Improved Disentangled Speech Representations Using Contrastive Learning in Factorized Hierarchical Variational Autoencoder

Yuying Xie, Thomas Arildsen, Zheng-Hua Tan
Department of Electronic Systems, Aalborg University, Denmark
yuxi@es.aau.dk, tari@its.aau.dk, zt@es.aau.dk

***Abstract*—Leveraging the fact that speaker identity and content vary on different time scales, factorized hierarchical variational autoencoder (FHVAE) uses different latent variables to symbolize these two attributes. Disentanglement of these attributes is carried out by different prior settings of the corresponding latent variables. For the prior of speaker identity variable, FHVAE assumes it is a Gaussian distribution with an utterance-scale varying mean and a fixed variance. By setting a small fixed variance, the training process promotes identity variables within one utterance gathering close to the mean of their prior. However, this constraint is relatively weak, as the mean of the prior changes between utterances. Therefore, we introduce contrastive learning into the FHVAE framework, to make the speaker identity variables gathering when representing the same speaker, while distancing themselves as far as possible from those of other speakers. The model structure has not been changed in this work but only the training process, thus no additional cost is needed during testing. Voice conversion has been chosen as the application in this paper. Latent variable evaluations include speaker verification and identification for the speaker identity variable, and speech recognition for the content variable. Furthermore, assessments of voice conversion performance are on the grounds of fake speech detection experiments. Results show that the proposed method improves both speaker identity and content feature extraction compared to FHVAE, and has better performance than baseline on conversion.**

***Index Terms*—disentangled representation learning, contrastive learning, voice conversion**

I. Introduction

Disentangled representation learning [1]–[3] tries to extract features for representing the independent data attributes separately. Even though unsupervised learning is a hot topic nowadays, paper [4] revealed that unsupervised disentangled representation learning is fundamentally impossible, and thus supervised or weakly supervised data are generally needed. Applications of disentangled representation learning are broad, and one in speech is voice conversion [5]–[11].

The basic strategy of using disentangled representation learning on voice conversion contains two stages: training and conversion. Training makes neural networks learn to extract features to represent speaker identity and content separately. To make the generated audio sound like target speaker but retain same content from source speaker utterance, the converted utterances are generated from the speaker embedding of the target speaker and the content embedding from the source speaker. Speaker embedding and content embedding should contain as much corresponding information as possible but no information overlap, so as to get desired conversion performance. This strategy has been applied extensively in recent work [8]–[11]. However, most work has requirements on data. Specifically, AutoVC [8] carried out embedding disentanglement under an auto-encoder framework, by controlling the bottleneck dimension carefully to allow only content information to pass through. Speaker embedding in this work is extracted from a pre-trained speaker encoder according to [12], and the segment length for extraction is between 3.5s to 4.5s. The authors of [9] use an encoder and a codebook for content embedding extraction, while speaker embedding is got as the difference between the encoder output and the codebook output. Trimming the silence of speech with a fixed threshold is an essential pre-processing in this work. The authors of [10] used a deterministic style encoder and a statistical content encoder to extract speaker embedding and content embedding separately. Contrastive predictive coding is used on style encoder output and the framework is trained end-to-end. The segment length is required between 2s and 4s.

Factorized hierarchical variational autoencoder (FHVAE) [5] assumes that speaker identity changes between sequences (longer than 200ms), while content varies between segments (200ms in experiments). Compared with other works, FHVAE neither requires lengthy utterances or pre-processing, nor a pre-trained model for extracting speaker embeddings. However, FHVAE has its own limitations. To equip sequential latent variables with representation capability, the prior adopts a Gaussian distribution with mean which changes between utterances and a fixed small variance which makes sequential latent variables are similar within the utterance. However, the relationship of sequential variables between utterances has not been considered.

