IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 17, 2020, accepted July 26, 2020, date of publication July 30, 2020, date of current version August 13, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3012893

Disentangled Speaker and Nuisance Attribute Embedding for Robust Speaker Verification

WOO HYUN KANG, (Student Member, IEEE), SUNG HWAN MUN, (Student Member, IEEE), MIN HYUN HAN, (Student Member, IEEE), AND NAM SOO KIM, (Senior Member, IEEE)
Department of Electrical and Computer Engineering and INMC, Seoul National University, Seoul 08826, South Korea

Corresponding author: Nam Soo Kim (nkim@snu.ac.kr)

This work was supported by the BK21 Plus program of the Creative Research Engineer Development for IT, Seoul National University in 2020; in part by the Korean National Police Agency. [Project Name: Real-time speaker recognition via voiceprint analysis / Project Number: PA-J000001-2017-101].

ABSTRACT Over the recent years, various deep learning-based embedding methods have been proposed and have shown impressive performance in speaker verification. However, as in most of the classical embedding techniques, the deep learning-based methods are known to suffer from severe performance degradation when dealing with speech samples with different conditions (e.g., recording devices, emotional states). In this paper, we propose a novel fully supervised training method for extracting a speaker embedding vector disentangled from the variability caused by the nuisance attributes. The proposed framework was compared with the conventional deep learning-based embedding methods using the RSR2015 and VoxCeleb1 dataset. Experimental results show that the proposed approach can extract speaker embeddings robust to channel and emotional variability.

INDEX TERMS Speech embedding, speaker verification, domain disentanglement, deep learning.

I. INTRODUCTION

Speaker verification is the task of verifying the claimed speaker identity based on the given speech samples and has become a key technology for personal authentication in many commercial applications, forensics and law enforcement [1]. Commonly, an utterance-level fixed-dimensional vectors (i.e. embedding vectors) are extracted from the enrollment and test speech samples and then fed into a scoring algorithm (e.g., cosine distance, probabilistic linear discriminant analysis) to measure their similarity or likelihood of being spoken by the same speaker. Over the past years, the i-vector framework has been one of the most dominant approaches for speech embedding [2], [3]. The widespread popularity of the i-vector framework in the speaker verification community can be attributed to its ability to summarize the distributive pattern of the speech with a relatively small amount of training data in an unsupervised manner.

In recent years, various methods have been proposed utilizing deep learning architectures for extracting embedding vectors and have shown better performance than the i-vector framework when a large amount of training data is available [4]. In [5], a deep neural network (DNN) for frame-level speaker identification was trained and the averaged activation from the last hidden layer, namely, the d-vector, was taken as the embedding vector for text-dependent speaker verification. In [4], [6], a speaker identification model consisting of a frame-level network and a segment-level network was trained and the hidden layer activation of the segment-level network (i.e. x-vector) was extracted as the embedding vector. In [7], long short-term memory (LSTM) layers were adopted to capture the contextual information within the d-vector, and the embedding network was trained to directly optimize the verification score (e.g., cosine similarity) in an end-to-end fashion. The end-to-end d-vector framework was further enhanced in [8] by applying different weight (i.e. attention) to each frame-level activation while obtaining the d-vector, which enables the embedding network to attend more on the frames with relatively higher amount of speaker-dependent information. In [9], a generalized end-to-end loss function, which optimizes the embedding vector to move towards the centroid of the true speaker while departing away from the centroid of the most confusing speaker, was introduced to train the end-to-end d-vector system more efficiently. In [10]

The associate editor coordinating the review of this manuscript and approving it for publication was Yongming Li.

and [11], a variational autoencoder (VAE)-based architecture was trained in an unsupervised manner to extract an embedding vector for short-duration speaker verification. Despite their success in well-matched conditions, the deep learning-based embedding methods are vulnerable to the performance degradation caused by mismatched conditions (e.g., channel, noise) [12].

In real life applications, numerous factors can contribute to the mismatches in speaker verification [1]. Especially in forensic situations, channel mismatch often occurs since police officers usually acquire voice recordings using various recording devices (e.g., hidden microphones, mobile phones) [13]. Such variation in recording devices is known to cause variability to the speech distribution, which leads to low speaker identification or verification performance.

Recently, many attempts have been made to extract an embedding vector robust to mismatched conditions. Conventionally, various researches focused on adapting the back-end scoring model (e.g., PLDA) [14] or training the embedding network with an augmented dataset containing various nuisance variability [15]. These methods are proven to be effective when the dataset for the target condition (e.g., noisy evaluation domain) is scarce, but since these methods do not intervene during the embedding extraction, their performance may be bottlenecked by the speaker discriminative capability of the embedding network. Unlike the aforementioned domain adaptation techniques, there have been several methods which aim to directly disentangle the undesired variability while extracting the speaker embeddings. In [12], [16], inspired by the usage of gradient reversal strategy in image classification [17], [18] and robust speech recognition [19], [20], the embedding networks were trained to minimize the speaker classification error while maximizing the error of the subtask (e.g., noise or channel type classification) with the use of gradient reversal layer. Although the gradient reversal strategy has shown meaningful improvement in performance, domain adversarial training using gradient reversal layer is known to be very unstable and sensitive to hyper-parameter setting [21]. In [22], the embedding network was trained to maximize the error of a subtask (i.e. noise type classification) by using an adversarial training strategy similarly to the generative adversarial network (GAN) [23]. The speaker embedding network and the noise classification network are trained competitively; the noise classification network is trained to discriminate the noise type correctly, and at the same time the embedding network is trained to discriminate the speaker while having high uncertainty on the noise type. When training the speaker embedding network, bit-inverted one-hot labels (i.e. anti-labels) were used for noise classification, which would force the embedding network to output a wrong noise label equally. Though the anti-label strategy has proven its strength in noise-robust speaker embedding [22], adversarial training is known to be extremely unstable and difficult [24].

In this paper, we propose a novel approach to disentangle the nuisance attribute information from the speaker embedding vector without the use of gradient reversal or adversarial training. The proposed method employs an embedding network similar to the conventional methods (e.g., d-vector and x-vector). However, unlike the conventional embedding networks, which produce a single embedding vector per utterance, the proposed embedding network simultaneously extracts a speaker- and nuisance attribute-dependent (e.g., recording device-, emotion-dependent) embedding vectors, hence we call the proposed technique joint factor embedding (JFE). In the JFE technique, the embedding network is trained in a fully supervised manner simultaneously with the speaker and nuisance attribute (e.g., channel, emotion) discriminator networks where each discriminator is trained to take the embedding vector as input and identify their respective targets. Analogous to the conventional speaker embedding systems, the proposed embedding network is trained to produce a speaker embedding vector with high speaker discriminability. On the other hand, to disentangle the non-speaker information from the speaker embedding vector, we propose two different ways to increase the nuisance attribute uncertainty inherent in the speaker embedding vector. One way is to train the embedding network to extract a speaker embedding vector to maximize the entropy in nuisance attribute identification, and the other is to decrease the relevancy between the speaker and nuisance embedding vectors by minimizing the mean absolute Pearson's correlation (MAPC) [25].

In order to evaluate the performance of the proposed system in a realistic scenario, we conducted a set of experiments using two datasets:

- RSR2015 Part 3 dataset: a random digits strings speaker verification corpus consisting of speech samples recorded from 6 different hand-held devices [26], [27].
- VoxCeleb1 dataset: a text-independent speaker verification corpus consisting of speech samples with 8 different emotional states [28].

