GR0: SELF-SUPERVISED GLOBAL REPRESENTATION LEARNING FOR ZERO-SHOT VOICE CONVERSION

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

[1]Princeton University [2]Adobe Research

ABSTRACT

Research in generative self-supervised learning (SSL) has largely focused on *local* embeddings for tokenized sequences. We introduce a generative SSL framework that learns a *global* representation that is disentangled from local embeddings. We apply this technique to jointly learn a global speaker embedding and a zero-shot voice converter. The converter modifies recorded speech to sound as if it were spoken by a different person while preserving the content, using only a short reference clip unavailable to the model during training. Listening experiments conducted on an unseen dataset show that our models significantly outperform SOTA baselines in both quality and speaker similarity for various datasets and unseen languages.

Index Terms— generative self-supervised global representation learning, cross-lingual zero-shot voice conversion

1. INTRODUCTION

Pretrained deep neural networks recently enabled a variety of impressive applications in NLP, imaging, speech recognition, among many other fields. Generative self-supervised pretraining is a form of unsupervised learning that forms a representation by reconstructing data or properties of the data without human labels. In NLP and speech, self-supervised learning (SSL) is often formulated around sequence prediction for discrete tokens, e.g. using context to predict missing text [1], discretized sound [2], or audio codec tokens [3]. The trained model then can be repurposed for downstream tasks such as text generation [4, 2], speech recognition [4], or speaker classification [5]. The primary advantage of SSL is the ability to leverage in-the-wild data. Research has shown significant performance gains by pretraining a large SSL model with crawled Internet data in order to improve subsequent models trained by supervised learning, e.g. wav2vec for speech recognition [6] and SEER for few-shot image classification [7].

While generative SSL has shown success in *local* representation learning, we introduce an approach to learning a *global* representation from unlabelled audio data, for generation. A global representation is a fixed-dimension real-valued vector that represents a certain property that is consistent across the entire input data, e.g., the topic of a paragraph – or in our applications, the vocal characteristics of an individual speaker. Conventional methods often use supervised contrastive learning (which requires labels) to learn such embeddings, e.g., d-vectors in learned speaker embedding [8]. Another approach, contrastive SSL, learns a common embedding for data that share the same global characteristics, for example instance-to-instance contrast in SimCLR [9]. However, unlike generative SSL, these models do not have the ability to generate instances after training [10].

This paper proposes an SSL framework to learn a global representation from unlabelled in-the-wild data via a generative process. We show its application to zero-shot *voice conversion* (VC) – the task of converting an utterance made by one person so that it sounds like a different (reference) voice, using a model trained without the voice of either speaker. Zero-shot VC methods can be broadly classified into two categories: one is based on auto-encoders [11, 12, 13], and the other is based on speech recognition and resynthesis [14, 15, 16]. Apart from these, NANSY [17] utilizes perturbation on speech components to learn disentangled components through synthesis. Style Tokens [18] trains a separate model to infer style tokens from reference audio samples while our approach learns style representation, audio-to-style, style-to-audio jointly without the need of text. VALL-E [19] is a language-model-based TTS method that can be used for VC but it doesn't preserve prosody from the input speech while ours does. The main difference between our method and these approaches is that these approaches often use a fixed pretrained speaker encoder to describe speaker information, whereas **GR0** jointly learns the speaker representation and generation via a single SSL framework while ensuring that the learned style representation contains no local information from the input data.

The key idea is as follows (Figure 1): we use *local* feature extractors to extract the non-global information of an input sequence, for example, speech content. Next we train a *global* encoder that we call a **GR0**-encoder, together with a decoder to reconstruct the input clip. Unlike Style Tokens [18], **GR0** is trained using a *second* sequence (clip) that shares the same global properties as the first sequence (e.g. speaker identity) but contains different local information (e.g. speech content). During training, the decoder reconstructs the input clip using its extracted local features coupled with the global embedding from the second clip. If the reconstruction is successful, the learned global embedding contains information invariant across local embeddings (hence disentangled) while capturing the information shared by both sequences. This SSL framework simultaneously learns a global embedding and a generator. One can control generation by modifying the global embedding to achieve VC. Our experiments with zero-shot VC show that the jointly-learned embedding and generator outperform baseline methods using contrastive pretrained embeddings.

