CONTROLLABLE SPEECH REPRESENTATION LEARNING VIA VOICE CONVERSION AND AIC LOSS

Yunyun Wang[1] *Jiaqi Su*[1] *Adam Finkelstein*[1] *Zeyu Jin*[2]

[1]Princeton University [2]Adobe Research

ABSTRACT

Speech representation learning transforms speech into features that are suitable for downstream tasks, e.g. speech recognition, phoneme classification, or speaker identification. For such recognition tasks, a representation can be lossy (non-invertible), which is typical of BERT-like self-supervised models. However, when used for synthesis tasks, we find these lossy representations prove to be insufficient to plausibly reconstruct the input signal. This paper introduces a method for invertible and controllable speech representation learning based on disentanglement. The representation can be decoded into a signal perceptually identical to the original. Moreover, its disentangled components (content, pitch, speaker identity, and energy) can be controlled independently to alter the synthesis result. Our model builds upon a zero-shot voice conversion model AutoVC-F0, in which we introduce alteration invariant content loss (AIC loss) and adversarial training (GAN). Through objective measures and subjective tests, we show that our formulation offers significant improvement in voice conversion sound quality as well as more precise control over the disentangled features.

Index Terms— representation learning, voice conversion

1. INTRODUCTION

Representation learning transforms complex signals into features more suitable for downstream tasks. Self-supervised learning such as BERT [1] and its successor GPT-3 [2] play a key role in state-of-the-art methods in language processing and computer vision. For audio, BERT-like self-supervised models like Wav2Vec [3] and HuBERT [4] have demonstrated success in speech speech recognition, phoneme classification and speaker identification [5, 6, 7, 8]. The learned representation provides a more robust solution to these downstream tasks while dramatically reducing the need for labeled data. However, the encoded representations suffer from several drawbacks. First, they are typically lossy (non-invertible) as the training process does not optimize for reconstruction. While not problematic for the aforementioned recognition tasks, we show that the resulting representation is not suitable for synthesis tasks due to information loss. Second, the fine-tuning steps are essential to improve the learned representation for a downstream task, but also make the model domain-specific. Fine-tuning for English phoneme classification, for example, improves the performance on that task, but makes the model less generalizable to other languages. Finally, models such as Wav2Vec focus on encoding one aspect of speech signal, especially after fine-tuning. While the representation fine-tuned on phonemes performs well for speech content-related tasks, it is less suitable for other aspects of speech such as the recording environment, quality, prosody, and speaking style. This paper introduces a strategy for speech representation learning that is both invertible (making it suitable for synthesis tasks) and controllable – disentangled components can be manipulated independently, including content, pitch, speaker identity, and energy.

Our approach is inspired by recent work in *voice conversion*. Many-to-many voice conversion has been a challenging task due to the lack of parallel speech data. Existing approaches are generally based on either speech recognition or auto-encoders. Speech recognition-based methods combine automatic speech recognition (ASR) and text-to-speech (TTS) directly [9, 10] or synthesizing from intermediate features from speech recognition [11, 12]. Assuming ASR is near perfect, the converted speech often sounds natural and clean. However, this method is often language-specific, and detailed information about the speaker and prosody is lost during recognition. The autoencoder-based separates and manipulates speaker and content information by bottleneck constraints [13, 14, 15] or cross domain feature disentanglement [16]. They tend to preserve prosody and accent, and can be language-independent; but the synthesis quality tends to reduce due to information loss at bottleneck and disentanglement.

Our approach specifically builds on advances in zero-shot voice conversion – especially AutoVC [13] and AutoVC-F0 [14, 15] – that disentangle speaker identity and pitch from speech content, while achieving perceptually plausible reconstructions (meaning they are invertible). They are also naturally language-agnostic. AutoVC uses an encoder to extract a code (content) from the mel-spectrogram of the input speaker, and reconstructs the mel-spectrogram by combing the code, speaker identity embedding and pitch in the decoder. The AutoVC paper shows that invertibility coupled with a properly tuned bottleneck guarantees perfect disentanglement, meaning the output can be controlled by altering different representations independently. However, the bottleneck for the code space is sensitive to the architecture and must be carefully crafted – architectures that deviate from their original design tend to result in a significant reduction in reconstruction quality.