Contrastive learning intends to make representations symbolising the same class become as similar as possible, while making representations symbolising different classes become as dissimilar as possible [13]–[15]. This idea fits speaker embedding learning, and there also exists some work using contrastive learning in speaker embedding extraction, like [9], [12].

Inspired by the idea behind contrastive learning and related work, this paper applies contrastive learning in the FHVAE

The work of Yuying Xie is supported by China Scholarship Council.

framework. Compared with FHVAE only concerning information within one utterance, contrastive learning brings cross-utterance information during training to enhance latent variable representation capability. The proposed method does not change the framework (and thus does not increase the model complexity during test), but the training strategy. Speech recognition and speaker verification are used to analysis the extracted latent variables and fake speech detection for converted utterances.

II. Factorized Hierarchical Variational Autoencoder

The nature of speech shows that speaker identity and content vary on different time scales: speaker identity changes between sequences, while content varies faster, within segments. To utilize this fact in speech signal disentangled representation, FHVAE assumes two variables follow a sequence-dependent prior and a sequence-independent prior respectively. Latent variables representing speaker identity and linguistic content in the graphical model are denoted sequential variable and segmental variable in the following.

Specifically, we denote the speech feature (e.g. log-magnitude spectrogram) dataset $\boldsymbol{D} = \{\boldsymbol{X}^{(i)}\}_{i=1}^{M}$, where i represents the i-th sequence in the dataset, and the dataset contains M utterances. Assume each utterance $\boldsymbol{X}^{(i)}$ contains $N^{(i)}$ segments: $\boldsymbol{X}^{(i)} = \{\boldsymbol{x}^{(i,n)}\}_{n=1}^{N^{(i)}}$. $z_1^{(i,n)}$ and $z_2^{(i,n)}$ denote segmental latent variable and sequential latent variable in order, and superscript (i, n) expresses that the variables are for the n-th segment of the i-th utterance. The following notation will omit the superscript (i) for simplicity as training happens on segment-scale. Besides, $p(\cdot)$ and $q(\cdot)$ represent prior and posterior distributions, while θ and ϕ represent parameters in the generative model and the inference model, respectively.

The generative model in FHVAE assumes data $\boldsymbol{X}$ for each sequence is generated in the following process: (1) latent variable $\boldsymbol{\mu}_2$ is drawn from prior $p_\theta(\boldsymbol{\mu}_2) = \mathcal{N}(\boldsymbol{\mu}_2|\boldsymbol{0}, \boldsymbol{I})$; (2) sequential latent variables $\{\boldsymbol{z}_2^{(n)}\}_{n=1}^{N}$ and segmental latent variables $\{\boldsymbol{z}_1^{(n)}\}_{n=1}^{N}$ are drawn from priors $p_\theta(\boldsymbol{z}_2|\boldsymbol{\mu}_2) = \mathcal{N}(\boldsymbol{z}_2|\boldsymbol{\mu}_2, \sigma_{\boldsymbol{z}_2}^2\boldsymbol{I})$ and $p_\theta(\boldsymbol{z}_1) = \mathcal{N}(\boldsymbol{z}_1|\boldsymbol{0}, \boldsymbol{I})$.

The inference model in FHVAE assumes all posteriors of latent variables: $q_\phi(\boldsymbol{\mu}_2^{(i)})$, $q_\phi(\boldsymbol{z}_2|\boldsymbol{x})$ and $q_\phi(\boldsymbol{z}_1|\boldsymbol{x}, \boldsymbol{z}_2)$ are Gaussian distributions. Means and variances of $q_\phi(\boldsymbol{z}_2|\boldsymbol{x})$ and $q_\phi(\boldsymbol{z}_1|\boldsymbol{x}, \boldsymbol{z}_2)$ are from the neural network, while the mean $\widetilde{\boldsymbol{\mu}}_2^{(i)}$ of $q_\phi(\boldsymbol{\mu}_2^{(i)}) = \mathcal{N}(\boldsymbol{\mu}_2^{(i)}|\widetilde{\boldsymbol{\mu}}_2^{(i)}, \boldsymbol{I})$ is regarded as a parameter in the inference model.

