Disentangling Speaker and Content in Pre-trained Speech Models with Latent Diffusion for Robust Speaker Verification

Zhe Li[1], *Man-Wai Mak*[1], *Jen-Tzung Chien*[2], *Mert Pilanci*[3], *Zezhong Jin*[1], *Helen Meng*[4]

[1]Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University
[2]Institute of Electrical and Computer Engineering, National Yang Ming Chiao Tung University
[3]Department of Electrical Engineering, Stanford University
[4]Systems Engineering & Engineering Management, The Chinese University of Hong Kong

lizhe.li@connect.polyu.hk

Abstract

Disentangled speech representation learning for speaker verification aims to separate spoken content and speaker timbre into distinct representations. However, existing variational autoencoder (VAE)–based methods for speech disentanglement rely on latent variables that lack semantic meaning, limiting their effectiveness for speaker verification. To address this limitation, we propose a diffusion-based method that disentangles and separates speaker features and speech content in the latent space. Building upon the VAE framework, we employ a speaker encoder to learn latent variables representing speaker features while using frame-specific latent variables to capture content. Unlike previous sequential VAE approaches, our method utilizes a conditional diffusion model in the latent space to derive speaker-aware representations. Experiments on the VoxCeleb datasets demonstrate that our method effectively isolates speaker features from speech content using pre-trained speech representations.

Index Terms: Disentanglement, diffusion models, VAE, pre-trained speech models, speaker verification

1. Introduction

A speech signal is represented as a one-dimensional waveform. Despite its apparent simplicity, a speech waveform encodes a wealth of high-level information such as phonemes, tone, emotion, gender, and speaker identity. However, attributes like speaking style, prosody, recording conditions, and noise levels are challenging to annotate [1, 2, 3]. To extract accurate speaker representations, existing methods employ phonetic content representations as a reference for speaker embeddings. Specifically, these methods include: (1) leveraging pre-trained automatic speech recognition (ASR) models [4, 5, 6] and (2) utilizing jointly trained multi-task models with additional modules for content representation [7, 8]. These approaches demonstrate that incorporating content representations enhances speaker recognition performance.

However, both strategies face limitations in practical applications. Pre-trained ASR models significantly increase model size and computational complexity during inference. Meanwhile, joint training with additional modules requires either a separate dataset with both text labels and speaker identities or a unified dataset containing both, which is often costly and challenging to obtain.

Disentangled representations have garnered considerable attention in recent research due to their ability to capture distinct variations in data generation. These variations often carry semantic meaning, facilitating the removal of irrelevant factors and reducing sample complexity for downstream learning tasks. In speaker verification, an ideal disentangled representation can isolate time-invariant features (e.g., speaker characteristics) from dynamic information (e.g., speech content). Moreover, downstream tasks such as speech recognition and speaker classification can benefit significantly from these representations by utilizing the separated components to improve representation learning.

Recent studies have investigated disentangled representation learning through variational autoencoders (VAEs) [9, 10] and generative adversarial networks (GANs). Models such as SpeechTripleNet [11], AnnealVAE [12], and JointVAE [13] set channel capacity for distinct latent variables to promote disentanglement. InfoGAN [14] divided the latent space and incorporated a mutual information regularization term into the standard GAN loss to enhance disentanglement. Similarly, Mathieu *et al.* [15] partitioned the encoding space into style and content components, employing adversarial training to encourage data points within the same class to share content representations while maintaining diverse style features.

Denoising diffusion models have recently demonstrated superior performance in disentangled representation learning, offering more stable training and higher representation fidelity compared to GANs and VAE-based models. Diffusion models address the challenge of representing complex, high-dimensional probability distributions by decomposing the problem into T incremental steps. At each step, the model transforms the noise data from a simpler distribution (e.g., the simplest Gaussian prior at $t = T$) to a more complex one (e.g., the real data distribution at $t = 0$). This iterative inference and denoising paradigm enables the model to map a simple distribution to a complex one through gradual refinement over many steps. However, the latent variables produced by diffusion models often lack high-level semantics and other desirable properties, such as speaker features.