The experimental results show that the proposed method outperforms the conventional disentanglement methods (i.e. gradient reversal, anti-label) in terms of equal error rate (EER). Moreover, the proposed system performed better than the conventional x-vector on short duration speech samples, which is likely to lack significant phonetic information.

The contributions of this paper are as follows:

- We propose a new method to train a speaker embedding network robust to nuisance attributes, which can be done easily without the use of adversarial training or gradient reversal learning.
- We compared the proposed speaker embedding technique with conventional methods for multi-device and emotional speaker verification.
- We experimented the proposed speaker embedding technique on speech utterances with various durations.

The rest of this paper is organized as follows: We first briefly describe the conventional embedding network architecture and disentanglement methods based on gradient reversal and anti-label in Section II. In Section III, the newly

proposed JFE scheme is presented. The experiments and results are shown in Section IV. Finally, Section V concludes the paper.

II. DEEP LEARNING-BASED SPEAKER EMBEDDING

A. DEEP EMBEDDING NETWORK

Two of the most widely used speaker embedding techniques are the LSTM-based d-vector [9] and the TDNN (time-delay DNN)-based x-vector system [4]. In both frameworks, given a speech utterance $\mathbf{X}$ with T frames, a sequence of frame-level acoustic features $\{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ extracted from $\mathbf{X}$ is fed into the frame-level network. In the d-vector system, one of most widely used technique for text-dependent speaker recognition, the frame-level network is composed of LSTM layers, which helps capture the temporal correlation. On the other hand, the frame-level network of the x-vector system consists of TDNN layers, which is often used for text-independent speaker recognition. Once the frame-level outputs $\{\mathbf{h}_1, \ldots, \mathbf{h}_T\}$ are obtained, they are aggregated to obtain an utterance-level representation. One way of aggregating the frame-level outputs is to compute the weighted average as

$$\omega = \sum_{t=1}^{T} \alpha_t \mathbf{h}_t \tag{1}$$

where $\alpha_t \in [0, 1]$ is a normalized weight, which is computed by

$$\alpha_t = \frac{\exp(e_t)}{\sum_{t=1}^{T} \exp(e_t)}. \tag{2}$$

In (2), the frame-level score (i.e. attention) e_t is computed as follows:

$$e_t = \mathbf{v}_t^{\mathsf{T}} \tanh(\mathbf{W}_t \mathbf{h}_t + \mathbf{b}_t) \tag{3}$$

where $\mathbf{v}_t$, $\mathbf{W}_t$, and $\mathbf{b}_t$ are trainable parameters and superscript T indicates transpose operation. By using different weight for each frame, speech frames with relatively higher speaker-relevancy can contribute more to the embedding vector.

The embedding network is trained by either minimizing the speaker identification loss [5] or directly optimizing the verification performance (i.e. end-to-end speaker verification) [9]. In the first case (i.e. embedding network trained for identification), as shown in Fig. 1a, a feed-forward neural network for classifying the speakers in the training set is trained jointly with the embedding network. The speaker classification network takes the utterance-level representation ω as input and has an N-dimensional softmax output $\tilde{\mathbf{y}}(\omega)$ where N corresponds to the number of training speakers. Given the one-hot speaker label $\mathbf{y}$, the embedding and classification networks are trained to minimize the following cross-entropy loss function:

$$\mathbf{L}_{spkr} = -\sum_{n=1}^{N} \mathbf{y}_n \log \tilde{\mathbf{y}}_n(\omega) \tag{4}$$

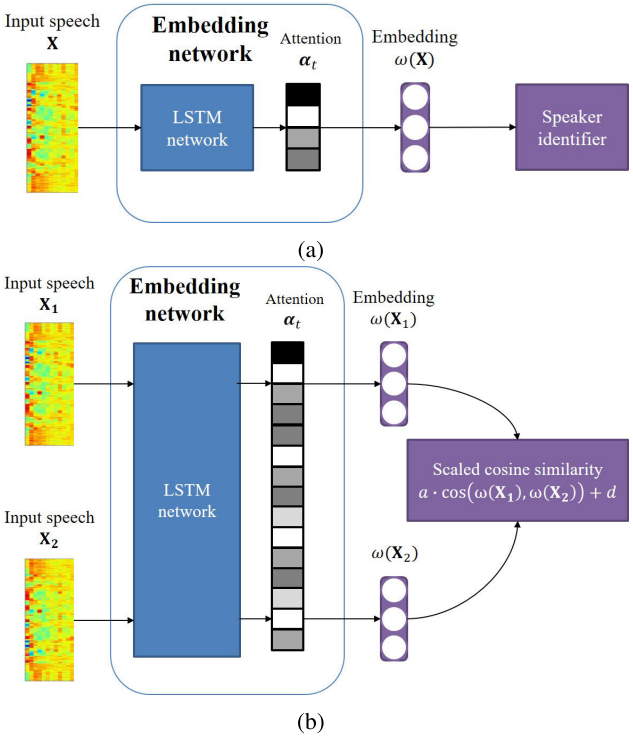

FIGURE 1. (a) LSTM-based d-vector system trained with softmax loss. (b) LSTM-based d-vector system trained with end-to-end loss.

where $\mathbf{y}_n$ and $\tilde{\mathbf{y}}_n(\omega)$ are the n^{th} components of $\mathbf{y}$ and $\tilde{\mathbf{y}}(\omega)$, respectively.

For training the end-to-end speaker verification system (i.e. embedding network trained for verification), a mini-batch of $J \times K$ utterances is fed into the embedding network where the mini-batch is composed of J speakers, and each speaker has K utterances. As depicted in Fig. 1b, the scaled cosine similarity between each embedding vector and the centroid of the embedding vectors from each speaker are computed by

$$\mathbf{S}_{jk,i} = a \cdot \cos(\omega_{jk}, \mathbf{c}_i) + d \tag{5}$$

where a and d are trainable parameters, and $\cos(\omega_{jk}, \mathbf{c}_i)$ is the cosine similarity between the utterance-level representation extracted from the k^{th} utterance of the j^{th} speaker ω_{jk} and the centroid of the i^{th} speaker's utterance-level representations $\mathbf{c}_i$ ($1 \le j, i \le J$ and $1 \le k \le K$). For each utterance-level representation ω_{jk} in the mini-batch, the embedding network is trained to maximize the following end-to-end loss function:

$$\mathbf{L}_{e2e} = \mathbf{S}_{jk,j} - \log \sum_{i=1, i \neq j}^{J} \exp(\mathbf{S}_{jk,i}). \tag{6}$$

The end-to-end system is known to outperform the softmax method when a large amount of dataset is used for training [6], [7].

Once the embedding network is trained, the utterance-level representation ω [9], or the hidden layer activation of the

speaker classification network [4] can be used as the speaker embedding vector.

B. CONVENTIONAL DISENTANGLEMENT METHODS

Recently, disentangling various non-speaker factors (e.g., channel type, noise type, noise-level) from the embedding vector has become an important issue in speaker verification [12], [16], [22]. Most of the techniques developed to address this issue are based on the multi-task learning (MTL) approaches [29] where the embedding network is trained to optimize in two tasks: main task (i.e. speaker classification) and subtask (e.g., channel classification) as shown in Fig. 2a. The objective of the MTL-based disentanglement technique is to achieve the best performance in the main task while degrading the performance in the subtask.

