HIFI-SVC: FAST HIGH FIDELITY CROSS-DOMAIN SINGING VOICE CONVERSION

Yong Zhou, Xiangju Lu

iQIYI Inc, China

ABSTRACT

This paper presents HiFi-SVC, a small cross-domain singing voice conversion model for generating high-fidelity 22.05 kHz singing voices. Building on state-of-the-art neural vocoder HiFi-GAN and a convolution-based module for modeling F0, HiFi-SVC can be trained end-to-end with either speech or singing data, achieving better voice similarity on two of the datasets than FastSVC while using slightly smaller number of parameters. We also propose a pitch adjustment method for improving conversion quality.

Index Terms— Singing voice conversion, phonetic posteriorgrams, pitch modelling

1. INTRODUCTION

Singing voice conversion (SVC) aims to convert a piece of singing voice to the voice of another speaker or singer while preserving the other properties of the original singing such as lyrics and rhythm. It is more challenging than speech voice conversion (VC) since singing voices typically carry more expressive elements and pitch contours.

Since parallel data are rare and expensive to obtain, most of recent research in VC focuses on non-parallel training data. Several approaches have been proposed for non-parallel data training. CycleGAN-VC [1] and its variants [2][3][4] focus on learning a mapping function that preserves linguistic information. CycleGAN learns forward and inverse mappings simultaneously using adversarial and cycle-consistency losses. Another line of research works by separating speaker information from linguistic content [5][6]. Phonetic posteriorgrams (PPGs) were first proposed in [7] for voice conversion and have since been widely used for both speech VC [8][9] and SVC [10][11][12]. PPGs are one type of speaker-invariant linguistic representation of voice that is extracted from a pretrained automatic speech recognition (ASR) model. The advantage of the PPG-based approach is that there is no need to disentangle linguistic content from speaker information. We focus on PPG-based approach in this work.

Due to the lack of large public singing datasets, cross-domain SVC becomes an attractive research area. UCD-SVC [11] is an end-to-end generative model for cross-domain any-to-many SVC that uses loudness feature, fundamental frequency (F0), and linguistic feature as input. Noting UCD-SVC's huge number of parameters and complex training process, FastSVC [12] was proposed as a light-weight solution to cross-domain SVC by building on WaveGrad [13]. It uses a Conformer-based [14] phoneme recognizer as linguistic extractor and incorporates F0 and loudness feature into the FiLM [15] conditioning module. FastSVC achieves better mean opinion score (MOS) in naturalness and similarity than UCD-SVC using only 3.2% of total number of parameters of UCD-SVC. FastSVC is a fairly compact model with only 2.9M parameters, excluding the ASR model.

There is room for improvement of FastSVC in terms of voice similarity and audio volume. Firstly, the audio samples from FastSVC, sampled at 16 kHz, tend to be perceptually coarse. Secondly, many of FastSVC samples tend to be louder than their corresponding source singing audio[1]. The audio volume of its output samples seems to be at the same level. The job of voice conversion is to only convert the identity of the speaker while preserving all other properties including audio volume. FastSVC has room for improvement in terms of audio volume. To improve on these issues, we argue that SVC can be done by building on a high-fidelity neural vocoder other than WaveGrad. HiFi-GAN [16] is a fast and high-fidelity neural vocoder able to generate 22.05 kHz audio 167.9 times faster than real-time on a single V100 GPU. By combining HiFi-GAN and a new module for modeling F0, we can achieve better voice similarity than FastSVC while using slightly smaller number of parameters.

2. PROPOSED METHOD

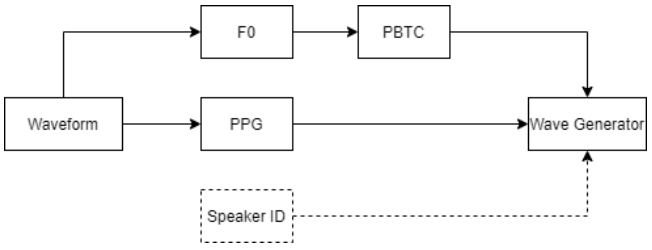

Fig. 1. Overall Diagram of HiFi-SVC

The overall diagram of HiFi-SVC is demonstrated in Fig.

