Federated Group Knowledge Transfer (FedGKT) [16] was originally proposed to address the challenges of training with non-IID datasets; it can also be applied to train a large multi-modal model in a distributed manner since it allows us to train device/modality specific encoders till convergence which are then aggregated centrally using knowledge distillation. However, FedGKT assumes that edge specific labels are also shared with the server which enables them to train a server side classifier till convergence, but these labels may not always be available, or not permitted to be shared freely with the server. For example, patients may prefer not to share their diagnosis information (class labels) outside the hospital. In our work, we therefore avoid label sharing entirely by using a contrastive objective to train the device specific encoders, and show strong model performance without sharing any edge specific labels with the central server.

Split Federated Learning (SFL) [19] is a recent work which attempts to bridge SL and FL. However, this also carries heavy communication costs due to the combined overheads of split learning and federated learning, which limits its applications to practical settings. Further, in our experiments, they do not demonstrate any gains in accuracy compared to our baselines.

Finally, while federated learning has been studied extensively in uni-modal settings, multi-modal federated learning has recently gained attention from several works [4, 24, 25, 5].

C. LEARNING ALGORITHM

Detailed training steps are as follows:

Step 1 (server): At the beginning of each global round, the server samples the edge devices and sends the global model $\mathcal{F}_g(\cdot)$ parameterized by θ_g and $\mathbf{z_T}$ to each device.

Step 2 (edge): After receiving θ_g and the server side embeddings $\mathbf{z}_T^k$, the edge device trains the global $\mathcal{F}_g(\cdot)$ on its local image data $\mathbf{X_I}$. The learning objective is a combination of cross-entropy loss $\mathcal{L}_{CE}$ and the cross-modal contrastive loss $\mathcal{L}^{T \to I}$. We use the weight parameter β to set the importance of the cross-modal contrastive loss. The combined loss for the

image global model is shown below:

$$\mathcal{L}_{glob} = \mathcal{L}_{CE}(\mathcal{F}_g(\theta_g; \mathbf{X}_I^k), y^k) + \beta \mathcal{L}^{T \to I}(\mathbf{z_I}, \mathbf{z_T}) \quad (3)$$

Step 3 (edge): Next, the edge device trains its local model using $\mathbf{X_T}$. Here, the training objective is a weighted sum between the cross-entropy loss $\mathcal{L}_{CE}$ and the contrastive loss $\mathcal{L}^{S \to L}$.

$$\mathcal{L}_{loc} = \mathcal{L}_{CE}(\mathcal{F}_k(\theta_k; \mathbf{X}_T^k), y^k) + \beta \mathcal{L}^{S \to L}(\mathbf{z'_T}, \mathbf{z_T}) \quad (4)$$

Step 4 (server): Finally, the server receives the trained global model and edge generated embeddings $\mathbf{z'_T}$. The server then aggregates the global model using weighted average and also trains the server side model $\mathcal{F}_s(\cdot)$ using the contrastive loss $\mathcal{L}^{L \to S}$:

We summarize the full training algorithm in Algorithm 1 and a pictorial representation of the same in Figure 3 in Appendix C.

Figure 3 shows a pictorial representation of our full learning algorithm, which is listed in detail in Algorithm 1.

D. EXPERIMENT DETAILS

D.1. Dataset

D.1.1. IEMOCAP

The IEMOCAP database [17] was collected using multi-modal sensors to capture motion, audio, and video of human interactions. The original corpus contains 10,039 utterances from ten subjects (five male and five female) expressing various categorical emotions from improvised and scripted scenarios. Following [26], we focus on the improvised sessions. We use the four most frequent emotion labels: neutral, sad, happiness, and anger for training the SER model due to the data imbalance in other labels.

D.1.2. MSP-Improv

The MSP-Improv corpus [18] is a multi-modal emotion recognition data set captured from improvised scenarios. The data is collected from 12 speakers (six male and six female) and includes audio and textual data of utterances spoken in different recording conditions.