The structure of FHVAE contains three modules: encoder 1 (denoted Enc_1 in the following) and encoder 2 (denoted Enc_2) for extraction of latent variables $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, and decoder (denoted Dec) for data reconstruction and conversion. The data flow of the FHVAE framework is shown as below:

$$\boldsymbol{z}_2 = \mathrm{Enc}_2(\boldsymbol{x}) \tag{1}$$

$$\boldsymbol{z}_1 = \mathrm{Enc}_1(\boldsymbol{x}, \boldsymbol{z}_2) \tag{2}$$

$$\boldsymbol{y} = \mathrm{Dec}(\boldsymbol{z}_1, \boldsymbol{z}_2) \tag{3}$$

and $\boldsymbol{y}$ denotes the generated data.

The objective function of FHVAE contains four terms: log-likelihood loss to measure the reconstruction performance; the KL divergence to calculate the distance between prior and posterior of $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$; the log-likelihood loss of mean of $\boldsymbol{z}_2$. The mathematical formulation is:

$$\begin{aligned}
&\mathcal{L}_{orig}^{(i,n)} \\
&= \mathbb{E}_{q_\phi(\boldsymbol{z}_1^{(i,n)}, \boldsymbol{z}_2^{(i,n)}|\boldsymbol{x}^{(i,n)})}\left[\log p_\theta(\boldsymbol{y}^{(i,n)}|\boldsymbol{z}_1^{(i,n)}, \boldsymbol{z}_2^{(i,n)})\right] \\
&- \mathbb{E}_{q_\phi(\boldsymbol{z}_2^{(i,n)}|\boldsymbol{x}^{(i,n)})}\left[D_{KL}(q_\phi(\boldsymbol{z}_1^{(i,n)}|\boldsymbol{x}^{(i,n)}, \boldsymbol{z}_2^{(i,n)})\|p_\theta(\boldsymbol{z}_1^{(i,n)}))\right] \\
&- D_{KL}(q_\phi(\boldsymbol{z}_2^{(i,n)}|\boldsymbol{x}^{(i,n)})\|p_\theta(\boldsymbol{z}_2^{(i,n)}|\tilde{\boldsymbol{\mu}}_2^{(i)})) \\
&+ \frac{1}{N^{(i)}}\log p_\theta(\tilde{\boldsymbol{\mu}}_2^{(i)})
\end{aligned} \tag{4}$$

When applied in voice conversion, superscript src and tar denote the latent variables from source speaker and target speaker in the following, and superscript con denotes the variables prepared for voice conversion. For sequential latent variable $\boldsymbol{z}_2$, the new latent variable $\boldsymbol{z}_2^{con}$ is generated by shifting the mean of $\boldsymbol{z}_2$ from $\boldsymbol{\mu}_2^{src}$ to $\boldsymbol{\mu}_2^{tar}$:

$$\boldsymbol{z}_2^{con} = \boldsymbol{z}_2^{src} - \boldsymbol{\mu}_2^{src} + \boldsymbol{\mu}_2^{tar} \tag{5}$$

And the converted utterance is generated as:

$$\boldsymbol{y}^{con} = \mathrm{Dec}(\boldsymbol{z}_1^{src}, \boldsymbol{z}_2^{con}) \tag{6}$$

The decoder uses the the new sequential latent variable z_2^{con} and the segmental latent variable z_1^{src} from the source speaker to generate the converted utterance.

