ZERO-SHOT SINGING VOICE CONVERSION

Shahan Nercessian
iZotope, Inc.
shahan@izotope.com

ABSTRACT

In this paper, we propose the use of speaker embedding networks to perform zero-shot singing voice conversion, and suggest two architectures for its realization. The use of speaker embedding networks not only enables the capability to adapt to new voices on-the-fly, but also allows for model training on unlabeled data. This not only facilitates the collection of suitable singing voice data, but also allows networks to be pretrained on large speech corpora before being refined on singing voice datasets, improving network generalization. We illustrate the effectiveness of the proposed zero-shot singing voice conversion algorithms by both qualitative and quantitative means.

1. INTRODUCTION

Singing voice conversion (SVC) is the transformation of a singing performance from one vocalist to that of another. It can be used for creative manipulations of the voice that go far beyond traditional time stretching and pitch/formant shifting [1]. SVC methods must learn to disentangle speaker content from acoustic features [2], while accurately preserving input phonetic and pitch information in the converted output. Relative to similar methods applied to speech, the singing voice exhibits a larger pitch range and generally slower transitions between phonetic units, which conversion networks must be able to accommodate for [3, 4].

Most approaches to SVC rely on some form of vocoder which synthesizes vocal waveforms. The SVC task then becomes one of transforming vocoder features from a performance of a source singer to that of some target voice. Unlike approaches to voice conversion on speech, which usually leverage neural vocoders such as WaveNet [5] or WaveRNN [6] as their back-end speech synthesizer, SVC and singing synthesis algorithms tend to use hand-designed vocoders such as WORLD [7] for acoustic modeling and synthesis of the voice (with some exceptions, as in [8]). This is because they explicitly separate pitch from timbral components [3, 4, 9]. Accordingly, it is possible to learn timbral transformations while preserving pitch, which is not usually guaranteed when using neural vocoder [2]. This may come at the expense of reduced expressivity relative to neural vocoders, but is considered to be acceptable given its pitch-preserving characteristics [4].

Generative Adversarial Networks (GANs) [3, 9, 10] and Variational Autoencoders (VAEs) [11] have become popular choices for learning transformations of vocoder features for both SVC and singing synthesis. Different strategies have been investigated to model several target voices, and specifically, to adapt systems for new voices not seen during model training. One such strategy involves assigning a random embedding to the unseen voice, and resuming model training on data of the unseen voice so as to update this embedding and perform any necessary refinements to the model [12, 13]. More recently, conversion algorithms in the speech domain have used pretrained speaker embedding networks designed for speaker verification tasks [14] in order to encode speaker identity [15]. These approaches have the advantage that, upon having trained the speaker embedding network on many speakers, conversion algorithms can be adapted to new voices in a zero-shot manner, requiring no further training of the model and with as few as one sample of an unseen voice.

In this paper, we adapt zero-shot voice conversion methodologies [15] utilizing speaker embedding networks for the application of SVC. We use the WORLD vocoder and suggest two architectures for carrying out zero-shot SVC. We show that the zero-shot nature of the algorithm allows for SVC on unlabeled data. Moreover, we posit that SVC systems are amenable to initial training on large speech datasets which are more widely available, followed by model adaptation on smaller singing voice datasets. To the best of our knowledge, this is the first work to tackle zero-shot SVC. Unlike singing synthesis algorithms, such as [4, 10, 13], it does not require predefined annotations of phonetic transitions or pitch, as this information is extracted from acoustic features of the source performance.

The remainder of this paper is structured as follows: We suggest architectures for zero-shot SVC in Section 2. We evaluate model performance via qualitative and quantitative means in Section 3. Lastly, we draw conclusions and allude to future work in Section 4.

2. SVC ALGORITHMS

We use the WORLD vocoder for analysis and synthesis of singing voices due to its ability to separate timbral and pitch components. Specifically, the system decomposes a vocal signal into a harmonic spectral envelope and an aperiodicity envelope, based on a tonality-gated estimate

Attribution: S. Nercessian, "Zero-Shot Singing Voice Conversion", in *Proc. of the 21st Int. Society for Music Information Retrieval Conf.,* Montréal, Canada, 2020.