D.1.3. UPMC Food101

The UPMC Food101 dataset [27] consists of web pages with textual recipe descriptions for 101 food labels automatically retrieved online. Each page was matched with a single image, where the images were obtained by querying Google Image Search for the given category. Examples of food labels are Filet Mignon, Pad Thai, Breakfast Burrito and Spaghetti Bolognese. The web pages were processed with html2text[1] to obtain the raw text.

[1] github.com/aaronsw/html2text

Algorithm 1 PartialFL

Server Initialize: θ_g^0, θ_s
for $k \in \{1, 2, ..., K\}$ **do**
  Client Initialize: θ_k
  Upload $\mathbf{X_T^k}$ **to server**
end for
Server Executes:
for $k \in \{1, 2, ..., K\}$ **do**
  $\mathbf{z_T^k} \leftarrow \mathcal{F}_s(\theta_s; \mathbf{X_T^k})$
end for
for Each round $t = 0, ..., T-1$ **do**
  ## Step 1: Sample and distribution
  Sample edge devices $\mathcal{S} \in \{1, 2, ..., K\}$
  Distribute θ_g^t and $\mathbf{z_T^k}$
  ## Step 2 and 3: Client training
  for Each edge device $k \in \mathcal{S}$ in parallel **do**
    $\theta_{g,k}^t, \theta_k \leftarrow$ **ClientTrain(**θ_g^t**,**θ_k**,**$\mathbf{z_T^k}$**,**$\mathcal{D}^k$**)**
    $\mathbf{z_T'^k} \leftarrow \mathcal{F}_k(\theta_k; \mathcal{D}^k)$
    Upload $\theta_{g,k}^t$, $\mathbf{z_T'^k}$ **to server**
  end for
  ## Step 4.1: Server trains θ_s
  for Each edge device $k \in \mathcal{S}$ **do**
    $\theta_s \leftarrow$ **ServerTrain(**θ_s^t**,** $\mathbf{z_T'^k}$**,** $\mathbf{X}_T^k$**)**
  end for
  for $k \in \{1, 2, ..., K\}$ **do**
    $\mathbf{z_T^k} \leftarrow \mathcal{F}_s(\theta_s; \mathbf{X_T^k})$
  end for
  ## Step 4.2: Server aggregates $\theta_{\mathbf{g,k}}^{\mathbf{t}}$
  $\theta_{\mathbf{g}}^{\mathbf{t+1}} \leftarrow \frac{1}{|\mathcal{S}|} \sum_{k \in \mathcal{S}} \theta_{\mathbf{g,k}}^{\mathbf{t}}$
end for
function CLIENTTRAIN**(**θ_g**,** θ_k**,** $\mathbf{z_T}$**,** $\mathcal{D}$**)**
  for Local epoch e from 0 to $E-1$ **do**
    for Iteration i from 0 to $I-1$ **do**
      Sample mini-batch $\mathbf{l}$ from $\mathcal{D}$
      $\theta_g \leftarrow \theta_g - \eta \nabla_{\theta_g} \mathcal{L}_{glob}(\theta_g; \mathcal{D}^{\mathbf{l}}, \mathbf{z_T^l})$
      $\theta_k \leftarrow \theta_k - \eta \nabla_{\theta_k} \mathcal{L}_{local}(\theta_k; \mathcal{D}^{\mathbf{l}}, \mathbf{z}_T^{\mathbf{l}})$
    end for
  end for
  return θ_g, θ_k, $\mathbf{z'}_T$
end function
function SERVERTRAIN**(**θ_s**,** $\mathbf{z'_T}$**,** $\mathbf{X_T}$**)**
  for Iteration i from 0 to $I-1$ **do**
    Sample mini-batch $\mathbf{l}$ from $\mathbf{X_T}$
    $\theta_s \leftarrow \theta_s - \eta \nabla_{\theta_s} \mathcal{L}_{server}(\theta_s; \mathbf{X_T^l}, \mathbf{z_T'^l})$
  end for
  return θ_s
end function