To overcome the aforementioned limitations, we condition a denoising diffusion implicit model (DDIM) [16] on speaker features and propose a disentangled sequential model that leverages the capabilities of the DDIM to learn multi-level representations. Specifically, we employ a learnable speaker encoder to capture utterance-level speaker characteristics while the DDIM decodes and models the Gaussian variations in data. A latent vector represents the speaker features, and additional frame-level latent vectors capture dynamic information such as speech content. The DDIM's forward and generative processes are conducted within the joint latent space of speaker features and speech content. We applied our proposed disentanglement method to the speech representations generated by the pre-trained models WavLM [17] and HuBERT [18]. Experiments conducted on the speaker verification datasets VoxCeleb demonstrate that our method can effectively extract accurate speaker embeddings.

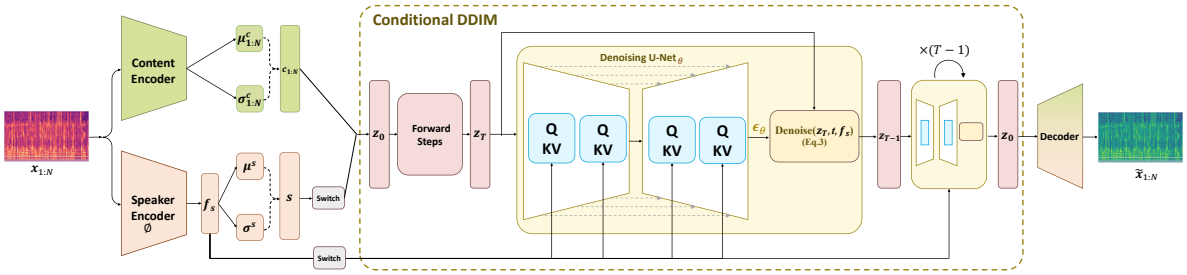


Figure 1: *The autoencoder comprises a speaker encoder, a content encoder, a conditional DDIM, and a speech decoder. The speaker encoder utilizes an ECAPA-TDNN [19] to transform the input speech* $\boldsymbol{x}_{1:N}$ *into a speaker representation* $\boldsymbol{f}_s$*, which is further transformed to* $\boldsymbol{\mu}^s$ *and* $\boldsymbol{\sigma}^s$ *through two linear heads. Similarly,* $\boldsymbol{\mu}^c_{1:N}$ *and* $\boldsymbol{\sigma}^c_{1:N}$ *can be obtained from a long short-term memory (LSTM) network with two linear heads. The "Switch" module changes the dimension of input vectors. For notational simplicity, we use the same symbols before and after the change of dimension. The dotted brace represents Gaussian sampling, which is performed by a reparameterization trick [20]. A conditional DDIM that serves as both a stochastic encoder* $\boldsymbol{z}_0 \rightarrow \boldsymbol{z}_T$ *and a deterministic decoder* $\boldsymbol{z}_{t-1} = Denoise(\boldsymbol{z}_t, \boldsymbol{f}_s, t)$*.* $\boldsymbol{z}_0 \in \mathbb{R}^{2D\times N}$*, where* D *is the dimension of* $\boldsymbol{c}_i$ *and* $\boldsymbol{s}$*. Similarly,* $\boldsymbol{z}_T, \boldsymbol{z}_{T-1}, \ldots, \boldsymbol{z}_1$ *have the same dimensions as* $\boldsymbol{z}_0$*.*

2. Methodology

We denote the input speech sequence as $\boldsymbol{x}_{1:N} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)$, where $\boldsymbol{x}_i$ is the filter-bank feature vector corresponding to the i-th frame, and N is the number of frames in the sequence. To facilitate an informative global latent representation $\boldsymbol{z}_T$ for the decoding process, we introduce a conditional DDIM decoder, represented by $p_\theta(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t, \boldsymbol{f}_s)$. This decoder is conditioned on an auxiliary latent variable $\boldsymbol{f}_s$ obtained through a speaker encoder $\boldsymbol{f}_s = \text{Enc}_\phi(\boldsymbol{x}_{1:N})$, which maps the entire input sequence $\boldsymbol{x}_{1:N}$ to speaker representation $\boldsymbol{f}_s$. $\boldsymbol{f}_s$ is fed into two linear heads to produce the mean vector $\boldsymbol{\mu}_s$ and standard deviation vector $\boldsymbol{\sigma}_s$. Then, the speaker vector $\boldsymbol{s}$ is obtained by sampling the Gaussian distribution defined by these mean and standard deviation vectors. The module "Switch" in Fig. 1 changes the dimension of $\boldsymbol{s}$ by repeating it N times so that the resulting matrix can be concatenated with $\boldsymbol{c}_{1:N}$. We refer to the network in Fig. 1 as **D**isentengled **L**atent **D**iffusion–**A**uto**E**ncoder (DLD–AE).