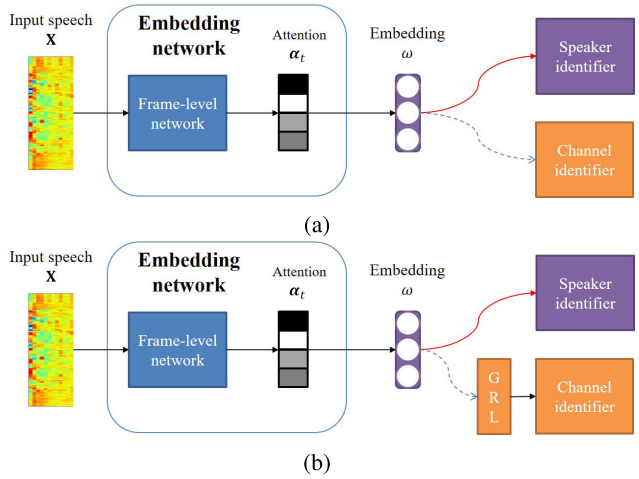

FIGURE 2. (a) Standard multi-task learning (MTL) architecture. (b) Domain adversarial training via gradient reversal layer (GRL).

1) GRADIENT REVERSAL STRATEGY

One way to achieve this is the gradient reversal strategy, which has shown meaningful performance in channel-robust [16] and noise-robust [12] speaker verification. As shown in Fig. 2b, the gradient reversal strategy adds a gradient reversal layer (GRL) [17] between the subtask network and the embedding network. Let θ_{emb}, θ_{main}, θ_{sub} denote the parameters for the embedding, main task, and subtask networks. The GRL performs identity transformation on the input during forward propagation and reverses the gradient by multiplying a negative scalar $-\lambda$ during backpropagation. When jointly training the networks, the parameters are updated as

$$\theta_{emb} \leftarrow \theta_{emb} - l \cdot \left(\frac{\partial \mathbf{L}_{main}}{\partial \theta_{emb}} - \lambda \frac{\partial \mathbf{L}_{sub}}{\partial \theta_{emb}}\right), \tag{7}$$

$$\theta_{main} \leftarrow \theta_{main} - l \cdot \left(\frac{\partial \mathbf{L}_{main}}{\partial \theta_{main}}\right), \tag{8}$$

$$\theta_{sub} \leftarrow \theta_{sub} - l \cdot \left(\frac{\partial \mathbf{L}_{sub}}{\partial \theta_{sub}}\right) \tag{9}$$

where l, $\mathbf{L}_{main}$, and $\mathbf{L}_{sub}$ are the learning rate, loss functions for the main task and subtask, respectively. For extracting a channel-robust embedding for speaker verification, $\mathbf{L}_{main}$ would be the speaker cross-entropy $\mathbf{L}_{spkr}$ defined in (4), and $\mathbf{L}_{sub}$ would be the channel cross-entropy which can be computed as follows:

$$\mathbf{L}_{chan} = -\sum_{m=1}^{M} \mathbf{r}_m \log \tilde{\mathbf{r}}_m(\omega) \tag{10}$$

where M is the number of different channels (e.g., recording devices) in the training set, $\mathbf{r}_m$ and $\tilde{\mathbf{r}}_m(\omega)$ are the m^{th} component of the one-hot channel label $\mathbf{r}$ and channel classifier's softmax output $\tilde{\mathbf{r}}(\omega)$, respectively.

2) ANTI-LOSS STRATEGY

Another way to achieve disentanglement is by training the embedding network and the subtask network in a competitive manner via adversarial training [22]. The subtask network is trained to classify the channel identity correctly given the embedding vector as in (10). On the other hand, the main task and embedding networks are trained to discriminate the speaker by minimizing (4) but not to perform well on the subtask. In order to ensure high uncertainty on the subtask, [22] introduces anti-label when computing the cross-entropy for the subtask. The anti-label is obtained by flipping each bit in the one-hot label vector. This indicates that for channel disentanglement, the anti-loss can be computed as follows:

$$\mathbf{L}_{anti-dev} = -\sum_{m=1}^{M} (1 - \mathbf{r}_m) \log \tilde{\mathbf{r}}_m(\omega). \tag{11}$$

By minimizing $\mathbf{L}_{anti-dev}$ and $\mathbf{L}_{speaker}$ simultaneously, the embedding network would be trained to produce a speaker discriminative embedding vector which is robust to channel variability.

III. JOINT FACTOR EMBEDDING

A. JOINT FACTOR EMBEDDING NETWORK ARCHITECTURE

Analogous to the conventional disentanglement techniques [12], [16], [22], the proposed method is based on the MTL framework. However, as depicted in Fig. 3, unlike the standard MTL embedding system, the embedding network of the proposed framework extracts two different embedding vectors simultaneously: speaker embedding ω_{spkr} and nuisance embedding ω_{nuis}. The speaker embedding vector ω_{spkr} is trained to be dependent solely on the speaker variability while the nuisance embedding vector ω_{nuis} is trained to be dependent on the nuisance (e.g., channel, emotion) variability only. When obtaining ω_{spkr} and ω_{nuis}, different weights are used for aggregating the frame-level outputs as

$$\omega_{spkr} = \sum_{t=1}^{T} \alpha_{spkr,t} \mathbf{h}_t, \tag{12}$$

$$\omega_{nuis} = \sum_{t=1}^{T} \alpha_{nuis,t} \mathbf{h}_t \tag{13}$$

where $\alpha_{spkr,t}$ and $\alpha_{nuis,t}$ are the speaker and nuisance weights for attention, respectively, which are obtained as in (2).

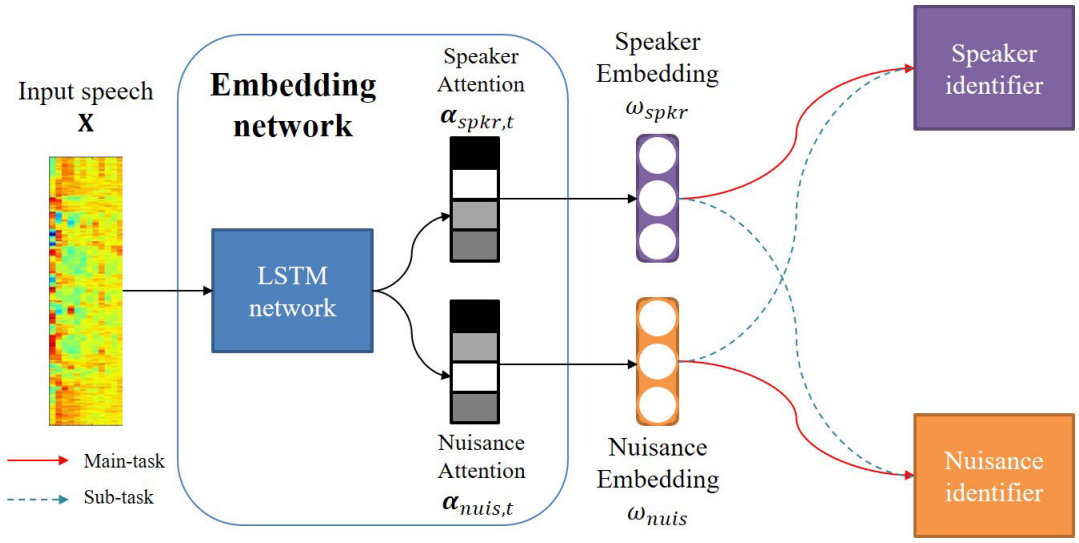

FIGURE 3. The architecture of the proposed joint factor embedding system.

TABLE 1. Main tasks and subtasks for the embedding vectors of the joint factor embedding scheme.