[1]This is most obvious in some of their A2M-ID samples

1. PPGs are extracted from a linguistic extractor. Pitch tracking models extract F0 values from the audio. For the multi-speaker/singer scenario, speaker IDs are also needed for input.

2.1. F0 Extraction

Any pitch tracking methods can be used. We found that CREPE [17] yielded better results for the conversion than WORLD [18], so the former is used.

2.2. Linguistic Extractor

The purpose of linguistic extractor is to extract speaker-invariant content representation. We leverage the specific Conformer model used in FastSVC, trained with the 960-hour LibriSpeech corpus [19], to extract PPGs.

2.3. F0 Modeling

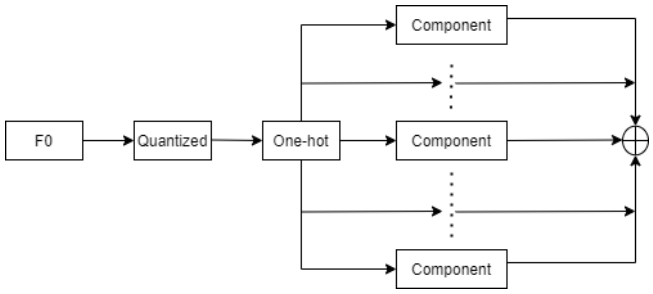

Fig. 2. Overall Diagram of PBTC

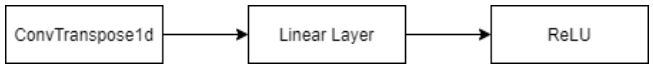

Fig. 3. Diagram of PBTC Component

F0 modeling is a crucial part for singing voice synthesis and conversion. We leverage a neural construct called PBTC introduced in [20] to model F0. As its authors put it, PBTC is a parallel bank of 1D transposed convolutions designed to generate the harmonic structure in the characteristic of F0. To this end, ten components are arranged in a parallel manner to process the input to PBTC. Each component is a 1D transposed convolution followed by a linear layer and ReLU. Each component is configured with a different dilation for their convolution[2]. Since different dilations result in different output sizes, their outputs are projected to identical size by a linear layer. The outputs from these components are summed as the PBTC output. The input to PBTC is one-hot embeddings of the quantized F0 values. We refer to the output of the PBTC module as F0 embeddings. PBTC is illustrated in Fig. 2 and its component in Fig. 3.

[2]We use dilation 1,3,5...19 for these ten components

2.4. Waveform Generator

The waveform generator is an enhanced version of the HiFi-GAN generator. Specifically, the generator output is now conditioned on input F0 and PPGs, instead of mel-spectrograms. Conditioning on F0 enables the output to preserve the melody of the source singing.

F0 embeddings are concatenated with the output of the waveform generator prenet along the feature dimension. For the multi-speaker/singer scenario, the output of a speaker embedding table is concatenated as well. These concatenated embeddings are passed to a number of layers that consist of an upsampling module and a multi-receptive field (MRF) fusion module. These layers are the same as those in the HiFi-GAN generator, except that some of the configuration parameters are adjusted for the SVC task.

Since the output of the waveform generator is the same as the HiFi-GAN generator, the same discriminators and training losses are used [16].

2.5. Pitch Adjustment

When performing singing voice conversion, the pitch of the source singing is used as one of the inputs. However, when the source voice and the target (reference) voice have a huge pitch difference, applying the original pitch might cause undesirable effects.