2.1. Speaker Encoder

We utilize an ECAPA-TDNN model [19] to transform the input speech sequence $\boldsymbol{x}_{1:N}$ to a representative vector $\boldsymbol{f}_s$. This vector captures crucial speaker information for the DDIM decoder (the yellow boxes in Fig. 1), expressed as $p_\theta(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t, \boldsymbol{f}_s)$, to perform the denoising process and predict the output latent vector $\tilde{\boldsymbol{z}}_0 \equiv (\boldsymbol{z}_0, \boldsymbol{f}_s)$. By conditioning DDIM on an enriched information vector $\boldsymbol{f}_s$, we enhance the efficiency and accuracy of the denoising operation, ultimately leading to a more reliable generation of latent representations.

2.2. Content Encoder

We employ an LSTM with two linear heads as the content encoder to transform $\boldsymbol{x}_{1:N}$ into $\boldsymbol{c}_{1:N}$, where $\boldsymbol{c}_i$ denotes the dynamic state learned at frame i. We assume that each $\boldsymbol{c}_i$ depends on the preceding dynamic variables, denoted as $\boldsymbol{c}_{<i} \equiv \{\boldsymbol{c}_0, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_{i-1}\}$, with $\boldsymbol{c}_0 = \boldsymbol{0}$.

2.3. Reverse Diffusion Process

Our proposed conditional DDIM's reverse process utilizes the input $\tilde{\boldsymbol{z}}_{t-1} \equiv (\boldsymbol{z}_t, \boldsymbol{f}_s)$, which comprises the DDIM encoder's output and speaker representation, to generate an output latent vector. Using a denoising U-Net, each block of the conditional DDIM decoder models the probability distribution $p_\theta(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t, \boldsymbol{f}_s)$:

$$p_\theta(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t, \boldsymbol{f}_s) = \begin{cases} \mathcal{N}(\boldsymbol{z}_{t-1}; f_\theta(\boldsymbol{z}_1, 1, \boldsymbol{f}_s), \sigma_1^2 \boldsymbol{I}) & \text{if } t = 1, \\ q_\sigma(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t, f_\theta(\boldsymbol{z}_t, t, \boldsymbol{f}_s)) & \text{otherwise} \end{cases}, \tag{1}$$

where σ_1 is set to 0. Following the approach in Song *et al.* [16], the inference distribution q_σ in Eq. 1 is defined as follows:

$$q_\sigma = \mathcal{N}\Bigg(\boldsymbol{z}_{t-1}; \sqrt{\alpha_{t-1}} f_\theta(\boldsymbol{z}_t, t, \boldsymbol{f}_s) + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \frac{\boldsymbol{z}_t - \sqrt{\alpha_t} f_\theta(\boldsymbol{z}_t, t, \boldsymbol{f}_s)}{\sqrt{1 - \alpha_t}}, \sigma_t^2 \boldsymbol{I}\Bigg) \tag{2}$$

where σ_t is set to 0. We implement f_θ in Eqs. 1 and 2 using a noise prediction network $\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t, \boldsymbol{f}_s)$:

$$f_\theta(\boldsymbol{z}_t, t, \boldsymbol{f}_s) \equiv \text{Denoise}(\boldsymbol{z}_t, t, \boldsymbol{f}_s) = \frac{\boldsymbol{z}_t - \sqrt{1 - \alpha_t}\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t, \boldsymbol{f}_s)}{\sqrt{\alpha_t}}, \tag{3}$$

where $\boldsymbol{\epsilon}_\theta$ is implemented by a U-Net as shown in Fig 1.

The training process involves optimizing the $\mathcal{L}_{DDIM}$ loss with respect to parameters θ and ϕ:

$$\mathcal{L}_{DDIM} = \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{z}_0, \boldsymbol{\epsilon}_t} \left[\| \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t, \boldsymbol{f}_s) - \boldsymbol{\epsilon}_t \|_2^2 \right], \tag{4}$$

where $\boldsymbol{f}_s = \text{Enc}_\phi(\boldsymbol{x}_{1:N})$, $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, $\boldsymbol{z}_t = \sqrt{\alpha_t}\boldsymbol{z}_0 + \sqrt{1 - \alpha_t}\boldsymbol{\epsilon}_t$, and T is an integer, e.g., 100. Note that this simplified loss function optimizes the DDIM but does not optimize the actual variational lower bound.

