Random Cycle Loss and Its Application to Voice Conversion

Haoran Sun, Dong Wang, *Senior Member, IEEE,* Lantian Li, Chen Chen, and Thomas Fang Zheng, *Senior Member, IEEE*

Abstract—Speech disentanglement aims to decompose independent causal factors of speech signals into separate codes. Perfect disentanglement benefits to a broad range of speech processing tasks. This paper presents a simple but effective disentanglement approach based on cycle consistency loss and random factor substitution. This leads to a novel random cycle (RC) loss that enforces analysis-and-resynthesis consistency, a main principle of reductionism. We theoretically demonstrate that the proposed RC loss can achieve independent codes if well optimized, which in turn leads to superior disentanglement when combined with information bottleneck (IB). Extensive simulation experiments were conducted to understand the properties of the RC loss, and experimental results on voice conversion further demonstrate the practical merit of the proposal. Source code and audio samples can be found on the webpage http://rc.cslt.org.

Index Terms—Cycle consistency, information disentanglement, information bottleneck, voice conversion.

1 Introduction

Human beings can effectively disentangle information involved in speech signals [1], [2]. This capability endows the downstream functionalities of human auditory systems, e.g., identifying speaker trait and recognizing speech content [3], [4]. Researchers have proposed multiple disentanglement models to simulate this capacity, and have applied them to various speech processing tasks, such as speech recognition [5], speech synthesis [6], [7], speaker recognition [8] and emotion recognition [9].

Among all the tasks, voice conversion (VC) [10] is a particular case that would benefit substantially if speech information could be well disentangled. The goal of VC is to modify speech signals uttered by one speaker to make them sound like being enunciated by another speaker, while keeping the content unchanged. Information disentanglement is a natural approach to achieve that goal: if one could represent content and speaker trait by separate codes, then VC could be simply achieved by changing the speaker code while keeping the content code unaltered.

In this section, we will first review the existing research on speech disentanglement, and then present our motivation for a novel random cycle (RC) loss, and describe how it is applied to the VC task.

1.1 Speech disentanglement

According to the identifiability problem [11], it is not possible to learn disentangled codes without any prior on models and training schemes. Depending on how the prior is involved, we can roughly divide existing speech disentanglement approaches into two categories: (1) **structure prior** that designs models to represent different factors by codes at different locations in the model structure; (2) **information regularization** that designs appropriate regularization to control information of different factors flowing to different codes. Note that we have intentionally distinguished the concept of *factor* and *code*, with the former referring to the underlying causes of the observed speech signals, and the latter referring to the representations that are derived by the disentangling model. The goal of speech disentanglement is to let the codes represent the factors, ideally with a one-to-one mapping.

1.1.1 Structure prior

Early research formulated probabilistic models that use codes with carefully designed priors and conditionals to fit the underlying factors in speech signals. For instance, the famous i-vector model [12], that was extensively used in speaker recognition, disentangles factors of phonetic content and speaker trait by designing a probabilistic model that emphasizes the different temporal scopes of the two factors. A key shortcoming of this approach is that to keep tractability, the model cannot be complex.

More recent research is based on deep neural nets. For example, Hsu et al. presented a disentanglement model based on VAE [5], [13], which decomposes speech signals into factors of content, speaker and speaking style by designing a probabilistic model in the latent space. Wang et al. [14] presented a similar architecture, though no explicit probabilistic models were designed in the latent space.

Overall, the central idea of these models is to design appropriate structures, either probabilistic or neural, to account for different properties of different speech factors.

• *H. Sun, D. Wang, L. Li, C. Chen and T. F. Zheng are with the Center for Speech and Language Technologies (CSLT), BNRist at Tsinghua University, Beijing 100084, China.*
E-mail: {sunhr,lilt,cchen}@cslt.org, wangdong99@mails.tsinghua.edu.cn, fzheng@tsinghua.edu.cn

This work was supported by the National Natural Science Foundation of China (NSFC) under the project No.62171250. Dong Wang and Lantian Li are the corresponding authors. Manuscript received April 19, 2005; revised August 26, 2015.

1.1.2 Information regularization

Information regularization is another way to achieve disentanglement, by designing various regularization methods that enforce information of a particular factor flowing to a designed code.

A strong regularization could be achieved through explicit supervision. For example, [9] designed a cascaded learning procedure that introduces supervision for content, speaker and emotion sequentially, hence enforcing codes of these three factors to be extracted successively. In most cases, however, full supervision is not available. In this situation, information bottleneck (IB) [15] accompanied by reconstruction loss is often used to control the information flow [16], [17], [18]. The main idea of the IB-based approach is that if the capacity of all information channels are limited, the code of each channel will exhibit information selectivity, in order to recover the input as good as possible with the limited code bandwidth.

It was found that when the IB is well designed, perfect disentanglement can be achieved under mild assumptions [18]. However in most cases, designing an appropriate IB is not trivial. To improve disentanglement, some authors designed explicit regularization on mutual information (MI) amongst codes, for instance [14], [16].

1.2 Motivation

In this paper, we focus on the IB-based disentanglement approach, due to its noticeable success in VC and other applications [18]. Although it was shown that perfect disentanglement can be obtained by this approach, IB design is often challenging, especially when multiple factors are involved. Moreover, explicit MI regularization [14], [16] may not perform as expected. Firstly, most of the regularization methods involve minmax optimization, which is notoriously unstable. Secondly, most MI regularization is derived from an upper-bound of MI, rather MI it self. If the bound is loose, the regularization is not effective. Thirdly, it was reported that MI regularization does not necessarily lead to disentangled codes, as argued in [19]. This is also demonstrated in our simulation study, refer to Section 4.

We present a novel random cycle (RC) loss to improve speech disentanglement. The main design is a random factor substitution (RFS) operation and a cycle consistency loss. Specifically, we hope to minimize $||C' - \hat{C}'||^2$, where $C' = RFS(C)$ represents the code constructed by randomly substituting some components of the code C of the input speech, and $\hat{C}'$ is obtained by re-encoding the speech signal reconstructed from C'. The RC loss was inspired by the analysis-resynthesis principle of reductionism [20], [21], [22], which states that the world can be decomposed into atomic factors, and the factors can be recomposed to form new things (material, concept, functionality, etc.). These new things are valid and so can be decomposed following the same rule as applied to the existing things. For speech disentanglement, this means that perfectly disentangled codes will represent the true factors that generate the speech, and therefore can be randomly recomposed to form new and valid speech, which in turn can be decomposed into those compositional factors, leading to *random cycle consistency*.

We will show theoretically and empirically that the RC loss, if perfectly optimized, can induce *independent* codes, and when combined with the IB-based approach, enforces *disentangled* codes. This interesting property lends the RC loss as a simple but effective approach to improving speech disentanglement. Compared to existing MI regularization methods based on adversarial loss or MI loss, the RC loss does not require any extra regressor/classifer, and does not play any minmax game, hence simpler in implementation and more stable in model training.

We will apply the RC loss to improve voice conversion, and test it with two VC models: AutoVC [18] and SpeechFlow [23]. These two models are representatives of the IB-based approach. They are based on auto-encoders (AEs), and disentangle speech into separate factors through IB design. While AutoVC aims to disentangle speech into content and speaker trait, SpeechFlow tries to disentangle speech into more factors: content, timbre, rhythm and pitch. For clarity in description, we will denote AutoVC and SpeechFlow trained with RC loss by **CycleVC** and **CycleFlow** respectively.

1.3 Paper structure

In summary, our contributions are three-fold: (1) Propose the RC loss and analyze its properties theoretically; (2) Present a simulation study to demonstrate the behavior of the RC loss; (3) Apply the RC loss to the VC task, to demonstrate its practical usage.

We first summarize the related work in Section 2, and propose the RC loss in Section 3. Section 4 presents an extensive simulation study, and Section 5 and Section 6 present the empirical results on the VC task. Finally Section 7 concludes the paper and presents some future research directions. Note that a preliminary version (see supplementary materials) has been published in the Odyssey 2022 workshop, and the current paper involves substantial extension in both theory and experiments.

2 Related works

2.1 Information disentanglement

Decomposing patterns/signals into compositional elements has a long history, represented by multiple classical algorithms including the famous principle component analysis (PCA) [24] and independent component analysis (ICA) [25]. In essence, these models target for *factorization*, i.e., seeking for statistically independent components z_i to recover pattern x by $p(x) = \prod p(z_i)$. Bengio argued that a more useful decomposition is *disentanglement* [26], which seeks for codes that correspond to the underlying causal factors. He argued that disentanglement is the essential way to achieve robustness and generalization in machine learning. Recently, Higgins presented a formal definition for disentanglement [27]. They drew connection between disentanglement and symmetric transformation.

Many researchers explored unsupervised learning models to disentangle information factors. The famous models include Info-GAN [28], β-VAE [29], FactorVAE [30], β-TCVAE [31]. A common theme among these models is to make the codes constrained in capacity but still being representative and ideally being mutually independent; hence

more possibly representing the main factors. However, there is no guarantee that the resultant codes really correspond to the true generating factors. This problem is theoretically analyzed in [11], which showed that unsupervised models without any inductive bias cannot ensure disentanglement, in other words *unidentifiable*. Since then, weakly/partially supervised models have been more prevalent. Typical approaches include data augmentation [32], using auxiliary variables [33], and imposing temporal or physical constraints [34], [35], [36]. All these models introduce auxiliary information to improve identifiability.

2.2 Voice conversion

Voice conversion (VC) [10] aims to change speaker identify of a speech signal. Conventional VC approaches require parallel data — that is, utterances with the same content but spoken by both the source and target speakers. Representative models include GMMs [10], [37], [38], neural nets [39], [40], [41], [42], [43], and NMF [44], [45]. Collecting parallel data is clearly costly, so people have tried to study approaches with non-parallel data. One research line relies on distribution matching. The VC models based on CycleGAN [46], [47], [48] and StartGAN [49] are examples of this paradigm. The training criterion of these model is to match the distributions of the converted speech and the true speech, rather than matching pairs of frames or utterances, thereby circumventing the need for parallel data.

Another research line is based on speech disentanglement. The belief is that once speech content and speaker trait can be well disentangled, it would be easy to achieve VC by replacing the speaker code. Encoder-Decoder modeling is the general architecture for this type of VC methods, where the encoder produces codes corresponding to individual factors, and the decoder collects these codes to produce converted speech. Representative models include VAE [50], CVAE [17], VQVAE [51], [52], and AdaIN-VC [53]. A key issue with this approach is information entanglement, i.e., information related to different factors might be mixed in the same code. To address this problem, a possible solution is to use large-scale pre-trained models to extract codes sensitive to a particular factor. For instance, using speech recognition (ASR) models [54], [55] or self-supervised models (e.g., Wav2Vec [56], [57] or HuBERT [58]) to generate content codes, and using pre-trained speaker embedding models to generate speaker codes [18]. However, there is no guarantee that the pre-trained models produce information-purified codes. For example, it is well known that speaker embeddings carry more information than just speaker traits [59].

