HHS Public Access

Author manuscript

IEEE/ACM Trans Audio Speech Lang Process. Author manuscript; available in PMC 2022 January 01.

Published in final edited form as:

IEEE/ACM Trans Audio Speech Lang Process. 2021 ; 29: 1204–1219. doi:10.1109/taslp.2021.3061885.

Meta-learning with Latent Space Clustering in Generative Adversarial Network for Speaker Diarization

Monisankha Pal [Member, IEEE], **Manoj Kumar [Member, IEEE]**, **Raghuveer Peri [Member, IEEE]**, **Tae Jin Park [Member, IEEE]**

Signal Analysis and Interpretation Laboratory, University of Southern California, Los Angeles, USA

So Hyun Kim,

Center for Autism and the Developing Brain, Weill Cornell Medicine, USA

Catherine Lord,

Semel Institute of Neuroscience and Human Behavior, University of California Los Angeles, USA

Somer Bishop,

Department of Psychiatry, University of California, San Francisco, USA

Shrikanth Narayanan [Fellow, IEEE]

Signal Analysis and Interpretation Laboratory, University of Southern California, Los Angeles, USA

Abstract

The performance of most speaker diarization systems with x-vector embeddings is both vulnerable to noisy environments and lacks domain robustness. Earlier work on speaker diarization using generative adversarial network (GAN) with an encoder network (ClusterGAN) to project input x-vectors into a latent space has shown promising performance on meeting data. In this paper, we extend the ClusterGAN network to improve diarization robustness and enable rapid generalization across various challenging domains. To this end, we fetch the pre-trained encoder from the ClusterGAN and fine tune it by using prototypical loss (meta-ClusterGAN or MCGAN) under the meta-learning paradigm. Experiments are conducted on CALLHOME telephonic conversations, AMI meeting data, DIHARD-II (dev set) which includes challenging multi-domain corpus, and two child-clinician interaction corpora (ADOS, BOSCC) related to the autism spectrum disorder domain. Extensive analyses of the experimental data are done to investigate the effectiveness of the proposed ClusterGAN and MCGAN embeddings over x-vectors. The results show that the proposed embeddings with normalized maximum eigengap spectral clustering (NME-SC) back-end consistently outperform the Kaldi state-of-the-art x-vector diarization system. Finally, we employ embedding fusion with x-vectors to provide further improvement in diarization performance. We achieve a relative diarization error rate (DER) improvement of 6.67% to 53.93% on the aforementioned datasets using the proposed fused embeddings over x-vectors. Besides, the

mp_323@usc.edu.

MCGAN embeddings provide better performance in the number of speakers estimation and short speech segment diarization compared to x-vectors and ClusterGAN on telephonic conversations.

Index Terms—

ClusterGAN; MCGAN; NME-SC; speaker diarization; speaker embeddings; x-vector

I. Introduction

Speaker diarization [1], the task of determining "who spoke when" in a multi-speaker audio stream, has a wide range of applications such as in information retrieval, speaker-based indexing, meeting annotations, and conversation analysis [2]. Present-day diarization systems typically comprise four components: (a) A speech segmentation module that removes the non-speech parts using a speech activity detector (SAD) and segments the speech part into multiple speaker-homogeneous short segments [3]; (b) A speaker representation (embedding) extractor that maps the segments into fixed-dimensional *speaker embeddings* such as i-vectors [4], [5], d-vectors [6], [7], [8] and *x-vectors* [9], [3], [10]; (c) A clustering module that determines the number of constituent speakers in an audio recording and clusters the extracted embeddings into these speakers [11], [12]; (d) A re-segmentation module that refines the clustering results [3].

For embedding extraction, typically i-vectors have been obtained through total variability space projection [13]. However, recently significant performance improvement has been shown using deep neural network embeddings such as d-vectors with architectures such as LSTM [7], [14], CNN [15]; and x-vectors with time-delay neural network (TDNN) [3], [16]. The combination of different embeddings, e.g., c-vectors using 2D self-attentive structure, has also been proposed to exploit the complementary merits of each embedding [17].

In terms of clustering, most of the existing algorithms that have been used in speaker diarization are unsupervised. Among them, agglomerative hierarchical clustering (AHC) [3] and spectral clustering (SC) [18] using pairwise embedding similarity measurement techniques like cosine distance [7], [11], PLDA [19] and using an LSTM [12] are the most popular. Similarly, other unsupervised clustering methods such as Gaussian mixture model [4], [15], mean-shift [5], k-means [20], and links [21] have also been adopted for speaker diarization. Moreover, clustering depends on tuning hyperparameters like stopping threshold (for AHC), the p-value for binarization of affinity matrix (for SC). However, more recently, an auto-tuning and improved version of the spectral clustering approach on x-vectors using cosine similarity measure, which is called as *normalized maximum eigengap spectral clustering (NME-SC)* was introduced in [11]. Despite the success of these speaker clustering algorithms, speaker diarization remains a challenging task due to the wide heterogeneity and variability of audio data recorded in many real-world scenarios [22].

The other approach for speaker clustering has been based on supervised methods. A fully supervised speaker diarization framework, named UIS-RNN was proposed in [8]. Although this model for clustering produces excellent performance in telephone conversations, its performance deteriorates in a more challenging multi-domain database like DIHARD-II

[23]. To improve the UIS-RNN diarization performance further, a novel sample-mean loss function to train the RNN has been introduced very recently [23]. Efforts have been made to automatically deal with speaker-overlapping speech segments and directly optimize an end-to-end neural network based on diarization errors [24]. The network is trained in a supervised manner using a permutation free objective function. The diarization performance was further enhanced by introducing a self-attention based end-to-end neural network [25]. Although the above methods do not rely on clustering and can directly compute the final diarization outputs using a single network, they assume that the number of speakers is known apriori or at least bounded to two speakers. Along these lines, the performance of deep embedded clustering, which was originally proposed in [26], was incorporated and modified for speaker clustering in diarization task [20]. The limitation of this work is that a good estimate of the number of speakers is needed for its evaluation.

While performance of tasks such as speech and speaker recognition have improved significantly due to supervised deep learning approaches, most of the speaker clustering is yet to take advantage of similar techniques. The main problem that hinders in making clustering a supervised task is associated with the fact that speaker labels are ambiguous (e.g., both “112233” and “223311” sequences of labels are equally correct for the same diarization session). In our earlier proposed work, we incorporated *ClusterGAN* to non-linearly transform DNN-based speaker embeddings into a low-dimensional latent space better suited for clustering [27]. The proposed ClusterGAN, which exploits the GAN latent space with the help of an encoder network, was trained with a combination of adversarial loss, latent variable recovery loss, and clustering-specific loss. Although the proposed system showed significant performance improvement over x-vector based state-of-the-art in meeting and child-adult interaction corpora, its performance was not tested against telephone conversations and a broader set of multi-domain data.

In this work, a ClusterGAN network which was originally proposed for image clustering [28], is adopted and modified for the speaker clustering task in the speaker diarization framework. The GAN and the encoder network are trained jointly in a supervised manner with clustering-specific loss and latent embeddings are extracted using the trained encoder to perform unsupervised clustering at the back-end. Two main advantages of GAN-based latent space clustering are the interpretability and interpolation in the latent space [28]. We use ClusterGAN-trained encoder network as initialization to further fine-tune it with meta-learning based *prototypical loss* function [29], [30]. This is represented as *meta-ClusterGAN or MCGAN* in this paper. The prototypical network was introduced for the few-shot image classification task [29] and is the state-of-the-art approach on a few-shot image classification benchmark. The motivation behind using proto-learning for our task is that it has a simpler inductive bias in the form of speaker prototypes and can perform rapid generalization to new speakers or types of data not seen while training. The prototypical loss trained for learning a metric space to mimic the test scenario will be beneficial in capturing information related to both generalization and clustering objectives.

The main contributions of this paper are: (a) A novel speaker diarization framework based on prototypical learning; (b) Extensive multi-domain experimental evaluation and analysis of the proposed diarization system on various challenging speaker diarization corpora; (c)

Demonstration of the use of novel speaker embeddings that outperform x-vectors through analysis across various challenging scenarios.

II. Related work

A. Deep clustering algorithms

Using deep neural networks to non-linearly transform the input data into cluster-friendly representation along with dimension reduction is commonly known as deep clustering [31]. Recent deep clustering methods on image data using autoencoder networks like deep embedded clustering (DEC) [26] achieve impressive clustering performance. Generative modeling based approaches like variational deep embedding [32], information maximizing GAN (InfoGAN) [33], GAN mixture model [34], [35] learn latent representation space and can interpolate to generate new samples from the data distribution. In all these algorithms, the deep neural network is usually trained on two types of losses: representation loss or network loss and clustering-specific loss. The network loss is essential for network initialization and is used to learn feasible latent features. The different network losses are reconstruction loss of autoencoder, variational loss of a variational autoencoder, and adversarial loss of GANs. On the other hand, clustering-specific loss helps to learn representations suitable for clustering. The option for clustering-specific losses are assignment losses like k-means loss [36], cluster assignment hardening loss [26], agglomerative clustering loss [37], spectral clustering loss [38] or regularization losses such as locality preserving loss, cluster classification loss [31]. Different ClusterGANs proposed for image data clustering adopt adversarial loss in GAN and clustering-specific loss like balanced self-paced entropy minimization loss [39] or cluster classification loss [28]. Very recently, few deep clustering approaches like transformer-based discriminative neural clustering model [40], deep clustering loss in end-to-end neural speaker diarization [25], deep embedded clustering [20], and ClusterGAN [27] have been used for speaker diarization. Although multifarious deep clustering approaches have been successfully applied for image data clustering, their application toward speaker diarization has been limited mainly due to the problem of the unknown number of speakers in a given diarization session.

B. Meta-learning algorithms

Inspired by human learning of new categories (classes) given just a very few examples, the meta-learning model trained over a large variety of learning tasks can adapt or generalize well to potentially unseen tasks [41]. It is also known as learning-to-learn, which learns on a given task and also across tasks. In the computer vision literature, there are three common approaches to meta-learning: metric-based [42], model-based [43], and optimization-based [44]. Metric learning aims at learning a metric or distance function over the embedding space. Among metric-learning based approaches, Siamese networks [45] and triplet networks [46] for learning speaker embeddings have been proposed for speaker recognition [42], [47], [48] and speaker diarization [49], and have yielded promising performances. The prototypical network that learns a metric space by computing prototype representation of each class is a state-of-the-art approach for few-shot image classification tasks [29]. Along these lines, prototypical loss to optimize a speaker embedding model for the speaker

verification task was explored in [30], [48], [50]. The resulting model provides superior performance to triplet loss based models. More recently, the use of protonets for child-adult audio classification task was explored in [51]. Our proposed approach uses prototypical loss (PTL) to fine-tune the encoder of ClusterGAN for robust speaker embedding extraction in the speaker diarization framework.