	Main task	Subtask
ω_{spkr}	Speaker classification	Nuisance classification
ω_{nuis}	Nuisance classification	Speaker classification

The reason why we use separate attention weights for obtaining ω_{spkr} and ω_{nuis} is that we assume that frames with high speaker-dependent information are not always guaranteed to have high nuisance attribute-dependent information. For instance, speaker-dependent information will be high on speech frames, while channel-dependent information will be rather consistent across all frames since even non-speech frames are affected by the recording channel. Once the embedding vectors are extracted, both ω_{spkr} and ω_{nuis} are fed into the speaker and nuisance classification networks.

B. TRAINING FOR JOINT FACTOR EMBEDDING

1) DISCRIMINATIVE TRAINING

As described in Table 1, the embedding vectors ω_{spkr} and ω_{nuis} are trained with different main task and subtask specifications. In order to maximize the discriminability on their main tasks, the following cross-entropy loss functions are minimized:

$$\mathbf{L}_{s-s,CE} = -\sum_{n=1}^{N} \mathbf{y}_n \log \tilde{\mathbf{y}}_n(\omega_{spkr}), \tag{14}$$

$$\mathbf{L}_{c-c,CE} = -\sum_{m=1}^{M} \mathbf{r}_m \log \tilde{\mathbf{r}}_m(\omega_{nuis}). \tag{15}$$

By minimizing (14) and (15) simultaneously, the embedding network is trained to produce ω_{spkr} with high speaker-dependent information and ω_{nuis} with high nuisance attribute-dependent information. Moreover, the attention weights $\alpha_{spkr,t}$ and $\alpha_{nuis,t}$ will be trained to focus on the frames with more meaningful information on their main tasks.

2) DISENTANGLEMENT TRAINING

In this paper, we propose two types of loss functions to perform disentanglement in the subtasks of the embedding vectors ω_{spkr} and ω_{nuis}. One way for disentanglement is to directly maximize the entropy (or uncertainty) on their subtasks while training. For ω_{spkr} and ω_{nuis}, the entropies [30] on their subtasks can be computed as

$$\mathbf{L}_{s-c,E} = -\sum_{n=1}^{N} \tilde{\mathbf{y}}_n(\omega_{nuis}) \log \tilde{\mathbf{y}}_n(\omega_{nuis}), \tag{16}$$

$$\mathbf{L}_{c-s,E} = -\sum_{m=1}^{M} \tilde{\mathbf{r}}_m(\omega_{spkr}) \log \tilde{\mathbf{r}}_m(\omega_{spkr}). \tag{17}$$

By maximizing (16) and (17), the uncertainty of the outputs in the subtasks will be maximized, leading the conditional distribution of the subtask classes to approach uniform.

Another way to perform disentanglement is to regularize the embedding vectors ω_{spkr} and ω_{nuis} so as to have low correlation instead of directly maximizing the uncertainty on their subtasks. This can be achieved by maximizing the negative MAPC [25], which can be computed across the mini-batch by

$$\mathbf{L}_{nMAPC} = -\frac{1}{F}\sum_{f=1}^{F} \frac{|cov(\omega_{spkr,f}, \omega_{nuis,f})|}{std(\omega_{spkr,f})std(\omega_{nuis,f})} \tag{18}$$

where cov is the covariance, std is the standard deviation, and F, $\omega_{spkr,f}$, $\omega_{nuis,f}$ are the dimensionality of the embedding vectors, f^{th} element of ω_{spkr} and ω_{nuis}, respectively. Since zero correlation indicates that the two variables are not related, by minimizing the MAPC between ω_{spkr} and ω_{nuis},

the relevancy between the two embedding vectors can be reduced.

The proposed JFE system is trained by simultaneously minimizing the discriminative losses (i.e. cross-entropy) depicted in (14) and (15), while maximizing the disentanglement loss in (16), (17), (18). In short, the embedding network is trained to minimize the following loss function:

$$\mathbf{L}_{JFE} = \mathbf{L}_{s-s,CE} + \mathbf{L}_{c-c,CE} - \mathbf{L}_{s-c,E} - \mathbf{L}_{c-s,E} - \mathbf{L}_{nMAPC}. \quad (19)$$

By optimizing the JFE network, the speaker embedding vector ω_{spkr} is trained to be speaker discriminative while having high uncertainty on the nuisance attribute, and the nuisance embedding vector ω_{nuis} aims to be nuisance attribute discriminative while having high uncertainty on the speaker.

IV. EXPERIMENTS

A. CHANNEL DISENTANGLEMENT EXPERIMENTS

1) DATABASE

In order to evaluate the performance of the proposed technique for a real-life application of speaker verification where multiple recording devices are involved for enrollment and testing, a set of experiments were conducted based on the RSR2015 dataset [26], [27], which is a speaker verification dataset recorded using 6 different hand-held devices (i.e. 1 Samsung Nexus, 2 Samsung Galaxy S, 1 HTC Desire, 1 Samsung Tab, 1 HTC Legend). For training the embedding networks, we used the *background* and *development* subsets of the RSR2015 dataset Part 3, consisting of utterances (recorded from all six devices) spoken by 194 speakers (100 male and 94 female speakers).

The evaluation was performed according to the RSR2015 Part 3 (random digits string) protocol [27] where 106 speakers (57 male and 49 female speakers) are involved. From the RSR2015 Part 3 evaluation dataset, the 10-digits strings of sessions 1, 4, 7 were used for enrollment and the 5-digits strings of sessions 2, 3, 5, 6, 8, 9 were used for testing.

2) EXPERIMENTAL SETUP

To investigate the effects of the proposed JFE strategy on different embedding architecture, two types of frameworks were used for embedding extraction: d-vector and x-vector. For the d-vector-based systems, a single 512-dimensional unidirectional LSTM layer with a projection layer [31] (projected to 256-dimension) was used. By aggregating the LSTM outputs via a weighted average as described in (1), 256-dimensional embedding vectors were obtained. Each classification networks (i.e. speaker and channel identifier) consisted of a single 256-dimensional rectified linear unit (ReLU) hidden layer and a softmax output layer where the output size corresponds to the number of speakers or devices within the training set (e.g., 194-dimensional softmax output for speaker classifier and 6-dimensional softmax output for channel classifier). The acoustic features used in the d-vector-based systems were 19-dimensional Mel-frequency cepstral coefficients (MFCCs) and the log-energy extracted at every 10 ms, using a 20 ms Hamming window. Together with the delta and delta-delta of the 19-dimensional MFCCs and the log-energy, the frame-level feature used in our experiments was a 60-dimensional vector.

For the x-vector-based systems, 5 TDNN layers were used as the frame-level network as in the Kaldi x-vector recipe [4]. The frame-level output of the last TDNN layer were aggregated via attention pooling (1) and followed by a ReLU layer, resulting in a 512-dimensional embedding vector. The classification networks in the x-vector-based systems consisted of a single 512-dimensional rectified linear unit (ReLU) hidden layer and a softmax output layer. The acoustic features used in the x-vector-based systems were 30-dimensional MFCCs extracted at every 10 ms, using a 20 ms Hamming window.

The implementation of the embedding systems was done via Tensorflow [32] and trained using the ADAM optimization technique [33] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. All the experimented networks were trained with learning rate 0.001 and batch size 32 for 12,000 iterations. Cosine similarity was used for computing the verification scores in the experiments.

In our experiments, EER was evaluated as the performance measure. The EER indicates the error when the false alarm rate (FAR) and the false reject rate (FRR) are the same.