The contributions of this work include: (1) We introduce **GR0**, a general SSL framework for disentangling global conditions. (2) We apply **GR0** to zero-shot VC – unseen speaker, utterance and dataset. (3) Experiments show that both applications achieve higher quality and speaker similarity than SOTA baselines. (4) Experiments also show that the learned global embedding is more effective for encoding speaker characteristics than a pretrained embedding based on contrastive learning that relies on the same decoder design. (5) We also introduce a data preprocessing approach that can produce a large single-speaker dataset for SSL from in-the-wild data. A subset of the listening examples from our experiments may be found here: `https://pixl.cs.princeton.edu/pubs/Wang_2024_GSG/`

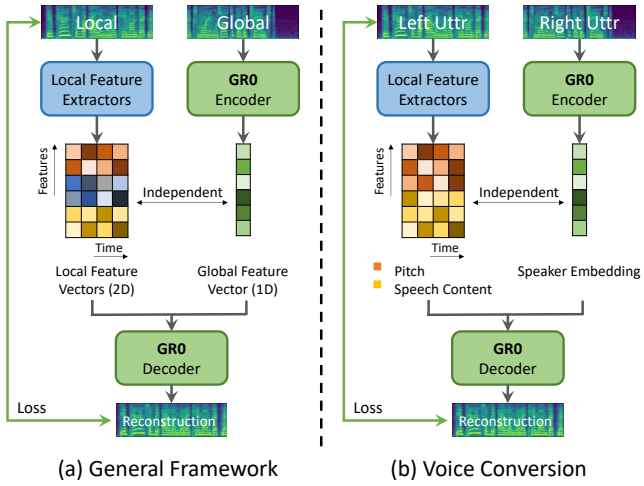

Fig. 1. An illustration of **GR0**. Voice conversion (b) is implemented based on the general framework (a). The green modules are trainable and the blue modules are fixed feature extractors. The extracted local feature vectors are independent of learned global feature vector because the two inputs share the same global information (speaker identity) but different local information (speech content).

2. METHOD

Our method employs generative self-supervised learning to capture a global representation (e.g., speaker identity). For training, it requires two sequences with shared global information (e.g., same utterance excerpts) but differing local content. The first sequence undergoes local feature extraction (such as F0 and speech content) that produces a representation excluding the global information we aim to model. The second sequence goes through an encoder we call **GR0**-encoder that outputs a fixed-sized global embedding. Our hypothesis is that for any pairs of sequences that share a global feature, if a decoder can reconstruct the first sequence based on the local features and global embedding, the learned global embedding (1) effectively captures the shared global features and (2) remains independent of the local features. The choice of the two sequences and local features is how we control the definition of the global embedding. For instance, in voice conversion, if the local condition represents speaker-invariant content tied to phonemes, the global condition will encompass the speaker's identity and intonation patterns. However, if we include F0 in the local condition, the intonation pattern will no longer be considered as global information. Using two non-overlapping excerpts from the same single-speaker utterance, we can train on in-the-wild data without speaker labels.

2.1. General Framework

Our model (Figure 1) consists of two trainable modules: a global content encoder $\mathrm{Enc}_{\mathrm{GR0}}(\cdot)$ and a decoder $\mathrm{Dec}_{\mathrm{GR0}}(\cdot,\cdot,\cdot)$ for reconstruction. Collectively, they are referred to as the generator G. We use two input audios x_1 and x_2 which have shared global information but different local information. Our **GR0**-encoder summarizes the global information shared by the two inputs (e.g., voice characteristics for utterances from the same speaker) from mel-spectrogram x_2 and outputs a 1-dimensional global embedding vector g with a size of 256, inspired by the formulation of Resemblyzer [8]. The global embedding is then repeated over the time axis and concatenated with the local content embeddings obtained from x_1. The decoder learns to reconstruct the mel-spectrogram of x_1 from the concatenated embeddings. We choose a transformer-based decoder for our final model after comparison (website Appendix). We adopt a postnet from previous work [11, 12] to add details to the generated mel-spectrogram from the decoder. Finally, we synthesize the output audio from the output mel-spectrogram using vocoder [20].