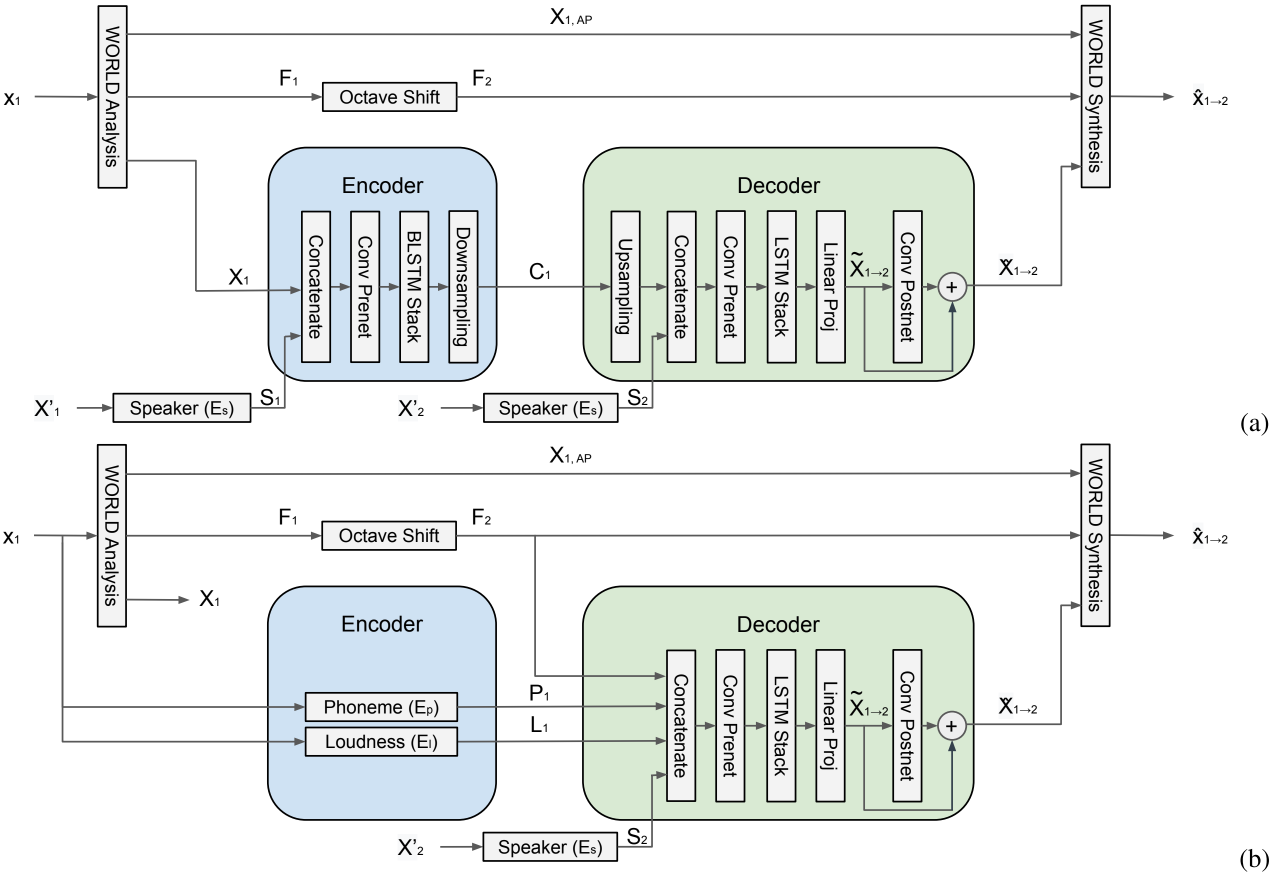

Figure 1. (a) Adapted AutoVC and (b) fixed encoder network architectures for zero-shot SVC.

of fundamental frequency. The conversion task primarily involves transformation of the harmonic spectral envelope, leaving the aperiodicity envelope unchanged. As in [4], we reduce the dimensionality of harmonic spectral envelopes to 60 coefficients at each time step, using truncated frequency warping in the cepstral domain with an allpole warping coefficient $\alpha = 0.45$ [16]. We consider two different architectures for SVC, as illustrated in Figure 1, drawing inspiration from [2, 15, 17, 18].