Our proposed model ameliorates these concerns. We introduce a new alteration invariant content (AIC) loss that maps together the code-spaces of two utterances with the same speech content spoken by different people. We also add adversarial training to further improve the quality of the output mel-spectrogram. The AIC loss avoids speaker identity from leaking through, thus allowing us to use a larger bottleneck size for better synthesis quality. The model is able to synthesize speech reliably, and provides independent control of content, speaker identity, pitch and energy. Finally the model itself is language independent.

We describe objective experiments showing that our method can provide effective control over pitch and energy. In addition we describe subjective experiments showing that our method is able to reconstruct plausible voice – while providing independent control of speaker identity, comparing favorably with baseline methods. Our trained models, as well as listening examples, are available here: `https://pixl.cs.princeton.edu/pubs/Wang_2022_CSR/`

2. METHOD

The basis of our approach originates from a source-filter view of speech signal. The source is either periodic signal or noise, characterized by VUV (voiced vs unvoiced) and F0 (aka pitch, in Hz or semitones). The filter contains content which relates to muscle control that produces the filter of words and speaking style, referring to the acoustic properties of the speaker's vocal tract. Conceptually these aspects can be manipulated independently and are sufficient in determining the original speech. This conceptual model has been supported in AutoVC-F0 as their architecture disentangles F0 and speaker identity from the rest of the information (content) allowing manipulation of each representation without altering the others.

We denote i-th speaker's voice identity as a vector s_i and u_j as the j-th utterance content code which contains the content and rhythm of the speech. Let $A_{s_i}^{u_j}$ be the real audio of speaker s_i speaking the content u_j with F0 sequence $f_{s_i}^{u_j}$. Note that u_j is unique to every speaker s_i as every speaker speaks differently, even for the same text. Let $M_{s_i}^{u_j} = \text{melspec}(A_{s_i}^{u_j})$ be the mel-spectrogram of corresponding audio $A_{s_i}^{u_j}$.

An ideal content encoder E should be able to extract u_j from $M_{s_i}^{u_j}$ and an ideal decoder D should be able to reconstruct $M_{s_i}^{u_j}$ given u_j, s_i and $f_{s_i}^{u_j}$:

$$E(M_{s_i}^{u_j}) = u_j, D(u_j, s_i, f_{s_i}^{u_j}) = M_{s_i}^{u_j} \tag{1}$$

We aim to design E and D such that (1) reconstructed speech $D(E(M_{s_i}^{u_j}), s_i, f_{s_i}^{u_j})$ and $M_{s_i}^{u_j}$ are perceptually identical, and (2) altering s and f makes decoded speech sounds like Speaker s' speaking the same content with a new F0 f'. To achieve this, we propose several objectives (Sec.2.2): mel-spec loss, self content loss, AIC loss, and adversarial loss.

2.1. Model

Our model (Fig. 1) builds upon the AutoVC-F0 [14] decoder. Unlike the AutoVC-F0, we design a convolution-based encoder instead of LSTM-based encoder, to extract code space decoupled from long-term context dependency. This obtains a more interpretable code space with clearer correspondence to local content (i.e. phonemes).

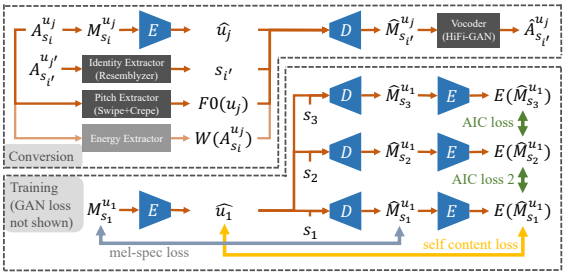

Fig. 1. The framework of our autoencoder model, shown for voice conversion (above) with training architecture (below). Two types of AIC loss are shown in green.

Encoder. The encoder consists of 6 stacks of convolution layers with kernel size 5 and stride 1, each following groupnorm of group size 32 and ReLU activation. It takes in 80-coefficient mel-spectrogram, and outputs a sequence of codes at the same temporal resolution as the input. The convolution channel sizes are $[80, 512, 512, 512, 512, 512, \textit{neckDim}]$. AutoVC-F0 uses a resolution of $32 \times \frac{1}{8}$ for the bottleneck. In experiments, we choose a *neckDim* of 8 to include more information with a larger code space.