Loss Function. The reconstruction loss is the sum of two L2 losses, one for the mel-spectrogram prediction before the postnet, and one for the final output mel-spectrogram:

$$\mathcal{L}_{recon} = \mathbb{E}_s\left[\|X_1 - G'(s)\|_2^2\right] + \mathbb{E}_s\left[\|X_1 - G(s)\|_2^2\right]$$

where s is (x_1, x_2) the pair of input audios, X_1 is the mel-spectrogram of the audio x_1 for reconstruction, G is the generator, and G' is the generator without the postnet. The generator extracts local information from x_1 and learns global information from x_2 in order to reconstruct X_1.

GAN Training. In our experiments, minimizing L2 loss almost perfectly reconstructs the input mel-spectrogram, but the audio quality is limited by the commonly observed over-smoothing effect where unpredictable noises collapse into the average. Therefore, we apply adversarial training after the model is sufficiently converged with the reconstruction loss to improve the generation quality further. The discriminator is adapted from the SpecGAN discriminator [21]. We use the hinge loss objective [22, 23] together with the feature matching loss [24, 25]:

$$\begin{aligned}\mathcal{L}_{adv}(D;G) &= \mathbb{E}_{X_1}\left[\min(0, 1 - D(X_1))\right] \\ &\quad + \mathbb{E}_s\left[\min(0, 1 + D(G(s)))\right] \\ \mathcal{L}_{FM}(G;D) &= \mathbb{E}_s\left[\frac{1}{N}\|D_F(X_1) - D_F(G(s))\|_1\right] \\ \mathcal{L}_{adv}(G;D) &= \mathbb{E}_s\left[-D(G(s))\right]\end{aligned}$$

where D is the discriminator, and D_F is the discriminator's feature maps, defined as the activations of the layers before the output layer in the discriminator. N is the total number of features in the discriminator feature layer. Our final loss is calculated as below, where we set $\lambda_{adv} = 10^5$ and $\lambda_{FM} = 10$ to scale the objectives to the same range:

$$\begin{aligned}\mathcal{L}_G &= \mathcal{L}_{recon} + \lambda_{adv}\left(\mathcal{L}_{\mathrm{adv}}(G;D) + \lambda_{FM}\mathcal{L}_{FM}(G;D)\right) \\ \mathcal{L}_D &= \lambda_{adv}\mathcal{L}_{adv}(D;G)\end{aligned}$$

2.2. Voice Conversion

Our design is depicted in Figure 1. The generator G incorporates two fixed modules for local information extraction: a pre-trained content encoder $\mathrm{Enc}_{\mathrm{local}}(\cdot)$ and a pitch extractor $\mathrm{Pitch}(\cdot)$. Given an input audio waveform x, we divide it into two halves x_1 and x_2. The first half is for extracting its local representations including the speech content $C = \mathrm{Enc}_{\mathrm{local}}(x_1)$ and the pitch (i.e. F0) $P = \mathrm{Pitch}(x_1)$; the second half is used for learning a global representation $g = \mathrm{Enc}_{\mathrm{GR0}}(x_2)$.