III. Background

A. Generative adversarial network (GAN)

The standard GAN is formulated as an adversarial mini-max game between two neural networks: a generator (G) and a discriminator (D) [52]. The generator aims to create a map from latent space to data space, i.e., $G:\mathbf{Z}\rightarrow\hat{\mathbf{x}}$. It takes random noise $\mathbf{z}$ sampled from $p_{\mathbf{z}}$ and synthesizes data similar to original data to fool the discriminator. The discriminator is considered to be a mapping from the data space to a real value $D:\mathbf{x}\rightarrow\mathbb{R}$. It takes real data $\mathbf{x}$ sampled from $p_{\mathbf{x}}^{r}$ and aims to distinguish the real data from the generator produced samples. Training the GAN is equivalent to optimizing the following objective function

$$\min_{G}\max_{D} U_{\mathrm{GAN}}(D,G)=\mathbb{E}_{\mathbf{x}\sim p_{\mathbf{x}}^{r}}[\log D(\mathbf{x})]-\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}[\log(1-D(G(\mathbf{z})))] \quad (1)$$

Although GANs can learn to mimic any data distribution, they are difficult to train due to the mode collapse problem [53]. To address this issue, several variants of GANs such as Wasserstein GAN (WGAN) [53], and improved WGAN [54] (IWGAN) have been proposed in the literature.

B. Prototypical networks

Deep metric-learning based approaches were developed within the meta-learning paradigm to address generalization in few-shot learning. Among metric-learning based approaches, prototypical networks, or protonets, apply a simpler inductive bias (in the form of class prototypes) as compared to other metric-learning based methods and shown to achieve state-of-the-art few-shot performance on image classification [29] and natural language processing tasks [55]. The key assumption is that there exists an embedding in which samples from each class cluster around a single prototype representation of that class. Protonets learn a non-linear transformation into an embedding space, where every class is represented by its prototype, sample mean of its support set in the embedding space. During inference, an embedded query sample is assigned to its nearest prototype.

Protonet is trained episodically, where each episode is one mini-batch consisting of N_C categories randomly sampled from total K categories. The mini-batch also contains a labeled set of examples (*support* set S) and unlabeled data (*query* set Q) to predict classes. Consider the support set S of N labeled examples as $S=\{\mathbf{x}_i, y_i\}_{i=1}^{N}$, where each sample $\mathbf{x}_I$ is a D-dimensional feature vector and the corresponding label $y_i\in\{1,\ldots,K\}$. We denote $S_k\subseteq S$ as the set of examples labeled with class k. The protonet learns a non-linear mapping $f_{\boldsymbol{\Psi}}:\mathbb{R}^D\rightarrow\mathbb{R}^M$. The M-dimensional prototype of each class is computed as the mean of the embedded support points belonging to that class

$$\mathbf{p}_k = \frac{1}{|S_k|} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_\psi(\mathbf{x}_i) \tag{2}$$

where ψ is the learnable parameters of the protonet.

During training, every query sample $\{\mathbf{x}_j, y_j\} \in Q$ is classified against K classes based on a soft-max over the distances to each class prototypes in the new embedding space:

$$p_{\boldsymbol{\psi}}(y = y_j \mid \mathbf{x}_j) = \frac{\exp\left(-d\left(f_{\boldsymbol{\psi}}(\mathbf{x}_j), \mathbf{p}_{y_j}\right)\right)}{\sum_{k'} \exp\left(-d\left(f_{\boldsymbol{\psi}}(\mathbf{x}_j), \mathbf{p}_{k'}\right)\right)} \tag{3}$$

where $d(.)$ represents a distance function. Learning proceeds by minimizing the loss function $L_{\text{PTL}} = -\log p_{\boldsymbol{\psi}}(y = y_j|\mathbf{x}_j)$ of the true class $y = y_j$.

IV. Proposed speaker diarization system

An overview of our proposed speaker diarization system is shown in Fig. 1. The non-speech part in a given multi-speaker conversation is removed first by using a speech activity detection (SAD) system. Our diarization system uses Kaldi[1] style uniform segmentation and the segments are embedded into a fixed-dimensional vector using a time-delay neural network (TDNN), which is commonly known as x-vector [16]. The proposed meta-ClusterGAN (MCGAN) is developed on top of x-vectors to perform deep latent space clustering for speaker diarization. The motivation behind introducing MCGAN is to non-linearly transform the input x-vectors (trained with categorical cross-entropy loss) into another embedding suitable for speaker clustering and that can generalize well to new classes (here, speakers) not seen while training. As shown in Fig. 1, the proposed MCGAN training has two phases: (a) parameter initialization using a ClusterGAN, trained with clustering-specific loss in GAN latent space (MCGAN pre-training), and (b) inducing robustness to the initialized encoder in ClusterGAN by further fine-tuning it with meta-learning based prototypical loss (MCGAN fine-tuning). We describe each of the modules in the diarization pipeline below.

A. Segmentation

In this paper, our proposed system uses oracle SAD for all the analysis and experiments, following common practice in the speaker diarization literature [3], [8], [56]. Therefore, our approach starts with a temporal uniform segmentation of 1.5 sec with an overlap of 1 sec between two adjacent segments. This denser segmentation gives more number of samples while evaluating a diarization session and it helps in clustering.

B. Speaker embedding vector

The speaker embedding vectors used to train the MCGAN models are x-vectors, which are fixed-length representation using a TDNN from variable-length utterances. In this approach, MFCCs are first extracted at frame-level and input to a TDNN for supervised training using

[1] https://kaldi-asr.org/

the categorical cross-entropy loss based on the speaker labels. The statistics pooling layer inside the TDNN architecture is used to convert frame-level features into a segment-level embedding. The detailed procedure of x-vector extraction is concisely described in [3], [16]. In this paper, we use Kaldi-based pre-trained x-vectors.

C. Meta-ClusterGAN (MCGAN) pre-training/ClusterGAN training

We pre-train the MCGAN encoder using ClusterGAN training since it can decipher the original data representation by exploiting the GAN latent space. The learned encoder in ClusterGAN can generate embeddings in another space while maintaining the separable properties among the classes. ClusterGAN comprises three components: generator (G), discriminator (D) and encoder (E). The complete ClusterGAN architecture is shown in a red dashed rectangle in Fig. 2 and its training procedure is described in detail below.

1) Motivation: Although the main focus of speaker clustering is to separate out the original data into speakers, it would be more appealing and easier if it could be simply accomplished with dimensionality reduction. Deep clustering models can non-linearly transform the input data into a cluster-friendly representation with dimensionality reduction and have the capacity to deal with large scale datasets. GAN is a powerful class of deep generative model, which has the ability to capture high-dimensional real data distributions and can impute missing data. Moreover, the latent space of GAN has good interpretability and interpolation ability [28]. The use of GAN latent space with an inference network for supervised, semi-supervised, and unsupervised tasks has been explored in [33], [57], [58]. GAN with an inference network as the classifier is employed for semi-supervised classification task in [59]. Along these lines, ClusterGAN is designed specifically for clustering to utilize the GAN latent space, using an inference network and clustering-specific loss, to preserve cluster structure in the disentangled latent variables.

2) Adversarial training: ClusterGAN adopts adversarial training of GANs for the clustering task. In this work, we incorporate improved WGAN [54] (IWGAN) as our GAN network. The objective function of this adversarial game between G and D

$$\min_{G}\max_{D} U_{\text{IWGAN}}(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{\mathbf{x}}^{r}}[D(\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}[D(G(\mathbf{z}))] + \lambda \cdot \text{GP} \tag{4}$$

where λ denotes the gradient penalty coefficient and GP represents the gradient penalty term [54]. The gradient penalty term can be expressed as

$$\text{GP} = \mathbb{E}_{\hat{\mathbf{x}}\sim p_{\hat{\mathbf{x}}}}\left[\left(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1\right)^2\right] \tag{5}$$

where $\hat{\mathbf{x}} = \epsilon\mathbf{x} + (1-\epsilon)G(\mathbf{z})$ and ϵ is a random number uniformly sampled in between 0 and 1. As shown in Fig. 2, we employ pre-trained x-vectors as real data input to the GAN discriminator.

3) Mixture of discrete and continuous latent variables: One possible way to perform clustering in the latent space is to back-project the data into the GAN latent space and then cluster it. The latent vectors for GANs trained with different priors such as

Gaussian or uniform distribution usually lead to bad clustering [60]. Although the latent space may contain useful information about the data, the distance geometry does not reflect any form of clustering. To combat this issue, boosting the latent space using categorical variables ($\mathbf{z}_c$) to form non-smooth geometry is essential. The discrete variable $\mathbf{z}_c$ (using the original speaker label) as a mixture with the continuous random variable ($\mathbf{z}_n$) will restrict the GAN generator to produce each mode only generating samples from a corresponding category in the real data. A similar type of latent variable structure within a GAN generator for learning disentangled and meaningful representation was employed in InfoGAN [33]. However, ClusterGAN has been reported to be superior to InfoGAN for clustering [28]. Furthermore, continuity in the latent space is also required for good interpolation objective and GANs have good interpolation ability. Therefore, our latent variable $\mathbf{z}$ is a concatenation of $\mathbf{z}_n$ and $\mathbf{z}_c$ shown in Fig. 2. In this work, we use $\mathbf{z}_n \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_{d_n})$, where we chose a small value of variance (σ) as 0.10 to make the clusters separated. We use $\mathbf{z}_c$ as a one-hot encoded vector by using the original speaker labels in the training data. Thus, our ClusterGAN training is supervised in nature. The mixture of $\mathbf{z}_n$ and $\mathbf{z}_c$ as the prior enables clustering in the latent space.

4) Inverse mapping network: Inverse mapping from data space to latent space is a non-trivial problem, as it requires the inversion of the generator, which is a multi-layered non-linear model. The work proposed in [60], [61], tackles this issue by solving an optimization problem in $\mathbf{z}$ to recover the latent vectors using $\mathbf{z}^* = \text{argmin}_{\mathbf{z}} L(G(\mathbf{z}), \mathbf{x}) + \lambda \|\mathbf{z}\|_p$, where L is a suitable loss function, λ is a regularization constant and $\|\cdot\|_p$ denotes the norm. However, this optimization is non-convex in $\mathbf{z}$ and there exist multiple $\mathbf{z}$ values to describe a single real data $\mathbf{x}$ [28], [61]. To mitigate these issues, the stochastic clipping of $\mathbf{z}$ at each iteration step was used in [60]. However, the above approaches are not amenable to clustering. In this work, we train a separate encoder (E) network (shown in Fig. 2) alongside the GAN network to learn the inverse mapping function of the generator, estimating discriminative latent embeddings for the real data. For every mini-batch, we sample $\mathbf{z}_c$ as the speaker labels of the corresponding real data, and sample $\mathbf{z}_n$ from a normal distribution. Moreover, to enforce precise recovery of $\mathbf{z}_n$, we compute the numerical difference between $\mathbf{z}_n$ and corresponding encoder output $\hat{\mathbf{z}}_n$. We empirically found that instead of mean square error, cosine distance is more suitable in the embedding space for distance calculation. The objective function related to this task is

$$\min \text{COS}(G, E) = \frac{1}{m} \sum_{i=1}^{m} \left[1 - \frac{E\left(G\left(\mathbf{z}_n^i\right)\right) \cdot \mathbf{z}_n^i}{\|E\left(G\left(\mathbf{z}_n^i\right)\right)\| \|\mathbf{z}_n^i\|} \right] \tag{6}$$

where m is the mini-batch size.