2.1 AutoVC

The first architecture is an adaptation of the AutoVC architecture [15] for singing voice, which operates on harmonic spectral envelopes extracted from WORLD (instead of Mel spectrograms which are ultimately fed into a WaveNet vocoder as in the original work). It is comprised of a speaker embedding network $E_s(\cdot)$ which takes as input a Mel spectogram and generates a single fixed-dimensional speaker embedding, a content encoder $E(\cdot)$ which takes as input the harmonic spectral envelope and speaker embedding from a source utterance and generates a latent encoding, and a decoder network $D(\cdot)$ which constructs the converted harmonic spectral envelope from a latent encoding and target speaker embedding.

The input to the encoder is the harmonic spectral envelope X_1 computed from a source utterance x_1. This is concatenated with a source speaker embedding $S_1 = E_s(X'_1)$ at each time step, where X'_1 is a Mel spectrogram of the same or potentially different utterance x'_1 from the same source speaker. The encoder consists of a convolutional prenet, comprised of three 1D convolutional layers with 512 output channels and kernel size 5, each followed by batch normalization and ReLU activation. This result is passed through two bidirectional LSTM layers with forward and backward cell dimensions of 32, yielding an encoding of dimension 64. This is temporally downsampled by 32, yielding the content encoding C_1. The inclusion of S_1 in the encoder network helps the encoder to more easily learn a speaker-independent encoding.

The decoder begins by upsampling the latent encoding C_1 to its original temporal resolution. Given the Mel spectrogram X'_2 of some utterance x'_2 from the same target speaker as the target utterance x_2, the speaker embedding $S_2 = E_s(X'_2)$ is concatenated with the upsampled encoding. The concatenated features pass through a convolutional prenet similar to that in the encoder, followed by three LSTM layers with cell dimension 1024. The outputs of the LSTM layer are linearly projected to dimension 60, serving as an initial estimate $\tilde{X}_{1\to2}$ of the converted harmonic spectral envelope. This initial estimate is refined by means of a convolutional postnet consisting of five 1D convolutional layers of kernel size 5. Batch normalization and Tanh are applied to the first four layers, and they each output 512 channels. The final layer applies no activation and outputs 60 channels. The converted harmonic spectral envelope $\hat{X}_{1\to2}$ is produced by adding the output of the postnet to $\tilde{X}_{1\to2}$.

During training, we set $x_1 = x_2$, $S_1 = S_2$, and accordingly, $X_1 = X_2$, $\tilde{X}_{1\to1} = \tilde{X}_{1\to2}$, and $\hat{X}_{1\to1} = \hat{X}_{1\to2}$. The objective function used for training AutoVC is

$$\begin{aligned}\mathcal{L} =&\mathrm{E}[|X_1 - \hat{X}_{1\to1}|_2^2] + \\ &\mu\mathrm{E}[|X_1 - \tilde{X}_{1\to1}|_2^2] + \\ &\lambda\mathrm{E}[|E(X_1, S_1) - E(\hat{X}_{1\to1}, S_1)|_1]\end{aligned} \tag{1}$$

The first term is the reconstruction loss between the original and reconstructed harmonic spectral envelopes. The second term is a reconstruction loss between the original and initially estimated harmonic spectral envelopes, which empirically helps model convergence. The third term is a latent regressor loss [19] penalizing differences in encodings between the original and converted harmonic spectral envelopes. In practice, hyperparameters μ and λ can be set to 1 [15]. The model is trained as an autoencoder, with the hope that its bottleneck will be small enough to disentangle speaker identity but large enough to allow for an accurate reconstruction.

During inference, S_2 can be set to the speaker embedding of some target singer to perform a conversion. Given a source pitch contour F_1 extracted during WORLD analysis, the target pitch contour F_2 should be adjusted to accommodate the register of the target singer, and therefore, $F_2 = F_1 + F_{\Delta 1\to2}$. The pitch shift $F_{\Delta 1\to2}$ can be determined automatically by measuring the median pitches of source and target performances, and taking their difference rounded to the nearest octave. The aperiodicity spectral envelope of the source performance $X_{1,AP}$ is used as is. The converted audio waveform $\hat{x}_{1\to2}$ is computed by feeding the transformed harmonic spectral envelope, source aperiodicity spectral envelope, and target pitch contour F_2 as input to the WORLD synthesis engine.