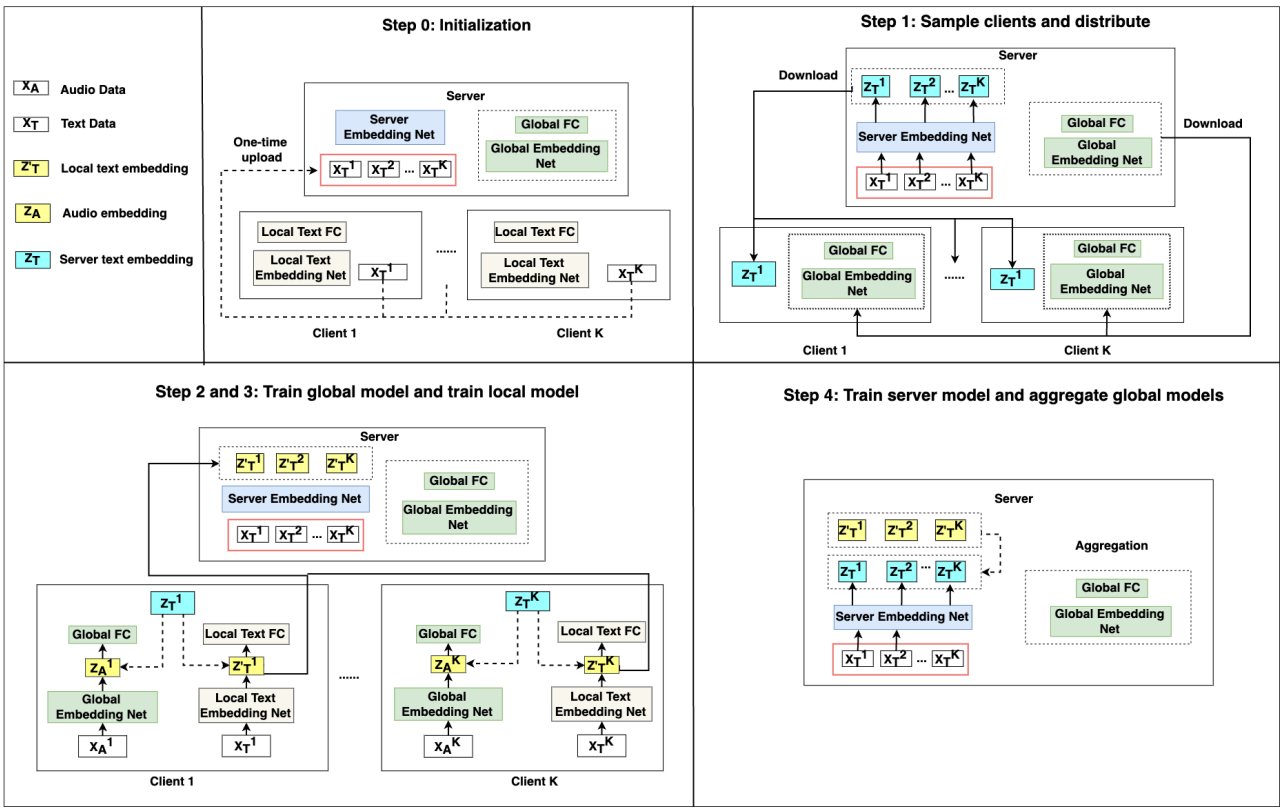

Fig. 3. Training steps of the PartialFL framework. Training steps 1, 2, 3, and 4 are repeated in each global training round.

Table 1. Statistics of data sets used in this study.

	Train	Dev	Test	
Food101	58545	6556	21695	
	Neu.	**Hap.**	**Sad**	**Ang.**
IEMOCAP	1099	947	608	289
MSP-Improv	2072	1184	739	585

D.2. Data Statistics

Data statistics are shown in Table 1.

D.3. Models and Features

D.3.1. Audio

The model we use for the SER task is similar to [28]. The network has 2 main components: an embedding network and an emotion classifier. The embedding network of the audio data is a set of convolution layers proposed in [29], followed by a Gated Recurrent Unit (GRU) layer [30]. We obtain an utterance-level audio representation by applying mean pooling to the output of the GRU layer. We then use a projection layer to compute the audio embedding. Finally, the emotion

Table 2. Ablation study: comparisons between PartialFL and FL on the Food101 data set under different data heterogeneity settings.