For the speech content, we use the last layer output of wav2vec 2.0 (`wav2vec2-large-960h-lv60-self`) fine-tuned with CTC loss [4], which based on our experiment, does not contain speaker information. We use nearest neighbor interpolation to match our time resolution to that of wav2vec. Other speaker invariant speech features such as HuBERT [2] or linguistic features in NANSY [17] can also be an option. For the pitch, we use CREPE [26] to extract the fundamental frequency (F0) for all utterances. Then we compute the F0 range to guide

SWIPE [27] to extract the F0 from the speech. This practice ameliorates the known double-frequency issue in SWIPE and also overcomes an issue of CREPE where the F0 contour is over-smoothed. Then we compute the log of F0 and normalize it using the mean and standard deviation of this utterance following this format $(\log \text{F0} - \text{mean}(\log \text{F0})) \,/\, (4\ \text{std}(\log \text{F0}))$ and clip the value between 0 and 1. We quantize the range of 0-1 into 256 bins, and create a 257-dimensional one hot vector to present the F0 with one additional dimension as a binary indicator for voiced/unvoiced.

Intuitively, if the input audio is clean and the local representations are phonetic information and F0 (pitch), the disentangled global embedding should contain speaker information that is invariant of content and F0. Our experiments in Section 3.1 verify that the learned global embedding captures sufficient vocal characteristic information for driving a voice conversion task. Thus, we refer to this embedding as a speaker embedding. Next, a decoder combines the speaker embedding and the local embeddings to reconstruct the first half of the input mel-spectrogram: $\hat{X_1} = \text{Dec}_{\text{GR0}}(g, C, P)$.

3. EXPERIMENTS

Podcast Dataset. We derived our training dataset from the Spotify Podcast dataset [28] which consists of 47K hours of transcribed audio over 100K podcast episodes. To eliminate music, noise, and speaker cross-talk, we developed a data processing pipeline to obtain single-speaker samples: for every file, we first label find frames with speech using voice activity detector (VAD) [29]. We extract continuous utterances that are longer than 15 seconds and contain a single speaker. We used NISQA [30] to further assess the audio quality [31] retain samples have MOS of 4.3 or higher. This step effectively removes the bulk of multi-speaker and non-clean data. This strict NISQA threshold leaves us 6127 hours of audio; to remove remaining noise and unify the audio quality, we apply HiFiGAN-2 [32] and bandwidth extension (BWE) [33] on the filtered utterances to obtain clean audio. Compared to the other datasets, the Podcast Dataset has greater speaker variety and more natural-sounding speech, but it contains no speaker labels. This procedure can also be used for ASR-oriented dataset to distill clean audios from in-the-wild data.

Mel-Spectrogram. We use the same mel-spectrogram format as from the HiFiGAN vocoder [20] for consistency in the vocoding stage, with 256 hop size, 1024 window size and FFT size, 80 bins, 0 min frequency, 8000 max frequency, and on a log scale. We use 22050 sample rate throughout the paper. Note that we do not normalize the value of the mel-spectrogram during the training.

Model Training. We train our voice conversion model on 8 Nvidia A100 GPUs using a batch size of 128 and the Adam optimizer with a learning rate of $3e$-5. During training, we randomly sample an excerpt of mel-spectrogram with a shape of 1024×80 from a random utterance, where we use the first half (512×80) for generation and the latter half (512×80) for global summarization. We train our models 1M steps before adding GAN loss and feature matching loss. With GAN training, the generator and discriminator are trained with a learning rate of $1e$-5 for another 100k steps.

3.1. Evaluation Settings

For zero-shot voice conversion, the quality of the converted voice and the similarity between the target voice and the converted voice indicate how well the speaker embeddings can capture speaker characteristics. Therefore, we compare our approach to the supervised speaker embedding method Resemblyzer [8], as often used in prior voice conversion works. We also demonstrate that a voice converter trained using **GR0** outperforms other zero-shot voice converters. To truly test generalizability, our evaluation is performed on unseen datasets and cross-language conversions with unseen language.

DAPS voices. We use DAPS [34] as the evaluation dataset because none of our models or baselines were trained on DAPS. DAPS ensures consistent text across speakers, so it is more convenient to compare the results with the real recordings. We use the first 5 male and female speakers from DAPS respectively, and the first sentence of their 5 audio recordings, totaling 50 utterances. We name the 10 speakers $s_0, s_1, ..., s_9$ and the 5 utterances from each speaker $u_0, u_1, ..., u_4$. For each pair of speakers (s_i, s_j) $0 \leq i, j \leq 9$, we convert the content source $s_i u_k$ to target speaker s_j using reference $s_j u_{(k+1)\%5}, k = 0, 1, ..., 5$, totaling $10 \times 10 \times 5 = 500$ pairs of voice conversion samples.