To achieve more disentangled codes, various regularization methods have been proposed. For example, some authors introduced an adversarial discriminator on the converted speech, to enforce that the generated speech sounds like the target speaker [60], [61], or an adversarial classifier in the latent space, to ensure that codes corresponding to different factors are less dependent [62], [63]. Other authors designed more information-inspired regularization to suppress MI between codes of different factors, e.g. [14], [16]. These regularization methods generally improve VC performance. However, the additional regressor/classifier and minmax training leads to increased complexity, and MI is not guaranteed to be reduced [19].

Recently, Qian et al. [18] presented a simple VC model based on the vanilla auto-encoder (AE), called AutoVC. In this model, speaker identity is used as condition, and the latent code, if the dimensionality is well designed, will represent and only represent speech content. The authors explained their model based on the IB theory [15], and showed mathematically that if speech signals can be regarded as an ergodic stationary order-τ Markov process with bounded second moment, with an appropriate IB setting, speaker and content information can be perfectly disentangled. This theoretical guarantee makes the simple AutoVC a strong competitor of more complex models such as CVAE [17] and starGAN [49]. Following the same IB principle, Qian et al. extended AutoVC to SpeechFlow [23], with more speech factors considered, including timbre, rhythm, pitch and content.

Although attractive in theory, the IB-based models like AutoVC and SpeechFlow rely on careful IB design. A poorly designed IB leads to either information entanglement or information loss. Unfortunately, IB design is often difficult and requires much trial-and-error, in particular when the model involves multiple factors like SpeechFlow. To address this problem, researchers have to go back to the conventional MI regularization methods, including adversarial loss [63], [64] and MI loss [14]. While performance improvement was reported, the intrinsic problems of MI regularization discussed above remain.

The RC loss proposed in this paper aims at the same goal of purifying information load of speech codes with IB-based models, but tackles it from a different way: it enforces cycle consistency that a perfect disentanglement model should anyway satisfy, rather than explicit MI regularization.

2.3 Cycle consistency loss

Cycle consistency loss has been known as a key ingredient in CycleGAN [65], [66], a model that has been used in VC [46]. The same loss was also employed in CycleVAE [67], [68], where the primary loss is maximum likelihood rather than adversarial loss. For both CycleGAN and CycleVAE, the cycle is back and forth from one speaker to another. The cycle in the RC loss is quite different: it is back and forth from the code space to the observation space. Cycle loss was also presented in AutoVC [18], where it was used to bound the entropy of each dimension in the latent code, rather than information purification.

The idea of RC loss also appears in general machine learning literature. For example, [19] found that in VAE, ensuring random cycle consistency leads to better disentangled codes in image processing. However, they used random sampling rather than RFS, that could synthesize data off the manifold. In [69], a cycle loss is presented to encourage class-sensitive codes producing the same classification result when the residual code is changed. They found with this cycle loss, the residual code becomes more class independent. This cycle loss can be regarded as a partial RC loss, with RFS on the residual code only. In summary, cycle consistency loss, with or without RFS, has been used in information disentanglement, however most existing work

uses it as an ad-hoc regularization. In this paper, we will present a theoretical, systematic and in-depth study on this topic.

3 Random Cycle Loss

3.1 IB-based information disentanglement

Our study starts from a formal definition for speech disentanglement, and show how an IB-based approach can provide a solution. Part of the analysis is inspired by [23].

Let X denote a fixed-length speech segment, and assume it is generated from an underlying process involving M independent factors $F = [F_1, ..., F_M]$, formulated by $X = g(F)$, where g is a one-to-one mapping whose domain involves all the combinations of $\{F_i\}$. The information disentanglement task is defined as follows: Design an encoder f that represent X as a set of latent codes $Z = [Z_1, Z_2, ..., Z_M]$, so that each Z_i contains and only contains information of a corresponding factor F_i. Mathematically, perfect disentanglement can be formulated follows:

$$MI(Z_i; F_i) = H(F_i) \quad (1)$$
$$MI(Z_i; F_{\neq i}) = 0 \quad (2)$$

where $H(\cdot)$ and $MI(\cdot;\cdot)$ denote (differential) information entropy and mutual information respectively, and

$$X = g(F) \quad (3)$$
$$Z = f(X). \quad (4)$$

We show that with an auto-encoder architecture, disentanglement can be achieved by setting IB capacity and information bias for each code Z_i. Firstly, Theorem 1 (proof in Appendix A) shows that if Z_i contains full information of F_i, setting appropriate IB guarantees perfect disentanglement.

Theorem 1. *Assume that Z_i contains full information of F_i, i.e.,*

$$MI(Z_i; F_i) = H(F_i). \quad (5)$$

If the IB for each code Z_i is precisely set as follows:

$$H(Z_i) = H(F_i), \quad (6)$$

then Z_i are mutually independent and Z_i contains information of F_i only, i.e., $MI(Z_i; F_{\neq i}) = 0$.

Theorem 1 states that an IB-based approach can solve the disentanglement problem. However, it relies on the assumption $MI(Z_i; F_i) = H(F_i)$, which is just a partial goal of the disentanglement task. One may think this can be simply satisfied by feeding to the encoder of Z_i with full knowledge of F_i. However, that is not true as the encoder may not propagate the required information to Z_i. To solve the problem, one can set explicit **information bias** on the input of Z_i and try to recover X from Z by a decoder h. This is formally presented as Theorem 2 (proof in Appendix B).

Theorem 2. *Suppose that (1) for any factor F_i, only one code Z_i steadily receives full information of F_i, and (2) if any information about F_i is lost by Z_i, none of other codes or code sets can* always *provide the complement. With this assumption, $||X - \hat{X}||^2 = 0$ ensures $MI(F_i; Z_i) = H(F_i)$.*

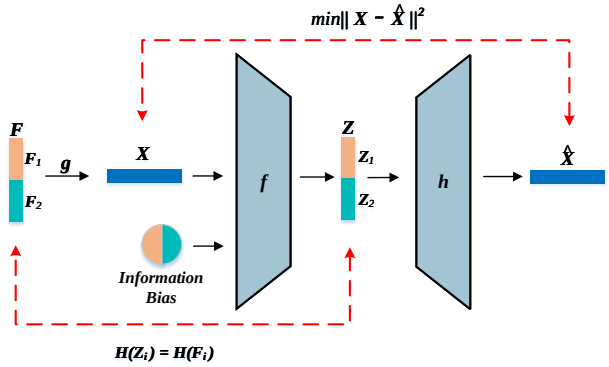


Fig. 1: Illustration for IB-based information disentanglement. With the IB setting $H(Z_i) = H(F_i)$ and appropriate information bias, an AE model that seeks for minimum reconstruction loss $||X - \hat{X}||^2$ will induce disentangled codes.

Combining Theorem 2 and Theorem 1, we conclude that if the IB and information bias are well settled, disentanglement can be achieved with an auto-encoder. This is the theoretical foundation for the IB-based disentanglement approach. Fig.1 presents a graphical illustration for the approach.

The final question is, how to design information bias to make sure only Z_i receives full information of F_i? A known approach is via information corruption. For example, in SpeechFlow [23], random resampling is used to corrupt rhythm information when encoding content and pitch, to ensure that only the rhythm encoder can receive the full rhythm information. We note that the information bias mentioned above is nothing but a particular form of inductive bias mentioned in [11], and is essential to gain identifiability.

3.2 Random cycle loss

Although theoretically sound, the IB-based disentanglement approach requires appropriate setting for the IB and information bias, which is not easy in practice. On the one hand, setting IB for Z_i by controlling the dimensionality is difficult. This is because we have no prior knowledge about the suitable dimensions, therefore mostly relying on trial-and-error. This is particularly the case when the codes are continuous, as continuous variables hold infinite entropy in theory. Some authors chose discrete variables [51], [52] to alleviate this problem, but trial-and-error is still required. On the other hand, setting information bias is not any easier. For example, it was shown that random resampling might be ineffective in corrupting undesired information [70]. All these problems lead to imperfect disentanglement. To tackle the problem, a common practice is to design explicit MI regularization, such as adversarial loss [64] and MI loss [14]. However, as mentioned already, these MI regularizations increase model complexity and training instability, and do not guarantee better disentanglement.

In this paper, we present a new regularization called **random cycle (RC) loss** to improve information disentanglement with IB-based models. The core idea is simple: we

randomly pick up codes from different and independent utterances, and recombine them to compose a new code. After an additional decoding-encoding path, we hope the resultant codes are exactly the same as those we picked up. The RC loss is defined as the Euclidean distance between the codes before and after the decoding-encoding process.

Before any theoretical discussion, we show how the RC loss is implemented. The full process is shown as the following path in Fig. 2: (1) 1st round encoding (black solid line); (2) random factor substitution (blue dashed line); (3) speech reconstruction (yellow dashed line); (4) 2nd round encoding (red dashed line); (5) loss calculation.

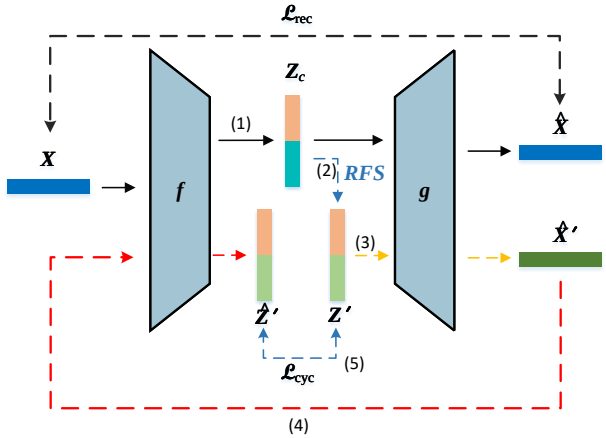


Fig. 2: The IB-based speech disentanglement model with RC loss. f and g are encoder and decoder respectively. Solid lines denote the encoding and decoding for the original utterance. Dashed lines denote the process of random substitution and cyclic decoding-encoding. $\mathcal{L}_{rec}$ and $\mathcal{L}_{cyc}$ represent the reconstruction loss and the RC loss, respectively.

We use two data samples, denoted by X^1 and X^2, to demonstrate the computing process. Assume that there are two latent codes Z_1 and Z_2.