Classification Modality	Exp. Setting	Alpha	Top5 Accuracy
Image	FL	1.0	51.08%
	PartialFL		**53.03%**
	FL	0.5	48.27%
	PartialFL		**51.20%**
	FL	0.1	36.20%
	PartialFL		**39.54%**
Image+Text	FL	1.0	56.04%
	PartialFL		**58.62%**
	FL	0.5	52.91%
	PartialFL		**55.94%**
	FL	0.1	40.68%
	PartialFL		**43.74%**

Table 3. Ablation study: Impact of missing modalities in some edge devices; in the Food-101 dataset, the Global model uses Image modality only and we report Top 5 Accuracy; In IEMOCAP and MSP-Improv, we use the Audio only Global model and report UAR.

Dataset	α	q	Scores
Food-101	**1.0**	100%	58.62%
		50%	58.04%
		25%	57.58%
		0%	56.04%
	0.5	100%	51.2%
		50%	50.94%
		25%	50.31%
		0%	48.27%
	0.1	100%	39.54%
		50%	39.47%
		25%	38.65%
		0%	36.20%
IEMOCAP	**-**	100%	61.30%
		50%	59.37%
		0%	56.93%
MSP-Improv	**-**	100%	45.14%
		50%	44.97%
		0%	43.90%

classifier takes the audio embedding and estimates emotion labels using a set of dense layers. Inputs to the embedding network are 80 dimensional Mel filter bank (MFB) features using 25ms Hamming window with step size of 10ms. We further apply z-normalization to these features within each speaker before feeding them to the embedding network.

D.3.2. Image

Similar to the SER model, the image model consists of an embedding network and a classifier. The image embedding network is a set of dense layers followed by a projection layer. We use the image embedding to generate the prediction output. We use the MobileNetV2 [31] pre-trained model to extract representations of raw images which are passed to the embedding network. We use this model since it is small enough to fit most edge devices.

D.3.3. Text

For the text model, we use an embedding network which maps DistilBert [7] sentence representations into a lower-dimensional text embedding. We use a considerably larger embedding network (compared to the edge) in our server. The text model on edge also includes a classifier module (fully connected layer with an outer softmax layer) to generate predictions using local text embeddings.

D.3.4. Multi-modal Model

In the multi-modal global model, we concatenate audio (or image) representations with text representations and pass them through a projection layer to obtain the multi-modal embedding. The classifier then uses this multi-modal embedding to generate the predictions for corresponding tasks.

E. ADDITIONAL RESULTS

E.1. Results on UPMC Food-101 dataset

We report top-1 and top-5 accuracies in this data set. Model performances and sizes for this dataset are shown in Figure 4.

E.1.1. Uni-modal global model (image only)

Firstly, the centralized image model shows the best performance, largely thanks to a larger model size. Furthermore, we can observe that SL consistently yields better performance than FL. Under FL settings, we find that FedProx outperforms FedAvg, and the top-5 accuracy of FL baseline is about 4% higher using the FedProx algorithm when compared with FedAvg. PartialFL outperforms the FL baseline by 1.95% by making use of the shared data modality. SL outperforms PartialFL by only 1.76%, but at a significant communication overhead.

E.1.2. Multi-modal global model (image+text)

Similar to the previous setting, we observe the centralized multi-modal models showing better performance compared to other baselines and FedProx outperforms FedAvg in both FL and PartialFL experiments. Our proposed PartialFL approach improves the global model performance by 2.58% comparing to the FL baseline.

E.2. Impact of non-IID settings

In this experiment, we explore how PartialFL performs under a non-IID setting with data heterogeneity, which is frequently encountered in most real world FL applications. We simulate non-IID edge devices by controlling the α parameter [32] when creating the data sets, which controls the concentration of the Dirichlet distribution in allocating proportion of label samples to each device (so a small alpha corresponds to more imbalanced data distribution, and hence higher data heterogeneity). We perform this experiment only on the Food-101 data set since the speech data sets are small (with each speaker as a separate edge device). Table 2 presents model comparisons between the PartialFL and FL baselines. We only present results with FedProx, as it showed better performance than