VCTK voices. We also use VCTK dataset [35] to evaluate the cross-dataset generalizability of various methods. Unlike DAPS, which is collected in the USA, The speakers in the VCTK dataset often have British accent. Note that some of the baselines are trained on the VCTK dataset and are likely to perform better on VCTK than other datasets. We hope to demonstrate that our method maintains the same performance regardless of what dataset is used. We select the last 5 male speakers and the last 5 female speakers for testing because the last 10 gender-balanced speakers in VCTK are often held as test speakers. We avoid short utterances and use 1 long utterances (005) for testing. Thus we have 10×10 pairs of samples in total.

Cross language Conversion We also challenge our model with cross-language conversion. Similar to the setting of YourTTS [14], we convert between the 10 speakers in VCTK and 10 speakers in MLS Portuguese data [36]. We use the label EN→PT to convert from a source Portuguese utterance to a target English speaker, resulting a Portuguese utterance. Likewise, the label PT→EN denotes converting from a source English speech to a target Portuguese speaker, resulting an English utterance. The conversion between 10 speakers from VCTK and 10 speakers from MLS Portuguese gives us 10×10 pairs conversions for EN→PT and PT→EN each.

Baselines. We compare our model with a few notable zero-shot voice conversion baselines that are also based on representation learning. **AutovcF0** [11] achieves zero-shot voice conversion through an encoder-decoder architecture with a certain sized bottleneck. We trained AutovcF0 with the original paper's setting on VCTK. **AutovcAIC** [12] builds upon AutovcF0 and adds alteration invariant content loss (AIC loss) to lift the bottleneck restriction and improve the generation quality. We implemented the model AutovcAIC and trained an additional AutovcAIC model on the Spotify Podcast dataset from scratch as **AutovcAIC-spotify**, to investigate the difference brought by training data. **YourTTS** [14] is a state-of-the-art multi-speaker TTS model and can also be used in zero-shot voice conversion. We adopt the released multilingual voice conversion model from YourTTS GitHub repository. Finally, as a global representation baseline, we evaluate **Resemblyzer** that learns speaker identity embedding through contrastive learning [8] and is widely used as speaker encoder in recognition and synthesis tasks. We use the same architecture, replacing our learnable **GR0**-encoder with a pre-trained Resemblyzer as a baseline for comparison.

3.2. Mean-Opinion-Scores (MOS) Evaluation

For every pair of source and target voice in Section 3.1, we obtain test samples from our methods and various baseline methods (Sec-

