Adversarial-AutoVC: Using Adversarial Loss to Improve Disentanglement for Voice Conversion

Jan Brukner[1], *Kateřina Žmolíková*[1], *Bolaji Yusuf*[1,2], *Lukáš Burget*[1]

[1]Speech@FIT, Faculty of Information Technology, Brno University of Technology, Czechia
[2]Bogazici University, Department of Electrical and Electronics Engineering, Turkey

ibrukner@fit.vut.cz

Abstract

The task of zero-shot voice conversion is to convert utterance from one speaker to another using only small amount of target speaker data. One of ways to achieve this is to disentangle speech into speaker and content dependent information. We use speaker classifier to adversarially train voice conversion model to disentangle these speech attributes. Further, we used Generative Adversarial Network to improve naturalness of resulting speech. We evaluated our systems using automatic speaker verification and automatic speech verification methods. We achieved significant improvement in spoofing rate with minor degradation of word error rate.

Index Terms: voice conversion, adversarial training, GAN

1. Introduction

Voice conversion (VC) is the task of changing characteristics of an utterance of one speaker so that it sounds like the same utterance spoken by another speaker. Such techniques can be used in the entertainment industry, for dubbing, speaker anonymization or for data augmentation in other speech processing fields [1]. Inevitably, voice conversion can be used for various malicious reasons when attempting to fake someone's identity. Development of effective countermeasures will be necessary to deal with this problem.

Voice conversion techniques can be categorized based on how general they are. One-to-one models achieve conversion only for one pair of speakers on which they were trained, many-to-many models change characteristics to and from any speaker in a training data-set. The hardest scenario for voice conversion system is any-to-any (zero-shot) conversion, where the goal is to be able to modify voice *of* and *to* any speaker (i.e. including those not present in the data for training the VC model). Typically, two main approaches are used for zero-show VC: One is direct conversion [2, 3], where speech characteristics are changed continuously in several layers of neural network. The second approach is based on disentanglement, where speech is decomposed into mutually independent components. These components are typically speaker characteristics in form of speaker embedding (this part of the system is frequently trained separately) and some form of content characteristics, which is in form of sequence of content embeddings. Recent publications also try to use prosody information and fundamental frequency contour [4].

One of pioneering zero-shot voice conversion models is AutoVC [5]. AutoVC system uses an *autoencoder* (AE) architecture where the decoder is conditioned on a speaker embedding. It relies on *disentanglement* speaker- and content-related properties of voice, with the aid of a properly designed bottleneck layer. The key idea is that only content information is allowed to flow through the *content encoder*, while all speaker-related information (e.g. timbre) is represented by a speaker embedding, on which the decoder is conditioned. Theoretically, if the bottleneck size is set to be 'just right', the encoder should remove all source speaker traces from the input utterance, retaining only speaker-independent, content-related features. These content-only features are then concatenated with a speaker embedding vector of the target speaker as input to the decoder.

In this work, we modify AutoVC by using a speaker classifier which to classifies speaker from the bottleneck features. This classifier should improve level of disentanglement by adversarially forcing linguistic related part of the system to not produce any information that can be used to classify the source speaker. We hypothesize that this modification will allow us to use larger bottlenecks and to improve naturalness of resulting speech.

Furthermore, we address the issue of over-smooth outputs which often plagues autoencoders, and degrades the quality of the resulting output, by putting the generator into GAN setting which should improve naturalness of produced speech.

2. Method

AutoVC, as autoencoder, consists two main networks, *Content encoder* – E_C and *Decoder* – D. E_C takes mel-spectrogram – $\mathbf{X}$ as input. To help E_C to adapt to source speaker, it also takes source speakers embedding as input. Finally, to achieve information reduction, output of E_C sub-sampled in time-domain resulting into matrix of *content codes* – $\mathbf{Z}$. After encoding, the resulting content

codes are upsampled by copying to match the original time resolution. Architecture of content encoder consists of three convolutional layers with ReLU and batch normalization, followed by two BLSTM layers.