2.2 Fixed encoder model

The second architecture is similar to AutoVC, but replaces the encoder $E(\cdot)$ with a number of conditioning signals, such as those found in [2]. By design, these conditioning signals encode the input in a speaker-independent way using explicit features, akin to the timbre transfer networks in [18]. We capture linguistic content using phonetic posteriorgrams (PPGs) extracted from a phoneme classifier $E_p(\cdot)$, as in [17]. The classifier passes 40 Mel frequency cepstral coefficients (MFCCs) per frame through two bidirectional LSTMs with 128 units per direction. A final dense layer with softmax activation yields the classifier output, which is compared to ground truth labels using a categorical cross-entropy loss during training. We trained the network on the TIMIT dataset [20], using its prescribed training and test sets. The dataset consists of audio and sample level timestamps of phonetic transitions from one of 61 classes (including a silence class). The output of the phoneme classifier is, therefore, a 61-dimensional vector at each time frame. The classification accuracy on the test set is 65%, which is found to be sufficient to act as a speaker-independent representation of linguistic content. We extract loudness information (L) using the computational steps $E_l(\cdot)$ as in [21]. We compute an A-weighted power spectrum, which puts greater emphasis on higher frequencies. The result is aggregated across all frequencies and converted to decibels to produce a loudness value (in dbA) at each time step. Lastly, we include the target pitch contour F_2.

The decoder concatenates the target pitch contour F_2, $P_1 = E_p(x_1)$, $L_1 = E_l(x_1)$ with the target speaker embedding S_2. The inclusion of these different conditioning signals attempts to account for timbral changes which may vary as a function of the pitch and dynamics of a particular performance, while instructing the decoder of its underlying linguistic content. The decoder network is almost identical to that in AutoVC, except that we remove the upsampling operation as we no longer need to construct an information bottleneck for speaker disentanglement. We refer to this architecture as the fixed encoder model, because all conditioning signals are either computed without a neural network, or using a pretrained neural network whose weights are frozen during the training of the decoder network. The training objective is similar to that in Eqn. (1), except that the third term is no longer applicable and is therefore removed. Note that in this case, the source harmonic spectral envelope X_1 is never actually passed as input to the network, but is used as a target for reconstruction during training.

2.3 Architecture comparisons

We notionally discuss the potential advantages and disadvantages associated with the architectures proposed here. The main advantage of the AutoVC architecture is that it does not rely on a dedicated training set for extracting phonetic information. This information is learned by the encoder itself during model training. This could potentially have better implications for cross-lingual applications, in the case that the set of phoneme labels of a dataset itself introduces a language bias [22]. It does, however, incur some risk, as the encoder is solely responsible for learning all timbral variations in vocalization. It also requires a temporal downsampling/upsampling of its encoding to create an information bottleneck for speaker disentanglement, which has some additional latency implications in the decoder. The fixed encoder architecture is computationally less intensive, as the phoneme classifier is substantially smaller than the encoder network in AutoVC. It also avoids the need for temporal downsampling/upsampling. The main disadvantages of this architecture is the reliance on data to train a phoneme classifier, as well as the fact that its expressivity is limited to that provided by its conditioning signals.

2.4 Universal Background Model (UBM)

While we could simply train SVC networks "from scratch" on singing voice datasets, we consider leveraging the interesting fact that the use of speaker embeddings for encoding vocal identity (instead of one-hot labels) allows the system to be trained on unlabeled data. Arguably, any "clean" speech or singing voice clip could now be used for training SVC systems. It is generally understood that there is significantly more speech data than their is singing voice data for research purposes. Borrowing nomenclature from the speech recognition community, an initial pretraining on large speech corpora is like training a UBM [23] from which other networks can be adapted for the more specific SVC task. We would hope that such a model would serve as a better initial condition for training a SVC network than random weights, and that the resulting system would at the very least generalize to more voices.

3. EXPERIMENTAL RESULTS

3.1 Experimental setup

Two datasets are used for training conversion networks in this work. We use the VCTK corpus, which consists of over 40 hours of speech from 109 speakers [24]. This corpus serves as both a supervised speaker dataset to compare performance between supervised and (unsupervised) zero-shot networks, as well as a sufficiently large dataset for training a UBM for further model fine tuning. As in [15], we retain 90% of the data of each speaker for training, and save the remainder as a test set. Additionally, we use a proprietary and unlabeled dataset consisting of 7 hours of singing voice data, which we simply call the SVC dataset. Again, we retain 90% of the data for training, and save the remainder as a test set. Note that the lack of labels in this dataset poses no problem for zero-shot network training.