5) Clustering-specific loss: We introduce a clustering-specific loss to learn cluster-friendly representation. For that, we employ cross-entropy (CE) loss, which is computed between $\mathbf{z}_c$ and the soft-max layer output $\widehat{\mathbf{Z}}_c$ of the encoder network. This loss along with the GAN mini-max objective and the latent variable recovery loss in $\mathbf{z}_n$ encourages clustering in

the latent space and also increases discriminative information. We minimize the cross-entropy between the predicted result and the ground truth as

$$\min \mathrm{CE}(G, E) = \frac{1}{m} \sum_{i=1}^{m} \left[p\left(\mathbf{z}_{c,i}^{k}\right) \log p\left(E\left(G\left(\mathbf{z}_{c,i}^{k}\right)\right)\right) \right] \tag{7}$$

where the first term is the empirical probability that the embedding belongs to the *k*-th speaker, and the second term is the predicted probability that the encoder produced embedding belongs to the *k*-th speaker.

Algorithm 1 ClusterGAN training. Default values: $\lambda = 10$, $m = 128$, $n_{\text{critic}} = 5$, $\alpha = 1\text{e}{-4}$, $\beta_1 = 0.5$, $\beta_2 = 0.9$

Require: λ: gradient penalty coefficient; α: learning rate; m: batch size; N_{it}: number of iterations; n_{critic}: number of critic iterations for each generator iteration; α, β_1, β_2: Adam hyper-parameters

for $it = 1$ to N_{it} **do**
  for $\tau = 1$ to n_{critic} **do**
    Sample $\{\mathbf{x}^{(i)}\}_{i=1}^{m}$, a batch of x-vectors
    Update the discriminator parameters by
    $$\theta \leftarrow \text{Adam}[\nabla_\theta \{\frac{1}{m} \sum_{i=1}^{m} w_1 \cdot [D_\theta(\mathbf{x}^{(i)}) - D_\theta(G_\phi(\mathbf{z}^{(i)})) + \lambda \cdot \text{GP}]\}, \theta, \alpha, \beta_1, \beta_2]$$
  end for
  Sample $\{\mathbf{z}^{(i)}\}_{i=1}^{m}$, a batch of latent vectors
  Update the generator and encoder parameters by
  $$\phi, \psi \leftarrow \text{Adam}[\nabla_{\phi,\psi} \{\frac{1}{m} \sum_{i=1}^{m} -w_1 \cdot D_\theta(G_\phi(\mathbf{z}^{(i)})) + w_2 \cdot \text{COS}(G_\phi, E_\psi) + w_3 \cdot \text{CE}(G_\phi, E_\psi)\}, \phi, \psi, \alpha, \beta_1, \beta_2]$$
end for

6) Joint training: The GAN and the encoder networks training in this approach involves joint parameter updates. The final training objective has the following form:

$$\min_{G,E} \max_{D} [w_1 \cdot U_{\text{IWGAN}}(D, G) + w_2 \cdot \text{COS}(G, E) + w_3 \cdot \text{CE}(G, E)] \tag{8}$$

Weights w_2 and w_3 represent relative significance of preserving continuous and discrete portions of the latent variable. Algorithm 1 lists the whole ClusterGAN training procedure.

D. MCGAN fine-tuning

Thus far we have discussed the training procedure of ClusterGAN, which is considered as a pre-training part of MCGAN training. In the second phase of MCGAN training, we discard the generator and discriminator, and fine-tune the pre-trained encoder with meta-learning based prototypical loss.

1) Motivation: The motivation behind fine-tuning the encoder with prototypical loss is that it has good generalization ability at test-time to new classes (unseen during training) given only a handful of examples of each new class [29]. Similar to this setting, in speaker diarization, a trained model for embedding extraction is asked to do clustering among unseen speakers within an audio stream. This is close to a metric learning task, where input

audio must be mapped to a discriminative embedding space. Furthermore, a speaker embedding such as x-vector is trained on a speaker classification loss, which is not explicitly designed to optimize embedding similarity. Metric learning related losses such as contrastive loss [62] and triplet loss [46] can resolve the above issues. Nonetheless, these methods require careful pair or triplet selection, which is sometimes time-consuming and performance-sensitive. In this context, prototypical loss trained for learning a metric space to mimic the test scenario might be handy in capturing information related to both generalization and clustering objectives.

2) Episode training: The encoder or the protonet in the MCGAN is trained episodically, where each episode is one mini-batch consisting of N_C categories randomly sampled from total K categories (here, speakers). Suppose the whole training set $\mathscr{D} = \left\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{N_{tr}}, y_{N_{tr}}\right\}$, where each $y_i \in \{1, \ldots, K\}$. Here, K is the total number of speakers in the training set. We iterate through each episode and in each episode, we randomly sample N_C speakers from total K speakers. For each chosen speaker, N_S number of random samples is selected as the support set and from the rest of the samples of that particular speaker, N_Q number of samples is selected as the query set without replacement. The supports are used to construct the class prototypes using Eq. (2) and the prototypical loss is computed with weight updates based on the query samples according to Eq. (3) of Section III-B. In Eq. (3), the choice of $d(.)$ can be arbitrary. However, it is shown in [29] that the squared Euclidean distance, which is a particular class of distance function known as Bregman divergence, is good for the clustering problem, and the training algorithm is equivalent to modeling the supports using Gaussian mixture density estimation. Therefore, we also use Euclidean distance as our distance function for proto-learning in the embedding space. The loss function for each mini-batch is the negative log probability for the true class via gradient descent. The prototypical loss within a mini-batch can be written as

$$L_{\mathrm{PTL}} = \sum_{\left\{\mathbf{x}_j, y_j \in Q\right\}} -\log p_{\boldsymbol{\psi}}\left(y = y_j \mid \mathbf{x}_j\right) \tag{9}$$

To increase robustness, instead of using a fixed total number of speakers, we randomly choose the total number of speakers within an episode. We fine-tune the pre-trained encoder by freezing its first two hidden layers and training it with prototypical loss for every episode. The MCGAN fine-tuning procedure is shown in the dashed blue rectangle in Fig. 2. The episodic training procedure is summarized in Algorithm 2.

E. MCGAN testing

After completion of offline training, only the trained encoder model in MCGAN is used to produce the proposed latent embeddings for the input x-vectors of a given test diarization session (shown in Fig. 1). The concatenated latent embeddings ($\mathbf{z}_n$ and $\mathbf{z}_c$) for ClusterGAN or logits for MCGAN are clustered using k-means or NME-SC, and speaker labels of each audio segment are obtained.

Algorithm 2 Meta-learning training for prototypical networks. N_{tr} = number of labeled examples in the training set, K = total number of speakers in the training set, $N_C \leq K$ is the number of speakers per episode, N_S = number of support examples per chosen speaker, N_Q = number of query examples per chosen speaker. $\mathcal{R}(S, N)$ denotes a set of N elements sampled uniformly at random from set S, without replacement.

Require: The whole training set $\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{D}_k$, where $\mathcal{D}_k$ represents the subset of $\mathcal{D}$ containing all elements such that $\{(\mathbf{x}_i, y_i); y_i = k\}$

$N_C \leftarrow \mathcal{R}(\{10, 20, \ldots, 150\}, 1)$ ▷ Randomly select total number of speakers in an episode

$V \leftarrow \mathcal{R}(\{1, \ldots, K\}, N_C)$ ▷ Randomly select speakers in an episode

for k in $\{1, \ldots, N_C\}$ **do**

  $S_k \leftarrow \mathcal{R}(\mathcal{D}_{V_k}, N_S)$ ▷ Supports

  $Q_k \leftarrow \mathcal{R}(\mathcal{D}_{V_k} \backslash S_k, N_Q)$ ▷ Queries

  $\mathbf{p}_k \leftarrow \frac{1}{N_S} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_\psi(\mathbf{x}_i)$ ▷ Prototypes

end for

$L_{\text{PTL}} \leftarrow 0$

for k in $\{1, \ldots, N_C\}$ **do**

  for $(\mathbf{x}_j, y_j)$ in Q_k **do**

$$L_{\text{PTL}} \leftarrow L_{\text{PTL}} + \frac{1}{N_C N_Q}[-d(f_\psi(\mathbf{x}_j), \mathbf{p}_{y_j}) + \log \sum_{k'} \exp(-d(f_\psi(\mathbf{x}_j), \mathbf{p}_{k'}))]$$ ▷ Loss update

  end for

end for

F. Normalized Maximum Eigengap Spectral Clustering (NME-SC)

We adopt NME-SC[2] as our spectral clustering method in this paper for speaker diarization evaluation in the unknown number of speakers condition. The NME-SC algorithm can auto-tune parameters of the clustering and also provides improved speaker diarization performance as compared to traditional spectral clustering approaches. As reported in [11], the steps to perform NME-SC are: (a) Construct affinity matrix (**A**) based on cosine similarity values between the segment embeddings. (b) Binarize **A** based on a p-value by converting the p-largest elements in each row of **A** to 1 and else to 0. (c) Perform symmetrization on the binarized affinity matrix $\mathbf{A}_p$ to obtain $\overline{\mathbf{A}}_p$ and compute the Laplacian matrix $\mathbf{L}_p$ as $\mathbf{L}_p = \mathbf{D}_p - \overline{\mathbf{A}}_p$, where $\mathbf{D}_p$ is a diagonal matrix and $\mathbf{D}_p = \sum_{j=1}^{n} \overline{\mathbf{A}}_{p,ij}$ (d) Perform eigen decomposition on $\mathbf{L}_p$ and create eigengap vector ($\mathbf{e}_p$). (e) Perform NME analysis to estimate the optimum value $\hat{p}$ and number of clusters k for a given session. (f) Select the k-smallest eigenvalues and the corresponding eigenvectors to construct a matrix $\mathbf{P} \in \mathbb{R}^{n \times k}$. (g) Cluster the row vectors of **P** using k-means algorithm. The details of the NME-SC algorithm is precisely described in [11].

G. Diarization algorithm

We can summarize the proposed diarization algorithm as follows:

[2] https://github.com/tango4j/Python-Speaker-Diarization

1. On the training data, employ oracle SAD to remove the non-speech segments and apply uniform speaker-homogeneous segmentation of speech of fixed size 1.5 sec with an overlap of 1 sec between two adjacent segments.
2. Extract x-vectors from each speech segment using Kaldi-based pre-trained model.
3. *MCGAN pre-training:* As described in Section IV-C, train the ClusterGAN until convergence with x-vectors as input to the GAN discriminator. For each mini-batch, sample z_n from a normal distribution and z_c using the original speaker label.
4. *MCGAN fine-tuning:* Discard the ClusterGAN G and D networks, and fine-tune the E-network using prototypical loss described in Section IV-D.
5. While testing a diarization session, extract speaker embeddings from the trained (after MCGAN pre-training and fine-tuning) encoder after uniform segmentation and x-vector extraction.
6. Clustering on the proposed embeddings using NME-SC algorithm described in Section IV-F.