Decoder. The decoder first stacks all input features (content code, speaker embedding, pitch and possibly energy) along the feature dimension and feeds the resulting sequence to a LSTM with hidden size 512. Then it goes through 3 stacks of convolution layers similar to the encoder with channel sizes $[512, 512, 512, 512]$ following by 2-layer LSTM with hidden size 1024. Finally, the linear projection layer projects the features from dimension 1024 to 80. To add more details to the output mel-spectrogram, we add one postnet identical to the implementation of AutoVC-F0. The output of the decoder and the output of the postnet are added together as the final output of the decoder. The postnet has 6 convolution layers similar to encoder with channel sizes $[80, 512, 512, 512, 512, 512, 80]$.

Generator and Discriminator. We notice that the over-smoothing effect caused by mel-spec loss is limiting the audio quality. Thus we use an adversarial network to refine the mel-spectrogram and reduce artifacts. The whole encoder-decoder architecture acts as the generator and the discriminator uses the same architecture from SpecGAN [17].

2.2. Objectives

During each iteration of training, we randomly select a speaker s_1 and a segment of real utterance $M_{s_1}^{u_1}$. We extract the content code $\hat{u}_1$ using the encoder and generate three mel-spectrograms $\hat{M}_{s_1}^{u_1}, \hat{M}_{s_2}^{u_1}, \hat{M}_{s_3}^{u_1}$ of the same code with three different speaker identities using the decoder. We apply the following objectives among the synthesized mel-spectrograms to force the encoder to disentangle speaker identity information and the decoder to reconstruct realistic speech.

Mel-spec Loss. Mel-spec loss is a L2 loss between the real mel-spectrogram and the self-to-self reconstructed mel-spectrogram. By restricting the synthesized self-to-self mel-spectrogram close to the real one, it ensures the reconstruction quality of the decoder.

$$\mathcal{L}_M = L2\left(M_{s_1}^{u_1}, \hat{M}_{s_1}^{u_1}\right) \tag{2}$$

Self Content Loss. Self content loss is a L1 loss between the extracted code of the real input and that of the self-to-self reconstructed output. It is used in AutoVC [13] and shares a similar idea to cycle consistency loss [18]. It enhances the robustness of the encoder as it is invariant to self-to-self reconstruction.

$$\mathcal{L}_S = L1\left(E(M_{s_1}^{u_1}), E(\hat{M}_{s_1}^{u_1})\right) \tag{3}$$

Alteration Invariant Content (AIC) Loss. AIC loss is our proposed content loss to improve synthesis quality and model robustness. In the AutoVC [13, 14] setting, the bottleneck between the encoder and decoder should be carefully crafted such that the code space fully contains content information necessary for a perfect reconstruction of the utterance, while no information in the condition leaks through. As such, AutoVC reduces the time resolution of the extracted codes, which leads to phoneme inaccuracy and low synthesis quality. Our AIC loss prevents speaker identity from leaking through by forcing the code spaces of any two different speakers to be close, and therefore permits a large bottleneck at the original time resolution that better preserves details of the input content. Specifically, during training, we convert a content code to two randomly selected speakers and calculate their L1 content code distance. We focus on invariance to alternation in speaker identity for AIC loss because our experiment shows pitch and energy are already well disentangled from the code space even without the AIC loss.

$$\mathcal{L}_C = L1\left(E(\hat{M}_{s_2}^{u_1}), E(\hat{M}_{s_3}^{u_1})\right) \tag{4}$$

Adversarial Loss. We adopt the hinge loss and feature matching loss from MelGAN [19] as our adversarial loss. The generator hinge loss and feature matching loss are combined with weight $1, 10$ to form the adversarial generator loss. We sum up the four losses with weight $1, 100, 100, 10$ respectively as our full objective for the generator, and use the hinge loss for the discriminator.

3. EXPERIMENTS

The model is trained from scratch using the mel-spec loss, self content loss and AIC loss for 400k iterations with a learning rate 10^{-4}. Next, we add GAN loss into training and train an additional 400k iterations with learning rate 10^{-5} for the generator and 10^{-6} for the discriminator. We update the discriminator and generator in turn each iteration. We use a batch size of 4 and Adam optimizer [20] on a GeForce RTX 3090 GPU. We use the VCTK dataset [21] for training and evaluation. The last 10 speakers and first 10 utterances for each speaker are held for testing. Each clip is preprocessed through a Butterworth highpass filter at 30Hz to remove low frequency noise.