The target speaker embedding is concatenated to each frame of $\mathbf{Z}$ to form the input to the decoder. Which produces mel-spectrogram $\tilde{\mathbf{X}}$ with content from $\mathbf{Z}$ and speaker identity from the embedding. In attempts to address the issue of over-smooth outputs, AutoVC uses *postnet* [6] is applied and computed residuals of the signal is added with the mel-spectrogram predicted by decoder forming the final output $\hat{\mathbf{X}}$. Whole AutoVC system is shown in figure 1.

Speaker embeddings are extracted using a separate network trained for speaker verification task. Original AutoVC uses *d-vectors* [7] for this purpose. In our experiments, we used publicly available implementation of d-vectors – Resemblyzer[1].

AutoVC is trained on reconstruction of the input:

$$\mathcal{L}_{PSNT} = \mathbb{E}\left[\|\hat{\mathbf{X}} - \mathbf{X}\|_2^2\right], \tag{1}$$

To speed up the training, MSE between input and output of the decoder is computed:

$$\mathcal{L}_{D} = \mathbb{E}\left[\|\tilde{\mathbf{X}} - \mathbf{X}\|_2^2\right], \tag{2}$$

Lastly, to preserve content of converted utterances, loss on content codes is introduced:

$$\mathcal{L}_{CD} = \mathbb{E}\left[\|E_C\left(\hat{\mathbf{X}}\right) - \mathbf{Z}\|_1\right] \tag{3}$$

Finally, total loss is computed as

$$\mathcal{L} = \mathcal{L}_{PSNT} + \mathcal{L}_{D} + \mathcal{L}_{E}, \tag{4}$$

To further improve naturalness of resulting speech, we incorporate discriminator network. There are many flavors of GANs, out of which we used WGAN [8] setting, together with differentiable augmentations [9]. Used augmentations are time-spectral masking and Gaussian noise.

Loss function used in this work is adopted from WGAN-GP[10].

2.1. AutoVC with adversarial loss

A major shortcoming of AutoVC system is the bottleneck size guessing game. Using larger bottleneck results in more natural and intelligible speech, but it also makes it harder to separate source speaker information ultimately causing worse target speaker similarity. On the other hand, smaller bottleneck ensures that speaker related information is discarded, but also discards some of the content.

[1]https://github.com/resemble-ai/Resemblyzer

We propose using an adversarial speaker classifier C on the content features $\mathbf{Z}$ as a way to directly remove speaker information, which then allows us to use larger bottleneck. We use a TDNN classifier network trained with the cross-entropy objective, which for an utterance with softmax output, $\mathbf{y} = [y_1, y_2, \ldots, y_K]$ maximizes:

$$\mathcal{L}_{SC} = \log p(\text{speaker} = c|\mathbf{y}) = \log y_c, \tag{5}$$

where c is the index of the correct speaker. Simultaneously, the encoder is trained to maximize the log-likelihood of the *other* speakers:

$$\mathcal{L}_{ADV} = \log p(\text{speaker} \neq c|\mathbf{y}) = \log \sum_{k \neq c} y_k = \log(1 - y_c). \tag{6}$$

This is analogous to the non-saturating form of the GAN objective [11]. Note that it results in a different update rule than conventional gradient reversal [12] which can be shown to maximize the negative log-likelihood of the correct speaker ($-\log y_c$). Moreover, the correct gradients can be computed in a single pass by noting that the gradient of the adversarial loss with respect to the encoder outputs $\mathbf{Z}$:

$$\frac{\delta \mathcal{L}_{ADV}}{\delta \mathbf{Z}} = \frac{\delta \mathcal{L}_{ADV}}{\delta \mathcal{L}_{SC}} \frac{\delta \mathcal{L}_{SC}}{\delta \mathbf{Z}} = \frac{\delta \log(1 - y_c)}{\delta \log y_c} \frac{\delta \mathcal{L}_{SC}}{\delta \mathbf{Z}} = -\frac{y_c}{1 - y_c} \frac{\delta \mathcal{L}_{SC}}{\delta \mathbf{Z}}. \tag{7}$$

This can be interpreted as gradient reversal with an adaptive weight computed from the softmax output. Intuitively, the term $\frac{y_c}{1-y_c}$ can be seen as an adaptive "gain control" which amplifies encoder gradients when the classifier correctly classifies gradients and attenuates them as the classifier gets better fooled. This can be seen by considering cases where the classifier becomes very good at predicting the correct speaker. then the gradient $\frac{\delta \mathcal{L}_{SC}}{\delta \mathbf{Z}}$ becomes close to zero and ordinary gradient reversal cannot correct the encoder. However, with our non-saturating formulation, under that same condition, the term $\frac{y_c}{1-y_c}$ tends toward infinity, resulting in substantial amplification of the gradients.