3) COMPARISON BETWEEN DIFFERENT DISENTANGLEMENT LOSS TERMS

In this experiment, we compare the performance of the speaker embeddings obtained from the d-vector-based JFE system trained with different disentanglement loss terms discussed in Section III. The experimented methods are as follows:

- *Only discriminative*: speaker embedding vector extracted from the JFE network trained only with the discriminative loss functions in (14) and (15) (which is essentially a multi-task learning for the embedding network to encode speaker and nuisance discriminative information),
- *Entropy*: speaker embedding vector extracted from the JFE network trained with the discriminative loss functions in (14), (15) and the entropy-based disentanglement losses in (16) and (17),
- *nMAPC*: speaker embedding vector extracted from the JFE network trained with the discriminative loss functions in (14), (15) and the negative MAPC-based disentanglement losses in (18),
- *Entropy* + *nMAPC*: speaker embedding vector extracted from the JFE network trained with the discriminative loss functions in (14), (15) and both the entropy-based and the negative MAPC-based disentanglement losses in (16), (17) and (18).

Table 2 gives the EER results obtained by using these embeddings. As shown in the results, the embedding extracted from the JFE networks trained with either *Entropy* or *nMAPC* for disentanglement greatly improved the

TABLE 2. EER (%) comparison between the speaker embedding vectors extracted from the joint factor embedding networks trained with various disentanglement losses.

Loss	EER [%]
Only discriminative	11.28
Entropy	9.61
nMAPC	9.25
Entropy + nMAPC	**8.43**

performance compared to *Only discriminative*, which is essentially a standard MTL embedding technique. This implies that both *nMAPC* and *Entropy* are capable of training the embedding network to produce speaker embedding vectors disentangled from non-speaker factors. Especially the *nMAPC* showed relative improvement of 17.99% compared to *Only discriminative*. The best verification performance was achieved by using both disentanglement loss terms (i.e. *Entropy* + *nMAPC*), yielding a relative improvement of 25.27% in terms of EER. From this, we could assume that *nMAPC* and *Entropy* are useful for disentangling the channel variability from the speaker embedding. The DET curves are depicted in Figure 4.

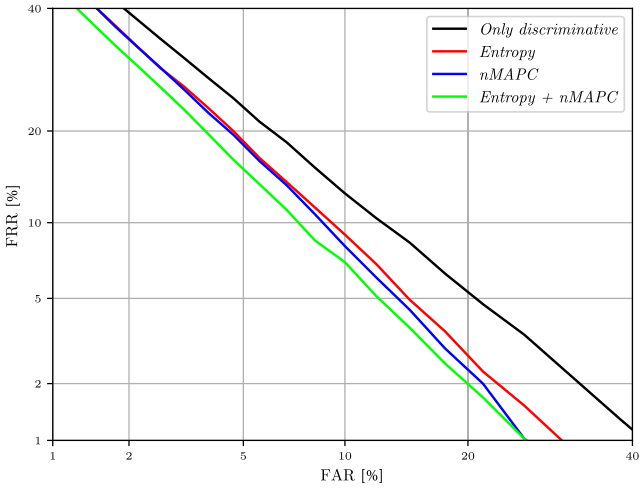

FIGURE 4. DET curves of the JFE systems trained with various disentanglement losses.

4) TRAINING ANALYSIS

In order to check if the training scheme of the proposed JFE system achieves our objective (i.e. maximizing the speaker discriminability and channel uncertainty in ω_{spkr}), we analyzed the training loss described in (14)-(17) of the d-vector-based JFE system. As shown in Fig. 5, due to the large difference in the unique number of speakers and devices (i.e. 194 speakers and 6 devices), the initial values for $\mathbf{L}_{s-s,CE}$ and $\mathbf{L}_{s-c,E}$ were higher than $\mathbf{L}_{c-c,CE}$ and $\mathbf{L}_{c-s,E}$. The cross-entropy losses (i.e. $\mathbf{L}_{s-s,CE}$ and $\mathbf{L}_{c-c,CE}$) decreased quickly toward 0 when the training iteration increases. On the other hand, the entropy losses (i.e. $\mathbf{L}_{s-c,E}$ and $\mathbf{L}_{c-s,E}$)

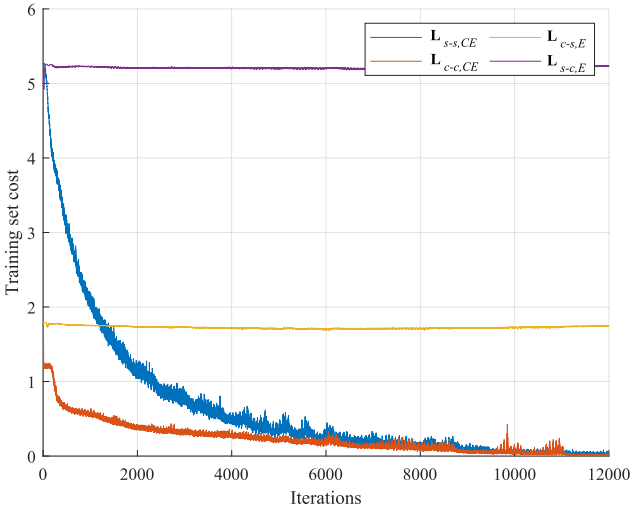

FIGURE 5. The joint factor embedding training loss values on each iteration.

stayed near at their initial values throughout the training. This indicates that the proposed training scheme increases the discriminability of the speaker and channel embeddings on their main tasks while keeping their uncertainty on the subtasks high as expected.

In Fig. 6, the t-SNE plots [34] of the speaker and channel embedding vectors of 10 speakers and 3 devices are shown. As can be seen in Figs. 6a and 6c, the speaker embedding vectors ω_{spkr} were well separated between different speakers but were highly overlapped when it comes to different devices. Meanwhile, as shown in Figs. 6b and 6d, the channel embedding vectors ω_{chan} were separately distributed in terms of

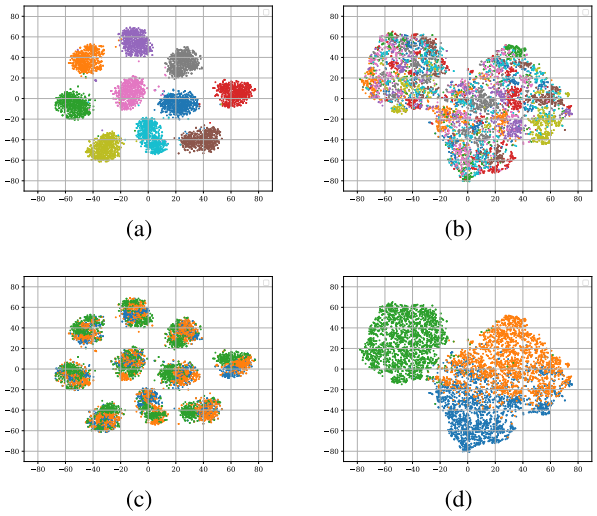

FIGURE 6. t-SNE plot of the speaker and channel embedding vectors extracted from 10 speakers and 3 devices. The x and y axis indicates the 1^{st} and 2^{nd} dimension of the 2D T-SNE projection, respectively. (a) and (c) are the t-SNE plots of the speaker embedding vectors, and (b) and (d) are the t-SNE plots of the channel embedding vectors. Different colors in (a) and (b) indicate different speakers, and different colors in (c) and (d) indicates different devices.

the device, while they were inseparable in terms of speakers. This confirms that the embedding vectors extracted from the proposed JFE system are discriminative on their main tasks, but are invariant with respect to their subtasks.