To produce accurate fundamental frequencies (F0), we use CREPE [22] to estimate the pitch range for the utterance and then guide SWIPE [23] to calculate the final log-F0. The log-F0s are then normalized into 257 bins with 256 effective pitches and 1 unvoiced bin. To compare how normalization affects the generated audio quality, we try two different normalization strategies: absolute normalization, and relative normalization. For frequency f Hz, absolute normalization normalizes f to $\frac{\log(f)-\log(f_{\min})}{\log(f_{\max})-\log(f_{\min})}$ with $f_{\min} = 40, f_{\max} = 400$ chosen. Similar to how AutoVC-F0[14] does speaker normalization, relative normalization normalizes f to $\frac{\log(f)-\text{mean}_{s_i}}{4\text{std}_{s_i}}$ where mean_{s_i} and std_{s_i} are the mean and standard deviation of the log frequency of speaker s_i.

For the identity extractor, we use the pretrained Resemblyzer [24], a version of speaker encoder from GE2E [25]. Resemblyzer produces a speakers embedding of dimension 256. The speaker identity embeddings of the utterances from the same speaker are averaged as the identity embedding for the speaker. The energy is calculated from raw waveform directly. We use the pretrained HiFi-GAN vocoder [26] to synthesize audio at 22050 sample rate from the predicted mel-spectrogram at the last step.

3.1. Ablation Study

We conduct ablation study on three design choices to justify our final approach: **(a)** Absolute pitch vs. relative pitch **(b)** GAN training vs. no GAN training **(c)** AIC loss vs. AIC loss 2 vs. no AIC loss. Variations of our models are trained and compared:

Our first main model is denoted as **Ours-AC** where **A** and **C** indicate that we're using **A**bsolute pitch and AI**C** loss. To explore the potential of **Ours-AC**, we also extend its output audio to 48k (**Ours-AC-48k**) using a bandwidth extension model [27]. **Ours-AC-noGAN** is **Ours-AC** without GAN training. **Ours-ACE** is **Ours-AC** with additional condition on **E**nergy. **Ours-RC** is with **R**elative pitch and AI**C** loss. **Ours-A** and **R** are two models with **A**bsolute pitch and **R**elative pitch respectively but without AI**C** loss. **Ours-AC2** is our second main model with **A**bsolute pitch and a different formulation of AI**C** loss which we call AI**C** loss **2**. Instead of choosing two different speakers s_2 and s_3 as in the original AIC loss, it uses one additional speaker s_2: $\mathcal{L}_{C2} = L1\left(E(\hat{M}^{u_1}_{s_1}), E(\hat{M}^{u_1}_{s_2})\right)$.

We compare different settings of our models through a MOS (Mean Opinion Score) test on quality and similarity using Amazon Mechanical Turk (Fig. 2). The subjects are required to pass a preliminary test before entering, making sure they have the suitable equipment and are able to distinguish audios of different quality. We add validation questions randomly during the test and filter out the subjects who answer randomly. In quality test, we collected 401 valid HITs across 176 unique workers, totaling 10426 answers for all the method conditions (including input and target). In similarity test, we collected 554 valid hits across 292 unique workers, totaling 14404 answers. Note that in similarity test, we set the high anchor as "two speakers sound like the same persons" and low anchor as "two speakers sound like two different persons" making it more difficult to get high ratings across board.

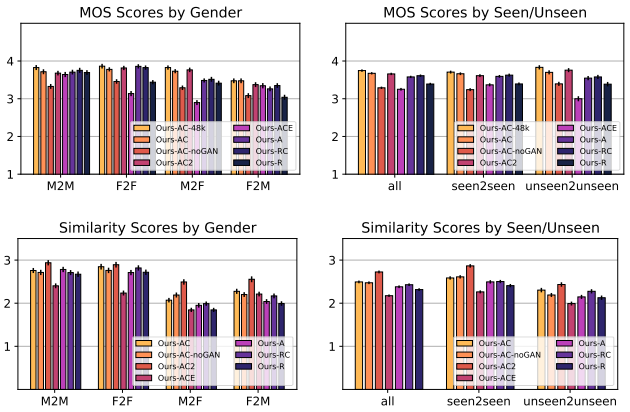

Fig. 2. Ablation. MOS scores (above) and similarity scores (below) model variants. Among these, we find that AIC loss, GAN training, and absolute pitch all play important roles in the complete model.