3. Experiments

For conversion from mel-spectrogram back to time domain, we used pre-trained MelGAN[2] [13] vocoder, which is based on TDNN network with temporal context 9 frames and 3 dilated convolutional layers with 2048 channels and kernel size 5 before pooling. Our discriminator network is adopted from StarGAN-VC2 [2], based on public implementation[3]. The original network also used

[2]https://github.com/descriptinc/melgan-neurips

[3]https://github.com/SamuelBroughton/StarGAN-Voice-Conversion-2

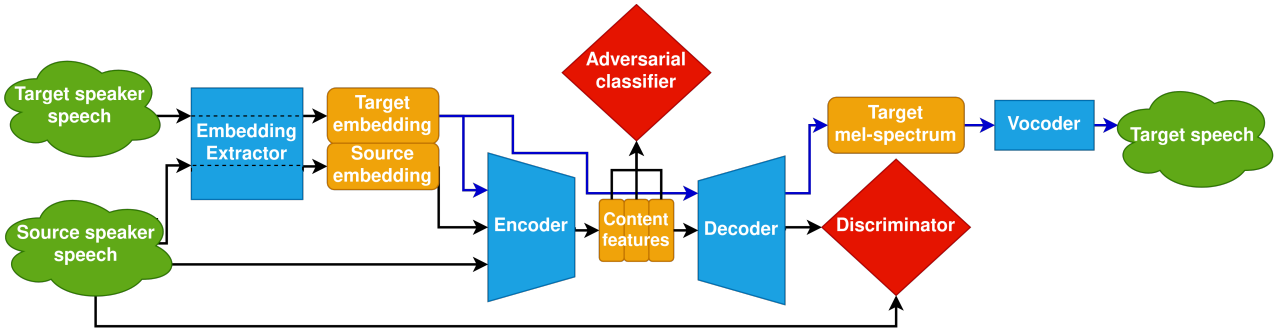

Figure 1: *Schematic diagram of modified AutoVC system. Note that decoder block consists of decoder itself and post-network. Red blocks are part of our modifications.*

speaker embeddings, we removed this part to change it into classic discriminator (or critic, as we use WGAN loss). AutoVC bottleneck size is 32 in frequency domain and sub-sampling period is 8 through all our experiments.

During development, we found that the adversarial classifier gets very quickly fooled during the adversarial training and becomes unable to discriminate between target speakers (i.e. it always predicts uniform distribution over the target speaker classes). This is the desirable final state fort he adversarial training suggesting that all speaker information should be removed from the content features. However, when we trained an independent speaker classifier on the content features (as a sanity check) we saw that it is still able to recognize the speakers. This is is similar to the *mode collapse* problem known from standard GAN training [14]. To address this issue, we used soft-max probabilities from this another classifier to construct an adaptive term.

3.1. Dataset

We used CSTR Voice Cloning toolkit (version 0.92) (VCTK) [15] dataset for training and evaluation of VC systems. VCTK consists of 110 speakers with around 400 utterances for each speaker. There are two tracks for each utterance *mic1* and *mic2*, from which we took only *mic1* version which corresponds to older VCTK version. Text transcriptions are also provided for each speaker (except for p315). Since our focus is to evaluate one-shot VC systems, we split speakers into three groups: 10 speakers are completely excluded from training (one-shot speakers), 10 speakers have half of utterances in train set (many-to-many speakers) and remaining 90 speakers are left for training. Train dataset is further divided into train and validation sets in 9:1 ratio.

In test set, female and male speakers are equally distributed and the same speakers as in [16] are taken[4]. Original utterances were downsampled to 22.05 Hz.