2.4. Disentangled Sequential Variational Autoencoder

We define the global latent representation as $\boldsymbol{z}_0 \equiv (\boldsymbol{s}, \boldsymbol{c}_{1:N})$. Our formulation is based on the intuition that sequence variations can be decomposed into time-dependent dynamic components $\{\boldsymbol{c}_i\}$ and a static component $\boldsymbol{s}$. We assume independence between the static variable $\boldsymbol{s}$ and the dynamic variables $\boldsymbol{c}_{1:N}$, implying $p(\boldsymbol{z}_0) = p(\boldsymbol{s}, \boldsymbol{c}_{1:N}) = p(\boldsymbol{s})p(\boldsymbol{c}_{1:N})$. The static component remains constant across all frames within a given utterance but differs across different utterances.

In a speech signal, the phonetic transcription governs the movement of the vocal tract and the produced sounds over time, while the speaker's identity remains fixed throughout an utterance. Based on these assumptions, we derive the following complete likelihood [21]:

$$\begin{aligned} p(\boldsymbol{x}_{1:N}, \boldsymbol{z}_0) &= p(\boldsymbol{x}_{1:N}, \boldsymbol{s}, \boldsymbol{c}_{1:N}) \\ &= p(\boldsymbol{s}, \boldsymbol{c}_{1:N}) p(\boldsymbol{x}_{1:N} | \boldsymbol{s}, \boldsymbol{c}_{1:N}) \\ &= p(\boldsymbol{s}) \left[\prod_{i=1}^{N} p(\boldsymbol{c}_i | \boldsymbol{c}_{<i}) p(\boldsymbol{x}_i | \boldsymbol{s}, \boldsymbol{c}_i) \right]. \end{aligned} \tag{5}$$

We define $p(\boldsymbol{s})$ in Eq. 5 as a standard Gaussian $\mathcal{N}(\boldsymbol{s}; \boldsymbol{0}, \boldsymbol{I})$. We assume that $p(\boldsymbol{c}_i | \boldsymbol{c}_{<i})$ follows a Gaussian distribution:

$$p(\boldsymbol{c}_i | \boldsymbol{c}_{<i}) = \mathcal{N}(\boldsymbol{c}_i; \boldsymbol{\mu}_i(\boldsymbol{c}_{<i}), \mathrm{diag}((\boldsymbol{\sigma}_i(\boldsymbol{c}_{<i}))^2)), \tag{6}$$

where $\boldsymbol{\mu}_i(\cdot)$ and $\boldsymbol{\sigma}_i(\cdot)$ can be modeled by an LSTM followed by two linear heads. Since both $\boldsymbol{\mu}_i(\cdot)$ and $\boldsymbol{\sigma}_i(\cdot)$ are conditioned on the temporal context $\boldsymbol{c}_{<i}$, their derivation at frame i requires access to the history $\boldsymbol{c}_{<i}$. To sample $\boldsymbol{c}_i$ from $p(\boldsymbol{c}_i | \boldsymbol{c}_{<i})$, $\boldsymbol{c}_{i-1}$ is first passed through the LSTM cells to forward one step, generating $\boldsymbol{\mu}_i(\cdot)$ and $\boldsymbol{\sigma}_i(\cdot)$ via linear transformation layers. The reparameterization trick is then applied to draw a sample from the resulting distribution [22, 23].

To derive latent representations solely from the observed data $\boldsymbol{x}_{1:N}$, where speaker characteristics and content are entangled, we aim to learn a posterior distribution $q(\boldsymbol{z}_0 | \boldsymbol{x}_{1:N})$ that disentangles these two components. Specifically, we use variational inference:

$$\begin{aligned} q(\boldsymbol{z}_0 | \boldsymbol{x}_{1:N}) &= q(\boldsymbol{c}_{1:N}, \boldsymbol{s} | \boldsymbol{x}_{1:N}) \\ &= q(\boldsymbol{c}_{1:N} | \boldsymbol{x}_{1:N}) q(\boldsymbol{s} | \boldsymbol{x}_{1:N}) \\ &= q(\boldsymbol{s} | \boldsymbol{x}_{1:N}) \prod_{i=1}^{N} q(\boldsymbol{c}_i | \boldsymbol{c}_{<i}, \boldsymbol{x}_{1:N}). \end{aligned} \tag{7}$$