We make use of an open-source speaker embedding network [1] pretrained to minimize the Generalized End-to-End Loss [14]. This speaker embedding network generates a 256-dimensional speaker embedding from a 40-band Mel spectrogram using an LSTM architecture and retaining only the output from the final time step. During training, we use an entire utterance for x_1', whereas x_1 is a 2 second cut from the same utterance. The speaker embedding network and the phoneme classifier are pretrained and frozen during the training of the conversion networks.

All models operate at 16 kHz with a frame rate of 12.5 ms, and were trained with a batch size of 2 using the ADAM optimizer and a learning rate of 10^{-3}. We train four configurations for each model architecture described here. The first configuration, VCTK (one-hot), is trained on the VCTK corpus using the labels provided by the dataset, which are converted to a one-hot representation and fed as S_1 to the network. This configuration serves as a baseline to compare against its zero-shot counterpart. The second configuration, VCTK (zero-shot), is trained on the VCTK corpus using speaker embeddings for S_1. The first two configurations are each trained for 150,000 steps. In the third configuration, SVC (zero-shot), we train zero-shot architectures on the SVC dataset for 500,000 steps. In the final configuration, VCTK→SVC (zero-shot), the second configuration is used as an initial state, and training is resumed for 350,000 steps on the SVC dataset (for a total of 500,000 steps). For audio examples, please visit the demo site [2] associated with this paper.

3.2 Performance assessment

We assess networks by both qualitative and quantitative means. The main goal of this paper is to illustrate that speaker embeddings networks can indeed be utilized for training zero-shot SVC networks. Since we are unaware of any other published methods for zero-shot SVC such as the ones introduced here, and in order to provide some form of comparative analysis, we focus our attention to analyzing differences in results between the training configurations outlined here. For our quantitative evaluation, we report the reconstruction loss for each network (the first term in Eqn. (1)), which when computed on harmonic spectral envelopes, effectively serves as a Mel cepstral distortion metric. For our qualitative evaluation, we conducted surveys with 15 participants within our organization who have some critical listening experience, and tabulated mean opinion scores (MOS). We conduct separate surveys for overall conversion quality and on similarity to the target voice. While we provide examples from both architectures in the supplementary material of this work, we restrict ourselves to samples generated from training variants of the fixed encoder architecture for subjective evaluations. The first reason for this restriction is simply to minimize the number of listening options so as not to overwhelm participants taking part in the survey. The second reason is because the inclusion of one-hot speaker labels for S_1 in the encoder network of AutoVC would require that input source samples come from its closed speaker set. Therefore, it is not practically possible to use the VCTK (one-hot) training configuration on AutoVC on singing voice examples without removing S_1 from the network, leading to a potentially unfair comparison.

The results of our quantitative analysis assessed on both the VCTK and SVC datasets are shown in Tables 1 and 2, respectively. Across both architectures, we can confirm that the replacement of one-hot labels with speaker embeddings does not dramatically hurt conversion performance.

	AutoVC	Fixed Encoder
VCTK (one-hot)	0.1837	**0.1882**
VCTK (zero-shot)	<u>**0.1634**</u>	0.1891
SVC (zero-shot)	0.2930	0.3590
VCTK→SVC (zero-shot)	0.2557	0.3232

Table 1. Reconstruction loss on the VCTK test set.

	AutoVC	Fixed Encoder
VCTK (one-hot)	N/A	N/A
VCTK (zero-shot)	0.3007	0.4314
SVC (zero-shot)	0.1650	0.1959
VCTK→SVC (zero-shot)	<u>**0.1439**</u>	**0.1850**

Table 2. Reconstruction loss on the SVC test set.

[1] https://github.com/CorentinJ/Real-Time-Voice-Cloning.