[4] http://speech.ee.ntu.edu.tw/~jery2243542/resource/model/is18/en_speaker_used.txt

3.2. Baselines

As baseline systems we use the original AutoVC[5] [5]. We use the original dimensionality (32) for the content features, but we experiment with different sub-sammpling of the content features (original sub-sampling period 32, and also 16 and 8) to obtain voice conversion of different quality. As a second baseline, we used the publicly available VC system FragmentVC[6] [17].

3.3. Evaluation setup

Traditionally, subjective measures, namely Mean Opinion Score (MOS) and speaker similarity, are used to evaluate VC systems. However, these methods, if done properly, need a lot of preparations ahead and often turn out to be costly. Also, once we start using VC systems to convert noisier data, human listeners may become unreliable due to low quality (and therefore MOS) of the original recordings. From large-scale evaluations done in Voice Conversion Challenges [18, 19, 20], it seems that MOS are well correlated with the speaker similarity scores, where human listeners were simply asked to asses whether a pair of recordings comes from the same speaker or not. Therefore, the speaker verification scores produced by trained automatic speaker verification (ASV) can serve as a good proxy for the human generated speaker similarity scores, leading to objective (rather than subjective) evaluation of the quality of VC system.

For our evaluation, we conduct automatic speaker verification (ASV) and automatic speech recognition (ASR) tests, which have been shown to be highly correlated with both subjective measures [21]. In attempt to make our evaluation also reproducible, we adopted models from *speechbrain*[7] [22] toolkit. Speechbrain comes conveniently with pretrained models, which are used in black-box manner for our purposes. We choose the following systems:

[5] https://github.com/auspicious3000/autovc
[6] https://github.com/yistLin/FragmentVC/
[7] https://github.com/speechbrain/speechbrain/

- TDNN x-vector model trained on VoxCeleb1&2 using Categorical Cross-Entropy Loss[8] [23]
- For ASR, CRDNN with CTC/Attention and RNNLM trained on LibriSpeech is used[9]

Regarding ASV, we evaluate how good the VC systems are in terms of fooling the ASV system by claiming somebody else's identity (i.e. spoofing atack). In particlular, we report *spoofing rate*, which is evaluated as follows:

For each speaker in the test set (which will serve as a target speaker), target speaker model is created by averaging embeddings extracted from all its original utterances (i.e without VC). Cosine similarity between a speaker models and an embedding extracted from an utterance is used as the similarity score. On the original utterances, we estimate the threshold that would correspond to the EER operating point and we fix this threasold. Now, we evaluated the cosine similarity between each utterance processed by the VC system and the corresponding target speaker model (i.e. the model corresponding to the claimed identity). The reported *spoofing rate* corresponds to the percentage of converted utterances that exceeded the threshold (i.e. managed to fool the ASV system).

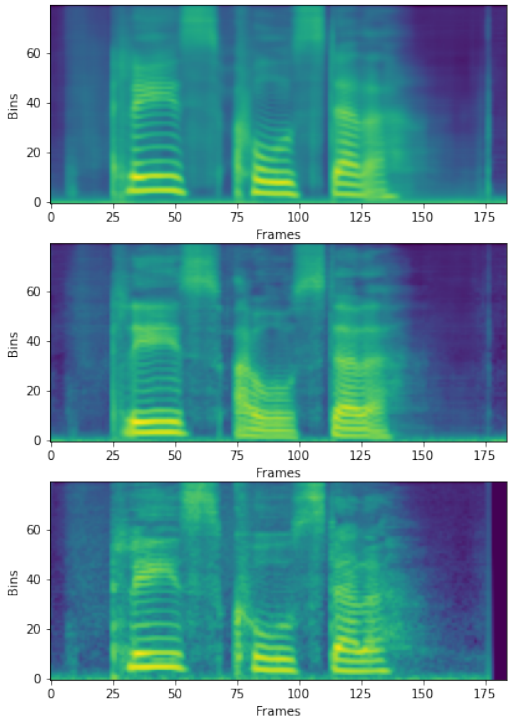

Figure 2: *Difference between mel-spectrogram outputs of AutoVC autoencoder without (top) and with (middle) GAN objective. For comparison, real mel-spectrogram is in the bottom.*

[8]https://huggingface.co/speechbrain/spkrec-xvect-voxceleb
[9]https://huggingface.co/speechbrain/asr-crdnn-rnnlm-librispeech

Table 1: *WER and spoofing results.*

System	WER [%]	Spoofing rate
Ground truth	6.9	-
MelGAN	9.6	-
AutoVC 32x32	25.83	0.17
AutoVC 32x8	12.44	0.01
AutoVC 32x16	17.13	0.02
FragmentVC	14.43	0.09
Ours	45.56	0.18
Ours + GAN	29.59	**0.38**

ASR results are more straightforward with computing Word Error Rate (WER) for all test utterances.