	DAPS		VCTK		EN→PT		PT→EN	
Exp.	**MOS**	**SIM-MOS**	**MOS**	**SIM-MOS**	**MOS**	**SIM-MOS**	**MOS**	**SIM-MOS**
GroundTruth	4.67 ± 0.04	4.26 ± 0.07	/	/	/	/	/	/
Source	/	1.87 ± 0.08	/	1.98 ± 0.14	/	1.76 ± 0.10	/	1.65 ± 0.10
Target	4.60 ± 0.04	/	4.55 ± 0.06	4.93 ± 0.03	4.77 ± 0.05	4.99 ± 0.01	4.42 ± 0.08	4.96 ± 0.03
Mismatch	/	1.03 ± 0.02	/	1.01 ± 0.01	/	1.02 ± 0.02	/	1.19 ± 0.06
AutovcF0	2.77 ± 0.05	2.10 ± 0.07	3.08 ± 0.09	2.42 ± 0.12	2.72 ± 0.10	2.19 ± 0.10	2.69 ± 0.08	1.98 ± 0.10
AutovcAIC	2.47 ± 0.05	1.87 ± 0.06	2.85 ± 0.09	2.10 ± 0.12	2.75 ± 0.09	1.97 ± 0.10	2.52 ± 0.09	1.73 ± 0.09
AutovcAIC-spotify	3.09 ± 0.06	1.91 ± 0.08	3.31 ± 0.10	2.01 ± 0.13	3.27 ± 0.09	1.70 ± 0.09	3.10 ± 0.10	1.73 ± 0.10
YourTTS	2.54 ± 0.05	2.28 ± 0.07	2.82 ± 0.08	2.58 ± 0.11	2.74 ± 0.09	2.32 ± 0.10	2.59 ± 0.08	2.05 ± 0.10
Resemblyzer	3.94 ± 0.05	3.49 ± 0.08	4.14 ± 0.07	3.62 ± 0.12	4.08 ± 0.08	3.09 ± 0.08	4.02 ± 0.08	3.09 ± 0.12
Ours-TFdecoder	3.89 ± 0.05	3.80 ± 0.07	4.19 ± 0.08	$\mathbf{3.98 \pm 0.11}$	4.08 ± 0.07	$\mathbf{3.56 \pm 0.11}$	4.08 ± 0.08	3.25 ± 0.12
Ours-TFdecoder+	$\mathbf{4.14 \pm 0.05}$	$\mathbf{3.82 \pm 0.07}$	$\mathbf{4.37 \pm 0.07}$	3.87 ± 0.11	$\mathbf{4.34 \pm 0.07}$	3.52 ± 0.11	$\mathbf{4.35 \pm 0.07}$	$\mathbf{3.30 \pm 0.09}$

Table 1. The quality (MOS) and similarity (SIM-MOS) scores for all methods with 95% confidence intervals. The first section of experiments are real samples from the datasets, where **Mismatch** stands for an utterance with opposite gender to the **Target**. The second section shows the baselines. The third section is a special baseline that combines our decoder with a pretrained speaker embedding **Resemblyzer** [8]. The forth section includes our model (**Ours-TFdecoder**) and its HiFiGAN-2 [32] enhanced version (**Ours-TFdecoder+**). Some entries are not included: **Source** and **Mismatch** quality scores are not collected; EN→PT and PT→EN do not have **GroundTruth** samples; VCTK **GroundTruth** and **Target** are the same; DAPS has **GroundTruth** available so we omitted the **Target**.

tion 3.2). Then we conducted listening tests on Amazon Mechanical Turk (AMT) to rate the quality and speaker similarity of these samples in the form of mean-opinion-scores (MOS) on a Likert scale of 1 to 5. To establish high and low anchors, we also ask listeners to rate MOS scores for the source voice sample, the target reference voice sample used to extract the global embedding, the groundtruth sample if available, a "mismatched" speaker sample that has the opposite biological gender of the target speaker and a heavily corrupted sample. Details about experiment designs and participation statistics are provided in website Appendix. Table 1 presents the results of the studies for the baselines and our top performing methods.

DAPS Baseline Comparison. Figure 2 visualizes the scores for DAPS from Table 1. We include **Mismatch** which is an utterance whose speaker has the opposite reported gender to the **Target** as a low anchor in the similarity test. Our method **Ours-TFdecoder** is on par with **Resemblyzer** in quality but surpasses all baselines in speaker similarity. This observation proves that we are able to learn a better speaker representation for synthesis than Resemblyzer. Resemblyzer uses supervised contrastive learning that requires speaker labels, while our method can easily expand to any speech dataset with proper data pre-processing, and thus is more flexible in comparison. **Ours-TFdecoder+** achieves the highest MOS and SIM-MOS out of all methods and is close to the ground truth, showing that our method significantly outperforms the state-of-the-art zero-shot voice conversion approaches. We also provide a gender-based analysis of the results and report the speaker identification accuracy using the learned embedding in the website Appendix.