[2] https://sites.google.com/izotope.com/ismir2020-audio-demo

In fact, we see that for the AutoVC architecture, VCTK (zero-shot) actually outperforms VCTK (one-hot), while offering the added functionality of zero-shot adaptation to new unseen voices. This result is consistent with the findings in [15]. We note that there is a significant degradation in performance assessed quantitatively when applying either VCTK (zero-shot) directly to singing voice samples, or when applying SVC networks directly to VCTK samples, highlighting that there is indeed a mismatch between the speech and singing voice domains. There is a consistent improvement when using our proposed adaptation strategy, with VCTK→SVC (zero-shot) outperforming SVC (zero-shot), both in the speech domain and more importantly, in the singing voice domain of interest. Overall, the best performing approach for singing voice based on this quantitative evaluation is AutoVC trained using VCTK→SVC (zero-shot) training configuration, though the computationally lighter, fixed encoder model does perform comparably well. It is worth noting that the VCTK (one-hot) configuration is not applicable for evaluation on the SVC dataset as it does not have the immediate ability to adapt to new voices.

The results of our qualitative analysis, converting singing voice performances using target voices from both the VCTK and SVC test sets are shown in Tables 3 and 4, respectively. First and foremost, we observe that speaker embeddings networks can, in general, be used for zero-shot SVC. We note that conversion networks trained on speech can be used on singing voice, but they have some trouble maintaining consistent spectral envelopes over prolonged vowels. Lastly, while not formally a part of the subjective evaluation, we informally observe comparable performance between architectures, with a preference towards one architecture over the other on a per-case basis.

With target voices from VCTK, there is no remarkable difference between networks trained using one-hot speaker labels or using zero-shot speaker embeddings, but the latter naturally allows adaptation to new voices. While SVC (zero-shot) is trained to be adapted to properties of singing voice, it is trained on less data and has been exposed to fewer voices. Though it was able to generate voices resembling the VCTK target voices due to its zero-shot nature, and worked comparably to other methods, it understandably received the lowest MOS in this case. The networks trained on the SVC dataset are more successful when using target voices from the SVC test set (and again, are better adapted to the time scale of phonetic transitions in singing). In this case, there is some degradation in quality for the system trained using the VCTK (zero-shot) configuration, and the VCTK (one-hot) configuration is not even applicable. We again see an improvement for networks trained using VCTK→SVC (zero-shot) relative to SVC (zero-shot) in this scenario. In fact, the VCTK→SVC (zero-shot) training configuration outperforms other methods in terms of overall quality for both VCTK and SVC target voices. The VCTK (zero-shot) and VCTK→SVC (zero-shot) training configurations are the top performers in terms of voice similarity for VCTK and SVC target voices, respectively.

	Quality	Similarity
VCTK (one-hot)	2.377	2.828
VCTK (zero-shot)	2.447	**3.051**
SVC (zero-shot)	2.289	2.549
VCTK→SVC (zero-shot)	**2.476**	2.664

Table 3. Mean opinion scores on singing voice with target voices from the VCTK test set with fixed encoder model.

	Quality	Similarity
VCTK (one-hot)	N/A	N/A
VCTK (zero-shot)	2.154	2.610
SVC (zero-shot)	2.477	2.772
VCTK→SVC (zero-shot)	**2.674**	**2.937**

Table 4. Mean opinion scores on singing voice with target voices from the SVC test set with fixed encoder model.

Lastly, we further exemplify the zero-shot nature of our proposed method by subjecting our system to target voices outside of the VCTK and SVC datasets. These examples were generated without any further training of models and using just 1-2 seconds of audio from a target voice in order to compute speaker embeddings. While quality and voice similarity could obviously be improved by further model fine tuning on more data from target voices, it is apparent that the system can generate reasonable conversions resembling the voices from the reference material in a zero-shot manner.

4. CONCLUSION

In this paper, we propose the application of speaker embedding networks for zero-shot SVC. We suggest two architectures for carrying out zero-shot SVC using the WORLD vocoder for modeling singing voice. Overall, we find that speaker embeddings can indeed be used directly for zero-shot SVC. Moreover, zero-shot networks replacing one-hot speaker labels with speaker embeddings perform as well as (or even better than) their supervised closed set counterparts, with the invaluable added benefits that they can be trained on unlabeled data and can potentially adapt to new voices without requiring further training. Furthermore, we show that there is some benefit to training zero-shot SVC networks by adapting an initial model trained on large amounts of speech data. In future work, we will investigate learning latent factors which can allow for further expressive manipulation of conversion results. While some initial progress to this end has been made using Gaussian Mixture VAEs (GMVAEs) [11], they have largely been limited to sung vowels. We can likely generalize this to more practical singing voice by utilizing the conditioning signals used in this work. We are also interested in replacing the WORLD vocoder with learned vocoders based on differentiable digital signal processing, as in [18, 25], in order to enable lightweight end-to-end training.