The speaker latent posterior follows a Gaussian distribution:

$$q(\boldsymbol{s} | \boldsymbol{x}_{1:N}) = \mathcal{N}\left(\boldsymbol{s}; \boldsymbol{\mu}^s(\boldsymbol{x}_{1:N}), \mathrm{diag}\{(\boldsymbol{\sigma}^s(\boldsymbol{x}_{1:N}))^2\}\right), \tag{8}$$

where the mean and standard deviation vectors, $\boldsymbol{\mu}^s(\cdot)$ and $\boldsymbol{\sigma}^s(\cdot)$, are modeled by an ECAPA-TDNN [19] with two linear layers. Similarly, we define:

$$q(\boldsymbol{c}_i | \boldsymbol{c}_{<i}, \boldsymbol{x}_{1:N}) = \mathcal{N}\left(\boldsymbol{c}_i; \boldsymbol{\mu}_i^c(\boldsymbol{x}_{1:N}, \boldsymbol{c}_{<i}), \mathrm{diag}\{(\boldsymbol{\sigma}_i^c(\boldsymbol{x}_{1:N}, \boldsymbol{c}_{<i}))^2\}\right), \tag{9}$$

where $\boldsymbol{\mu}_i^c(\cdot)$ and $\boldsymbol{\sigma}_i^c(\cdot)$ are obtained by passing the inputs $\boldsymbol{c}_{<i}$ and $\boldsymbol{x}_{1:N}$ through bidirectional LSTMs, followed by an RNN and two linear layers. The reparameterization trick is applied to sample $\boldsymbol{s}$ and $\{\boldsymbol{c}_i\}_{i=1}^N$.

Previous studies have introduced similar parameterizations of dynamic variables through recurrent networks [22, 23]. The standard approach for learning latent representations is to maximize the evidence lower bound (ELBO) [9, 24]:

$$\max_{p,q} \mathbb{E}_{\boldsymbol{x}_{1:N} \sim p_{\mathrm{D}}(\boldsymbol{x}_{1:N})} \left[\underbrace{\mathbb{E}_{q(\boldsymbol{z}_0 | \boldsymbol{x}_{1:N})} \log p(\boldsymbol{x}_{1:N} | \boldsymbol{z}_0)}_{\text{reconstruction term}} - \underbrace{\mathrm{KL}[q(\boldsymbol{z}_0 | \boldsymbol{x}_{1:N}) \parallel p(\boldsymbol{z}_0)]}_{\text{prior matching term}} \right], \tag{10}$$

where $p_{\mathrm{D}}(\boldsymbol{x}_{1:N})$ represents the empirical data distribution and $\mathrm{KL}[\cdot||\cdot]$ denotes Kullback-Leibler (KL) divergence. Under the assumption that $\boldsymbol{s}$ and $\boldsymbol{c}_{1:N}$ are mutually independent in the posterior, the KL-divergence term is simplified as

$$\begin{aligned} &\mathrm{KL}[q(\boldsymbol{z}_0 | \boldsymbol{x}_{1:N}) \parallel p(\boldsymbol{z}_0)] = \\ &\mathrm{KL}[q(\boldsymbol{s} | \boldsymbol{x}_{1:N}) \parallel p(\boldsymbol{s})] + \mathrm{KL}[q(\boldsymbol{c}_{1:N} | \boldsymbol{x}_{1:N}) \parallel p(\boldsymbol{c}_{1:N})], \end{aligned} \tag{11}$$

where the second term is approximated using sampled trajectories of the dynamic variables $\boldsymbol{c}_{1:N}$.

We define the **d**isentangled **s**equential **v**ariational **a**uto**e**ncoder (**DSVAE**) loss as the negative ELBO of the log-likelihood:

$$\begin{aligned} \mathcal{L}_{DSVAE} = &- \mathbb{E}_{p_{\mathrm{D}}(\boldsymbol{x}_{1:N})} \mathbb{E}_{q(\boldsymbol{z}_0 | \boldsymbol{x}_{1:N})} \Big[\log p(\boldsymbol{x}_{1:N} | \boldsymbol{z}_0) \Big] \\ &+ \mathrm{KL}\big[q(\boldsymbol{s} | \boldsymbol{x}_{1:N}) \parallel p(\boldsymbol{s})\big] \\ &+ \mathrm{KL}[q(\boldsymbol{c}_{1:N} | \boldsymbol{x}_{1:N}) \parallel p(\boldsymbol{c}_{1:N})\big]. \end{aligned} \tag{12}$$