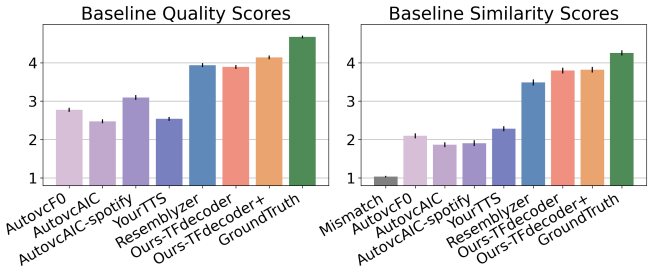

Fig. 2. MOS (left) and SIM-MOS (right) for VC.

VCTK Results. Table 1 shows the MOS and SIM-MOS for VCTK voices and the cross-language conversions. Evaluation on the VCTK dataset suggests the same trend as observed on the DAPS dataset. It is also worth noting that even though VCTK is used in training **AutovcF0**, **AutovcAIC**, and **YourTTS**, their performance on unseen speakers of VCTK does not catch up with our approach trained on larger yet unlabelled data. Post-processing using speech enhancement also proves to be helpful in improving quality. The similarity scores are in the same trend, with ours leading the scores. Using our learned speaker embedding with the decoder also improves upon using the pretrained speaker embedding from Resemblyzer, with a slight increase in quality and a large increase in speaker similarity.

Cross Language Conversions. In the cross-language scenario, our models consistently outperform all baselines. Unlike reported in YourTTS [14] that transferring from a reference PT sample reduces the MOS by a large margin ($4.20 \rightarrow 3.40$), our models are not affected by cross language in generation quality. The similarity scores are lowered in intra-lingual conversion for all methods, as it is more challenging for human listeners to associate speaker identities across languages. Thus we refer to the relative performance where our models significantly outperform all baselines.

4. CONCLUSION

In this paper, we propose a generative SSL framework for learning global representations, and apply it to zero-shot voice conversion. Listening tests suggest our model **GR0** enjoys improved performance in quality and speaker similarity over baseline models.

Here we discuss some limitations of our method, each of which suggests areas for future work. First, the SIM-MOS of our method still has a gap to ground-truth, mainly because we do not generate prosody (F0 & timing are already contained in the local features). Second, the synthesis quality is limited by the vocoder, which produces occasional artifacts and does not generate noise with fidelity. Third, CTC loss requires supervised data to train and is limited to the phoneme set. It may not generalize to all languages and other human vocalizations, such as laughter. We advocate for others to experiment with various content representation methods. Finally, we also believe this SSL framework may be applied to other tasks, for example, acoustic matching, transferring timbres of musical tones, or style transfer in other domains (imaging or NLP).

5. REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, et al., "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[2] Wei-Ning Hsu, Benjamin Bolte, et al., "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[3] Zalán Borsos, Raphaël Marinier, et al., "Audiolm: a language modeling approach to audio generation," *arXiv preprint arXiv:2209.03143*, 2022.

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

[5] Zhiyun Fan, Meng Li, Shiyu Zhou, and Bo Xu, "Exploring wav2vec 2.0 on speaker verification and language identification," *arXiv preprint arXiv:2012.06185*, 2020.

[6] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[7] Priya Goyal, Mathilde Caron, et al., "Self-supervised pretraining of visual features in the wild," *arXiv preprint arXiv:2103.01988*, 2021.

[8] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno, "Generalized end-to-end loss for speaker verification," in *ICASSP*. IEEE, 2018, pp. 4879–4883.

[9] Ting Chen, Simon Kornblith, et al., "A simple framework for contrastive learning of visual representations," in *ICML*. PMLR, 2020, pp. 1597–1607.

[10] Xiao Liu, Fanjin Zhang, et al., "Self-supervised learning: Generative or contrastive," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[11] Kaizhi Qian, Zeyu Jin, et al., "F0-consistent many-to-many non-parallel voice conversion via conditional autoencoder," in *ICASSP*. IEEE, 2020, pp. 6284–6288.