We propose a simple method for pitch adjustment to address this during inference. The pitch contour from the source singing is adjusted by the pitch contour of the reference audio as follows:

$$pitch_{src} = \log_2 pitch_{src} \tag{1}$$

$$pitch_{ref} = \log_2 pitch_{ref} \tag{2}$$

$$pitch_{adjusted} = pitch_{src} + mean(pitch_{ref}) - mean(pitch_{src}) \tag{3}$$

$$pitch_{adjusted} = 2^{pitch_{adjusted}} \tag{4}$$

$pitch_{src}$ and $pitch_{ref}$ are the raw pitch from the source singing and the reference audio respectively. Firstly, the log2 function is applied. Since they're both a 1D sequence, their mean values are computed. The difference of their mean values is used to adjust $pitch_{src}$ (Equation 3). The purpose of this adjustment is to preserve the melody embedded in $pitch_{src}$ while taking into account the pitch range difference between the two, so $pitch_{adjusted}$ won't be too far away from $pitch_{ref}$. Finally, exponentiation is applied to $pitch_{adjusted}$ (Equation 4).

3. EXPERIMENTS

3.1. Experimental Setup

Evaluations were done on three datasets: LJSpeech [21], VCTK [22] and NUS-48E [23]. For comparison with FastSVC, the same test scenarios were performed: any-to-one cross-domain (A2O-CD) SVC models trained with LJSpeech, any-to-many cross-domain (A2M-CD) SVC models trained with VCTK, any-to-many in-domain (A2M-ID) SVC models trained with NUS-48E and cross-lingual (CL) SVC.

16 kHz audio samples were used for extracting PPGs and F0, while 22.05 kHz samples were used as ground truth for training HiFi-SVC. Both PPGs and F0 were computed with a hop of 10ms. F0 values were quantized evenly into 329 bins.

The model was trained on randomly cropped audio clips of 22000 samples, which is roughly 0.9977 second of audio, with batch size of one. HiFi-SVC was optimized by ADAM optimizer with a learning rate of 0.0002 and a learning rate decay of 0.999. The hop size in the configuration of the official HiFi-GAN implementation was adjusted to 220 and upsampling rates were changed accordingly. The initial number of channels for upsampling was set to 128. Excluding the ASR model, numbers of parameters are shown in Table 1. The numbers for HiFi-SVC vary because different datasets cause different configurations such as number of speakers. HiFi-SVC is slightly smaller than FastSVC in terms of number of parameters.

Audio samples are available online[3].

Table 1. Number of Model Parameters

Model	# of Parameters
FastSVC	2.90M
HiFi-SVC	2.70-2.85M

3.2. Subjective Evaluation

Table 2. Mean Opinion Score

Scenario	Naturalness		Similarity	
	FastSVC	HiFi-SVC	FastSVC	HiFi-SVC
A2O-CD	**3.58±0.32**	3.42±0.34	3.53±0.28	**3.69±0.28**
A2M-CD	**3.43±0.38**	2.42±0.39	**3.31±0.37**	3.11±0.40
A2M-ID	3.78±0.30	**3.80±0.30**	3.46±0.37	**3.63±0.36**
CL	**3.03±0.29**	2.81±0.28	3.17±0.39	**3.24±0.40**

Since we didn't have a FastSVC implementation, we used its online audio samples for subjective evaluation. The results are shown in Table 2. FastSVC beats our proposed method by a large margin with the VCTK dataset while on other datasets, the results are much closer. Please check our online audio samples. We also provide audio samples to show the effects of pitch adjustment.

[3]https://zhouyong64.github.io/hifisvc-demo

3.3. Objective Evaluation

3.3.1. Naturalness

MOSNet [24] was used for objective evaluation of naturalness. As shown by Table 3, both FastSVC and HiFi-SVC beat UCD-SVC while FastSVC and HiFi-SVC results are close. Note the discrepancy between subjective and objective evaluations.

Table 3. Naturalness Scores from MOSNet

Scenario	UCD-SVC	FastSVC	HiFi-SVC
A2O-CD	2.75±0.06	**2.96±0.07**	2.84±0.03
A2M-CD	2.94±0.16	3.45±0.26	**3.67±0.29**
A2M-ID	2.50±0.09	2.81±0.11	**2.98±0.15**
CL	2.74±0.14	2.95±0.09	**3.01±0.07**

3.3.2. Voice similarity

Table 4. NUS-48E Test for Speaker Embeddings

Scenario	Reading-Singing Similarity
Same Person	**0.367±0.003**
Different Persons	0.068±0.001

Table 5. Voice Similarity

Scenario	Converted-Source	Converted-Reference	
		FastSVC	HiFi-SVC
A2O-CD	0.064	0.556	**0.634**
A2M-CD	0.117	0.266	**0.371**
A2M-ID	0.228	0.502	**0.619**
CL	0.140	0.443	**0.572**

For voice similarity, we conducted objective evaluation by using speaker embeddings and computing cosine distance. We used the ECAPA-TDNN [25] model implemented in SpeechBrain [26] trained with Voxceleb1 [27] and Voxceleb2 [28] datasets.