III. Proposed method

As shown in Section II, FHVAE assumes sequence-dependent and sequence-independent prior for sequential latent variable $\boldsymbol{z}_2$ and segmental latent variable $\boldsymbol{z}_1$, respectively. As linguistic content changes between segments but its statistic is global for all sequences, segmental-scale variable $\boldsymbol{z}_1$ has sequence-independent prior. For sequential latent variable $\boldsymbol{z}_2$, the mean $\boldsymbol{\mu}_2$ of its prior $p_\theta(\boldsymbol{z}_2|\boldsymbol{\mu}_2)$ is assumed drawn from a standard Gaussian distribution for each utterance, i.e. μ_2 changes between utterances. Thus prior $p_\theta(\boldsymbol{z}_2|\boldsymbol{\mu}_2)$ is sequence-dependent. The training target for the sequential latent variable is to make $\boldsymbol{z}_2$ become close to $\boldsymbol{\mu}_2$, and to other $\boldsymbol{z}_2$ from the same utterance in Euclidean space. This is carried out by setting a small fixed variance in the prior $p_\theta(\boldsymbol{z}_2|\boldsymbol{\mu}_2) = \mathcal{N}(\boldsymbol{z}_2|\boldsymbol{\mu}_2, \sigma_{\boldsymbol{z}_2}^2\boldsymbol{I})$, where $\sigma_{\boldsymbol{z}_2}$ equals 0.5 as in [5].

However, only encouraging latent variable $\boldsymbol{z}_2$ close within an utterance is relatively weak. The relationship of $\boldsymbol{z}_2$ between sequences should also be considered. Therefore, we introduce contrastive learning to improve sequential representation. The idea behind contrastive learning is to make representations more similar within the same class, and less similar between classes. And thus the cross-utterance information is tapped.

The framework of the proposed method is shown in Fig. 1. Same with FHVAE, this framework contains three modules: encoder 1 for segmental variable extraction, encoder 2 for sequential variable extraction, and decoder for reconstruction and conversion. The difference between the proposed method

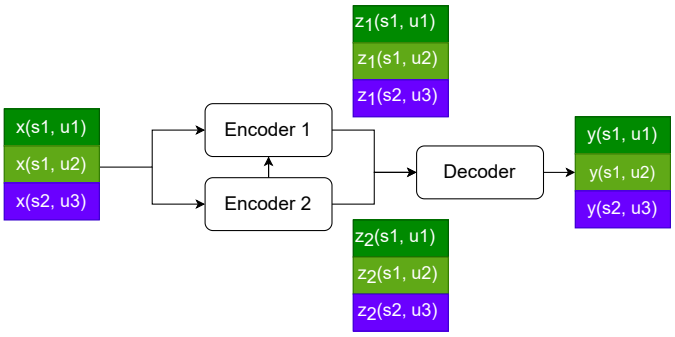


Fig. 1: Framework of the proposed method. Three different utterances are fed in the framework during every training step for contrastive learning.

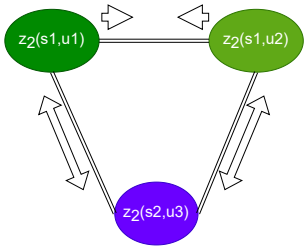


Fig. 2: Conceptual illustration of using contrastive learning on latent variable $\boldsymbol{z}_2$.

and FHVAE is: for every training step, the input contains speech features of three different utterances from two speakers: utterances 1 and 2 are from speaker 1, and their speech features denote $\boldsymbol{x}(\mathrm{s1}, \mathrm{u1})$ and $\boldsymbol{x}(\mathrm{s1}, \mathrm{u2})$ in Fig. 1; utterance 3 is from speaker 2, and is denoted $\boldsymbol{x}(\mathrm{s2}, \mathrm{u3})$. $(\mathrm{s}i, \mathrm{u}j)$ denotes the feature from the j-th utterance of speaker i in the following.