[12] Yunyun Wang, Jiaqi Su, Adam Finkelstein, and Zeyu Jin, "Controllable speech representation learning via voice conversion and aic loss," in *ICASSP*. IEEE, 2022, pp. 6682–6686.

[13] Yinghao Aaron Li, Ali Zare, and Nima Mesgarani, "Starganv2-vc: A diverse, unsupervised, non-parallel framework for natural-sounding voice conversion," *arXiv preprint arXiv:2107.10394*, 2021.

[14] Edresson Casanova, Julian Weber, et al., "Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone," in *ICML*. PMLR, 2022, pp. 2709–2720.

[15] Seung-won Park, Doo-young Kim, and Myun-chul Joe, "Cotatron: Transcription-guided speech encoder for any-to-many voice conversion without parallel data," *arXiv preprint arXiv:2005.03295*, 2020.

[16] Adam Polyak, Yossi Adi, et al., "Speech resynthesis from discrete disentangled self-supervised representations," *arXiv preprint arXiv:2104.00355*, 2021.

[17] Hyeong-Seok Choi, Juheon Lee, et al., "Neural analysis and synthesis: Reconstructing speech from self-supervised representations," *Advances in Neural Information Processing Systems*, vol. 34, pp. 16251–16265, 2021.

[18] Yuxuan Wang, Daisy Stanton, et al., "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *ICML*. PMLR, 2018, pp. 5180–5189.

[19] Chengyi Wang, Sanyuan Chen, et al., "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023.

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

[21] Chris Donahue, Julian McAuley, et al., "Adversarial audio synthesis," *arXiv preprint arXiv:1802.04208*, 2018.

[22] Jae Hyun Lim and Jong Chul Ye, "Geometric gan," *arXiv preprint arXiv:1705.02894*, 2017.

[23] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.

[24] Anders Boesen Lindbo Larsen et al., "Autoencoding beyond pixels using a learned similarity metric," in *ICML*. PMLR, 2016, pp. 1558–1566.

[25] Kundan Kumar, Rithesh Kumar, et al., "Melgan: Generative adversarial networks for conditional waveform synthesis," *Advances in neural information processing systems*, vol. 32, 2019.

[26] Jong Wook Kim, Justin Salamon, et al., "Crepe: A convolutional representation for pitch estimation," in *ICASSP*, 2018.

[27] 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.

[28] Ann Clifton, Sravana Reddy, Yongze Yu, et al., "100,000 podcasts: A spoken English document corpus," in *Proc. Internat. Conf. on Computational Linguistics*, 2020, pp. 5903–5917.

[29] Mirco Ravanelli, Titouan Parcollet, et al., "SpeechBrain: A general-purpose speech toolkit," 2021, arXiv:2106.04624.

[30] Gabriel Mittag, Babak Naderi, Assmaa Chehadi, and Sebastian Möller, "NISQA: A deep cnn-self-attention model for multidimensional speech quality prediction with crowdsourced datasets," *arXiv preprint arXiv:2104.09494*, 2021.

[31] Pranay Manocha, Zeyu Jin, et al., "Audio similarity is unreliable as a proxy for audio quality," *Interspeech*, 2022.

[32] Jiaqi Su, Zeyu Jin, et al., "HiFi-GAN-2: Studio-quality speech enhancement via generative adversarial networks conditioned on acoustic features," in *WASPAA*. IEEE, 2021, pp. 166–170.

[33] Jiaqi Su, Yunyun Wang, et al., "Bandwidth extension is all you need," in *ICASSP*. IEEE, 2021, pp. 696–700.

[34] Gautham J Mysore, "Can we automatically transform speech recorded on common consumer devices in real-world environments into professional production quality speech?—a dataset, insights, and challenges," *IEEE Signal Processing Letters*, vol. 22, no. 8, pp. 1006–1010, 2014.

[35] Junichi Yamagishi, Christophe Veaux, Kirsten MacDonald, et al., "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92)," 2019.

[36] Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, et al., "Mls: A large-scale multilingual dataset for speech research," *arXiv preprint arXiv:2012.03411*, 2020.