We first tested it on NUS-48E dataset to prove its validity. The audio in the dataset was split by silence to create about 12,000 utterances. Voice similarity was computed between reading and singing utterances, and those from the same person have a much larger voice similarity than those from different persons as shown by Table 4. For FastSVC samples and our samples, similarity was computed for pairs of source and reference audio, pairs of FastSVC result and its reference audio and pairs of HiFi-SVC result and its reference audio. As shown in Table 5, our results beat FastSVC in all four scenarios by a large margin in terms of voice similarity.

3.3.3. Pitch Adjustment

The proposed pitch adjustment was evaluated by MOSNet and speaker embeddings for naturalness and similarity respectively. As shown by Table 6, this adjustment improves voice similarity while making no difference for naturalness. Too large changes in pitch would make the voice sound like a different person so pitch adjustment should improve voice similarity. The result is consistent with this intuition.

Table 6. Scores with and without Pitch Adjustment

Scenario	Naturalness		Similarity	
	w/o	w/	w/o	w/
A2O-CD	2.93±0.05	2.93±0.03	0.61±0.02	**0.66±0.02**
A2M-ID	3.03±0.05	3.02±0.05	0.52±0.03	**0.66±0.01**

3.3.4. Audio volume

Table 7. Audio Volume Difference

Scenario	Converted-Source		
	FastSVC	HiFi-SVC	FastSVC 50%
A2O-CD	217.2	**43.7**	44.0
A2M-CD	174.3	**84.4**	49.0
A2M-ID	337.6	**55.3**	83.6
CL	**106.2**	356.1	329.8

Since loudness is a feature that can be objectively measured, we argue that it is a topic worth discussing. For example, what's the most ideal audio volume for the converted audio? Should it be a factor in assessing conversion quality?

We used the loudness function in Essentia [29], which computes the loudness of an audio signal defined by Steven's power law. We computed the absolute difference in loudness between converted audio and the source audio. As a sanity check for the correctness of Essentia's implementation, we also lowered the volume of FastSVC samples by 50% using the audio tool sox.

As shown by Table 7, except in the case of the CL scenario, FastSVC has a much larger audio volume difference with the source. In the case of CL, the source audio is also very loud, so FastSVC has the smallest difference. Also, when its audio volume was reduced 50%, the difference becomes much smaller. This proves that FastSVC outputs tend to be loud, regardless of their source audio.

4. CONCLUSION

We propose HiFi-SVC, a small singing voice conversion model that achieves better voice similarity on LJSpeech and NUS-48E datasets than FastSVC. We also address the issue of pitch adjustment and audio volume in the context of singing voice conversion.

5. REFERENCES

[1] Takuhiro Kaneko and Hirokazu Kameoka, "Parallel-data-free voice conversion using cycle-consistent adversarial networks," *arXiv preprint arXiv:1711.11293v2*, 2017.

[2] Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo, "CycleGAN-VC2: improved cyclegan-based non-parallel voice conversion," in *ICASSP*. IEEE, 2019.

[3] Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo, "CycleGAN-VC3: examining and improving cyclegan-vcs for mel-spectrogram conversion," in *INTERSPEECH*. The International Speech Communication Association, 2020.

[4] Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo, "MaskCycleGAN-VC: learning non-parallel voice conversion with filling in frames," in *ICASSP*. IEEE, 2021.

[5] Ju chieh Chou, Cheng chieh Yeh, and Hung yi Lee, "One-shot voice conversion by separating speaker and content representations with instance normalization," in *INTERSPEECH*. The International Speech Communication Association, 2019.