V. Database description

We evaluate our proposed speaker diarization system on five distinct and diverse databases covering many possible data types and domains that are encountered in real-world scenarios.

A. CALLHOME database

CALLHOME contains telephonic conversations recorded at 8 kHz sampling frequency. In speaker diarization literature, the NIST 2000 speaker recognition evaluation challenge disk-8 is referred to as CALLHOME [63]. It is a multi-lingual database covering six languages: English, Spanish, Arabic, Mandarin, Japanese, and German. The database comprises 500 conversations with the number of speakers in each session varying from 2 to 7. The telephone recordings range from 1 to 10 minutes in duration and the distribution of the number of speakers is given in [4], [5].

B. AMI database

AMI is a publicly available meeting corpus of 171 recordings, totalling about 100 hours of data[3]. The meetings are recorded at four different sites (Edinburgh, IDIAP, TNO, and Brno). We use the multiple close-talk microphone data post beamforming for our experiments. For our evaluation, we follow the official speech recognition partition of AMI database with TNO meetings excluded from dev and eval set. The same split is also used in [17]. The train and dev splits have two speaker overlap, however, there is no speaker overlap between train-eval and dev-eval splits. The database partition details are shown in Table I.

[3] http://groups.inf.ed.ac.uk/ami/download/

C. DIHARD-II database

The DIHARD-II database is from the DIHARD challenge conducted in 2019. It is a multi-domain database focused on difficult speaker diarization settings. The database is comprised of diverse recordings collected from domains like meeting speech, restaurant recordings, child language acquisition recordings, YouTube videos, clinical recordings, etc [64], [65]. The DIHARD challenge features two audio input conditions: single-channel and multi-channel. We evaluated our system on single-channel data with reference SAD, which is track 1 in the challenge. Moreover, the database has two subsets: development and evaluation. In this work, speaker diarization performance of proposed and other baseline systems are compared only on the development part of the database. The development set contains 192 recordings, and typically are of short duration (< 10 min) sessions, and with the number of speakers in each session varying between 1 and 10.

D. ADOS and BOSCC databases

The proposed system is also tested on two child-clinician interaction corpora obtained in a clinical context involving a sample of children with autism spectrum disorder (ASD). ADOS (Autism Diagnosis Observation Schedule) is a diagnostic tool based on expert clinical administration and observation that produces a diagnostic algorithm score to inform clinical diagnosis of ASD [66]. ADOS comprises 14 play-based conversational tasks, from within which we select data from two sub tasks: *Emotion and Social Difficulties and Annoyance* from 272 sessions for our evaluation. Each of these dyadic sub-sessions is of duration < 10 min. BOSCC (Brief Observation of Social Communication Change) is a behavioral observation based autism treatment outcome measure that uses play-based conversational segments of dyadic interaction between a child and an adult (e.g., examiner or caregiver) [67]. In this work, the diarization performance is tested on 24 BOSCC sessions that were collected in a clinical setting. A BOSCC session typically lasts for 12 minutes. The ADOS and BOSCC data considered here are from verbal children and adolescents with autism.

VI. Experimental setup

A. Speech segmentation

In all the experiments, we have used uniform segmentation (as followed in Kaldi) on the speech intervals specified by the oracle SAD. All the experiments reported in this paper use oracle SAD, which is also a common practice in speaker diarization research [3], [8], [56]. Since our focus is on the effectiveness of proposed embeddings in speaker clustering, we use oracle SAD to eliminate the chance of introducing undesirable error initially due to potential performance uncertainty in automated system SAD. For all the experiments, a sliding window of 1.5 sec duration and overlap of 1 sec is employed to produce speaker-homogeneous segments. Note that in this work no re-segmentation module is applied in the final processing step.

B. x-vector extraction

We use x-vectors from the CALLHOME[4] and Voxceleb[5] recipe as pre-trained audio embeddings for the 8 kHz and 16 kHz data, respectively. The x-vector dimensions are of 128 and 512 for 8 kHz and 16 kHz audio data, respectively.

C. ClusterGAN model specifications

We train two different ClusterGAN models to evaluate diarization performance on the different databases. To test speaker diarization performance in CALLHOME which contains 8 kHz telephonic conversations, we train the ClusterGAN network in a supervised manner based on AMI-train (downsampled to 8 kHz) and switchboard (NIST SRE 2000, disk-6) data. This is our M1 model given in Table II. The other model M2, which we employ for diarization performance evaluation on all other databases (AMI, DIHARD-II dev, ADOS, BOSCC) containing 16 kHz data is trained on AMI-train and ICSI data (shown in Table II). We use 60 beamformed ICSI [68] sessions with a total number of 46 speakers. The architectures details of generator (G), discriminator (D) and encoder (E) networks in ClusterGAN are shown in Table III. Moreover, we set the learning rate to 1e-4 and adopt Adam optimization with a mini-batch size of 128 samples to optimize the three networks. We choose the weights w_1, w_2, and w_3 as 1, 10, and 10, respectively, by tuning the diarization error rate (DER) on a held-out set for the 8 kHz model and AMI dev set for the 16 kHz model. It is to be noted that all the above-mentioned model specifications are kept the same for all the experiments reported in this paper.

D. MCGAN specifications

We fine-tune the prototypical network, i.e., the pre-trained encoder in ClusterGAN using Euclidean distance based prototypical loss, which is found to be more effective than cosine distance in [29]. We use the same encoder for embedding extraction for both support and query points; while x-vectors from the training data form the support and queries. We fine-tune the pre-trained encoder by freezing its first two hidden layers and then train it with prototypical loss. We develop support and query set from the same training data (shown in Table II) that are used to train the ClusterGAN (for both M1 and M2 models). Instead of using the fixed number of classes to construct all the episodes, we randomly choose the number of classes from 10 to 150 with intervals of 10 per training episode and found this approach is slightly more effective. The number of shots to use in the support set is selected by tuning the DER on the AMI dev set. We fix the number of supports and queries to 10 for all the experiments.

E. Baseline systems

We compare our proposed embeddings with different back-end clustering techniques against several baselines and state-of-the-art diarization systems in five different databases. Since our proposed system incorporates x-vectors as input features, we use Kaldi-based x-vectors with PLDA scoring and AHC clustering as our main baseline system. Furthermore, we show

[4] https://kaldi-asr.org/models/m6
[5] https://kaldi-asr.org/models/m7

results for x-vector embedding and k-means or spectral clustering (SC) as back-ends, and these are other baseline systems. For a fair comparison, we also report the results of our proposed embeddings with k-means and SC back-ends. It is to be noted that a few additional and in-domain (AMI) data was incorporated for MCGAN pre-training/fine-tuning (shown in Table II), and it was not used for the x-vector model training. This sort of practice of using additional data for model training has been used before in past speaker diarization work [8], [69], [70]. Moreover, this additional data is approximately 6% of the whole Voxceleb and augmented data that were used to train the Kaldi x-vector model. Therefore, we believe the proposed embeddings results for all the experiments are meaningful and comparable to the original TDNN x-vector embeddings. We also perform embedding fusion with x-vectors with k-means and SC back-end clustering. Note that for the oracle number of speakers we used fixed tuned p-value binarized SC [11], whereas for the estimated number of speakers we adopt NME-SC [11] for all the experiments in this paper. However, in the rest of the paper, we will refer to all systems adopting spectral clustering as SC.

F. Performance metrics

We evaluate the proposed speaker diarization system with NIST diarization error rate (DER) [71]. Following the approach described in [71], we use a collar of 0.25 sec for all the databases DER evaluation, except DIHARD-II, where zero collar is used according to the challenge criteria [64]. We ignore speaker overlap regions during scoring since neither x-vectors nor our proposed embeddings are trained to handle overlapping speech.

VII. Experimental results

A. Results and analysis on telephonic dataset

1) Importance of adversarial training and prototypical learning: We evaluate the importance of adversarial training in ClusterGAN and prototypical learning in our proposed speaker diarization system. To do so, we extract the embeddings from the following setup: (a) train a single encoder network with random initialization based on cross-entropy loss (E_{cross}) only using x-vectors from the training data as input, (b) train the single encoder based on the prototypical loss only (E_{proto1}) using x-vectors of the training data as input, (c) pre-train the single encoder with cross-entropy loss and re-training (training the entire network again after pre-training) it with prototypical loss (E_{proto2}), and (d) pre-training the encoder with cross-entropy loss and fine-tuning (training by freezing the first two hidden layers) it with prototypical loss (E_{proto3}). The ClusterGAN and MCGAN embeddings are extracted from the model M1 (shown in Table II). The results are summarized in Table IV for both known and estimated number of speakers with both k-means and SC back-end on the CALLHOME database.

By comparing the results of E_{cross} and ClusterGAN in Table IV, we can comment on the importance of adversarial training in our proposed diarization system setup. It is observed that ClusterGAN outperforms E_{cross} for both k-means and SC with known and estimated number of speakers cases. Therefore, it is important to perform adversarial training of GANs in our ClusterGAN. It is seen from the table that E_{proto1}, which is trained only on prototypical loss significantly reduces the DER values over E_{cross}. This indicates the

significance of prototypical loss based training. We can corroborate this fact by comparing the DERs of ClusterGAN with MCGAN, which is fine-tuning the ClusterGAN using prototypical loss. MCGAN yields significant performance improvement over ClusterGAN with both the clustering back-ends. Since for MCGAN adversarial training (pre-training) we used the speaker labels and cross-entropy loss, it is directly comparable to E_{proto3} which is based on pre-training the encoder only with cross-entropy loss and fine-tuning it with prototypical loss. The results in Table IV indicate that MCGAN is superior to E_{proto3} for both k-means and SC. Furthermore, MCGAN outperforms both E_{proto1} and E_{proto2} embeddings for both the back-ends. The performance of E_{proto1} and E_{proto2} embeddings are almost similar, and therefore it shows that cross-entropy pre-training for a single encoder doesn't improve performance with the prototypical loss for our diarization framework. Finally, MCGAN outperforms both E_{proto2} and E_{proto3} significantly, therefore it is clear that this improvement is due to the adversarial training setup in ClusterGAN training.

2) Ablation study: In this section, we report ablation experiments performed to examine the contribution of each component of our proposed system and demonstrate the feasibility of our framework. We compute DER for different embeddings including ClusterGAN, MCGAN, x-vector, x-vector + MCGAN with both k-means and spectral clustering (SC) as back-end clustering. Fig. 3 shows the difference in DER values between each embedding and our final proposed embedding (x-vector + MCGAN) with the two aforementioned clustering techniques on CALLHOME. The mean difference for all the sessions is shown between each scenario and the final proposed setting.

It can be observed from the figure that all the subcomponents contribute to improving DER performance. The effect of the various components on diarization performance on the CALLHOME dataset in increasing order is: ClusterGAN, x-vector, and MCGAN, for both k-means and SC back-ends. Moreover, the figure shows fine tuning ClusterGAN with prototypical loss (MCGAN) is important for achieving improved DER. It also demonstrates that prototypical training and embedding fusion are the key components to obtaining the best results on CALLHOME.