Moreover, in Fig. 7, the attention weights for the utterance speaking the sentence "only lawyers love millionaires" (i.e. 1^{st} sentence of the RSR2015 Part1 dataset) are shown. It is interesting to see that the difference between speaker attention weights α_{spkr} across the frames were quite dramatic, which indicates that α_{spkr} are likely to attend to certain frames. On the other hand, the channel attention weights α_{chan} were relatively consistent across all frames. These results strongly support our assumption that the frames with high speaker-dependent information are concentrated on specific frames while channel-dependent information is similar across the speech segment.

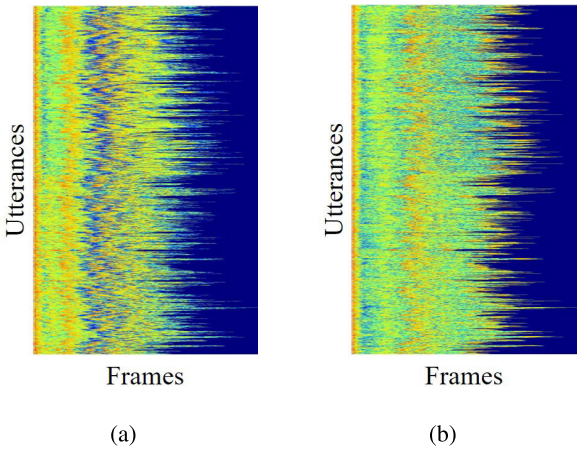

FIGURE 7. Attention weights of *d-vector (JFE)* for utterances speaking the sentence "only lawyers love millionaires." (a) Attention weights for the speaker embedding vector. (b) Attention weights for the channel embedding vector.

5) COMPARISON BETWEEN THE JOINT FACTOR EMBEDDING SCHEME AND CONVENTIONAL DISENTANGLEMENT METHODS

In this experiment, we compared the embedding vectors obtained from the proposed joint factor embedding scheme, with those obtained from the conventional disentanglement techniques discussed in Section II. The experimented training strategies are as follows:

- *Softmax*: embedding extracted from an embedding network trained with softmax objective in (4),
- *Gradient reversal*: embedding extracted from an embedding network trained with gradient reversal strategy as described in (7) where λ was set to be 0 in the beginning and linearly increased every iteration, reaching 1 at the end of the training as in [19],
- *Anti-loss*: embedding extracted from an embedding network trained with anti-loss as described in (11) using the same adversarial training strategy described in [22],
- *JFE (proposed)*: speaker embedding extracted from the proposed JFE system trained with the discriminative loss functions in (14) and (15) and both the entropy-based as shown in (16) and (17) and the negative MAPC-based disentanglement losses in (18).

Table 3 show the performance of the d-vector and x-vector-based systems trained with the methods described above. Generally, the *Anti-loss* disentanglement strategy has shown performance enhancement, achieving a relative improvement of 35.39% in terms of EER in the d-vector-based experiment. On the other hand, *Gradient reversal* method, showed only slightly improved or worse performance over *softmax*. Meanwhile, the speaker embedding extracted from the proposed JFE scheme yielded the best performance in all architectures (i.e., d-vector and x-vector), achieving a relative improvement of 18.39% in EER compared to that of *d-vector (softmax)*. This indicates that the proposed JFE system is capable of disentangling complicated corruptions (i.e. corruption via channel) introduced by different recording devices.

TABLE 3. EER (%) comparison between the speaker embedding vectors extracted from the proposed joint factor embedding and the other embedding techniques.

	Objective	EER [%]
d-vector	*Softmax*	10.72
	Gradient reversal	10.37
	Anti-loss	10.47
	JFE (proposed)	**8.43**
x-vector	*Softmax*	2.26
	Gradient reversal	5.87
	Anti-loss	1.46
	JFE (proposed)	**1.07**

In addition, Table 4 show the performance comparison between the state-of-the-art embedding techniques for random digit strings speaker verification (i.e., *DNN i-vectors* and *Uncertainty normalized HMM/i-vector*) [35] and the x-vector-based embedding network trained with the proposed JFE scheme. As shown in the results, *Uncertainty normalized HMM/i-vector* performs better than the *x-vector (softmax)* by a large margin. This is mainly attributed to the fact that the *Uncertainty normalized HMM/i-vector* is trained to model the within-digit variability and scored with prior knowledge on the set of digits being uttered within the test set. Therefore it is not surprising that the *x-vector (softmax)* performs worse

TABLE 4. Gender-dependent EER (%) comparison between the speaker embedding vectors extracted from the x-vector-based embedding systems and the state-of-the-art i-vector-based systems.

Methods	EER [%]	
	Male	Female
x-vector (Softmax)	2.09	2.48
DNN i-vectors [36]	1.70	2.69
Uncertainty normalized HMM/i-vector [36]	1.52	1.77
x-vector (GRL)	3.75	4.17
x-vector (Anti-loss)	1.25	1.66
x-vector (JFE)	**0.82**	**1.29**

than the HMM/i-vector system, since it is trained and evaluated with no information on the context. However, despite the innate disadvantage of the x-vector framework in random digits strings speaker verification, the proposed *x-vector (JFE)* outperformed the *Uncertainty normalized HMM/i-vector* with an relative improvement of 46.05% in terms of male trial EER.

6) DEVICE DISENTANGLEMENT IN DOMAIN-MISMATCH SCENARIO

In this experiment, we compared the performance of the conventional x-vector and the proposed JFE system in a cross-domain text-independent speaker verification scenario. More specifically, both embedding systems were trained using the entire RSR2015 dataset and evaluated on the VoxCeleb1 evaluation subset, which is a dataset collected from Youtube videos recorded from a wide variety of channel and environmental conditions (e.g., videos shot on hand-held devices, interviews from red carpets).

As depicted in Table 5, the embeddings extracted from systems trained with RSR2015 showed severe performance degradation. Such degradation was likely caused by the vast variety of channel and environmental conditions within the VoxCeleb1, which are known to cause high within-speaker variability of the extracted speaker embedding vectors. Although the RSR2015 dataset is recorded from multiple different devices, the number of recording devices is limited (i.e. 6 devices) and the speech samples are relatively noise-free since they were recorded in an office environment [26], [27]. Therefore training the embedding system using only the RSR2015 dataset may be insufficient to tackle the challenging condition of the VoxCeleb1 evaluation set. Hence the *x-vector* system trained only for speaker discrimination using RSR2015 showed a relative decrement of 94.83% in terms of EER compared to the network trained with the VoxCeleb1 training set. On the other hand, the degredation of the *JFE* system trained to disentangle the device factor from the speaker embedding was 71.55%, which outperformed the *x-vector* trained with the same dataset with a relative improvement of 11.95%. This indicates that even in a domain-mismatch scenario, the proposed *JFE* is able to alleviate the performance degradation caused by recording device variability.

TABLE 5. EER (%) comparison between the speaker embedding vectors extracted from the proposed joint factor embedding and the conventional x-vector framework evaluated on the VoxCeleb1 evaluation set.

Objective	Training data	EER [%]
x-vector (softmax)	VoxCeleb1	11.6
	RSR2015	22.6
x-vector (JFE)	RSR2015	19.9

B. EMOTION DISENTANGLEMENT

Emotion variability can cause severe performance degradation in speaker recognition [36], but emotion disentanglement has not been investigated as much as other nuisance attributes, such as noise or channel distortion. This may be due to the challenging nature of emotion disentanglement since unlike noise or channel, emotional variability is caused by the speaker's vocal tract, which also creates speaker variability. In this subsection, we apply the proposed JFE framework for disentangling the variability induced by the speaker's emotional state.