- **1st round encoding**: Firstly encode X^1 and X^2, resulting in two sets of codes: $Z^1 = \{Z_1^1, Z_2^1\}$ and $Z^2 = \{Z_1^2, Z_2^2\}$.
- **Random factor substitution (RFS)**: Randomly choose a code from Z^2, and use it to replace the corresponding code in Z^1. Suppose that the selected code is Z_2^2, we compose a new code set $Z' = \{Z_1^1, Z_2^2\}$.
- **Speech reconstruction**: Forward Z' to the decoder and produce the reconstructed speech $\hat{X}'$.
- **2nd round encoding**: Encode $\hat{X}'$ and obtain $\hat{Z}' = \{\hat{Z}_1', \hat{Z}_2'\}$.
- **Cycle loss computation**: The cycle loss is computed as follows:

$$\mathcal{L}_{cyc} = ||Z' - \hat{Z}'||^2. \tag{7}$$

The final loss for the disentanglement model combines the reconstruction loss and the RC loss shown in Eq.(7):

$$\mathcal{L} = \mathcal{L}_{rec} + \alpha * \mathcal{L}_{cyc} \tag{8}$$

where

$$\mathcal{L}_{rec} = ||X - \hat{X}||^2$$

and α is a hyperparameter to balance the contribution of the two losses.

3.3 Theoretical analysis

3.3.1 *Analysis 1: Code independence*

We show that under moderate conditions, the RC loss theoretically leads to independent codes.

Let's define $Z' = \{Z_1', Z_2'\}$ the codes after RFS. Now reconstruct $\hat{X}'$ from Z' and conduct the 2nd round encoding to get $\hat{Z}' = \{\hat{Z}_1', \hat{Z}_2'\}$. Our purpose is to let $\hat{Z}_1'$ be *fully* determined by Z_1' and $\hat{Z}_2'$ *fully* determined by Z_2'. This can be obtained by minimizing the conditional entropy $H(\hat{Z}_1'|Z_1')$ and $H(\hat{Z}_2'|Z_2')$, formulated by the following objective:

$$\begin{aligned}
\mathcal{L}_{ch} &= H(\hat{Z}_1'|Z_1') + H(\hat{Z}_2'|Z_2') \\
&= -\mathbb{E}_{Z_1',\hat{Z}_1'} \log p(\hat{Z}_1'|Z_1') - \mathbb{E}_{Z_2',\hat{Z}_2'} \log p(\hat{Z}_2'|Z_2') \\
&= -\mathbb{E}_{Z_1',Z_2'} \log p(\hat{Z}_1'|Z_1') - \mathbb{E}_{Z_2',Z_1'} \log p(\hat{Z}_2'|Z_2') \\
&= -\mathbb{E}_{Z_1'}\mathbb{E}_{Z_2'} \log p(\hat{Z}_1'|Z_1') - \mathbb{E}_{Z_2'}\mathbb{E}_{Z_1'} \log p(\hat{Z}_2'|Z_2') \,,
\end{aligned}$$

where we have employed the fact that $(\hat{Z}_1', \hat{Z}_2')$ is determined by (Z_1', Z_2'), and that Z_1' and Z_2' are independent. If we further assume that the conditional probabilities $p(\hat{Z}_1'|Z_1')$ and $p(\hat{Z}_2'|Z_2')$ are isotropic Gaussian with variation σ, then we have:

$$\begin{aligned}
\mathcal{L}_{ch} &= \mathbb{E}_{Z_1'}\mathbb{E}_{Z_2'}||\hat{Z}_1' - Z_1'||^2 + \mathbb{E}_{Z_2'}\mathbb{E}_{Z_1'}||\hat{Z}_2' - Z_2'||^2 + C(\sigma) \\
&\propto \mathbb{E}_{Z'}||\hat{Z}' - Z'||^2 \,,
\end{aligned}$$

where $C(\sigma)$ is a constant depending on σ. This is just the RC loss shown in Eq.(7). Therefore, minimizing the RC loss essentially reduces the conditional entropy of the recovered codes.

Further notice that Z_1' and Z_2' are independent, and in the case $\mathcal{L}_{ch} = 0$, $\hat{Z}_1'$ is totally dependent on Z_1', and $\hat{Z}_2'$ is totally dependent on Z_2', we have:

$$\begin{aligned}
p(\hat{Z}_1'|\hat{Z}_2') &= \int_{Z_1',Z_2'} p(\hat{Z}_1'|Z_1', Z_2', \hat{Z}_2') p(Z_1', Z_2'|\hat{Z}_2') \mathrm{d}Z_1' \mathrm{d}Z_2' \\
&= \int_{Z_1',Z_2'} p(\hat{Z}_1'|Z_1') \frac{p(\hat{Z}_2'|Z_1', Z_2') p(Z_1') p(Z_2')}{p(\hat{Z}_2')} \mathrm{d}Z_1' \mathrm{d}Z_2' \\
&= \int_{Z_1'} p(\hat{Z}_1', Z_1') \mathrm{d}Z_1' \int_{Z_2'} \frac{p(\hat{Z}_2'|Z_2') p(Z_2')}{p(\hat{Z}_2')} \mathrm{d}Z_2' \\
&= p(\hat{Z}_1') \,.
\end{aligned}$$

This result means that if the model has been well trained and the RC loss converges to zero, then the codes $\hat{Z}_1'$ and $\hat{Z}_2'$ are mutually independent. We highlight that this result is general and applies to any encoder-decoder process. Specifically, it does not rely on any IB setting.

3.3.2 Analysis 2: Compatibility with IB approach

The RC loss is fully compatible with the IB-based disentanglement model, in the sense that pursuing a low RC loss does not prevent the disentanglement model from pursuing its global optimum.

Let's assume the disentanglement model is perfectly trained with IB setting $H(Z_i) = H(F_i)$. Since $MI(Z_i; F_i) = H(F_i)$ (Eq.(1)), one immediately obtains $H(F_i|Z_i) = 0$ and $H(Z_i|F_i) = 0$. This means that the mapping $g \circ f$ between F and Z is one-to-one. Therefore, for any composed code $Z' = \{Z'_i\}$ by RFS, we can always identify a factor $F' = \{F'_i\}$ where each F'_i corresponds to Z'_i. Note that any combination of F'_i is in the domain of the generation function g, F' corresponds to a valid sample $X' = g(F')$. Since the disentanglement model is perfect, X' is exactly encoded as Z' and can be fully reconstructed by the decoder h, i.e., $\hat{X}' = h(Z') = X'$. Encoding the reconstructed data $\hat{X}'$ leads to: $\hat{Z}' = f(\hat{X}') = f(X') = Z'$. Therefore, the RC loss $||\hat{Z}' - Z'||^2 = 0$.

This result means that if one can train a perfect IB-based disentanglement model, the RC loss will naturally approach to 0. In other words, the RC loss does not posit anything conflicting to the goal of the disentanglement model; instead, it just reinforces a necessary condition that a perfect disentanglement model should satisfy.

3.3.3 Analysis 3: Code-space data augmentation

The entire path of the RC loss $h \circ f$ can be regarded as a code-space auto-encoder, where the encoder and decoder are swapped compared to the original model. With the new view, RFS plays a role of code-space data augmentation, by synthesizing new data with random recombination. As discussed in the previous analysis, if the disentanglement model is perfect, the new generated code C' corresponds to a unique factor F', which is a particular combination of F'_i. According to our assumption, any combination of F'_i is involved in the domain of the generation process g, ad so corresponds to a valid data X'. The RC loss encourages that the new data X' can be well represented by the model, i.e., X' can be well reconstructed.[1] In other words, the model takes X' as an extra training data. We conjecture this code-space data augmentation is particularly useful when the training set is small and contains limited cases of factor combination. Overall, the RC loss may improve model generalizability.

We highlight that the 'factor substitution' rather than 'factor resampling' is important for the RC loss. If the factors are randomly sampled or corrupted, e.g., by Gaussian, there is no guarantee that the synthesized code is valid, and in this situation enforcing RC loss could be less beneficial.[2] We also conjecture that the code-space data augmentation especially benefits to the voice conversion task, since it involves the same factor substitution operation as RFS during inference.

1. To make this clear, notice that $H(Z) = H(X)$, so the RFS code C' and the recovered code $\hat{C}'$ are deterministically mapped to X' and $\hat{X}'$ in the observation space respectively. Therefore, minimizing the RC loss is equal to recovering X' by $\hat{X}'$.

2. It may contribute to bound the entropy of each code dimension, as the identity loss in AutoVC. [18].

3.3.4 Analysis 4: Limit of RC loss

We notice that the RC loss by itself only ensures code independence (as shown in Analysis 1), but does not guarantee factor disentanglement. Essentially, this is because it is a symmetric loss and does not introduce any inductive bias. Therefore, it must co-operate with the IB approach, i.e., with reasonable settings for IB capacity and information bias, otherwise degenerated solutions may be obtained. For example, one can reduce the RC loss by setting a particular code to be constant. In this case, the constant code is indeed independent from others, but no disentanglement is obtained. Another example is the phenomenon of condition collapse. For example with CAE, the decoder may simply ignore the condition code. In this situation, the RC loss computed on the latent code is low but the information is fully entangled. We will show examples of degenerate situations in the simulation experiments; however on real speech data, we have not observed the problem.

3.3.5 Discussion

In essence, the RC loss encourages data synthesized by any factor combination being decomposable into the composite factors. This follows from an analysis-and-resynthesis principle of reductionism [20], [21], [22]. This principle has been widely adopted in many scientific fields, such as perception [71] and chemistry [72]. The primary belief is that once a phenomenon can be well explained by independent factors, recombining the factors can lead to a new and valid phenomenon, where the term 'valid' means that the new phenomenon can be explained in the same way as the existing observations. For example, once scientists know different materials are composed of atoms, it would be possible to construct new materials by combining atoms in new ways, and the new materials should can be decomposed into atoms following the same decomposition rule.

The reductionism origin leads to profound difference between RC loss and MI regularization such as adversarial loss and MI loss. Although RC loss does enforce independent codes hence MI reduction, MI reduction is not the way that RC loss takes to boost disentanglement. The true mechanism is the analysis-and-resynthesis principle and MI reduction is simply a consequence of the principle. Interestingly, we found in our experiments that RC loss is more effective in MI reduction than adversarial loss and MI loss, although it is not intentionally designed for that purpose.

4 Simulation Study

In this section, we present a simulation study to investigate behavior of the RC loss. We generate some 'speech like' one-dimensional sequences, for which the mean value represents 'speaker identify' and a Markov random process is designed to represent 'speech content'. By the simulated data, we explore how information is disentangled by IB-based conditional auto-encoders (CAEs), and understand how the MI regularization methods and the RC loss take effect.