Absolute Pitch vs. Relative Pitch. Absolute pitch helps to generate a more natural speech (**Ours-AC** vs.**Ours-RC**, **Ours-A** vs.**Ours-R**). The relative pitch normalizes pitch by speaker pitch distribution and leaves the decoder to decide the actual pitch range based on the speaker identity. Sometimes the decoder generates an unstable pitch, in other words, gender flip, for cross-gender scenario.

GAN Training vs. No GAN Training. Ours-AC has higher quality MOS score than **Ours-AC-noGAN** in all comparison scenarios, and **Ours-AC** is on par with **Ours-AC-noGAN** for similarity. Adding GAN resolves the over-smoothing issue caused by $L2$ loss and introduces details in the mel-spectrogram which encourages the vocoder to generate a cleaner audio. Noticeable artifacts can be easily picked up by human ear when not using GAN.

AIC Loss vs. AIC Loss 2 vs. No AIC Loss. Comparison between (**Ours-AC**, **Ours-A**) and (**Ours-RC**, **Ours-R**) shows that adding AIC loss can improve the generation quality, especially in cross-gender and unseen2unseen scenarios. AIC loss contributes more significantly in the relative pitch setting by reducing gender flip while absolute pitch is experiencing less of that issue. Using AIC loss (**Ours-AC**) scores slightly higher than AIC loss 2 (**Ours-AC2**) overall, but AIC loss 2 is more robust to unseen speakers and has better performance in similarity test. We also notice that AIC loss leads to a more stable training curve than AIC loss 2. In conclusion, **Ours-AC2** is a more balanced choice but **Ours-AC** can be useful in some scenarios too.

3.2. Voice Conversion

Based on the analysis above, we use **Ours-AC**, **Ours-AC2** and **Ours-AC-48k** for voice conversion task. We conduct the same tests as described in Sec.3.1 for our models and baselines. (Fig. 3)

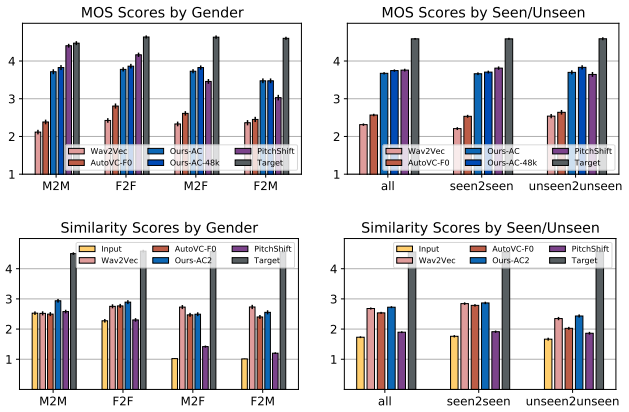

Fig. 3. MOS scores (above) and similarity scores (below) shows that our best models compare favorably with baselines across gender and seen/unseen speaker conversion cases.

PitchShift. We use the traditional pitch shifter PSOLA [28] to shift the mean absolute pitches of the input speaker to the mean absolute pitches of the target speaker. It is mainly for comparing the timbre difference between the models because it eliminates pitch difference.

AutoVC-F0. To match our experiment's setting, AutoVC-F0's [14] speaker identity extractor is changed to Resemblyzer. We notice a slight quality improvement from the original paper (20 speakers) on our training set (99 speakers).

Wav2Vec. The features of the input utterance are extracted using the large finetuned pretrained Wav2Vec 2.0 [3] model *wav2vec2-large-960h-lv60*, interpolated to match our time resolution and linearly projected to a code space of dimension 32 (much larger than ours). We then train our decoder on top of this fixed code space using mel-spec loss and self content loss. **Wav2Vec** constructs speaker identity reasonably. As is finetuned on phoneme recognition task, it naturally disentangles speaker information and thus leads to better speaker similarity in the decoder. However, it also loses other information besides phonemes that are crucial for high-quality speech. As a result, **Wav2Vec** representation leads to the lower audio quality, even though the same decoder architecture is used. **AutoVC-F0** scores low in MOS as well. It suffers from the low time resolution and sometimes blurs out a sequence of short phonemes. The generated F0 sometimes drifts off as its range is implicitly modeled using relative scales. **PitchShift** is a reference in which PSOLA [28] is used to alter the pitch without changing the timbre. Hence it has high quality score but low similarity scores. **Ours-AC** and **Ours-AC-48k** achieves a significantly higher MOS score and bandwidth extension improves audio quality noticeably. **Ours-AC2** wins by a small margin overall for the similarity test though is slightly behind **Ours-AC** in the MOS test. We conclude that **Ours-AC** and **Ours-AC2** have strength in quality and similarity respectively, and can be suitable for different tasks depending on the application.