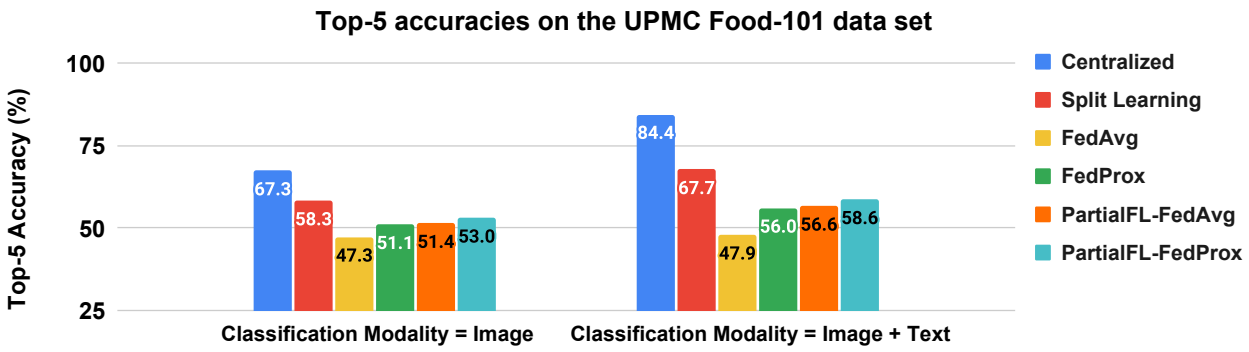

Fig. 4. Top-5 accuracies on test set for PartialFL compared to other baselines in the UPMC Food-101 data set. In all cases, we repeat the experiment 5 times with different seed numbers and report average performances. In PartialFL, we report best performance from different temperature values. We use $\alpha = 1.0$ in all federated experiments (FL, SL and PartialFL)

FedAvg. As expected, performances of both PartialFL and traditional FL decrease as α value reduces, but PartialFL consistently outperforms FL baselines in all settings. Furthermore, we observe that the PartialFL provides larger performance improvements against FL in non-IID settings (compared to the IID setting of $\alpha = 1.0$), by leveraging the shared modality.

E.3. Impact of missing modalities in edge devices

In practice, not every edge device may contain multi-modal data; for example, in commercial SLU systems, some devices are designed to only accept audio inputs while others may accept audio, video and text inputs. In this setting, we evaluate performance of PartialFL when the shareable modality is missing in a subset of edge devices. We define q as the percentage of the devices with multi-modal data, and we simulate the case with various values of q. $q = 0\%$ indicates the case where none of the edge devices contain the shareable modality, and is equivalent to the FL baseline. We explore $q \in \{0\%, 25\%, 50\%, 100\%\}$ and $q \in \{0\%, 50\%, 100\%\}$ in the Food-101 data set and emotion recognition data sets, respectively. Results are presented in Table 3. In general, we observe that fewer the number of edge devices with the shareable data, lower the performance of the global model. However, this decrease is not substantially large; for example, when $q = 25\%$, we find that the top-5 accuracy drops by around 1% in the Food-101 data set. These results suggest that PartialFL is robust to missing modalities in some devices and the performance decrease is not substantial.

E.4. Impact of temperature τ

The temperature parameter τ in the contrastive loss objectives defines the strength of penalties on the hard negative samples and often has a substantial impact on the final model performance [33]. In this study, we examine its impact on PartialFL. Table 4 shows performances of the uni-modal global model with different temperature parameters in all the datasets we explored. As we can see from the table, performance differences at different temperature values are close to each other ($< 1\%$) among all values of τ, suggesting that the PartialFL is fairly robust to this hyper-parameter.

Table 4. Ablation study: Impact of temperature (τ) on the PartialFL algorithm.

Model (metric)	Dataset	α	τ	Score
Image (top-5 acc)	Food-101	1.0	0.05	53.03%
			0.1	52.67%
			0.2	51.95%
		0.5	0.05	51.2%
			0.1	50.86%
			0.2	49.93%
		0.1	0.05	39.42%
			0.1	39.54%
			0.2	39.38%
Audio (UAR)	IEMOCAP	-	0.05	60.71%
			0.1	61.30%
			0.2	60.52%
	MSP-Improv	-	0.05	44.68%
			0.1	44.81%
			0.2	45.14%