3) Number of predicted speakers: In addition to DER, the mean absolute percentage deviation (MAPD) of the predicted number of speakers and percentage of the correct number of speaker estimation (POC) across all the sessions are also useful metrics in the context of estimating the number of speakers in speaker diarization. The lower the MAPD, and higher the POC, better the speaker estimation performance. The results on CALLHOME are summarized in Table V. It is evident from the table that MCGAN embeddings are more robust and accurate in estimating the number of speakers than ClusterGAN embeddings and x-vectors. The performance of fused embeddings is slightly worse than MCGAN. Therefore, it is expected that MCGAN will perform better than ClusterGAN and x-vector in the estimated number of speakers condition.

4) Overall performance evaluation: In this section, we present the experimental results on the whole CALLHOME evaluation set by using the tuned parameters of the different versions of our proposed diarization system. We compare the proposed system with other baselines and recent state-of-the-art diarization methods. The experimental results for

both known and unknown numbers of speakers are reported in Table VI. Note that for known or oracle number of speakers we use a fixed p-value which is tuned on Kaldi CALLHOME-1 held-out set and apply it to CALLHOME-2 and vice versa.

From Table VI column 3, we observe that for known number of speakers, the ClusterGAN embedding does not outperform x-vectors for both k-means and SC. However, we see that MCGAN embeddings which are extracted after fine-tuning the pre-trained encoder with prototypical loss provide superior performance over x-vectors for both k-means and SC back-ends. MCGAN reduces average DER of ClusterGAN from 10.24% to 8.72% and from 7.62% to 6.01% for k-means and SC, respectively. Therefore, fine-tuning the protonet (E) with meta-learning related prototypical loss is useful for better generalization. We obtain further improvement in DER by incorporating embedding fusion between x-vector and MCGAN embeddings. We achieve the best DER of 5.73% for the known number of speakers and SC back-end, which is significantly better than the Kaldi x-vector state-of-the-art (average DER 7.12%) and also superior to the x-vector with SC (average DER 6.23%). The relative improvement of our final proposed system over Kaldi state-of-the-art is 19.52% for known number of speakers.

We show the diarization performance of all the systems for the estimated number of speakers in Table VI column 4. The number of speakers for k-means and SC is estimated using NME-SC. From Table VI column 4, we see a similar trend in performance for estimated number of speakers. The biggest improvement in DER for the proposed embeddings comes from MCGAN, embedding fusion, and most importantly with SC. For the same back-end setting, the proposed MCGAN and fused embeddings significantly outperform both ClusterGAN and x-vectors. It is important to note that surprisingly in many of the settings (except SC) we obtain reduced DER for the automatically estimated number of speakers case than for oracle number of speakers. This could be attributed to the fact that even though the number of clusters may be correct for the oracle case there might be inherent speaker confusions, whereas, for the estimated number of speakers, the clusters based on data-driven estimation may be purer even if the estimated number of clusters is not exactly correct. The embedding fusion between x-vector and MCGAN with SC back-end yields the best DER value of 6.76% for the estimated number of speakers with a relative improvement of 19.43% over the Kaldi x-vector system. The next best system–MCGAN with SC–produces a DER of 7.03%, which is also significantly better than the Kaldi x-vector and x-vector with SC back-end. We also present the recent best system's results that are reported in the literature on the CALLHOME evaluation set. Many of these systems use cross-validation to train or adapt their systems. However, without using any cross-validation, the proposed system outperforms all the recent diarization systems on CALLHOME.

5) Analysis of Experimental Results: We first break down the average DER on CALLHOME database according to the number of speakers. The corresponding DERs are plotted in a group bar plot in Fig. 4 for x-vector, MCGAN and fused (x-vector + MCGAN) embeddings with NME-SC back-end and estimated number of speakers. It is evident from the figure that our proposed MCGAN and fused embeddings achieve significantly better DER values than x-vector for two and three speaker cases. For four and five speakers, x-vector is better than MCGAN. However, the fused embeddings provide better performance

than x-vectors for most of the speaker conditions (two, three, four, and six) and this covers majority of the conversations in the dataset. In the seven-speaker condition, the fused system is not able to outperform x-vector, which is possibly an anomaly since the number of sessions containing seven speakers is only two in this database. The reason behind obtaining better results using fused embeddings with SC is attributed to the complementary merits of the x-vector and MCGAN embeddings, and the modeling power of the NME-SC algorithm on embeddings. We are speculating that the source of this complementary nature is due to the ClusterGAN encoder training from the generator output and not the exact x-vectors.

We extend the analysis by checking the effectiveness of our proposed system in a more challenging practical scenario namely diarization in short speech segment cases. Shorter segments usually provide low-quality speaker embeddings. To carry out this analysis, we chose conversations from the CALLHOME evaluation set that have a majority number of short duration (<= 2 sec and <= 2.5 sec) speech segments. Here, we select sessions that have more than 80% of short speech segments in the entire session. We find that number of such sessions is 58 (<= 2 sec) and 129 (<= 2.5 sec), respectively. We compute and plotted the mean DER of the selected sessions in Fig. 5 for x-vector, ClusterGAN, MCGAN, and fused (x-vector + MCGAN) embeddings with k-means and spectral clustering and estimated number of speakers. It is clear from the figure that among the four embeddings, MCGAN embeddings produce the lowest average DER for short speech segment sessions compared to x-vector, ClusterGAN, and fused embeddings, and for both the clustering techniques. The fused and ClusterGAN embeddings yield better performance than x-vector for most of the cases. We obtain worse DER values for <= 2 sec segments than <= 2.5 segments, which is not surprising. Finally, we can conclude that MCGAN embedding is more robust than the other embeddings in short speech segment scenarios.

B. Results and analysis on wide-band dataset

1) Performance comparison: Herein we evaluate and compare different versions of proposed embeddings against x-vector with different clustering techniques, and existing Kaldi state-of-the-art speaker diarization system. For better clarity, we report the average DER values across four different popular databases (AMI meeting corpus, DIHARD-II dev multi-domain database, and child-clinician interaction corpora: ADOS and BOSCC) for oracle SAD and estimated number of speakers only in Table VII. The number of speakers for both k-means and SC back-end clustering is estimated by using the NME-SC algorithm. For the Kaldi x-vector baseline, the number of speakers in a session is estimated based on optimized threshold on the PLDA scores [3]. It is to be noted that we use the model M2 (shown in Table II) that is trained on AMI-train and ICSI data to generate our proposed embeddings (ClusterGAN and MCGAN).

We show results for all the systems on AMI dev and eval sets for the estimated number of speakers in Table VII columns 3 and 4. It is clear from the table that for both AMI dev and eval sets, all the proposed embeddings (ClusterGAN, MCGAN, x-vector + ClusterGAN, x-vector + MCGAN) are superior to x-vector (except ClusterGAN in AMI eval set for SC back-end). MCGAN yields better performance compared to ClusterGAN for the k-means back-end. Moreover, as expected, the fused embeddings further improve diarization

performance. On the other hand, spectral clustering boosts diarization performance further for all the embeddings. This highlights the effectiveness of NME-SC over k-means. Besides, the embedding fusion provides further reduction in DER for SC back-end. Finally, using our proposed fused system, we achieve significantly better performance on AMI dev and eval set with absolute DER of 5.02% and 2.87%, respectively, outperforming Kaldi x-vector baseline diarization system by a significant margin.

To investigate the effectiveness of the proposed embeddings for speaker diarization in challenging multi-domain settings, we evaluated and report the average DER values on the DIHARD-II development database in Table VII column 5. This is to check the robustness of our embeddings in real-world noisy scenarios without training explicitly using separate noisy data or data augmentation. It is seen from the table that MCGAN is superior to ClusterGAN in the estimated number of speakers scenario. However, individually x-vector is better than ClusterGAN and MCGAN on this database. Nonetheless, both the fused embeddings outperform x-vector for k-means back-end. With SC back-end, we achieve significant improvement in performance for all the embeddings. We attain the best DER value of 17.75% by using x-vector + ClusterGAN embedding and SC back-end. Furthermore, the proposed fused systems are significantly better than the Kaldi x-vector diarization system, which was the baseline in the challenge. Thus, the proposed embeddings although extracted from the model trained on AMI train and ICSI data, are promising in terms of generalization, have complementary information to x-vectors, and can yield improved performance on a challenging multi-domain database in an embedding fusion set up with the spectral clustering back-end.

Finally, we evaluate the proposed method on two child-clinician interaction corpora from the domain of Autism Spectrum Disorder: ADOS and BOSCC. The diarization results are presented in Table VII column 6 and 7. We observe from the table that the Kaldi x-vector diarization system does not perform well on these two databases. The most probable reason behind this is that the PLDA model is trained on Voxceleb data and thus creating a significant domain mismatch. However, the x-vectors with k-means and SC perform reasonably well on both ADOS and BOSCC data than the Kaldi x-vector system. Among the proposed embeddings, ClusterGAN is superior to MCGAN both individually and also when fused with x-vectors. This is attributed to the better performance of ClusterGAN over MCGAN in the known number of speakers condition in general. A significant reduction in DER is seen while SC is employed as the clustering mechanism. The best achieved DER on ADOS and BOSCC datasets is 6.74% and 9.26%, respectively, and this is for the x-vector + ClusterGAN with SC system. We obtain a relative improvement of 53.06% and 57.31% over Kaldi x-vector on the ADOS and BOSCC databases, respectively. We note that although we expect better generalization from MCGAN due to meta-learning, ClusterGAN emerges as useful in these known number of (dyadic) speaker conditions, i.e., child and adult interlocutors.

2) DER analysis according to the domains in DIHARD-II: To understand how our proposed embeddings with spectral clustering behave in each specific domain of DIHARD-II dev set, we split the DER according to the context of the database. The results shown in Table VIII indicate high variability in performance across the domains. The proposed

embeddings (ClusterGAN and MCGAN) individually are not able to outperform x-vectors except on court and audiobooks data. However, the fused embeddings offer promising performance on most of the domains compared to the x-vectors. The worst performing domains for our embeddings are restaurant, webvideo, and child. The metadata analysis of DIHARD-II dev set in [70] shows that restaurant sessions are highly noisy and also contain a large number of speakers. On the other hand, the observed worse performance on child data sessions is because the children are 6–18 months old, and have high variability in their vocalizations; moreover, more than two speakers are often present in those sessions. It is intriguing to note that our embeddings perform well in the meeting domain. This is possible because the proposed embeddings were trained on meeting data. However, x-vectors that are extracted from Voxceleb trained model also perform well. The other domains on this dataset where we obtain noticeable improvements over the x-vector system include audiobooks, clinical, child, and socio_field.