The first term in Eq. 12 corresponds to the reconstruction loss, while the next two terms correspond to the KL divergence between the posterior and prior distributions of the time-variant content embeddings $\{\boldsymbol{c}_i\}_{i=1}^N$ (Eq. 7 and 9) and the time-invariant speaker embeddings $\boldsymbol{s}$ (Eq. 8), respectively. Specifically, the reconstruction loss is computed through the mean squared error (MSE) between the decoder outputs and inputs. The KL divergence terms can be computed analytically as both the priors and posteriors of $\boldsymbol{s}$ and $\{\boldsymbol{c}_i\}_{i=1}^N$ are assumed to be Gaussian distributed [25].

2.5. Model Training

To ensure a meaningful condition for the speaker embedding $\boldsymbol{s}$, we optimize the speaker encoder using AAM-Softmax [26]. To train the network, we define the total loss: the AAM-Softmax [26], DDIM loss (Eq. 4), and DSVAE loss (Eq. 12). The last one can be treated as regularization. The combination can be implemented as follows:

$$\mathcal{L}_{DLDAE} = \mathcal{L}_{AAM\text{-}Softmax} + \mathcal{L}_{DDIM} + \lambda \mathcal{L}_{DSVAE}, \tag{13}$$

where λ is a hyperparameter that regulates the impact of sequential disentanglement. During inference, only the speaker encoder is used to extract speaker embeddings.

3. Experiments and Results

3.1. Implementation Details

We trained our method on VoxCeleb2-dev and evaluated on the VoxCeleb1 [27, 28] datasets for speaker verification. Features were extracted using HuBERT-Large [18] and WavLM-Large [17], enhanced with SpecAugment [29]. The speaker encoder was ECAPA-TDNN [19], and we applied four augmentation types (room impulse, music, noise, babble) with a 0.6 probability each. Training used 3-second utterances and the batch size was 256. We employed AAM-Softmax [26] (margin=0.2, scale=30) and reduced the learning rate by 3% per epoch. Networks were optimized using Adam with CosineAnnealingWarmRestarts [30].

3.2. Comparing with Existing Methods

To evaluate the effectiveness of our proposed DLD–AE, we compare its performance with existing disentanglement techniques. As shown in Table 1, when using Fbank features, our DLD–AE (Row 3) outperforms the baseline ECAPA-TDNN (Row 1) and achieves competitive results compared to RecXi (Row 2). This demonstrates the effectiveness of our disentanglement framework in improving speaker verification performance. The results also show that our disentanglement technique is particularly effective when applied to pre-trained features, including HuBERT and WavLM features. For example,

Table 1: *Performance of the baseline models and the proposed DLD–AE on VoxCeleb1-O, VoxCeleb1-E, and VoxCeleb1-H. All experiments used ECAPA-TDNN as the speaker encoder and were trained on VoxCeleb2-dev. Results were obtained without AS-Norm [31, 32] nor quality-aware score calibration [33]. For RecXi, the results are based on the setting RecXi($\tilde{\phi}$, $\tilde{\phi}_{lin}$) in [34].*

Row	Input Feature	Disentanglement Method	VoxCeleb1-O		VoxCeleb1-E		VoxCeleb1-H	
			EER(%)	minDCF	EER(%)	minDCF	EER(%)	minDCF
1	Fbank	None	1.12	0.145	1.25	0.142	2.43	0.239
2		RecXi + $\mathcal{L}_{\text{ssp}}$ [34]	1.19	0.107	1.29	0.141	2.46	0.227
3		DLD–AE (Ours)	1.01	0.101	1.28	0.153	2.35	0.213
4	HuBERT	None	0.91	0.119	0.99	1.146	2.35	0.252
5		DLD–AE (Ours)	0.88	0.088	**0.91**	1.011	2.05	0.231
6	WavLM	None	0.85	0.113	1.12	0.091	2.06	0.197
7		DLD–AE (Ours)	**0.78**	**0.081**	**0.91**	**0.090**	**1.83**	0.191