1) DATASET

In order to evaluate the performance of the proposed technique for emotion disentanglement, a set of experiments were conducted based on the VoxCeleb1 dataset [28] and the emotion labels provided by the EmoVoxCeleb teacher system [37][1]. For training the embedding networks, we used the *development* subset of the VoxCeleb1 dataset, consisting of 148,642 utterances collected from 1,211 speakers. According to the emotion labels in EmoVoxCeleb, total 8 emotions are observed in the VoxCeleb1 dataset (i.e., neutral, happy, surprise, sad, angry, disgust, fear, contempt).

The evaluation was performed according to the original VoxCeleb1 trial list, which consists of 4,874 utterances spoken by 40 speakers. The duration of the trial utterances was between 3.97 seconds and 69.05 seconds.

2) EXPERIMENTAL SETUP

The acoustic features used in the experiments were 30-dimensional MFCCs extracted at every 10 ms, using a 20 ms Hamming window. The embedding networks were trained with segments consisting of 250 frames, using the ADAM optimization technique.

For the baseline x-vector framework and joint factor embedding system, 5 TDNN layers were used as the frame-level network according to the Kaldi x-vector recipe [4]. The TDNN outputs are aggregated as described in (1), and fed into the utterance-level classification network (i.e. speaker and emotion identifier). Each utterance-level classification network consisted of two 512-dimensional LeakyReLU hidden layers and a softmax output layer where the output size corresponds to the number of speakers or emotions within the training set. All the experimented networks were trained with learning rate 0.001 and batch size 256 for 74,321 iterations. Cosine similarity was used for computing the verification scores in the experiments.

3) COMPARISON BETWEEN THE JOINT FACTOR EMBEDDING SCHEME AND CONVENTIONAL EMBEDDING TECHNIQUES

In this experiment, we compare the embedding vectors obtained from the proposed joint factor embedding scheme and the conventional x-vector framework along with techniques reported in recent studies including VGG-M,

[1]The emotion labels provided by the EmoVoxCeleb teacher system can be downloaded from here: http://www.robots.ox.ac.uk/ vgg/research/cross-modal-emotions/.

TABLE 6. EER (%) comparison between the speaker embedding vectors extracted from the proposed joint factor embedding and the conventional methods. In the Data Augmentation column, X indicates embedding network trained with no augmentation and O indicates network trained with augmented training set.

Methods	Scoring	Data augmentation	EER [%]
i-vector [39]	PLDA	X	8.8
VGG [39]	Cosine similarity	X	7.8
Generalized end-to-end [40]	Cosine similarity	X	10.7
All-speaker hard negative mining end-to-end [40]	Cosine similarity	X	5.6
x-vector (softmax) [39]	Cosine similarity	X	11.3
	PLDA	X	7.1
		O	6.0
x-vector (our implementation)	PLDA	O	4.9
CNN-embedding [39]	Cosine similarity	X	7.3
	PLDA	X	5.9
		O	5.3
x-vector (JFE)	Cosine similarity	X	**6.8**
	PLDA	X	**5.4**
		O	**4.4**

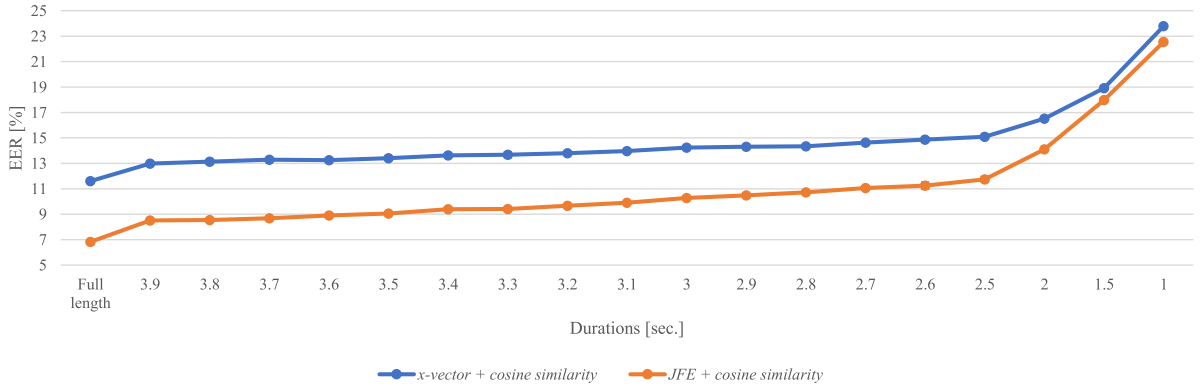

FIGURE 8. EER performance of the proposed joint factor embedding scheme and conventional x-vector on different duration utterances.

ResNet-34 and end-to-end verification systems [38], [39]. The experimented methods are as follows:

- *i-vector* [38]: the i-vector performance reported in [38],
- *VGG* [38]: the performance of the embedding extracted from VGG-M, which is a CNN architecture known to perform well on image and speaker classification, reported in [38],
- *Generalized end-to-end* [39]: the performance of the ResNet-34-based end-to-end speaker verification system trained with the generalized end-to-end loss (6) reported in [38],
- *All-speaker hard negative mining end-to-end* [39]: the performance of the ResNet-34-based end-to-end speaker verification system trained with the all-speaker hard negative mining loss, which is a modified version of the softmax loss for robust verification, reported in [38],
- *x-vector (softmax)* [38]: the x-vector performance reported in [38],
- *x-vector (our implementation)*: the performance of our implementation of *x-vector (softmax)*,
- *CNN-embedding* [38]: the performance of the embedding extracted from a CNN-based architecture reported in [38],
- *x-vector (JFE)*: the performance of the speaker embedding extracted from the proposed JFE system trained to disentangle the emotional factor using loss functions (14)–(18).

As shown in Table 6, the proposed *JFE* outperformed the conventional methods with both cosine similarity and PLDA backends. Especially when using PLDA as backend, the *JFE* achieved a relative improvement of 8.16% compared to the *x-vector (our implementation)* in terms of EER. Moreover, training the *JFE* with augmented training data described in [38] (i.e., noise and reverberation augmentation) further improved the performance. The results demonstrate that although the proposed *JFE* is composed of a simple x-vector-like network, it can provide embedding with higher speaker discriminative information than the systems with more complicated architecture.

In addition, we evaluated the conventional x-vector framework and the proposed joint factor embedding scheme on short duration speech samples. Each evaluation was done using randomly truncated trial utterances and the average EERs computed over three evaluations for each duration group are depicted in Fig. 8. As shown in the results, both the performance of the joint factor embedding framework

and the conventional x-vector were degraded as the duration decreased. This may be due to the lack of phonetically informative frames since a critical amount of speaker relevant information is contained in the phonetic characteristics [40]. However, the emotion disentangled speaker embedding obtained by the proposed *JFE* outperformed the conventional *x-vector* even with short duration speech segments.

V. CONCLUSION

In this paper, a novel approach for extracting an embedding vector robust to variability caused by nuisance attributes for speaker verification is proposed. In order to disentangle the nuisance variability from the speaker embedding vector, we introduce a JFE scheme where two types of embedding vectors are extracted, each dependent solely on the speaker or nuisance attribute, respectively. The proposed JFE network is trained simultaneously with the speaker and nuisance attribute classification networks where the speaker and nuisance embedding vectors are optimized to have good discriminability for their main task while having high uncertainty on their subtask.

To evaluate the performance of the embedding vector extracted from the proposed system in a realistic scenario, we conducted a set of speaker verification experiments using the RSR2015 dataset, which is composed of utterances recorded using multiple different hand-held devices, and VoxCeleb1 dataset, which is composed of various emotional speech utterances. From the results, it is shown that the proposed JFE scheme is capable of obtaining speaker embedding vectors with high speaker discriminability while showing robustness to channel and emotional variability. Moreover, we observed that the proposed embedding vector performs better than the conventional embedding technique with short duration speech segments.