VIII. Conclusions

We proposed new speaker embeddings by exploiting the latent space of GANs using ClusterGAN and by making the encoder in the ClusterGAN more robust and generalizable with the help of prototypical loss fine-tuning. We benchmarked the proposed embeddings individually and also fused with x-vectors within the speaker diarization framework. We investigated the effectiveness of the proposed embeddings by extensively evaluating them for speaker diarization across five different databases. We obtain a relative DER improvement of 19.43%, 71.47%, 19.77%, 53.06%, and 57.31% over the Kaldi x-vector baseline on CALLHOME, AMI-eval, DIHARD-II dev, ADOS, and BOSCC databases respectively. The key findings of this work can be summarized as follows:

- MCGAN embeddings outperform x-vectors and ClusterGAN embeddings significantly on telephonic data for both known and automatically estimated number of speaker conditions with both k-means and SC back-ends. They also perform better than ClusterGAN in the estimated number of speaker condition on meeting and multi-domain datasets.
- Analysis suggests that MCGAN embeddings are robust in the number of speakers estimation task and in diarizing sessions which have significant presence of short speech segments when compared to x-vectors, ClusterGAN and fused embeddings.
- Embedding fusion of x-vectors and the proposed embeddings improves diarization performance consistently for all the corpora considered. Therefore, we speculate that both the proposed embeddings have complementary information to the x-vectors. The proposed fused embeddings with NME-SC outperform the Kaldi x-vector system across all the wide-band datasets.

In the future, it would be worthwhile to investigate speech spectrograms directly instead of pre-trained embeddings as the input. The usage of other existing meta-learning algorithms can also be explored in the context of speaker diarization.

References

[1]. Anguera X, Bozonnet S, Evans N, Fredouille C, Friedland G, and Vinyals O, “Speaker diarization: A review of recent research,” IEEE Trans. Audio, Speech, Lang. Process, vol. 20, no. 2, pp. 356–370, 2012.

[2]. Vijayasenan D, Valente F, and Bourlard H, “An information theoretic approach to speaker diarization of meeting data,” IEEE Trans. Audio, Speech, and Lang. Process, vol. 17, no. 7, pp. 1382–1393, 2009.

[3]. Garcia-Romero D, Snyder D, Sell G, Povey D, and McCree A, “Speaker diarization using deep neural network embeddings,” in Proc. ICASSP, 2017, pp. 4930–4934.

[4]. Shum SH, Dehak N, Dehak R, and Glass JR, “Unsupervised methods for speaker diarization: An integrated and iterative approach,” IEEE Trans. Audio, Speech, Lang. Process, vol. 21, no. 10, pp. 2015–2028, 2013.

[5]. Senoussaoui M, Kenny P, Stafylakis T, and Dumouchel P, “A study of the cosine distance-based mean shift for telephone speech diarization,” IEEE/ACM Trans. Audio, Speech, and Lang. Process, vol. 22, no. 1, pp. 217–227, 2014.

[6]. Heigold G, Moreno I, Bengio S, and Shazeer N, “End-to-end text-dependent speaker verification,” in Proc. ICASSP IEEE, 2016, pp. 5115–5119.

[7]. Wang Q, Downey C, Wan L, Mansfield PA, and Moreno IL, “Speaker diarization with LSTM,” in Proc. ICASSP, 2018, pp. 5239–5243.

[8]. Zhang A, Wang Q, Zhu Z, Paisley J, and Wang C, “Fully supervised speaker diarization,” in Proc. ICASSP, 2019, pp. 6301–6305.

[9]. Snyder D, Ghahremani P, Povey D, Garcia-Romero D, Carmiel Y, and Khudanpur S, “Deep neural network-based speaker embeddings for end-to-end speaker verification,” in 2016 IEEE Spoken Language Technology Workshop (SLT). IEEE, 2016, pp. 165–170.

[10]. Sell G, Snyder D, McCree A, Garcia-Romero D, Villalba J, Maciejewski M, Manohar V, Dehak N, Povey D, Watanabe S et al., “Diarization is hard: Some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge,” in Proc. Interspeech, 2018, pp. 2808–2812.

[11]. Park TJ, Han KJ, Kumar M, and Narayanan S, “Auto-tuning spectral clustering for speaker diarization using normalized maximum eigengap,” IEEE Sign. Process. Lett, 2019.

[12]. Lin Q, Yin R, Li M, Bredin H, and Barras C, “LSTM based similarity measurement with spectral clustering for speaker diarization,” arXiv preprint arXiv:1907.10393, 2019.

[13]. Dehak N, Kenny PJ, Dehak R, Dumouchel P, and Ouellet P, “Front-end factor analysis for speaker verification,” IEEE Trans. Audio, Speech, Lang. Process, vol. 19, no. 4, pp. 788–798, 2010.

[14]. Wan L, Wang Q, Papir A, and Moreno IL, “Generalized end-to-end loss for speaker verification,” in Proc. ICASSP, 2018, pp. 4879–4883.

[15]. Zajíc Z, Hrúz M, and Müller L, “Speaker diarization using convolutional neural network for statistics accumulation refinement,” in Proc. Interspeech, 2017, pp. 3562–3566.

[16]. Snyder D, Garcia-Romero D, Sell G, Povey D, and Khudanpur S, “X-vectors: Robust DNN embeddings for speaker recognition,” in Proc. ICASSP, 2018, pp. 5329–5333.

[17]. Sun G, Zhang C, and Woodland PC, “Speaker diarisation using 2D self-attentive combination of embeddings,” in Proc. ICASSP, 2019, pp. 5801–5805.

[18]. Ning H, Liu M, Tang H, and Huang TS, “A spectral clustering approach to speaker diarization,” in Proc. ICSLP, 2006.

[19]. Park TJ, Kumar M, Flemotomos N, Pal M, Peri R, Lahiri R, Georgiou PG, and Narayanan S, “The Second DIHARD challenge: System Description for USC-SAIL Team,” in Proc. Interspeech, 2019, pp. 998–1002.

[20]. Dimitriadis D, “Enhancements for audio-only diarization systems,” arXiv preprint arXiv:1909.00082, 2019.

[21]. Mansfield PA, Wang Q, Downey C, Wan L, and Moreno IL, “Links: A high-dimensional online clustering method,” arXiv preprint arXiv:1801.10123, 2018.

[22]. Han KJ, Kim S, and Narayanan SS, “Strategies to improve the robustness of agglomerative hierarchical clustering under data source variation for speaker diarization,” IEEE Trans. Audio, Speech, Lang. Process, vol. 16, no. 8, pp. 1590–1601, 2008.

[23]. Fini E and Brutti A, “Supervised online diarization with sample mean loss for multi-domain data,” in Proc. ICASSP, 2020, pp. 7134–7138.

[24]. Fujita Y, Kanda N, Horiguchi S, Nagamatsu K, and Watanabe S, “End-to-end neural speaker diarization with permutation-free objectives,” in Proc. Interspeech, 2019, pp. 4300–4304.

[25]. Fujita Y, Kanda N, Horiguchi S, Xue Y, Nagamatsu K, and Watanabe S, “End-to-end neural speaker diarization with self-attention,” arXiv preprint arXiv:1909.06247, 2019.

[26]. Xie J, Girshick R, and Farhadi A, “Unsupervised deep embedding for clustering analysis,” in Proc. ICML, 2016, pp. 478–487.

[27]. Pal M, Kumar M, Peri R, Park TJ, Kim SH, Lord C, Bishop S, and Narayanan S, “Speaker diarization using latent space clustering in generative adversarial network,” in Proc. ICASSP, 2020, pp. 6504–6508.

[28]. Mukherjee S, Asnani H, Lin E, and Kannan S, “ClusterGAN: Latent space clustering in generative adversarial networks,” in Proc. AAAI, vol. 33, 2019, pp. 4610–4617.

[29]. Snell J, Swersky K, and Zemel R, “Prototypical networks for few-shot learning,” in Proc. NIPS, 2017, pp. 4077–4087.

[30]. Wang J, Wang K-C, Law MT, Rudzicz F, and Brudno M, “Centroid-based deep metric learning for speaker recognition,” in Proc. ICASSP IEEE, 2019, pp. 3652–3656.

[31]. Aljalbout E, Golkov V, Siddiqui Y, Strobel M, and Cremers D, “Clustering with deep learning: Taxonomy and new methods,” arXiv preprint arXiv:1801.07648, 2018.

[32]. Jiang Z, Zheng Y, Tan H, Tang B, and Zhou H, “Variational deep embedding: An unsupervised and generative approach to clustering,” arXiv preprint arXiv:1611.05148, 2016.

[33]. Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I, and Abbeel P, “InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets,” in Proc. NIPS, 2016, pp. 2172–2180.

[34]. Yu Y and Zhou W-J, “Mixture of GANs for Clustering.” in Proc. IJCAI, 2018, pp. 3047–3053.

[35]. Pal M, Kumar M, Peri R, and Narayanan S, “A study of semi-supervised speaker diarization system using gan mixture model,” arXiv preprint arXiv:1910.11416, 2019.

[36]. Yang B, Fu X, Sidiropoulos ND, and Hong M, “Towards k-means-friendly spaces: Simultaneous deep learning and clustering,” in Proc. ICML JMLR. org, 2017, pp. 3861–3870.

[37]. Yang J, Parikh D, and Batra D, “Joint unsupervised learning of deep representations and image clusters,” in Proc. CVPR, 2016, pp. 5147–5156.

[38]. Shaham U, Stanton K, Li H, Nadler B, Basri R, and Kluger Y, “Spectralnet: Spectral clustering using deep neural networks,” arXiv preprint arXiv:1801.01587, 2018.

[39]. Ghasedi K, Wang X, Deng C, and Huang H, “Balanced self-paced learning for generative adversarial clustering network,” in Proc. CVPR, 2019, pp. 4391–4400.

[40]. Li Q, Kreyssig FL, Zhang C, and Woodland PC, “Discriminative neural clustering for speaker diarisation,” arXiv preprint arXiv:1910.09703, 2019.

[41]. Ravi S and Larochelle H, “Optimization as a model for few-shot learning,” 2016.

[42]. Chen K and Salman A, “Learning speaker-specific characteristics with a deep neural architecture,” IEEE Trans. on Neural Networks, vol. 22, no. 11, pp. 1744–1756, 2011. [PubMed: 21954206]

[43]. Santoro A, Bartunov S, Botvinick M, Wierstra D, and Lillicrap T, “Meta-learning with memory-augmented neural networks,” in Proc. ICML, 2016, pp. 1842–1850.

[44]. Vinyals O, Blundell C, Lillicrap T, Wierstra D et al., “Matching networks for one shot learning,” in Proc. NIPS, 2016, pp. 3630–3638.

[45]. Koch G, Zemel R, and Salakhutdinov R, “Siamese neural networks for one-shot image recognition,” in ICML deep learning workshop, vol. 2. Lille, 2015.

[46]. Schroff F, Kalenichenko D, and Philbin J, “Facenet: A unified embedding for face recognition and clustering,” in Proc. CVPR, 2015, pp. 815–823.

[47]. Jati A, Peri R, Pal M, Park TJ, Kumar N, Travadi R, Georgiou PG, and Narayanan S, “Multi-task discriminative training of hybrid DNN-TVM model for speaker verification with noisy and far-field speech.” in Proc. Interspeech, 2019, pp. 2463–2467.

[48]. Chung JS, Huh J, Mun S, Lee M, Heo HS, Choe S, Ham C, Jung S, Lee B-J, and Han I, “In defence of metric learning for speaker recognition,” arXiv preprint arXiv:2003.11982, 2020.