[6] Yen-Hao Chen, Da-Yi Wu, Tsung-Han Wu, and Hung yi Lee, "AGAIN-VC: a one-shot voice conversion using activation guidance and adaptive instance normalization," in *ICASSP*. IEEE, 2021.

[7] L. Sun, K. Li, H. Wang, S. Kang, and H. M. Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *ICME*. IEEE Computer Society, 2016.

[8] S Liu, Y Cao, D Wang, X Wu, X Liu, and H Meng, "Any-to-many voice conversion with location-relative sequence-to-sequence modeling," in *IEEE/ACM Transactions on Audio Speech and Language Processing*, 2021.

[9] Zheng Lian and Zhengqi Wen, "Towards fine-grained prosody control for non-parallel voice conversion," *arXiv preprint arXiv:1910.11269v2*, 2020.

[10] Zhonghao Li, Benlai Tang, Xiang Yin, Yuan Wan, Ling Xu, Chen Shen, and Zejun Ma, "PPG-based singing voice conversion with adversarial representation learning," *arXiv preprint arXiv:2010.14804*, 2020.

[11] Adam Polyak, Lior Wolf, Yossi Adi, and Yaniv Taigman, "Unsupervised cross-domain singing voice conversion," *arXiv preprint arXiv:2008.02830*, 2020.

[12] Songxiang Liu, Yuewen Cao, Na Hu, Dan Su, and Helen Meng, "FastSVC: fast cross-domain singing voice conversion with feature-wise linear modulation," in *ICME*. IEEE Computer Society, 2021.

[13] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, , and W. Chan, "Wavegrad: Estimating gradients for waveform generation," *arXiv preprint arXiv:2009.00713*, 2020.

[14] Anmol Gulati, James Qin, Chung-Cheng Chiu, and Niki Parmar et al., "Conformer: Convolution-augmented transformer for speech recognition," in *INTERSPEECH*. The International Speech Communication Association, 2020.

[15] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "FiLM: visual reasoning with a general conditioning layer," *arXiv preprint arXiv:1709.07871*, 2017.

[16] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae, "HiFi-GAN: generative adversarial networks for efficient and high fidelity speech synthesis," in *NeurIPS*, 2020.

[17] Jong Wook Kim, Justin Salamon, Peter Li, and Juan Pablo Bello, "CREPE: a convolutional representation for pitch estimation," in *ICASSP*. IEEE, 2018.

[18] 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. 99, no. 7, pp. 1877–1884, 2016.

[19] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: an asr corpus based on public domain audio books," in *ICASSP*. IEEE, 2015, p. 5206–5210.

[20] Jacob J Webber, Olivier Perrotin, and Simon King, "Hider-Finder-Combiner: an adversarial architecture for general speech signal modification," in *INTERSPEECH*. The International Speech Communication Association, 2020.

[21] Keith Ito and Linda Johnson, "The LJ speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[22] Yamagishi Junichi, Veaux Christophe, and MacDonald Kirsten, "CSTR VCTK Corpus: english multi-speaker corpus for cstr voice cloning toolkit (version 0.92)," 2019.

[23] Z. Duan, H. Fang, B. Li, K. C. Sim, and Y. Wang, "The NUS sung and spoken lyrics corpus: A quantitative comparison of singing and speech," in *APSIPA ASC*. IEEE, 2013, p. 1–9.

[24] Chen-Chou Lo, Szu-Wei Fu, Wen-Chin Huang, Xin Wang, Junichi Yamagishi, Yu Tsao, and Hsin-Min Wang, "MOSNet: Deep learning based objective assessment for voice conversion," in *Proc. Interspeech 2019*, 2019.

[25] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, "ECAPA-TDNN: emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Interspeech 2020*, Helen Meng, Bo Xu, and Thomas Fang Zheng, Eds. 2020, pp. 3830–3834, ISCA.

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

[27] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *INTERSPEECH*, 2017.

[28] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *INTERSPEECH*, 2018.

[29] Dmitry Bogdanov, Nicolas Wack, and Emilia et al. Gómez, "Essentia: An open-source library for sound and music analysis," 10 2013.