4.1 Data generation

We generate each sample as follows, where T is the length of the sample:

$$x_c(t) = e(c) + z(t) + \epsilon(t) \quad t = 0, ..., T-1 \tag{9}$$

where c denotes class, and $e(c)$ denotes the center of class c, which we regard as the *class factor*; t denotes time, and $z(t)$ is the *content factor* which we assume follows an ergodic Markov chain. $\epsilon(t) \sim N(0, \sigma)$ is an additive Gaussian noise. All the quantities are scalars, and the resulting vector $[x_c(0), x_c(1), ..., x_c(T-1)]$ is regarded as a data sample. Note that the Markov assumption for $z(t)$ has been made in the theoretical analysis in AutoVC [18]. The detailed procedure is as follows:

- Define 10 classes whose centers $\{e(c)\}_{c=1}^{10}$ are evenly scattered on the real axis from 0 to 18.
- Define $z(t)$ as a Markov chain with 3 states whose values are 0.0, 0.5, 1.0 respectively. The chain always begins from state 1, and for each state, it must stay in that state for 3 steps; after that, it is possible to stay in the same state or transit to the next state, with probability 0.1 and 0.9 respectively. Allowed transitions from states 1, 2 and 3 are states 2, 3 and 1 respectively.
- To generate a data sample, we first randomly select a class c following a uniform distribution, and then run the Markov chain for $T = 50$ steps. At each step, record $z(t)$ and sample $\epsilon(t)$ from $N(0, 0.1)$. Finally, $x_c(t)$ is obtained by Eq.(9).

We generate 400 samples for each class, 200 for training and 200 for test. This amounts to a training set and a test set, each with 2000 samples. Mean-variance normalization is employed. Fig. 3 shows several data samples, each curve corresponding to a sample of a particular class.

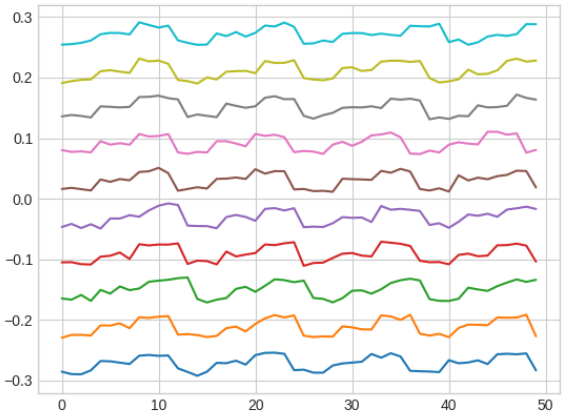


Fig. 3: Data samples generated by Eq.(9). Each curve represents a sample, and we choose one sample for each of the 10 classes.

4.2 CAE model

The model used in our experiments is a CAE. The dimensionality of the data space is 50, and the dimensionality of the latent space varies according to the test. The encoder and decoder are both three-layer fully-connected neural nets whose hidden layer consists of 64 units with the *tanh* activation function. The output of the encoder involves a *tanh* nonlinear transform, though the output of the decoder is linear. Unless explicitly stated, the model is trained for 20k iterations with batch size of 512, using the Adam optimizer [73] with a learning rate $1e^{-4}$.

We use the class label c as the conditional input of the decoder. More precisely, we learn an embedding matrix that maps $e(c)$ to a 20-dim *class code*, which is then concatenated with the output of the encoder and forwarded to the decoder. According to the theory of the IB-based approach developed in Section 3, the output of the encoder will mostly represent the content factor $z(t)$ if its capacity is well controlled. We therefore call the encoder output *content code*.

4.3 MI reduction

The first experiment examines how the RC loss purifies information in the content code, i.e., reduces MI between the content code and the class label. Since the content code is continuous, MI can not be computed directly. To solve the problem, we firstly cluster the content codes into 10 classes by k-means, and then apply the *normalized_mutual_info_score* function in the *sklearn* python package to compute MI between the cluster assignment and the class label.

Table 1 shows the results on both the training and test sets. Note that the k-means clustering is conducted with the training set, and the resultant clusters are used to compute the MI values on both the training and the test sets. It can be observed that the standard CAE can perform good reconstruction, and if the dimensionality of the content code is small, the MI can be reduced, indicating that the content code contains less class information. This conforms to the analysis in AutoVC [18] and our results in Section 3. However, this MI reduction happens only if the IB is tight, i.e., code dim = 1.

With the RC loss, the MI is significantly reduced if the IB is not over loose, while the reconstruction loss is not much sacrificed. This supports our theoretical analysis that RC loss is compatible with IB-based models and enforces independent codes. Note that if the IB is loose (code dim $\geq$ 25), the MI cannot be reduced even with the RC loss. This is consistent with our analysis about the limit of RC loss: it should be accompanied with a reasonable IB setting, otherwise degenerated solutions might be obtained. In this experiment specifically, if the dimensionality is over large, the CAE can achieve good reconstruction from the content code, hence ignoring the class code. In this case, imposing RC loss may worsen this 'condition collapse' problem, as ignoring the class code just reduces the RC loss.

4.4 Comparison with adversarial loss and MI loss

We now compare RC loss with two explicit MI regularization losses: adversarial loss and MI loss. For the adversarial loss, we design an extra classifier that accepts the content code and predicts the class label. The classifier is a three-layer neural net whose dimensionality of the hidden layer is half of that of the content code. A gradient reverse layer is inserted between the content code and the classifier to perform adversarial training, as in [62], [63], [64]. For the MI

TABLE 1: Reconstruction loss (Rec) and MI tested on CAE with/without RC loss. 'Code dim' represents the dimensionality of the content code. Note that 1.0 is the maximum value of MI computed by the *normalized_mutual_info_score* function.

		CAE		CAE + RC	
	Code dim	Rec	MI	Rec	MI
Train	1	$3.1e^{-5}$	0.128	$3.1e^{-5}$	0.117
	2	$3.0e^{-5}$	0.738	$3.1e^{-5}$	0.016
	4	$2.1e^{-5}$	1.000	$2.7e^{-5}$	0.178
	8	$1.4e^{-5}$	0.973	$2.0e^{-5}$	0.104
	25	$8.7e^{-6}$	1.000	$1.3e^{-5}$	1.000
Test	1	$3.1e^{-5}$	0.133	$3.0e^{-5}$	0.126
	2	$3.0e^{-5}$	0.742	$3.0e^{-5}$	0.025
	4	$2.1e^{-5}$	1.000	$2.7e^{-5}$	0.163
	8	$1.4e^{-5}$	0.975	$1.9e^{-5}$	0.102
	25	$8.9e^{-6}$	1.000	$1.4e^{-5}$	1.000

loss, we choose CLUB, an upper bound of MI as proposed in [74] and used in [14], [16].

The dimensionality of the content code is set to 8, and we train the model for 20k iterations. Fig. 4 (Top) presents the MI values during the training process, when different regularizations are employed. It can be seen that with the RC loss, the MI is quickly reduced to a low level, while with the adversarial loss and the MI loss, the MI value does not show a clear trend. This is a bit surprising, as the two losses were intentionally designed to reduce the MI. We attribute the phenomenon to the instability of the minmax game that the training process plays with the two losses.

In another experiment, we reduce the dimensionality of the content code to 2. The results are shown in Fig. 4 (Bottom). In this experiment, all the losses seem to contribute, especially with a heavily weighted MI loss (MI-10). The more substantial contributions of the adversarial loss and the MI loss in the dim-2 case compared to the dim-8 case may be attributed to the ease in model training. Nevertheless, the RC loss is more effective and stable than the two explicit regularization losses.

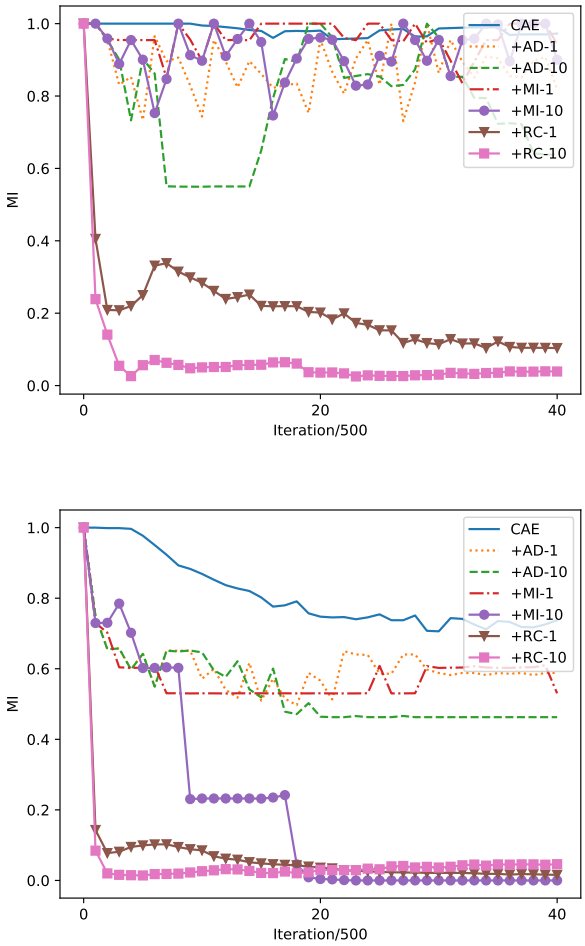


Fig. 4: MI between the content code and the class label, when trained with RC loss (RC), adversarial loss (AD) and MI loss (MI) as regularization. In the legend, the number appended to the name represents the weight on the regularization. The dimensionality of the latent code is 8 (Top) and 2 (Bottom) respectively.