Although the proposed technique showed great improvement over the conventional methods, since the proposed JFE is trained in a fully supervised manner, it requires labels for not only the speakers but also the nuisance attributes. Thus in our future study, we will expand the JFE technique to disentangle the non-speaker variability without the supervision of nuisance attribute labels. Moreover, we will improve the disentanglement performance by using more sophisticated methods for reducing the mutual information between the speaker and nuisance embedding vectors, rather than using a simple MAPC regularization.

REFERENCES

[1] J. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal Process. Mag.*, vol. 32, no. 6, pp. 74–99, Oct. 2015.
[2] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 4, pp. 788–798, May 2011.
[3] P. Kenny, "A small footprint i-vector extractor," in *Proc. Odyssey*, 2012, pp. 1–25.
[4] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5329–5333.
[5] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 4080–4084.
[6] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech*, Aug. 2017, pp. 999–1003.
[7] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 5115–5119.
[8] F. A. Rezaur rahman Chowdhury, Q. Wang, I. L. Moreno, and L. Wan, "Attention-based models for text-dependent speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5359–5363.
[9] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4879–4883.
[10] W. H. Kang and N. S. Kim, "Unsupervised learning of total variability embedding for speaker verification with random digit strings," *Appl. Sci.*, vol. 9, no. 8, p. 1597, Apr. 2019.
[11] W. H. Kang and N. S. Kim, "Adversarially learned total variability embedding for speaker recognition with random digit strings," *Sensors*, vol. 19, no. 21, p. 4709, Oct. 2019.
[12] Z. Meng, Y. Zhao, J. Li, and Y. Gong, "Adversarial speaker verification," in *Proc. Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6216–6220.
[13] D. Ramos, J. Gonzalez-Rodriguez, J. Gonzalez-Dominguez, and J. J. Lucena-Molina, "Addressing database mismatch in forensic speaker recognition with Ahumada III: A public real-casework database in Spanish," in *Proc. Interspeech*, 2008, pp. 1493–1496.
[14] X. Wang, L. Li, and D. Wang, "VAE-based domain adaptation for speaker verification," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Nov. 2019, pp. 535–539.
[15] P. S. Nidadavolu, S. Kataria, J. Villalba, and N. Dehak, "Low-resource domain adaptation for speaker recognition using cycle-gans," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2019, pp. 710–717.
[16] X. Fang, L. Zou, J. Li, L. Sun, and Z.-H. Ling, "Channel adversarial training for cross-channel text-independent speaker recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6221–6225.
[17] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 59, pp. 1–35, 2016.
[18] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by back-propagation," in *Proc. ICML*, 2015, pp. 1180–1189.
[19] Y. Shinohara, "Adversarial multi-task learning of deep neural networks for robust speech recognition," in *Proc. Interspeech*, Sep. 2016, pp. 2369–2372.
[20] A. Tripathi, A. Mohan, S. Anand, and M. Singh, "Adversarial learning of raw speech features for domain invariant speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5959–5963.
[21] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Process. Lett.*, vol. 22, no. 10, pp. 1671–1675, 2015.
[22] J. Zhou, T. Jiang, L. Li, Q. Hong, Z. Wang, and B. Xia, "Training multi-task adversarial network for extracting noise-robust speaker embedding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6196–6200.
[23] W. Shang, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 2672–2680.
[24] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," in *Proc. ICLR*, 2017, pp. 1–17.
[25] O. Morgen, "Representation learning for natural language," Ph.D. dissertation, Dept. Comput. Sci. Eng., Univ. Gothenburg, Gothenburg, Sweden, 2018.
[26] A. Larcher, K. A. Lee B. Ma, and H. Li, "The RSR2015: Database for text-dependent speaker verification using multiple pass-phrases," in *Proc. Interspeech*, 2012, pp. 2–5.
[27] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Text-dependent speaker verification: Classifiers, databases and RSR2015," *Speech Commun.*, vol. 60, pp. 56–77, May 2014.

[28] A. Nagrani, J. Son Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," 2017, *arXiv:1706.08612*. [Online]. Available: http://arxiv.org/abs/1706.08612
[29] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.
[30] J. T. Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," in *Proc. ICLR*, 2016, pp. 1–20.
[31] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. Interspeech*, 2014, pp. 338–342.
[32] M. Abadi *et al.*, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software. [Online]. Available: tensorflow.org
[33] D. P. Kingma and J. L. Ba, "ADAM, a method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–15.
[34] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[35] N. Maghsoodi, H. Sameti, H. Zeinali, and T. Stafylakis, "Speaker recognition with random digit strings using uncertainty normalized HMM-based i-Vectors," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 11, pp. 1815–1825, Nov. 2019.
[36] L. Chen and Y. Yang, "Emotional speaker recognition based on model space migration through translated learning," in *Proc. CCBR*, 2013, pp. 394–401.
[37] S. Albanie, A. Nagrani, A. Vedaldi, and A. Zisserman, "Emotion recognition in speech using cross-modal transfer in the wild," in *Proc. ACM Multimedia*, 2018, pp. 292–301.
[38] S. Shon, H. Tang, and J. Glass, "Frame-level speaker embeddings for text-independent speaker recognition and analysis of end-to-end model," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2018, pp. 1007–1013.
[39] H.-S. Heo, J.-W. Jung, I.-H. Yang, S.-H. Yoon, H.-J. Shim, and H.-J. Yu, "End-to-end losses based on speaker basis vectors and all-speaker hard negative mining for speaker verification," in *Proc. Interspeech*, Sep. 2019, pp. 2–16.
[40] T. Hasan, R. Saeidi, J. H. L. Hansen, and D. A. van Leeuwen, "Duration mismatch compensation for i-vector based speaker recognition systems," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 7663–7667.

WOO HYUN KANG (Student Member, IEEE) was born in Seoul, South Korea, in 1990. He received the B.S. degree in electronics engineering from Kookmin University, Seoul, in 2014. He is currently pursuing the Ph.D. degree in electrical engineering and computer science with Seoul National University (SNU).

His research interests include speaker recognition, machine learning, and signal processing.

SUNG HWAN MUN (Student Member, IEEE) was born in Incheon, South Korea, in 1993. He received the B.S. degree in electronics engineering from Inha University, Incheon, in 2017. He is currently pursuing the Ph.D. degree in electrical engineering and computer science with Seoul National University (SNU).

His research interests include speaker recognition, machine learning, and signal processing.

MIN HYUN HAN (Student Member, IEEE) was born in Seoul, South Korea, in 1992. He received the B.S. degree in electrical & electronic engineering from Yonsei University, Seoul, in 2018. He is currently pursuing the Ph.D. degree in electrical engineering and computer science with Seoul National University (SNU).

His research interests include speaker recognition, machine learning, and signal processing.

NAM SOO KIM (Senior Member, IEEE) received the B.S. degree in electronics engineering from Seoul National University (SNU), Seoul, South Korea, in 1988, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology, in 1990 and 1994, respectively.

From 1994 to 1998, he was a Senior Member of Technical Staff with the Samsung Advanced Institute of Technology. Since 1998, he has been with the School of Electrical Engineering, SNU, where he is currently a Professor. His research interests include speech signal processing, speech recognition, speech/audio coding, speech synthesis, adaptive signal processing, machine learning, and mobile communication.