Table 2: *Ablation study on VoxCeleb1-O. DSVAE [9] incorporates AAM-Softmax.*

Row	Input Feature	Disentanglement Method	VoxCeleb1-O	
			EER(%)	minDCF
1	HuBERT	None	0.91	0.119
2		DSVAE [9]	0.90	0.093
3		DLD–AE (w/o condition)	0.89	0.090
4		DLD–AE (Ours)	0.88	0.088
5	WavLM	None	0.85	0.113
6		DSVAE [9]	0.83	0.094
7		DLD–AE (w/o condition)	0.81	0.084
8		DLD–AE (Ours)	0.78	0.081

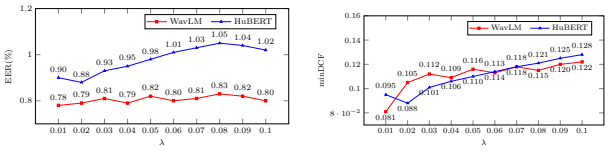
with the WavLM features, DLD–AE (Row 7) reduces the EER to 0.78% on VoxCeleb1-O, compared to 0.85% without disentanglement (Row 6). A similar trend is observed for minDCF. This improvement is attributed to our framework's ability to disentangle static speaker components, enhancing speaker recognition effectively. The improvement highlights the importance of modeling the dynamic contents in speech and disentangling the speaker and content representations.

3.3. Ablation Study

We conducted ablation experiments to investigate the importance of different components in the proposed DLD–AE. We also conducted experiments using DSVAE to perform the disentanglement, which is essentially a VAE-based disentanglement without the diffusion process. Results are shown in Table 2. Comparing Row 6 with Row 5 in Table 2 reveals that adding the VAE can slightly improve performance. However, a significant performance gain is observed when integrating the diffusion processes into the VAE (Row 7). The best performance is achieved when the diffusion processes are conditioned on the speaker embeddings (Row 8). The same conclusions are obtained regardless of which pre-trained models were used.

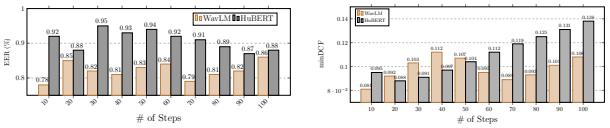
3.4. Impact of λ

The hyperparameter λ in Eq. 13 controls the extent of DLD–AE's contribution in the proposed framework. We analyze the impact of varying λ on SV performance. We selected λ ranging from 0.01 to 0.1, incrementing by 0.01 at each step. The results, shown in Fig. 2, indicate that for both EER and minDCF, when WavLM is used as the pre-trained model, the best performance is achieved at $\lambda = 0.01$. For HuBERT, the optimal result is observed at $\lambda = 0.02$, with performance declining as λ increases. These findings suggest that placing excessive emphasis on sequence decoupling may negatively impact the model's ability to learn discriminative speaker embeddings.



(a) *EER(%)* (b) *minDCF*

Figure 2: *Results on VoxCeleb1-O for different λ in Eq. 13, using WavLM Large and HuBERT Large as the PTMs.*



(a) *EER(%)* (b) *minDCF*

Figure 3: *Impact of diffusion steps on VoxCeleb1-O using WavLM-Large and HuBERT as pre-trained models.*

3.5. Impact of Diffusion Steps

In our work, we employ DDIM for diffusion and denoising, which substantially reduces the number of steps. Unlike standard DDPM, which often requires hundreds or even thousands of iterative steps, DDIM can generate high-quality samples in just a few dozen steps. This efficiency is achieved through an explicit inference process that reduces the random noise term, making each step more efficient and accurate. As illustrated in Fig. 3, the optimal performance is achieved with 10 steps, while for HuBERT, the best results are obtained using 20 steps.

4. Conclusions

This paper proposes a sequential disentanglement framework based on a latent diffusion model (DLD–AE) to decouple speaker traits from content factors, leveraging only speaker traits for speaker verification. Using WavLM and HuBERT as pre-trained models to extract frame-level features and the latent diffusion model for speaker-content disentanglement, our method achieves the best performance on the VoxCeleb1 test set. Experimental results demonstrate the effectiveness of incorporating sequential disentanglement with pre-trained models for extracting discriminative speaker embeddings.

5. Acknowledgment

This work was supported by the RGC of Hong Kong SAR, Theme-based Research Scheme (T45-407/19-N).

6. References

[1] A. Tjandra, R. Pang, Y. Zhang, and S. Karita, "Unsupervised learning of disentangled speech content and style representation," in *Proc. of Interspeech*, 2021, pp. 4089–4093.