[49]. Bredin H, “Tristounet: triplet loss for speaker turn embedding,” in Proc. ICASSP, 2017, pp. 5430–5434.

[50]. Kye SM, Jung Y, Lee HB, Hwang SJ, and Kim H, “Meta-learning for short utterance speaker recognition with imbalance length pairs,” arXiv preprint arXiv:2004.02863, 2020.

[51]. Koluguri NR, Kumar M, Kim SH, Lord C, and Narayanan S, “Meta-learning for robust child-adult classification from speech,” in Proc. ICASSP, 2020, pp. 8094–8098.

[52]. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, and Bengio Y, “Generative adversarial nets,” in Proc. NIPS, 2014, pp. 2672–2680.

[53]. Arjovsky M, Chintala S, and Bottou L, “Wasserstein GAN,” arXiv preprint arXiv:1701.07875, 2017.

[54]. Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, and Courville AC, “Improved training of Wasserstein GANs,” in Proc. NIPS, 2017, pp. 5767–5777.

[55]. Yu M, Guo X, Yi J, Chang S, Potdar S, Cheng Y, Tesauro G, Wang H, and Zhou B, “Diverse few-shot text classification with multiple metrics,” arXiv preprint arXiv:1805.07513, 2018.

[56]. Diez M, Burget L, Landini F, and Černockyỳ J, “Analysis of speaker diarization based on Bayesian HMM with eigenvoice priors,” IEEE/ACM Trans. Audio, Speech, and Lang. Process, vol. 28, pp. 355–368, 2019.

[57]. Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, and Chen X, “Improved techniques for training gans,” in Advances in neural information processing systems, 2016, pp. 2234–2242.

[58]. Wang X, Ghasedi Dizaji K, and Huang H, “Conditional generative adversarial network for gene expression inference,” Bioinformatics, vol. 34, no. 17, pp. i603–i611, 2018. [PubMed: 30423066]

[59]. Li C, Xu T, Zhu J, and Zhang B, “Triple generative adversarial nets,” Advances in neural information processing systems, vol. 30, pp. 4088–4098, 2017.

[60]. Lipton ZC and Tripathi S, “Precise recovery of latent vectors from generative adversarial networks,” arXiv preprint arXiv:1702.04782, 2017.

[61]. Creswell A and Bharath AA, “Inverting the generator of a generative adversarial network,” IEEE Trans. on neural networks and learning systems, 2018.

[62]. Chopra S, Hadsell R, and LeCun Y, “Learning a similarity metric discriminatively, with application to face verification,” in Proc. CVPR, vol. 1. IEEE, 2005, pp. 539–546.

[63]. Martin AF and Przybocki MA, “Speaker recognition in a multi-speaker environment,” in Proc. Speech Commun. and Tech, 2001.

[64]. Ryant N, Church K, Cieri C, Cristia A, Du J, Ganapathy S, and Liberman M, “Second DIHARD challenge evaluation plan,” Linguistic Data Consortium, Tech. Rep, 2019.

[65]. Sahidullah M, Patino J, Cornell S, Yin R, Sivasankaran S, Bredin H, Korshunov P, Brutti A, Serizel R, Vincent E et al., “The speed submission to DIHARD II: Contributions & lessons learned,” arXiv preprint arXiv:1911.02388, 2019.

[66]. Lord C, Risi S, Lambrecht L, Cook EH, Leventhal BL, DiLavore PC, Pickles A, and Rutter M, “The autism diagnostic observation schedule—generic: A standard measure of social and communication deficits associated with the spectrum of autism,” Journal of autism and developmental disorders, vol. 30, no. 3, pp. 205–223, 2000. [PubMed: 11055457]

[67]. Grzadzinski R, Carr T, Colombi C, McGuire K, Dufek S, Pickles A, and Lord C, “Measuring changes in social communication behaviors: preliminary development of the brief observation of social communication change (BOSCC),” Journal of autism and developmental disorders, vol. 46, no. 7, pp. 2464–2479, 2016. [PubMed: 27062034]

[68]. Janin A, Baron D, Edwards J, Ellis D, Gelbart D, Morgan N, Peskin B, Pfau T, Shriberg E, Stolcke A et al., “The ICSI meeting corpus,” in Proc. ICASSP, vol. 1, 2003, pp. I–I.

[69]. Novoselov S, Gusev A, Ivanov A, Pekhovsky T, Shulipa A, Avdeeva A, Gorlanov A, and Kozlov A, “Speaker diarization with deep speaker embeddings for DIHARD challenge II.” in Proc. Interspeech, 2019, pp. 1003–1007.

[70]. Lin Q, Cai W, Yang L, Wang J, Zhang J, and Li M, “DIHARD II is Still Hard: Experimental Results and Discussions from the DKU-LENOVO Team,” arXiv preprint arXiv:2002.12761, 2020.

[71]. Fiscus JG, Jerome A, Martial M, and Garofolo JS, “The Rich Transcription 2006 spring meeting recognition evaluation,” in International Workshop on Machine Learning for Multimodal Interaction. Springer, 2006, pp. 309–322.

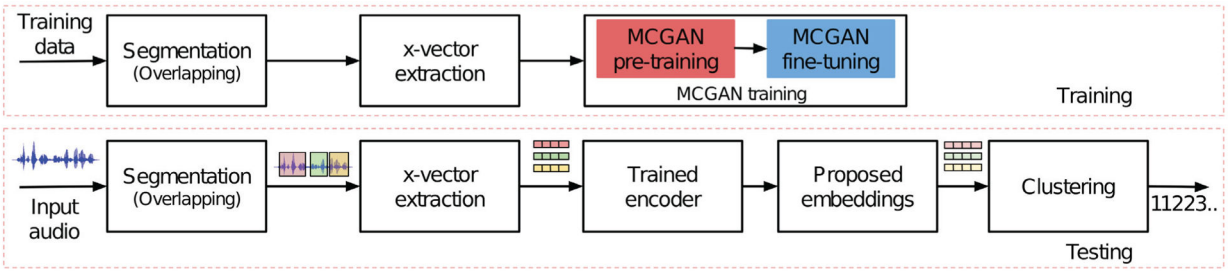

Fig. 1.
Skeleton of the proposed speaker diarization system.

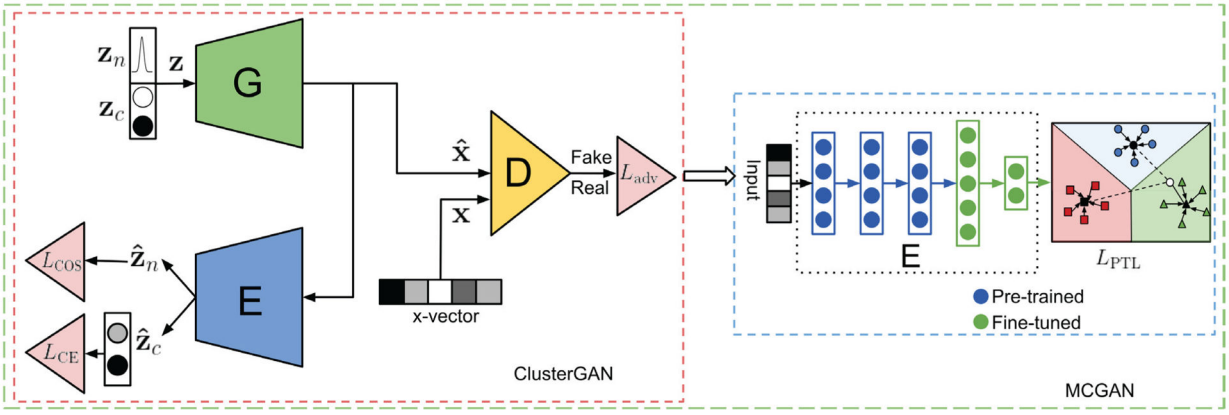

Fig. 2.

MCGAN system (shown in a green dashed rectangle). The ClusterGAN architecture shown in the red dashed rectangle is used for MCGAN pre-training. MCGAN fine-tuning part is shown in a blue dashed rectangle. Here, L_{adv}, L_{COS}, L_{CE} and L_{PTL} represent adversarial, cosine distance, cross-entropy and prototypical loss functions.

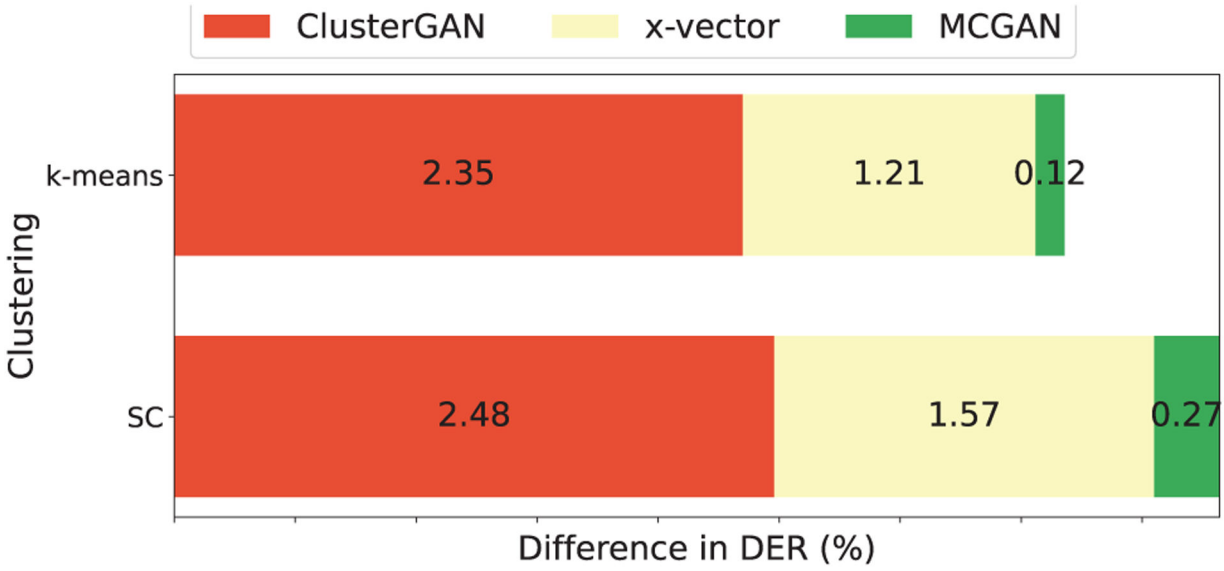

Fig. 3.

Difference in DER (%) in CALLHOME database between the final proposed system (fused) and the system trained only with specific components.