According to the data properties, the sequential latent variable $\boldsymbol{z}_2$ for utterances 1 and 2 should be naturally closer since they represent the same speaker. The sequential variable $\boldsymbol{z}_2$ for utterance 3 should be further from the other two since it represents another speaker. This enables contrastive learning for improving performance of latent variable $\boldsymbol{z}_2$ on speaker identity representation. The graphical illustration of using contrastive learning on $\boldsymbol{z}_2$ is shown in Fig. 2. The training target is to decrease the distance between $\boldsymbol{z}_2(\mathrm{s1}, \mathrm{u1})$ and $\boldsymbol{z}_2(\mathrm{s1}, \mathrm{u2})$, and increase the distances between $\boldsymbol{z}_2(\mathrm{s2}, \mathrm{u3})$ and the two former. In this work, L^2-norm is used as distance metric. The contrastive loss for $\boldsymbol{z}_2$ is shown as below:

$$\begin{aligned}\mathcal{L}_{cont} = \lambda \|\boldsymbol{z}_2(\mathrm{s1}, \mathrm{u1}) - \boldsymbol{z}_2(\mathrm{s1}, \mathrm{u2})\|_2^2 \\ - \beta \|\boldsymbol{z}_2(\mathrm{s1}, \mathrm{u1}) - \boldsymbol{z}_2(\mathrm{s2}, \mathrm{u3})\|_2^2 \\ - \beta \|\boldsymbol{z}_2(\mathrm{s1}, \mathrm{u2}) - \boldsymbol{z}_2(\mathrm{s2}, \mathrm{u3})\|_2^2\end{aligned} \tag{7}$$

Based on preliminary experiments and to make positive pair and negative pairs have same contribution in training, $\lambda = 0.01$, and $\beta = 0.005$ in this work.

The total loss function of the proposed method is:

$$\mathcal{L}_{total} = \mathcal{L}_{orig} - \mathcal{L}_{cont} \tag{8}$$

in which $\mathcal{L}_{orig}$ is the loss for FHVAE as shown in (4).

TABLE I: Latent variable evaluation.

Framework	Sequential			Segmental
	EER(%)	Accuracy(%)		WER(%)
		GRU	GRU+FC	
FHVAE	3.13	79.69	93.23	27.5
proposed	**2.73**	**85.94**	**96.88**	**26.3**

The voice conversion process is the same as for FHVAE which is explained in Section II.

IV. Experiments

A. Dataset

TIMIT: The training set, development set and core test set of TIMIT have been used, each containing respectively 462, 50, and 24 speakers. 8 utterances (labeled with 'SI' and 'SX') per speaker are used in the experiments. Log-magnitude spectrogram is used as input feature, with the window size and hop size equal to 25ms and 10ms. For conversion, one utterance is chosen randomly under label 'SI' for each speaker in test set, namely 24 content-different utterances are chosen. Conversion has been made pair-to-pair, and thus 576 converted utterances have been generated.

VCTK: VCTK dataset contains 110 speakers (62 females and 48 males). We randomly selected utterances from 88, 11 and 11 speakers as training set, development set and test set without overlap. As to conversion, 2 utterances are randomly selected for each speaker in test set, and 462 converted utterances are generated.

B. Baseline

The proposed method is compared with the original FHVAE framework, on latent variable extraction and voice conversion capability. Besides, AutoVC [8], as a state-of-the-art work mentioned in I, has been chosen as another baseline for comparison on conversion quality. The released model and vocoder from the authors of AutoVC have been used for converted utterances generation.