4.5 Conversion examples

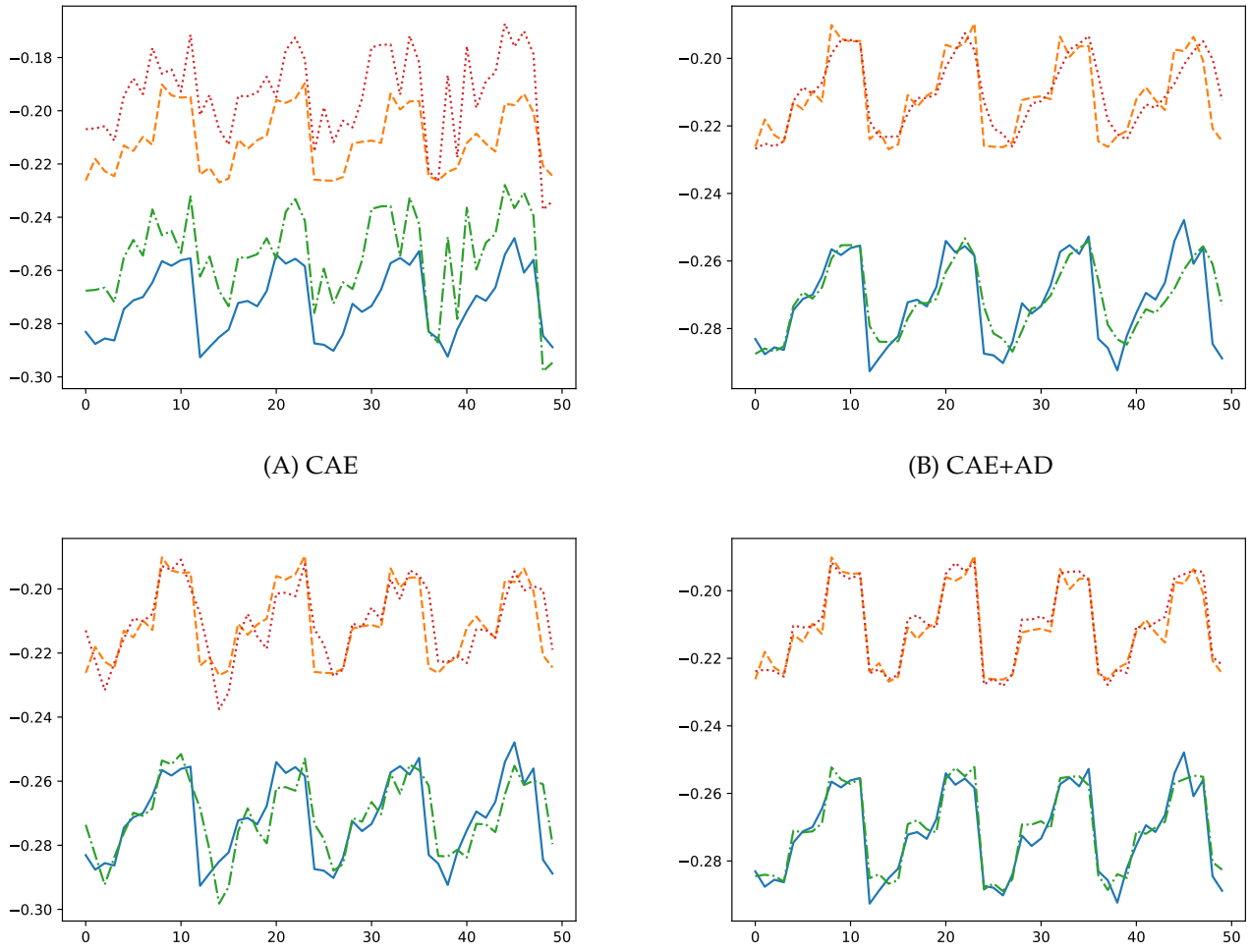
In this experiment, we choose two samples, one denoted by S and the other denoted by C. Using the CAE, we extract the content code from C and combine it with the class code of S, and then perform generation with the decoder. This simulates voice conversion.

Fig. 5 presents an example of the conversion. In this picture, the blue and orange curves represent the original sample S and C respectively. The content code of C is then combined with the class code of S to perform conversion, and the result S+C is shown as the green dot-dashed curve. We hope the converted sample matches S in mean value, while matches C in dynamic change. To show the match in dynamic change easily, we perform a mean-shift on the converted sample, resulting in S+C+Δ that is shown as the red dotted curve.

It can be seen that the conversion with the standard CAE largely fails, while with any of the regularizations, the conversion is better: at least the mean values of the converted samples match that of S, hence the class being successfully converted. Comparing the results with the three regularizations, it can be seen that with the RC loss, the converted sample with mean-shift (S+C+Δ) matches the content sample (C) best, suggesting that it preserves the content better than the adversarial loss and the MI loss. Moreover, we also found that quality of the conversion is more stable with the RC loss than with the two MI regularization losses, if we inspect the conversion results with models of different epoches. Videos of evidence are shown in the project web page and included in the supplemental as well.

5 Experiments on VC: CycleVC

In this experiment, we apply the RC loss to AutoVC [18], a simple VC model based on CAE. The main purpose of this experiment is to study the behavior of the RC loss on real speech data, rather than a full and complicated VC system. Therefore, we choose a medium sized training



(A) CAE (B) CAE+AD

(C) CAE+MI (D) CAE+RC

Fig. 5: An example of conversion with the conventional CAE and CAEs with adversarial loss (CAE+AD), MI loss (CAE+MI) and RC loss (CAE+RC). The weight for the AD/MI/RC loss is chosen to be 1, though other values show similar results. In each figure, the blue solid line shows the sample to provide class code (S); the orange dashed line shows the sample to provide content code (C); the green dot-dashed line shows the converted sample with class code from S and content code from C, denoted by S+C; the red dotted line is the mean-shift version of S+C, denoted by S+C+Δ. If the conversion is perfect, S+C will match S in mean value and C in dynamic change, so the curves of S+C+Δ and C should overlap.

set which allows us performing comprehensive parameter search. Moreover, we fully rely on objective metrics in system evaluation and comparison. Large-scale training and complete evaluation (both objective & subjective) will be taken in the next experiment with SpeechFlow, a successive version of AutoVC that offers more fine-grained control of disentanglement and conversion.

5.1 AutoVC and CycleVC

We firstly present the AutoVC model, and then describe how to apply the RC loss to the model. To assist the presentation, AutoVC regulated by RC loss will be called **CycleVC**.

Briefly, AutoVC is a CAE, where speaker vectors generated from a pre-trained speaker embedding model are used as the condition, and the latent code represents speech content. To avoid any confusion, we will use *speaker code* and *content code* to denote the speaker embedding and the latent code of the CAE respectively.

The main diagram is shown in Fig. 6. In the training phase (Fig. 6, left panel), a fixed-length speech segment X is fed to the speaker encoder E_s and produces speaker code Z_s. X and Z_s are then concatenated and fed to content encoder E_c, to produce content code Z_c. Z_c and Z_s are then concatenated and fed to the decoder D, to produce the reconstructed speech segment $\hat{X}$. The training criterion is reconstruction loss, i.e., $||X - \hat{X}||^2$. In the conversion phase (Fig. 6, right panel), the pipeline is almost the same as in the training phase, except that the speaker code fed to the decoder is produced from a reference speech segment X^r.

Fig. 7 shows the diagram of CycleVC, AutoVC regularized with the RC loss. Since the conversion phase is the same as AutoVC, only the training phase is shown. Moreover,

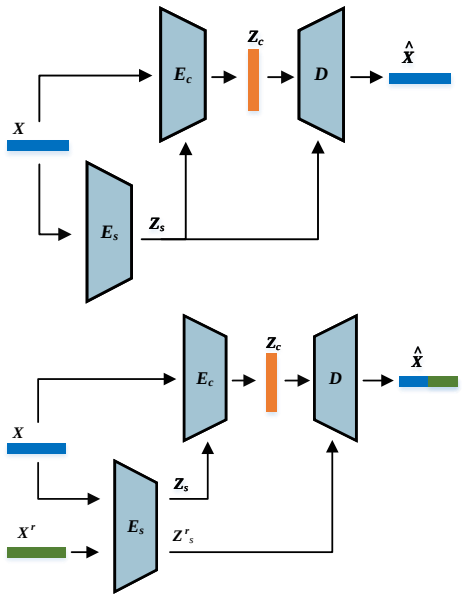


Fig. 6: AutoVC diagram in (Top) training phase and (Bottom) conversion phase.

we omit the path for the reconstruction loss as it is the same as AutoVC. In this diagram, two segments X and X^r are randomly selected to generate the content code Z_c and the speaker code Z_s^r respectively, and Z_c and Z_s^r are fed to the decoder, producing the reconstructed speech $\hat{X}$. The reconstructed speech is then fed to E_c and E_s to produce the 2nd round codes $\hat{Z}_s$ and $\hat{Z}_c$. The RC loss is then computed as $||\hat{Z}_c - Z_c||^2$. Note that a full RC loss should involve $||\hat{Z}_s - Z_s^r||$, though we assume the speaker encoder is fixed so just omit it.

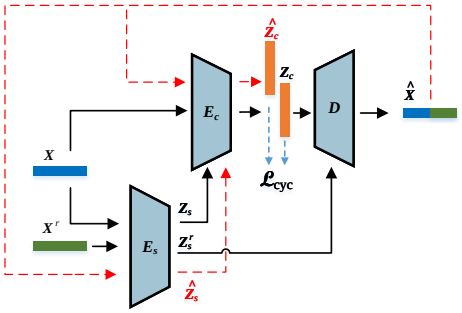


Fig. 7: CycleVC diagram for model training. Only path for the RC loss is shown.

5.2 Configurations

The CSTR VCTK corpus [75] is used to conduct the experiment. We choose 780 utterances from 20 speakers for model training, and 850 utterances from another 20 speakers are used for testing. There is no overlap between the training and test data, in both speaker and content.

Both the AutoVC and CycleVC systems are constructed using the source code published online[3]. The settings in the original repository are largely retained, including network structure, data processing steps, and the training scheme.

A pre-trained d-vector model [76] is used as the speaker encoder, which produces 256-dim speaker codes. The AutoVC/CycleVC models output 80-dimensional Mel spectrum, and a pre-trained HiFiGAN [77] is used to convert Mel spectrum to speech signals. The Adam optimizer [73] is used to train the model, with a batch size of 20 for $100k$ steps.[4]

We also implement the adversarial loss [62], [63] with the same source code and configuration. The extra classifier used for computing the adversarial loss is composed of 2 hidden layers, consisting of 256 and 128 units respectively. The input is the content code, and the output corresponds to the 20 speakers in the training data. A gradient reversal layer between the content code and the classifier is used to enable adversarial training. The model regulated by the adversarial loss is denoted by **ADVC**.

5.3 Evaluation metrics

Four objective metrics are used to evaluate the generated speech:

- **MOS**: The output of MOSNet [78] that approximates the mean opinion score (MOS) in subjective tests. This is used to evaluate the overall perceptual quality;
- **MCD**: Mel-cepstral distortion (MCD), to evaluate similarity on spectrum;
- **F0-PCC**: Pearson correlation coefficient (PCC) on F0 values, to evaluate similarity on pitch;
- **Spk-Sim**: Cosine similarity on speaker codes, to evaluate similarity on speaker trait.

5.4 Results for reconstruction

We firstly report the results on speech reconstruction. Besides the four metrics defined in the previous section, we also test MI between the content code and the speaker code, in order to evaluate how well the information is disentangled. The result is reported in the row MI(C; S). For both CycleVC and ADVC, the weight on the regularization term impacts system performance. We report the results with the setting for each system that leads to the lowest MI on the test set.

TABLE 2: Comparison among AutoVC, CycleVC and ADVC on reconstructed speech.