5. ACKNOWLEDGEMENTS

The author would like to thank François Germain and all the anonymous reviewers for their invaluable comments while preparing this paper, which dramatically improved the quality of this work.

6. REFERENCES

[1] K. Lent, "An efficient method for pitch shifting digitally sampled sounds," *Computer Music Journal*, vol. 13, no. 4, pp. 65–71, 1989.

[2] S. Nercessian, "Improved zero-shot voice conversion using explicit conditioning signals," in *Proc. of Interspeech 2020*, 2020, accepted.

[3] W. Zhao, W. Wang, Y. Sun, and T. Tang, "Singing voice conversion based on WD-GAN algorithm," in *Proc. of the 2019 IEEE 4th Advanced Information Tech., Electronic and Automation Control Conference (IAEAC)*, 2019, pp. 950–954.

[4] M. Blaauw and J. Bonada, "A neural parametric singing synthesizer," in *Proc. of Interspeech 2017*, 2017.

[5] A. van den Oord *et al.*, "WaveNet: A generative model for raw audio," *arXiv:1609.03499*, 2016.

[6] N. Kalchbrenner *et al.*, "Efficient neural audio synthesis," *arXiv:1802.08435*, 2018.

[7] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Transactions on Information and Systems*, vol. E99-D, no. 7, pp. 1877–1884, 2016.

[8] E. Nachmani and L. Wolf, "Unsupervised singing voice conversion," in *Proc. of Interspeech 2019*, 2019, pp. 2583–2587.

[9] B. Sisman, K. Vijayan, M. Dong, and H. Li, "SINGAN: Singing voice conversion with generative adversarial networks," in *Proc. of the 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2019, pp. 112–118.

[10] P. Chandna, M. Blaauw, J. Bonada, and E. Gomez, "WGANSing: A multi-voice singing voice synthesizer based on the wasserstein-gan," in *Proc. of the 27th European Signal Processing Conference*, 2019.

[11] Y. Luo, C. Hsu, K. Agres, and D. Herremans, "Singing voice conversion with disentangled representations of singer and vocal technique using variational autoencoders," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020.

[12] S. O. Arik *et al.*, "Deep voice 2: Multispeaker neural text-to-speech," *Advances in Neural Information Processing Systems*, vol. 30, pp. 2962–2970, 2017.

[13] M. Blaauw, J. Bonada, and R. Daido, "Data efficient voice cloning for neural singing synthesis," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019.

[14] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 4879––4883.

[15] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "AutoVC: Zero-shot voice style transfer with only autoencoder loss," in *Proc. of the International Conference on Machine Learning*, 2019.

[16] K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai, "Mel-generalized cepstral analysis - a unified approach to speech spectral estimation," in *Proc. of the International Conference on Spoken Language Processing*, 1994.

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

[18] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," in *Proc. of the International Conference on Learning Representations*, 2020, pp. 26–30.

[19] J. H. Lee, H. S. Choi, and K. Lee, "Audio query-based music source separation," in *Proc. of the International Society for Music Information Retrieval Conference*, 2019.

[20] J. S. Garapolo *et al.*, *TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1*. Philadelphia: Linguistic Data Consortium, 1993.

[21] L. Hantrakul, J. Engel, A. Roberts, and C. Gu, "Fast and flexible neural audio synthesis," in *Proc. of the International Society for Music Information Retrieval Conference*, 2019.

[22] Y. Zhou, X. Tian, H. Xu, R. K. Das, and H. Li, "Cross-lingual voice conversion with bilingual phonetic posteriorgram and average modeling," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 6790—6794.

[23] T. Hasan and J. H. L. Hansen, "A study on universal background model training in speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 1890–1899, 2011.

[24] C. Veaux, J. Yamagishi, and K. MacDonald, *CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit*. Edinburgh: The Centre for Speech Technology Research (CSTR), University of Edinburgh, 2016.

[25] X. Wang, S. Takaki, and J. Yamagishi, “Neural source-filter-based waveform model for statistical parametric speech synthesis,” in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 5916–5920.