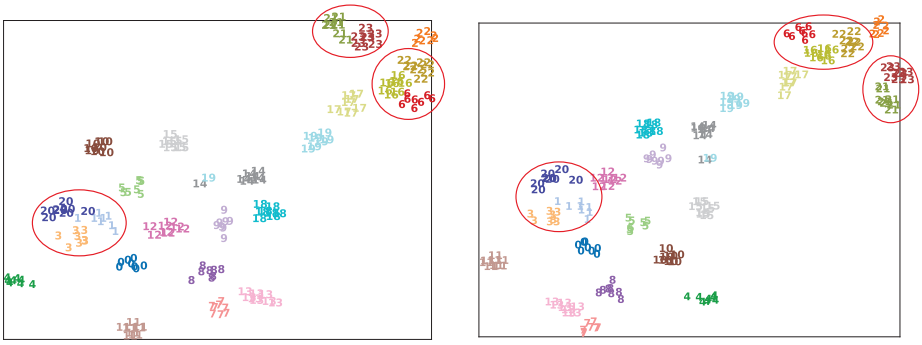
C. Implementation details

Framework structure and training details: The structure of encoders and decoders are the same as in FHVAE and in the proposed method. All encoders and decoders contain one LSTM layer with 256 units and a fully-connected layer. The dimensions of latent variables in both frameworks equal 32. For fair comparison, the training settings are also same for baseline and the proposed method. Batch size equals 768. Each batch contains data from three kinds of utterance equally in the proposed method and random data for baseline. Learning rate equals 10^{-4}, and the optimizer is Adam [16]. Vocoder used in this work is HifiGAN [17].

Segmental-level feature evaluation: The mean and variance of latent variable z_1 are used as the extracted segmental-level feature in the evaluation experiment. As the segmental-level feature assumed to represent linguistic content, speech recognition is used for evaluation. The speech recognition system is implemented by Kaldi toolbox [18] and experiment

TABLE II: Voice conversion evaluation

Framework	fake speech detection				
	F2F	M2M	F2M	M2F	All
FHVAE (TIMIT)	0.767	0.719	0.721	0.701	0.713
proposed (TIMIT)	**0.772**	**0.721**	**0.727**	**0.705**	**0.717**
AUTOVC (VCTK)	0.638	0.672	0.666	0.664	0.648
proposed (VCTK)	**0.680**	**0.713**	**0.679**	**0.685**	**0.678**



(a) FHVAE (b) The proposed method

Fig. 3: t-SNE plot of extracted sequential features from the compared frameworks. All parameter settings for both t-SNE plots are the same. The red circles indicate that the cluster dispersion is more obvious in the proposed method.

details are the same as in [19]. The results are shown in Table I under header 'Segmental'. 'WER' denotes word error rate (WER), and lower WER indicates better performance.

Sequential-level feature evaluation: The mean of z_2 has been used as the sequential-level representation. Speaker verification and speaker identification experiments have been done for fairness. For speaker verification, the equal error rate (EER) based on cosine similarity is displayed in Table I under the header 'EER(%)', the lower the better. Two neural network architectures have been applied for speaker identification: one implemented by 1-layer GRU and the other by 1-layer GRU with 512 units followed by a dense layer. The details of speaker identification are the same as in [19]. The results from the two classifiers can be found in Table I with header 'Accuracy(%)', under 'GRU' and 'GRU+FC' respectively. The higher accuracy means the better performance.

Voice conversion evaluation: Conversion evaluation is based on fake speech detection, using the open toolkit *Resemblyzer*[1] for speech quality and similarity evaluation. Results are shown in Table II under 'fake speech detection'. Higher score shows better performance. Specifically, fake speech detection has been done between intra-sexual conversion which shown with header 'F2F', 'M2M', and inter-sexual conversion under 'F2M' and 'M2F'. The overall evaluation results are shown under header 'All'. Demos[2] could be found on our website.

D. Results and analysis

Sequential-level feature: Table I shows consistent improvement from the proposed method in all experiments. Additionally, t-SNE [20] plots give visualization in Fig. 3. Each number k in Fig. 3 denotes one utterance from the k-th speaker. The clusters become more separated between different speaker classes in the proposed method. For instance, features from speaker 6, 22 and 16 look a little bit more separated in Fig. b, so do cluster 21 and 23. Cluster 20 and 1, 3 show more obviously the advantage of introducing contrastive learning to the FHVAE framework.

Segmental-level feature: The results in Table I show that the introduced contrastive learning not only improves sequential latent variable extraction, but also slightly improves segment latent variable extraction, indicating better disentanglement.