	32 dim			128 dim		
Metric	AutoVC	CycleVC	ADVC	AutoVC	CycleVC	ADVC
MOS (↑)	2.986	2.998	**2.999**	3.027	**3.067**	3.060
MCD (dB) (↓)	2.912	**2.851**	2.896	2.963	2.921	**2.911**
F0-PCC (↑)	**0.298**	0.275	0.284	0.346	0.370	**0.378**
Spk-Sim (↑)	0.712	**0.718**	0.712	0.740	**0.754**	0.753
MI(C; S)(↓)	0.158	**0.145**	0.155	0.249	**0.176**	0.195

The results are reported in Table 2, where the dimension of the content code is set to 32 and 128 respectively. Note that 32-dim is the default setting of AutoVC, representing

3. https://github.com/auspicious3000/autovc

4. In the original repository, the batch size was set to 2. To meet the request of the RFS operation in CycleFlow, we increased the batch size to 20. Our experiment showed that the increased batch size also benefits AutoVC.

a reasonable (tight and sufficient) IB setting. The results show that both CycleVC and ADVC can reduce MI between the content and speaker codes, though CycleVC is more effective. Moreover, almost on all the quality metrics, CycleVC and ADVC provide better performance than AutoVC, especially with the 128-dim setting where the IB is loose. This performance improvement should be attributed to the increased generalizability associated with the additional regualrization offered by the RC loss and adversarial loss. The relative advantage of CycleVC and ADVC is not clear in this experiment.

5.5 Results for conversion

The performance on the conversion task is shown in Table 3. Note that for F0-PCC and Spk-Sim, the pair for comparison is specified. There are several observations: (1) Comparing the AutoVC results of 32-dim and 128-dim, a noticeable change is that the 128-dim model leads to a larger F0-PCC between the source speech and the converted speech, reflecting the fact that more information of the source speech is retained. This redundant information leads to a worse MOS value, suggesting that the IB is loose. (2) CycleVC does not offer clear improvement in the 32-dim test, however in the 128-dim test, consistent performance improvement is observed. This is expected as the IB is tight in the 32-dim model, for which AutoVC can achieve a clean content code; in the 128-dim condition, the content code is more information entangled, so the RC loss provides more contribution. (3) In both the 32-dim and 128-dim cases, ADVC does not provide better performance than the AutoVC baseline. Considering that ADVC indeed reduces the MI value as shown in Table 2, it seems to suggest that a lower MI does not necessarily imply a better conversion. The clear improvement with the RC loss, therefore, should not be superficially explained by the MI reduction, but the improved analysis-and-resynthesis consistency, and perhaps the code-space data augmentation together.

TABLE 3: Comparison among AutoVC, CycleVC and ADVC on converted speech. 'CP' denotes to whom the converted speech will compare when computing the 'Metric'. 'S' denotes source speech, 'T' denotes target speech.

		32 dim			128 dim		
Metric	CP	AutoVC	CycleVC	ADVC	AutoVC	CycleVC	ADVC
MOS(↑)	-	**3.053**	2.977	3.031	3.045	**3.079**	3.042
F0-PCC(↑)	S	0.272	**0.279**	0.241	0.309	**0.331**	0.306
Spk-Sim(↑)	T	0.674	**0.675**	0.667	0.687	**0.700**	0.693

6 Experiments on VC: CycleFlow

In this experiment, we apply the RC loss to SpeechFlow [23], a more fine-grained VC model that decomposes speech signals to timbre, rhythm, content and pitch.

6.1 SpeechFlow and CycleFlow

SpeechFlow represents an input speech segment X with four codes: timbre Z_t, rhythm Z_r, content Z_c and pitch Z_f. To make sure different codes represent the corresponding factors, IB and the information bias are carefully designed. Specifically, a pre-trained d-vector model is used to produce Z_t, and a pitch extractor is used to produce input for Z_f. A *random resampling (RR)* operation is employed to corrupt rhythm information in both Z_c and Z_f. By these designs, information bias is established and once the model is well trained, different factors are represented by different codes.

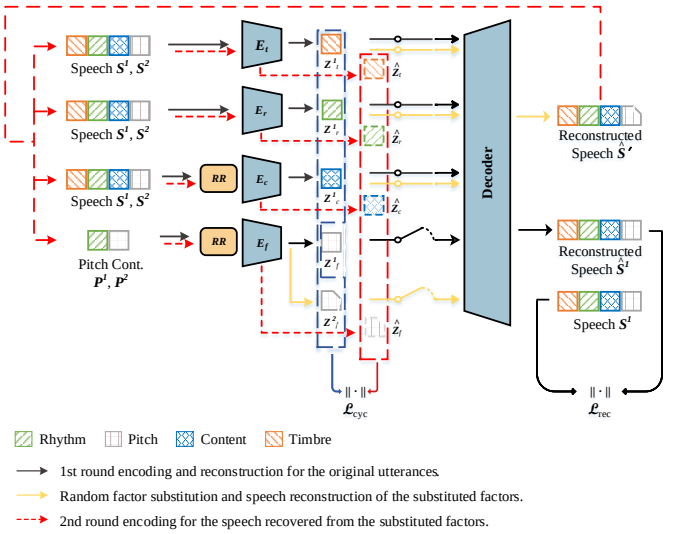


Fig. 8: The architecture of CycleFlow. Solid lines denote the path for reconstruction loss, and dashed lines denote the path for RC loss. Note that the pitch code Z^1_f is substituted by Z^2_f in the picture.

We apply the RC loss to improve SpeechFlow, and call the resultant model **CycleFlow**. The architecture is shown in Fig. 8. The main path of the RC loss is similar to that in CycleVC, and can be computed by an additional decoding-encoding path, as shown by the dashed lines in Fig. 8.

6.2 Data and configurations

We mostly follow the experimental settings in SpeechFlow [23]. Specifically, $27,500$ utterances of 100 speakers from VCTK are used for model training, and 1060 utterances of another 8 speakers from the same corpus are used to perform testing. No overlap exists between the training and test data, in both speaker and content.

SpeechFlow is reproduced using the source code published online[5], and the same code is adapted to implement CycleFlow. We mostly reuse the parameters of the original repository, including network structure, data processing steps, and the training scheme. Specifically, we use d-vector [76] to represent speaker timbre, SPTK[6] to extract F0 as the pitch value. The output of the model is 80-dimensional Mel spectrum, and a pre-trained WaveNet [79] is used to generate speech signal from Mel spectrum. The model is trained using the Adam optimizer [73] with a batch size of 16 for $200k$ steps. For comparison, we also implement an ADFlow model, another SpeechFlow variant regularized with adversarial loss as in [64]. We have tried our best to tune the weight of the adversarial loss, in order to get the best performance.

5. https://github.com/auspicious3000/SpeechSplit
6. http://sp-tk.sourceforge.net/

6.3 Quality of reconstruction

We firstly test the quality of reconstructed speech produced by different models. The results are shown in Table 4. It can be seen that CycleFlow substantially outperforms SpeechFlow. In comparison, ADFlow also performs better than SpeechFlow, but the improvement is marginal and much less significant compared to the one obtained by CycleFlow. As discussed in the AutoVC experiment, this result demonstrates that both the RC loss and adversarial loss can improve model generalizability, though RC loss is more effective.

TABLE 4: Quality comparison among SpeechFlow, CycleFlow and ADFlow on reconstructed speech.

Metric	SpeechFlow	CycleFlow	ADFlow
MOS (↑)	2.919	**3.124**	2.962
MCD (dB) (↓)	3.233	**3.121**	3.136
F0-PCC (↑)	0.647	**0.670**	0.650
Spk-Sim (↑)	0.849	**0.900**	0.849

6.4 Mutual information

In the second experiment, we investigate mutual information (1) between the original speech and the codes, and (2) between different codes. Ideally, we hope all these MI values as small as possible.

The results are shown in Table 5. It can be observed that in all the comparisons, CycleFlow achieves lower MIs than SpeechFlow. For row 1, 2 and 3, this means that all the encoders eliminate more irrelevant information from the original speech. For row 4, 5 and 6, this means that the codes are more mutually independent. Since CycleFlow can achieve comparable or even better reconstruction compared to SpeechFlow, the reduced MI should be attributed to better disentanglement (rather than information loss) with regularization of the RC loss. Finally, ADFlow does not show clear MI reduction, although that is the purpose of the adversarial loss. Considering the results of ADVC where MI is indeed reduced, we conjecture the failure of ADFlow is due to the more factors the model involves, which makes the minmax training even harder.

TABLE 5: MI between input speech and codes, and between pairs of codes. $\boldsymbol{S}$ denotes the original speech signal. $\boldsymbol{Z}_c$, $\boldsymbol{Z}_r$, $\boldsymbol{Z}_f$ denote codes for content, rhythm and pitch respectively.

No.	Factors	SpeechFlow	CycleFlow	ADFlow
1	$\boldsymbol{S}$ vs. $\boldsymbol{Z}_c$	0.509	**0.366**	0.525
2	$\boldsymbol{S}$ vs. $\boldsymbol{Z}_r$	0.713	**0.616**	0.704
3	$\boldsymbol{S}$ vs. $\boldsymbol{Z}_f$	0.552	**0.431**	0.498
4	$\boldsymbol{Z}_c$ vs. $\boldsymbol{Z}_r$	0.516	**0.340**	0.516
5	$\boldsymbol{Z}_c$ vs. $\boldsymbol{Z}_f$	0.446	**0.298**	0.438
6	$\boldsymbol{Z}_r$ vs. $\boldsymbol{Z}_f$	0.529	**0.495**	0.511

6.5 Voice conversion

In the third experiment, we conduct voice conversion using SpeechFlow, CycleFlow and ADFlow. Three tests are conducted: (1) Timbre conversion (timbre only); (2) Style conversion (pitch + rhythm); (3) Full conversion (timbre + pitch + rhythm). We do not intend to test pitch and rhythm separately, as it is not easy for human listeners to identify them individually.

6.5.1 Objective result

The objective results are reported in Table 6. It can be observed that in almost all the conversion tasks and on almost all the metrics, CycleFlow outperforms SpeechFlow, demonstrating the clear contribution of the RC loss. In particular, CycleFlow seems more superior in pitch transferring: F0-PCC in style conversion changes from 0.29 to 0.46. This observation conforms to the results in Table 5, where the MI between the content code and the original speech is significantly reduced with CycleFlow ($0.5093 \rightarrow 0.3659$), and the MI between the content and pitch codes is reduced as well ($0.4455 \rightarrow 0.2983$). This means that with the RC loss, the content code is significantly purified and involves much less pitch information, which making the pitch conversion easier. In comparison, ADFlow can generally improve MOS and F0-PCC, which coincides with the result in [64]. However, in terms of Spk-Sim, ADFlow does not offer any improvement. Comparing CycleFlow and ADFlow, CycleFlow works clearly better. The only metric on which ADFlow wins is the MOS value when converting speaking style, but this is probably due to the weak style conversion and weak timbre preservation.

TABLE 6: Comparison among SpeechFlow, CycleFlow and ADFlow on converted speech. 'CP' denotes to whom the converted speech will compare when computing the 'Metric' in the 'Conv' task.