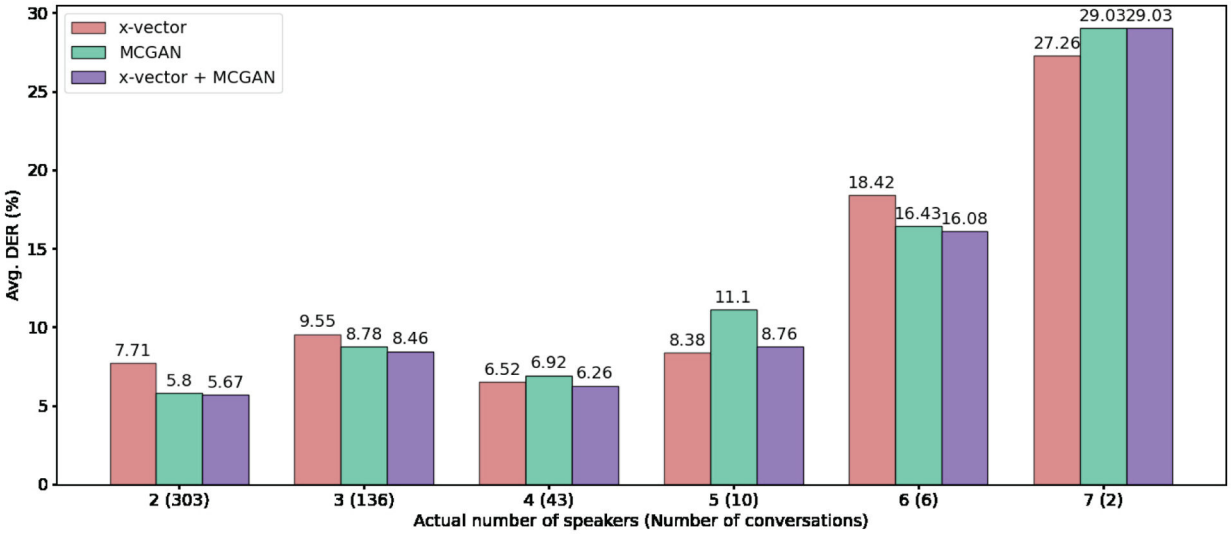

Fig. 4.

Avg. DER (%) analysis with respect to number of speakers in a diarization session.

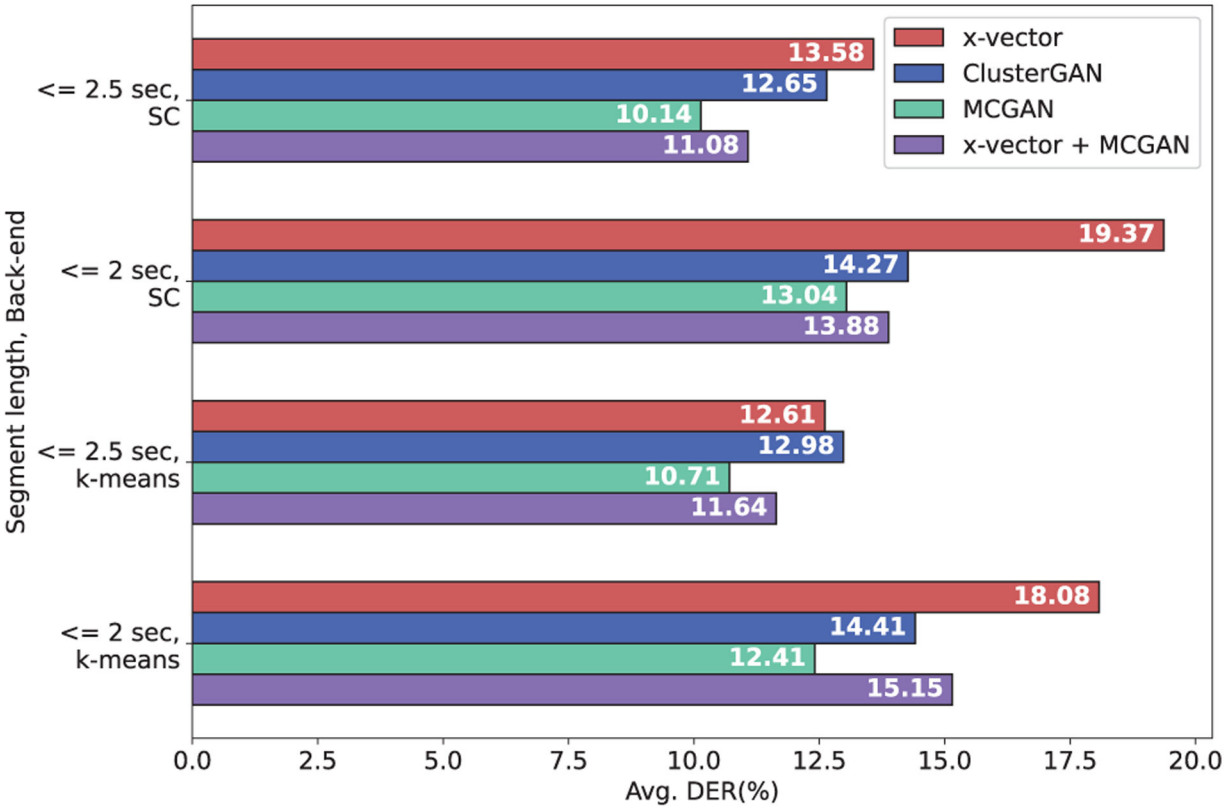

Fig. 5.
Average DER (%) analysis on short speech segment diarization sessions.

TABLE I

Details of the AMI data set used for our experiments.

	#Meetings	#Speakers
Train	136	155
Dev	14	17
Eval	12	12

TABLE II

Details of the data set used for our model pre-training/fine-tuning.

Model	MCGAN pre-training/fine-tuning on	Tested on
M1	Switchboard + AMI-train (downsampled)	CALLHOME
M2	AMI-train + ICSI	AMI-dev, AMI-eval, DIHARD-II dev, ADOS, BOSCC

TABLE III

ClusterGAN architecture details.

Generator (*G*)	Discriminator (*D*)	Encoder (*E*)
Input: Linear, $\mathbf{z} = (\mathbf{z}_n, \mathbf{z}_c) \in \mathbb{R}^{d_\mathbf{z}}$, $d_n^* = 90$, $d_c^\dagger = 932$ for 8 kHz model and 201 for 16 kHz model	Input: Linear, $\mathbf{x} \in \mathbb{R}^{d_\mathbf{x}}$	Input: Linear, $\hat{\mathbf{x}} \in \mathbb{R}^{d_\mathbf{x}}$
FC‡ 512 ReLU	FC 512 ReLU	FC 512 ReLU
FC 512 ReLU	FC 512 ReLU	FC 512 ReLU
Output: FC d_x linear for $\hat{\mathbf{x}}$	FC 512 ReLU	FC 1024 ReLU
	Output: FC 1 linear	Output: FC d_z linear for $\hat{\mathbf{z}}$. Softmax on last d_c to obtain $\hat{\mathbf{z}}_c$

*Dimension of $\mathbf{z}_n$,

†Dimension of $\mathbf{z}_c$,

‡Fully-connected

TABLE IV

Results (avg. DER (%)) on CALLHOME database for various training setups and proposed embeddings.

Embedding	Back-end			
	k-means		SC	
	Known #speakers	est. #speakers	Known #speakers	est. #speakers
E_{cross}	14.66	13.47	9.74	9.62
E_{proto1}	9.09	8.13	6.34	7.48
E_{proto2}	9.62	8.18	6.34	7.40
E_{proto3}	10.99	9.06	7.12	8.15
ClusterGAN	10.24	9.83	7.62	9.24
MCGAN	**8.72**	**7.60**	**6.01**	**7.03**

TABLE V

Results on CALLHOME database for MAPD of predicted speaker number and POC in estimating speaker number.

Metric	x-vector	ClusterGAN	MCGAN	Fusion (x-vector + MCGAN)
MAPD	12.54%	11.23%	**9.76%**	10.59%
POC	74.15%	72.14%	**75.55%**	75.35%

TABLE VI

Results on CALLHOME database for the baseline and proposed systems.

Embedding	Back-end	Avg. DER (%) (oracle SAD, known #speakers)	Avg. DER (%) (oracle SAD, est. #speakers)
x-vector	k-means	9.00	8.69
ClusterGAN		10.24	9.83
MCGAN		8.72	7.60
x-vector + ClusterGAN		8.98	8.77
x-vector + MCGAN		8.40	7.48
x-vector	SC	6.23	8.32
ClusterGAN		7.62	9.24
MCGAN		6.01	7.03
x-vector + ClusterGAN		6.22	7.70
x-vector + MCGAN		**5.73**	**6.76**
Wang et al. [7] d-vector	SC	-	12.00
Romero et al. [3] x-vector	AHC+VB*	-	9.90
Kaldi x-vector	PLDA+AHC+CV†	7.12	8.39
Zhang et al. [8] d-vector (5-fold)	UIS-RNN+CV	-	7.60
Park et al. [11] Kaldi x-vector	NME-SC	-	7.29

*Variational Bayes re-segmentation,

†Cross-validation

IEEE/ACM Trans Audio Speech Lang Process. Author manuscript; available in PMC 2022 January 01.

TABLE VII

Results (avg. DER in %) on AMI dev and eval, DIHARD-II dev, ADOS, and BOSCC databases for the baseline and proposed systems.

Col.* 1	Col. 2	Col. 3	Col. 4	Col. 5	Col. 6	Col. 7
Embedding	Back-end	AMI		DIHARD-II	ADOS	BOSCC
		Dev	Eval	dev set		
x-vector	k-means	12.64	12.26	23.38	12.35	14.73
ClusterGAN		11.34	11.51	26.03	10.21	10.59
MCGAN		7.09	6.09	25.71	9.71	14.67
x-vector + ClusterGAN		9.57	8.63	22.87	8.70	10.52
x-vector + MCGAN		6.47	8.76	22.53	9.10	13.22
x-vector	SC	6.42	6.23	18.88	8.51	11.99
ClusterGAN		6.41	8.16	21.84	6.75	9.32
MCGAN		5.10	5.38	21.16	9.96	13.21
x-vector + ClusterGAN		6.21	**2.87**	**17.75**	**6.74**	**9.26**
x-vector + MCGAN		**5.02**	4.92	18.59	9.18	12.17
Kaldi x-vector	PLDA+AHC	8.09	10.06	21.26	14.36	21.69

*Col. represents column number.

TABLE VIII

Diarization performance of the proposed and baseline systems in each specific domain on the DIHARD-II dev set.

System	audiobooks * (12)	broadcast_interview (12)	maptask (23)	socio_lab (16)	socio_field (12)	court (12)	clinical (24)	child (23)	webvideo (32)	meeting (14)	restaurant (12)
x-vector + SC	5.92	**4.56**	9.23	**7.87**	12.17	5.60	20.20	31.65	**33.74**	10.20	42.68
ClusterGAN + SC	1.98	4.75	9.50	8.75	15.54	4.67	24.48	31.77	37.37	24.94	54.04
MCGAN + SC	**1.59**	4.86	10.14	11.99	11.64	**4.66**	29.33	34.26	36.38	12.50	44.41
(x-vector + ClusterGAN) + SC	2.29	4.68	9.10	8.20	**9.20**	5.31	**16.37**	**29.69**	34.66	**9.90**	**40.46**
(x-vector + MCGAN) + SC	3.64	4.77	**8.92**	8.63	12.28	4.78	18.46	31.06	34.42	11.24	42.10

* The number of sessions within each domain