[2] W.-N. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen *et al.*, "Hierarchical generative modeling for controllable speech synthesis," in *Proc. of International Conference on Learning Representations (ICLR)*, 2021.

[3] C.-X. Gan, M.-W. Mak, W. Lin, and J.-T. Chien, "Asymmetric clean segments-guided self-supervised learning for robust speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 11 081–11 085.

[4] D. Liao, T. Jiang, F. Wang, L. Li, and Q. Hong, "Towards a unified conformer structure: from asr to asv task," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[5] D. Cai, W. Wang, M. Li, R. Xia, and C. Huang, "Pretraining conformer with asr for speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[6] Z. Song, L. He, P. Wang, Y. Hu, and H. Huang, "Introducing multilingual phonetic information to speaker embedding for speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 10 091–10 095.

[7] T. Liu, R. K. Das, M. Madhavi, S. Shen, and H. Li, "Speaker-utterance dual attention for speaker and utterance verification," in *Proc. of Interspeech*, 2020, pp. 4293–4297.

[8] T. Liu, R. K. Das, K. A. Lee, and H. Li, "Neural acoustic-phonetic approach for speaker verification with phonetic attention mask," *IEEE Signal Processing Letters*, vol. 29, pp. 782–786, 2022.

[9] L. Yingzhen and S. Mandt, "Disentangled sequential autoencoder," in *Proc. of International Conference on Machine Learning (ICML)*, 2018, pp. 5670–5679.

[10] J. Bai, W. Wang, and C. P. Gomes, "Contrastively disentangled sequential variational autoencoder," *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 10 105–10 118, 2021.

[11] H. Lu, X. Wu, Z. Wu, and H. Meng, "Speechtriplenet: End-to-end disentangled speech representation learning for content, timbre and prosody," in *Proc. of ACM International Conference on Multimedia*, 2023, pp. 2829–2837.

[12] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in β-vae," *arXiv e-prints*, p. arXiv:1804.03599, 2018.

[13] E. Dupont, "Learning disentangled joint continuous and discrete representations," *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.

[14] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, vol. 29, 2016.

[15] M. F. Mathieu, J. J. Zhao, J. Zhao, A. Ramesh, P. Sprechmann, and Y. LeCun, "Disentangling factors of variation in deep representation using adversarial training," *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, vol. 29, 2016.

[16] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. of International Conference on Learning Representations (ICLR)*.

[17] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[18] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[19] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," in *Proc. of Interspeech*, 2020, pp. 3830–3834.

[20] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. of International Conference on Learning Representations (ICLR)*, 2014.

[21] Y. Tu, M.-W. Mak, and J.-T. Chien, "Contrastive self-supervised speaker embedding with sequential disentanglement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[22] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, 2015.

[23] A. Goyal, A. Sordoni, M.-A. Côté, N. R. Ke, and Y. Bengio, "Z-forcing: training stochastic recurrent networks," in *Proc. of the International Conference on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 6716–6726.

[24] Y. Zhu, M. R. Min, A. Kadav, and H. P. Graf, "S3VAE: Self-supervised sequential VAE for representation disentanglement and data generation," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 6538–6547.

[25] J. Duchi, "Derivations for linear algebra and optimization," *Berkeley, California*, vol. 3, no. 1, pp. 2325–5870, 2007.

[26] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4690–4699.

[27] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," *Proc. of Interspeech 2017*, pp. 2616–2620, 2017.

[28] J. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," *Proc. of Interspeech 2018*, 2018.

[29] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. of Interspeech*, 2019, pp. 2613–2617.

[30] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," in *Proc. of International Conference on Learning Representations (ICLR)*, 2022.

[31] Z. N. Karam, W. M. Campbell, and N. Dehak, "Towards reduced false-alarms using cohorts," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 4512–4515.

[32] S. Cumani, P. D. Batzu, D. Colibro, C. Vair, P. Laface, V. Vasilakakis *et al.*, "Comparison of speaker recognition approaches for real applications." in *Proc. of Interspeech*, 2011, pp. 2365–2368.

[33] J. Thienpondt, B. Desplanques, and K. Demuynck, "The idlab voxsrc-20 submission: Large margin fine-tuning and quality-aware score calibration in dnn based speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5814–5818.

[34] T. Liu, K. A. Lee, Q. Wang, and H. Li, "Disentangling voice and content with self-supervision for speaker recognition," *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, pp. 50 221–50 236, 2023.