Metric	Conv	CP	SpeechFlow	CycleFlow	ADFlow
MOS(↑)	Timbre	-	2.959	**3.214**	2.994
	Style	-	2.944	2.987	**3.098**
	Full	-	2.924	**3.089**	3.056
F0-PCC(↑)	Timbre	S	0.441	**0.493**	0.471
	Style	T	0.286	**0.464**	0.349
	Full	T	0.348	**0.552**	0.375
Spk-Sim(↑)	Timbre	T	0.638	**0.781**	0.622
	Style	S	0.793	0.778	0.739
	Full	T	0.702	**0.805**	0.647

6.5.2 Subjective results

In the subjective test, we hired 26 Chinese listeners, each being assigned 150 test cases, divided into 5 test groups: (1) reconstruction quality; (2) style conversion; (3) timbre maintenance in style conversion; (4) full conversion with both style and timbre transfer; (5) quality of converted speech. For each evaluation, we presented listeners three utterances produced by SpeechFlow, CycleFlow and ADFlow respectively, and asked them to select which one is the best according to the specified metric. We used speech segments of 8 speakers in the test set (5 males and 3 females).

To guide the listeners how to proceed the tasks, an example was provided for each task. The utterance with the desired property was presented to the listener in each test case, to let them know what the conversion targets to. For example, in the (2) style conversion task, the target speech was presented to inform the target style; and in the (3) timbre maintenance task, the source speech was presented to inform the timbre that we wanted to maintain. Note that in all the tests, the content of the source and target speech are the same, in order to reduce the psychological load of the listeners and therefore a more accurate evaluation.

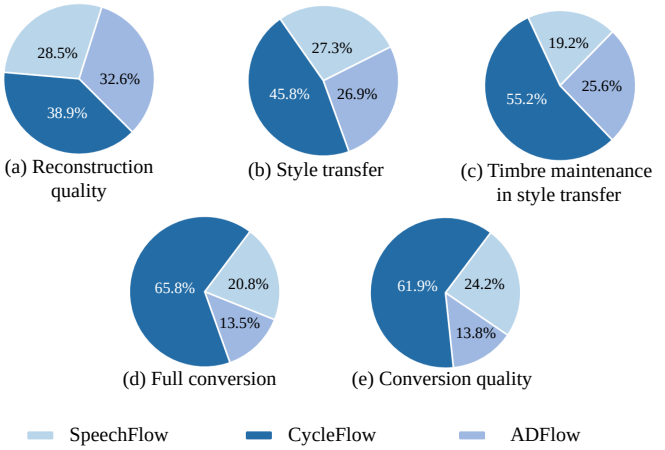


Fig. 9: Results of subjective evaluation on 5 listening tasks.

The results are shown in Fig. 9. It can be found that the three models obtained comparable results in the reconstruction task, and in all the four conversion tasks, CycleFlow clearly outperforms SpeechFlow and ADFlow. Note that ADFlow did not show much improvement over the SpeechFlow baseline. This is consistent with the results in the objective evaluation, and aligns with the argument in literature that adversarial loss does not necessarily improve disentanglement [19].

7 Conclusion

We proposed a novel random cycle (RC) loss to improve information disentanglement with the IB-based models. The core design is a combination of random factor substitution and cycle consistency loss. We demonstrated theoretically that the proposed RC loss is fully compatible with the IB-based disentanglement approach and leads to independent codes. In most cases (no degeneration happens), this strongly encourages better disentanglement. We tested the RC loss with simulation data and found that it is highly effective. We also applied the new loss to two popular voice conversion models, AutoVC and SpeechFlow, and observed significant performance improvement in reconstruction, disentanglement and conversion. Importantly, we found in nearly all the situations, RC loss outperforms adversarial loss and MI loss, two representatives of the MI regularization method that is widely used to promote disentanglement. It should be highlighted that the implementation of RC loss is simple and the additional computational cost is nearly negligible. Therefore, we believe the RC loss deserves more investigation and being employed in broader applications.

Vast amount of research work remains to be undertaken on this topic. To mention a few: how to avoid degenerate situations, how to combine and reconcile RC loss and other MI regularizations, how to prevent the negative impact on the reconstruction loss in model training, how the RC loss works in other applications. Finally, the foundation of the RC loss, i.e., the analysis-and-resynthesis principle, is a general belief of science. This principle deserves more investigation in machine learning, if the purpose is to explain physical data. From that perspective, the RC loss is just a small step, and other forms of models and methods that respect this principle could make more profound contribution.

Appendix A
Proof to Theorem 1

Proof. From Eq.(6) and Eq.(5), we have:

$$\begin{aligned} MI(Z_i;F_i) &= H(Z_i) - H(Z_i|F_i) && (10)\\ &= H(F_i) - H(F_i|Z_i) && (11)\\ &= H(F_i) && (12)\\ &= H(Z_i). && (13) \end{aligned}$$

Therefore

$$H(Z_i|F_i) = 0 \;\;;\;\; H(F_i|Z_i) = 0 \tag{14}$$

This equation shows that Z_i is fully determined by F_i and vice versa. **Since F_i are mutually independent, an immediate conclusion is that Z_i are mutually independent.**

For any factor that is different from F_i, denoted by $F_{\neq i}$, we have:

$$MI(Z_i;F_i,F_{\neq i}) = MI(Z_i;F_i) + MI(Z_i;F_{\neq i}|F_i) \tag{15}$$

Note that:

$$H(Z_i) \geq MI(Z_i;F_i,F_{\neq i}) \geq MI(Z_i;F_i) = H(Z_i) \tag{16}$$

This means:

$$MI(Z_i;F_i,F_{\neq i}) = H(Z_i). \tag{17}$$

Take Eq.(13) and Eq.(17) to Eq.(15):

$$MI(Z_i;F_{\neq i}|F_i) = 0 \tag{18}$$

Now compute $MI(Z_i;F_{\neq i}|F_i)$ as follows:

$$MI(Z_i;F_{\neq i}|F_i) = \mathbb{E}_{Z,F_i,F_{\neq i}} \ln \frac{p(Z_i,F_{\neq i}|F_i)}{p(Z_i|F_i)p(F_{\neq i}|F_i)} \tag{19}$$

Apply the fact that F_i and $F_{\neq i}$ are independent, and that F_i and Z_i determine each other:

$$\begin{aligned} MI(Z_i;F_{\neq i}|F_i) &= \mathbb{E}_{Z,F_i,F_{\neq i}} \ln \frac{p(Z_i,F_{\neq i},F_i)}{p(Z_i,F_i)p(F_{\neq i})} && (20)\\ &= \mathbb{E}_{Z,F_i,F_{\neq i}} \ln \frac{p(Z_i,F_{\neq i})}{p(Z_i)p(F_{\neq i})} && (21)\\ &= MI(Z_i;F_{\neq i}) && (22) \end{aligned}$$

Referring to Eq.(18), we therefore have:

$$MI(Z_i;F_{\neq i}) = 0. \tag{23}$$

This means Z_i contains no information of $F_{\neq i}$. In other words, it contains and only contains information of F_i.

□

APPENDIX B
PROOF TO THEOREM 2

Proof. Recall the assumption: one and only one code Z_i steadily receives full information of F_i; and if any information about F_i is lost by Z_i, none of other codes or code sets can *always* provide the complement. In other words, other codes or code sets may provide the complement in some instances, but cannot provide the complement all the time. Formally this can be stated as follows: if $H(F_i|Z_i) > 0$, then $H(F_i|Z_i, Z_{\neq i}) > 0$.

Now we can prove $H(F_i|Z_i) = 0$ by contraction.

If this is not the case, i.e., $H(F_i|Z_i) > 0$, according to the assumption, we have $H(F_i|Z_i, Z_{\neq i}) > 0$. Therefore:

$$H(F_i|\hat{S}) \geq H(F_i|Z_i, Z_{\neq i}) > 0 \tag{24}$$

where $\hat{S}$ is the reconstructed signal. Note that the original signal S fully determines F_i, i.e.,

$$H(F_i|S) = 0. \tag{25}$$

Eq.(24) and Eq.(25) indicate that $S \neq \hat{S}$, so the the reconstruction is not perfect, which is contrast to the condition that the solution is optimal.

Therefore, $H(F_i|Z_i) = 0$ must be held, which implies $MI(F_i; Z_i) = H(F_i)$.

□

ACKNOWLEDGMENTS

We sincerely thank Dr. RaviChander Vipperla from Amazon for his valuable comments and careful proof reading.

REFERENCES

[1] T. W. Picton, S. A. Hillyard, R. Galambos, and M. Schiff, "Human auditory attention: a central or peripheral process?" *Science*, vol. 173, no. 3994, pp. 351–353, 1971.

[2] L. A. Werner and G. C. Marean, *Human auditory development*. Routledge, 2019.

[3] J. F. Werker and S. Curtin, "PRIMIR: A developmental framework of infant speech processing," *Language learning and development*, vol. 1, no. 2, pp. 197–234, 2005.

[4] K. Johnson and M. J. Sjerps, "Speaker normalization in speech perception," *The handbook of speech perception*, pp. 145–176, 2021.

[5] W.-N. Hsu, Y. Zhang, and J. Glass, "Unsupervised learning of disentangled and interpretable representations from sequential data," *Advances in neural information processing systems*, vol. 30, 2017.

[6] S. Shechtman and A. Sorin, "Sequence to sequence neural speech synthesis with prosody modification capabilities," in *Proc. 10th ISCA Speech Synthesis Workshop*, pp. 275–280.

[7] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5180–5189.

[8] W. H. Kang, S. H. Mun, M. H. Han, and N. S. Kim, "Disentangled speaker and nuisance attribute embedding for robust speaker verification," *IEEE Access*, vol. 8, pp. 141 838–141 849, 2020.

[9] L. Li, D. Wang, Y. Chen, Y. Shi, Z. Tang, and T. F. Zheng, "Deep factorization for speech signal," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 5094–5098.

[10] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Transactions on speech and audio processing*, vol. 6, no. 2, pp. 131–142, 1998.

[11] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *international conference on machine learning*. PMLR, 2019, pp. 4114–4124.

[12] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[13] 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 *International Conference on Learning Representations*, 2018.

[14] D. Wang, L. Deng, Y. T. Yeung, X. Chen, X. Liu, and H. Meng, "VQMIVC: Vector quantization and mutual information-based unsupervised speech representation disentanglement for one-shot voice conversion," *arXiv preprint arXiv:2106.10132*, 2021.

[15] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *arXiv preprint physics/0004057*, 2000.

[16] S. Yuan, P. Cheng, R. Zhang, W. Hao, Z. Gan, and L. Carin, "Improving zero-shot voice style transfer via disentangled representation learning," in *International Conference on Learning Representations*, 2020.

[17] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2016, pp. 1–6.

[18] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "AutoVC: Zero-shot voice style transfer with only autoencoder loss," in *ICML*. PMLR, 2019, pp. 5210–5219.

[19] A. H. Jha, S. Anand, M. Singh, and V. Veeravasarapu, "Disentangling factors of variation with cycle-consistent variational auto-encoders," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 805–820.