Voice conversion: Results shown in Table II indicate that speech converted by the proposed method exhibits slight improvement on fake speech detection compared with the original FHVAE. It proves that contrastive learning strategy helps slightly on voice conversion performance of FHVAE. Besides, the experiment results also show that the proposed method works better than AutoVC.

V. Conclusion

As the constraint of sequential latent variable is relatively weak in FHVAE, we introduce contrastive learning to improve it. The proposed method not only considers the distance of sequential variables within one utterance, but also among utterances through contrastive learning. No more layers but only the learning strategy has been changed. Experiment results show that, compared with baseline, the proposed method improves both sequential latent variable and segmental latent variable extraction performance; meanwhile in voice conversion application, the proposed method also shows better performance.

[1]https://github.com/resemble-ai/Resemblyzer

[2]https://yuxi6842.github.io/contrastiveFHVAE.github.io/#traditional

REFERENCES

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[2] Adam Polyak, Yossi Adi, Jade Copet, Eugene Kharitonov, Kushal Lakhotia, Wei-Ning Hsu, Abdelrahman Mohamed, and Emmanuel Dupoux, "Speech Resynthesis from Discrete Disentangled Self-Supervised Representations," in *Proc. Interspeech*, 2021.

[3] Ranya Aloufi, Hamed Haddadi, and David Boyle, "Privacy-preserving voice analysis via disentangled representations," in *Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop*, 2020, pp. 1–14.

[4] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *International Conference on Machine Learning*. PMLR, 2019, pp. 4114–4124.

[5] W. Hsu, Y. Zhang, and J. Glass, "Unsupervised learning of disentangled and interpretable representations from sequential data," in *NIPS*, 2017.

[6] Xintao Zhao, Feng Liu, Changhe Song, Zhiyong Wu, Shiyin Kang, Deyi Tuo, and Helen Meng, "Disentangling content and fine-grained prosody information via hybrid asr bottleneck features for voice conversion," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7022–7026.

[7] Chak Ho Chan, Kaizhi Qian, Yang Zhang, and Mark Hasegawa-Johnson, "Speechsplit2. 0: Unsupervised speech disentanglement for voice conversion without tuning autoencoder bottlenecks," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6332–6336.

[8] Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson, "Autovc: Zero-shot voice style transfer with only autoencoder loss," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5210–5219.

[9] Huaizhen Tang, Xulong Zhang, Jianzong Wang, Ning Cheng, and Jing Xiao, "Avqvc: One-shot voice conversion by vector quantization with applying contrastive learning," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 4613–4617.

[10] Janek Ebbers, Michael Kuhlmann, Tobias Cord-Landwehr, and Reinhold Haeb-Umbach, "Contrastive predictive coding supported factorized variational autoencoder for unsupervised learning of disentangled speech representations," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3860–3864.

[11] Jiachen Lian, Chunlei Zhang, and Dong Yu, "Robust disentangled variational speech representation learning for zero-shot voice conversion," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6572–6576.

[12] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.

[13] Raia Hadsell, Sumit Chopra, and Yann LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. IEEE Computer Society, 2006, vol. 2, pp. 1735–1742.

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[15] Achintya Kumar Sarkar, Zheng-Hua Tan, Hao Tang, Suwon Shon, and James Glass, "Time-contrastive learning based deep bottleneck features for text-dependent speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1267–1279, 2019.

[16] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[17] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17022–17033, 2020.

[18] D. Povey and A. Ghoshal, "The kaldi speech recognition toolkit," in *IEEE Workshop on ASRU*, 2011.

[19] Yuying Xie, Thomas Arildsen, and Zheng-Hua Tan, "Disentangled speech representation learning based on factorized hierarchical variational autoencoder with self-supervised objective," in *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2021, pp. 1–6.

[20] Laurens Van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.