3.3. Analysis

Pitch/Energy Control. We test our model's pitch control ability by conditioning the decoder on an linearly increasing F0. Unlike **Wav2Vec** and our models, **AutoVC-F0** cannot shift precisely to an absolute pitch scale. Hence, we take the target pitch contour, normalize it using the same strategy **AutoVC-F0** normalizes pitch and use it as condition for **AutoVC-F0**. We calculate the objective scores (Tab. 1) using $L1$ distance in semitone and Hz for the voiced part. We add the VUV (Voice vs Unvoiced) error which denotes the portion that a voiced segment is misclassified as unvoiced or vice versa. **Ours-AC2** and **PitchShift** give the best pitch control performance overall. For **Ours-ACE**, we are able to control the energy by conditioning on a new energy sequence. We calculate the $L2$ distance between the energy of the generated audio and the target energy. The test audios have an energy mean of 0.0386. When conditioning on $\frac{1}{2}$ of the original energy, we get a distance of 0.0049. When conditioning on a non-zero constant, we get a distance of 0.0160. Note that this larger distance is due to the unvoiced part. When conditioning on constant 0, we get a distance of 0.0061 and the generated audio is mainly noise.

Method	L1 (in semitone)	L1 (in Hz)	VUV error
AutoVC-F0	2.318	32.997	0.136
Wav2Vec	0.302	9.679	0.119
Ours-AC	0.312	12.281	0.082
Ours-AC2	0.213	**3.051**	**0.068**
PitchShift	**0.196**	3.385	0.082

Table 1. Objective scores for pitch control. **PitchShift** and our method perform best for this task, outperforming other baselines.

Code Space. Fig. 4 shows the code space of **Ours-AC2** (our best model) for two speakers uttering the same sentence (a,d); and that of the same two utterances where the pitch is shifted linearly using PSOLA [28] (b,e); with transcriptions (c,f). Each dimension of the code space appears as a horizontal band, sorted vertically by overall energy (only top 4 shown), values normalized to $[0, 1]$, and visualized with the "jet" colormap (blue=0, red=1). Constant-value bands are dark blue. The content codes are similar for the different speakers (a,d), and are almost identical for the modified pitches (b,e). This observation shows that our code space is invariant to pitch changes and speaker identity changes. Despite having 8 dimensions in the bottleneck, the model uses only 3 of them for content code; dimensions 4-8 are constant. This shows that our AIC loss succeeds at limiting additional information leaking into the code space.

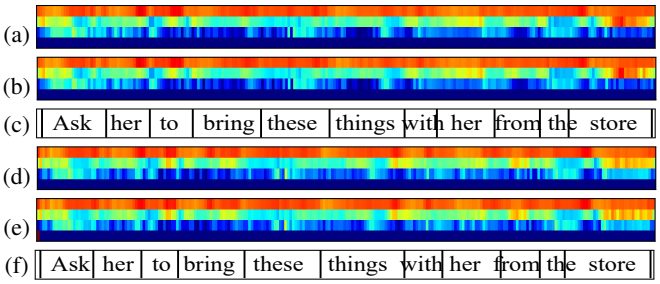

Fig. 4. Code space comparisons: the same sentence uttered by two different speakers (a,d); after pitch shift (b,e); and transcription (c,f).

4. CONCLUSION

This paper proposes an invertible speech representation learning model based on voice conversion. We show through experiments and analysis that the learned representation disentangles speech components (content, pitch, speaker identity, energy), which can be controlled separately to synthesize high-quality speech. Although the proposed model outperforms baselines, our listening test shows a gap in voice similarity among all tested voice conversion methods. Thus, one potential avenue for future work is to disentangle prosody for more accurate synthesis. There are a few other downstream tasks for future study, such as speech recognition using content code, disentangling prosodic style, and building multi-speaker text-to-speech synthesis by generating content codes and F0.

5. REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conf. of No. American Chapter of Assn. for Comp. Linguistics: Human Language Technologies, Volume 1*, 2018, pp. 4171–4186.