[20] R. H. Jones *et al.*, *Reductionism: Analysis and the fullness of reality*. Bucknell University Press, 2000.

[21] P. S. Churchland, *Neurophilosophy: Toward a unified science of the mind-brain*. MIT press, 1989.

[22] P. W. Anderson, "More is different: broken symmetry and the nature of the hierarchical structure of science." *Science*, vol. 177, no. 4047, pp. 393–396, 1972.

[23] K. Qian, Y. Zhang, S. Chang, M. Hasegawa-Johnson, and D. Cox, "Unsupervised speech decomposition via triple information bottleneck," in *ICML*. PMLR, 2020, pp. 7836–7846.

[24] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[25] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[26] 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.

[27] I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, and A. Lerchner, "Towards a definition of disentangled representations," *arXiv preprint arXiv:1812.02230*, 2018.

[28] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," *Advances in neural information processing systems*, vol. 29, 2016.

[29] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-VAE: Learning basic visual concepts with a constrained variational framework," in *International Conference on Learning Representations*, 2016.

[30] H. Kim and A. Mnih, "Disentangling by factorising," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2649–2658.

[31] R. T. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud, "Isolating sources of disentanglement in variational autoencoders," *Advances in neural information processing systems*, vol. 31, 2018.

[32] S. Liu, L. Zhang, X. Yang, H. Su, and J. Zhu, "Unsupervised part segmentation through disentangling appearance and shape," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8355–8364.

[33] A. Hyvarinen, H. Sasaki, and R. Turner, "Nonlinear ICA using auxiliary variables and generalized contrastive learning," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 859–868.

[34] H. Hälvä and A. Hyvarinen, "Hidden Markov nonlinear ICA: Unsupervised learning from nonstationary time series," in *Conference on Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 939–948.

[35] D. Klindt, L. Schott, Y. Sharma, I. Ustyuzhaninov, W. Brendel, M. Bethge, and D. Paiton, "Towards nonlinear disentanglement in natural data with temporal sparse coding," *arXiv preprint arXiv:2007.10930*, 2020.

[36] A. Dundar, K. Shih, A. Garg, R. Pottorff, A. Tao, and B. Catanzaro, "Unsupervised disentanglement of pose, appearance and background from images and videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[37] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[38] E. Helander, T. Virtanen, J. Nurminen, and M. Gabbouj, "Voice conversion using partial least squares regression," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 912–921, 2010.

[39] L.-H. Chen, Z.-H. Ling, L.-J. Liu, and L.-R. Dai, "Voice conversion using deep neural networks with layer-wise generative training," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1859–1872, 2014.

[40] T. Nakashika, T. Takiguchi, and Y. Ariki, "Voice conversion based on speaker-dependent restricted boltzmann machines," *IEICE TRANSACTIONS on Information and Systems*, vol. 97, no. 6, pp. 1403–1410, 2014.

[41] S. Desai, A. W. Black, B. Yegnanarayana, and K. Prahallad, "Spectral mapping using artificial neural networks for voice conversion," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 954–964, 2010.

[42] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 4869–4873.

[43] T. Nakashika, T. Takiguchi, and Y. Ariki, "High-order sequence modeling using speaker-dependent recurrent temporal restricted boltzmann machines for voice conversion," in *Fifteenth annual conference of the international speech communication association*, 2014.

[44] R. Takashima, T. Takiguchi, and Y. Ariki, "Exemplar-based voice conversion using sparse representation in noisy environments," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 96, no. 10, pp. 1946–1953, 2013.

[45] Z. Wu, T. Virtanen, E. S. Chng, and H. Li, "Exemplar-based sparse representation with residual compensation for voice conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1506–1521, 2014.

[46] T. Kaneko and H. Kameoka, "Parallel-data-free voice conversion using cycle-consistent adversarial networks," *arXiv preprint arXiv:1711.11293*, 2017.

[47] T. Kaneko and H. Kameoka, "CycleGAN-VC: Non-parallel voice conversion using cycle-consistent adversarial networks," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 2100–2104.

[48] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "CycleGAN-VC2: Improved cyclegan-based non-parallel voice conversion," in *2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2019, pp. 6820–6824.

[49] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "StarGAN-VC: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 266–273.

[50] M. Luong and V. A. Tran, "Many-to-many voice conversion based feature disentanglement using variational autoencoder," *arXiv preprint arXiv:2107.06642*, 2021.

[51] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[52] D.-Y. Wu, Y.-H. Chen, and H.-y. Lee, "VQVC+: One-shot voice conversion by vector quantization and U-net architecture," in *Interspeech*, 2020, pp. 4691–4695.

[53] J.-c. Chou and H.-Y. Lee, "One-shot voice conversion by separating speaker and content representations with instance normalization," in *Interspeech*, 2019, pp. 664–668.

[54] F.-L. Xie, F. K. Soong, and H. Li, "A KL divergence and dnn-based approach to voice conversion without parallel training sentences." in *Interspeech*, 2016, pp. 287–291.

[55] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *2016 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2016, pp. 1–6.

[56] J.-h. Lin, Y. Y. Lin, C.-M. Chien, and H.-y. Lee, "S2VC: A framework for any-to-any voice conversion with self-supervised pretrained representations," *arXiv preprint arXiv:2104.02901*, 2021.

[57] W.-C. Huang, Y.-C. Wu, and T. Hayashi, "Any-to-one sequence-to-sequence voice conversion using self-supervised discrete speech representations," in *2021 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2021, pp. 5944–5948.

[58] F. Kreuk, A. Polyak, J. Copet, E. Kharitonov, T.-A. Nguyen, M. Rivière, W.-N. Hsu, A. Mohamed, E. Dupoux, and Y. Adi, "Textless speech emotion conversion using decomposed and discrete representations," *arXiv preprint arXiv:2111.07402*, 2021.

[59] R. Peri, H. Li, K. Somandepalli, A. Jati, and S. Narayanan, "An empirical analysis of information encoded in disentangled neural speaker representations," in *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, 2020, pp. 194–201.

[60] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "ACVAE-VC: Non-parallel many-to-many voice conversion with auxiliary classifier variational autoencoder," *arXiv preprint arXiv:1808.05092*, 2018.

[61] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks," *arXiv preprint arXiv:1704.00849*, 2017.

[62] J.-c. Chou, C.-c. Yeh, H.-y. Lee, and L.-s. Lee, "Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations," in *Interspeech*, 2018, pp. 501–505.

[63] O. Ocal, O. H. Elibol, G. Keskin, C. Stephenson, A. Thomas, and K. Ramchandran, "Adversarially trained autoencoders for parallel-data-free voice conversion," in *2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2019, pp. 2777–2781.

[64] J. Wang, J. Li, X. Zhao, Z. Wu, and H. Meng, "Adversarially learning disentangled speech representations for robust multi-factor voice conversion," *arXiv preprint arXiv:2102.00184*, 2021.

[65] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*, 2017, pp. 2223–2232.

[66] S. Lee, B. Ko, K. Lee, I.-C. Yoo, and D. Yook, "Many-to-many voice conversion using conditional cycle-consistent adversarial networks," in *2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2020, pp. 6279–6283.

[67] P. L. Tobing, Y.-C. Wu, T. Hayashi, K. Kobayashi, and T. Toda, "Non-parallel voice conversion with cyclic variational autoencoder," in *Interspeech*, 2019, pp. 674–678.

[68] K. Matsubara, T. Okamoto, R. Takashima, T. Takiguchi, T. Toda, Y. Shiga, and H. Kawai, "High-intelligibility speech synthesis for dysarthric speakers with LPCNet-based TTS and CycleVAE-based VC," in *2021 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2021, pp. 7058–7062.

[69] M. Samarin, V. Nesterov, M. Wieser, A. Wieczorek, S. Parbhoo, and V. Roth, "Learning conditional invariance through cycle consistency," in *DAGM German Conference on Pattern Recognition*. Springer, 2021, pp. 376–391.

[70] K. Qian, Y. Zhang, S. Chang, J. Xiong, C. Gan, D. Cox, and M. Hasegawa-Johnson, "Global rhythm style transfer without text transcriptions," *arXiv preprint arXiv:2106.08519*, 2021.

[71] J. A. Fodor and Z. W. Pylyshyn, "Connectionism and cognitive architecture: A critical analysis," *Cognition*, vol. 28, no. 1-2, pp. 3–71, 1988.

[72] F. J. Stevenson, *Humus chemistry: genesis, composition, reactions*. John Wiley & Sons, 1994.

[73] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[74] P. Cheng, W. Hao, S. Dai, J. Liu, Z. Gan, and L. Carin, "Club: A contrastive log-ratio upper bound of mutual information," in *International conference on machine learning*. PMLR, 2020, pp. 1779–1788.

[75] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," *University of Edinburgh. CSTR*, 2017.

[76] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE international con-*

ference on acoustics, speech and signal processing (ICASSP). IEEE, 2014, pp. 4052–4056.
[77] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 022–17 033, 2020.
[78] C.-C. Lo, S.-W. Fu, W.-C. Huang, X. Wang, J. Yamagishi, Y. Tsao, and H.-M. Wang, "MOSNet: Deep learning-based objective assessment for voice conversion," in *Interspeech*, 2019, pp. 1541–1545.
[79] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

Haoran Sun received the B.Sc. degree in the department of automation from Tsinghua University in 2018, and M.Sc. degree in computer science from Tsinghua University in 2022. His research interest is speech analysis and voice conversion.

Dong Wang received the B.Sc. and M.Sc. degrees in computer science from Tsinghua University in 1999 and 2002. He received the Ph.D. degree (supported by a Marie Curie fellowship) from CSTR, University of Edinburgh, in 2010. He was employed with Oracle China during 2002 to 2004, and IBM China during 2004 to 2006. He joined CSTR, University of Edinburgh, in 2006 as a Research Fellow. From 2010 to 2011, he was with EURECOM as a Postdoctoral Fellow, and from 2011 to 2012, was a Senior Research Scientist with Nuance. He is now an Associate Professor with Tsinghua University, Beijing, China.

Lantian Li received the B.Sc. degree from China University of Mining and Technology, Beijing in 2013. He received the Ph.D. degree from the Department of Computer Science, Tsinghua University in 2018. Since 2018, he has been with the Center for Speech and Language Technology (CSLT), Tsinghua University as a postdoctoral fellow. His research interest is speaker recognition with machine learning methods.

Chen Chen recieved the B.Sc. degree in the department of automation from Tsinghua University in 2018, and is now a master student with the department of computer science and technology at Tsinghua University. His research interest is audio-visual multi-modal processing with machine learning methods.

Thomas Fang Zheng received the Ph.D. degree in computer science and technology from Tsinghua University, Beijing, China, in 1997. He is now a Research Professor and Director of the Center for Speech and Language Technologies, Tsinghua University. His research focuses on speech and language processing.