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, et al., "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 1877–1901.

[3] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Info. Proc. Sys.*, 2020, vol. 33, pp. 12449–12460.

[4] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, et al., "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *arXiv preprint arXiv:2106.07447*, 2021.

[5] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli, "wav2vec: Unsupervised pre-training for speech recognition.," in *Interspeech 2019*, 2019, pp. 3465–3469.

[6] Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James R. Glass, "An unsupervised autoregressive model for speech representation learning.," in *Interspeech 2019*, 2019, pp. 146–150.

[7] Jan Chorowski, Ron J. Weiss, Samy Bengio, and Aaron van den Oord, "Unsupervised speech representation learning using wavenet autoencoders," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2041–2053, 2019.

[8] Andy T. Liu, Shu wen Yang, Po-Han Chi, et al., "Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6419–6423.

[9] Seung Won Park, Doo young Kim, and Myun chul Joe, "Cotatron: Transcription-guided speech encoder for any-to-many voice conversion without parallel data.," in *Interspeech 2020*, 2020, pp. 4696–4700.

[10] Wen-Chin Huang, Tomoki Hayashi, Yi-Chiao Wu, et al., "Voice transformer network: Sequence-to-sequence voice conversion using transformer with text-to-speech pretraining," in *Interspeech 2020*, 2020, pp. 4676–4680.

[11] Adam Polyak, Yossi Adi, Jade Copet, et al., "Speech resynthesis from discrete disentangled self-supervised representations," in *Interspeech 2021*, 2021.

[12] Yinghao Aaron Li, Ali Zare, and Nima Mesgarani, "StarGANv2-VC: A diverse, unsupervised, non-parallel framework for natural-sounding voice conversion," in *Interspeech*, 2021.

[13] Kaizhi Qian, Yang Zhang, Shiyu Chang, et al., "AutoVC: Zero-shot voice style transfer with only autoencoder loss," in *ICML*, 2019, pp. 5210–5219.

[14] Kaizhi Qian, Zeyu Jin, Mark Hasegawa-Johnson, and Gautham J. Mysore, "F0-consistent many-to-many non-parallel voice conversion via conditional autoencoder," in *ICASSP*, 2020, pp. 6284–6288.

[15] Kaizhi Qian, Yang Zhang, Shiyu Chang, et al., "Unsupervised speech decomposition via triple information bottleneck," in *ICML*, 2020, vol. 1, pp. 7836–7846.

[16] Wen-Chin Huang, Hao Luo, Hsin-Te Hwang, et al., "Unsupervised representation disentanglement using cross domain features and adversarial learning in variational autoencoder based voice conversion," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 4, pp. 468–479, 2020.

[17] Chris Donahue, Julian J. McAuley, and Miller S. Puckette, "Adversarial audio synthesis," in *International Conference on Learning Representations*, 2018.

[18] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[19] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, et al., "Melgan: Generative adversarial networks for conditional waveform synthesis," in *Advances in Neural Information Processing Systems*, 2019, vol. 32, pp. 14881–14892.

[20] Diederik P. Kingma and Jimmy Lei Ba, "Adam: A method for stochastic optimization," in *ICLR 2015 : International Conference on Learning Representations 2015*, 2015.

[21] Christophe Veaux, Junichi Yamagishi, and Kirsten MacDonald, "Superseded - CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," *The Rainbow Passage which the speakers read out can be found in the International Dialects of English Archive.*, 2016.

[22] Jong Wook Kim, Justin Salamon, Peter Li, and Juan Pablo Bello, "Crepe: A convolutional representation for pitch estimation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 161–165.

[23] John G. Harris and Arturo Camacho, "Swipe: a sawtooth waveform inspired pitch estimator for speech and music," *Journal of the Acoustical Society of America*, vol. 124, no. 3, pp. 1638–1652, 2007.

[24] "Resemblyzer," https://github.com/resemble-ai/Resemblyzer.

[25] 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)*, 2018, pp. 4879–4883.

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

[27] Jiaqi Su, Yunyun Wang, Adam Finkelstein, and Zeyu Jin, "Bandwidth extension is all you need," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 696–700.

[28] F. Charpentier and M. Stella, "Diphone synthesis using an overlap-add technique for speech waveforms concatenation," in *ICASSP '86. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1986, vol. 11, pp. 2015–2018.