For evaluation, we used first 10 utterances from each speaker for conversions. Speaker embeddings were generated from concatenated utterances with total length of 10 seconds.

3.4. Results

Our results are in Table 1. It shows, that adversarial classifier is in fact capable to increase level of disentanglement. Our adversarial system is capable to match spoofing rate of AutoVC with tuned bottleneck while our generator is AutoVC 32x8. Interestingly, even though our bottleneck size is larger, it produces worse WER than AutoVC with smaller one. Using GAN helps quite a lot with word error rate, but rather counter-intuitively hugely increases spoofing rate.

In figure 2, we can see, the difference between oversmooth output of autoencoder and more realistic output from the network trained with GAN objective.

4. Conclusions

We propose adversarial classifier on AutoVC system, which operates directly on features we want to be speaker agnostic. We have shown that adversarial training can improve level of disentanglement and further improve voice conversion systems. We have shown that this method combined with GAN objective greatly improves spoofing rate of the VC system. In future work, we will focus on solving the mode collapse like problem and further improve naturalness in terms of word error rate.

5. Acknowledgements

The work was supported by Czech National Science Foundation (GACR) project "NEUREM3" No. 19-26934X. Part of high-performance computation run on IT4I supercomputer and was supported by the Ministry of Education, Youth and Sports of the Czech Republic through e-INFRA CZ (ID:90140).

6. References

[1] T. Glarner, J. Ebbers, and R. Häb-Umbach, "Voice conversion based speaker normalization for acoustic unit discovery," *arXiv preprint arXiv:2105.01786*, 2021.

[2] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Stargan-vc2: Rethinking conditional methods for stargan-based voice conversion," *Proc. Interspeech 2019*, pp. 679–683, 2019.

[3] Y. A. Li, A. Zare, and N. Mesgarani, "StarGANv2-VC: A Diverse, Unsupervised, Non-Parallel Framework for Natural-Sounding Voice Conversion," in *Proc. Interspeech 2021*, 2021, pp. 1349–1353.

[4] K. Qian, Y. Zhang, S. Chang, M. Hasegawa-Johnson, and D. Cox, "Unsupervised speech decomposition via triple information bottleneck," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 7836–7846. [Online]. Available: https://proceedings.mlr.press/v119/qian20a.html

[5] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "Autovc: Zero-shot voice style transfer with only autoencoder loss," *arXiv preprint arXiv:1905.05879*, 2019.

[6] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 4779–4783.

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

[8] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.

[9] S. Zhao, Z. Liu, J. Lin, J.-Y. Zhu, and S. Han, "Differentiable augmentation for data-efficient gan training," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7559–7570, 2020.

[10] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, 2017.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[12] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning*. PMLR, 2015, pp. 1180–1189.

[13] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brebisson, Y. Bengio, and A. Courville, "Melgan: generative adversarial networks for conditional waveform synthesis," pp. 14 910–14 921, 2019.

[14] V. Kushwaha, G. Nandi *et al.*, "Study of prevention of mode collapse in generative adversarial network (gan)," in *2020 IEEE 4th Conference on Information & Communication Technology (CICT)*. IEEE, 2020, pp. 1–6.

[15] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, "Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2016.

[16] J.-c. Chou, C.-c. Yeh, H.-y. Lee, and L.-s. Lee, "Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations," *arXiv preprint arXiv:1804.02812*, 2018.

[17] Y. Y. Lin, C.-M. Chien, J.-H. Lin, H.-y. Lee, and L.-s. Lee, "Fragmentvc: Any-to-any voice conversion by end-to-end extracting and fusing fine-grained voice fragments with attention," pp. 5939–5943, 2021.

[18] T. Toda, L.-H. Chen, D. Saito, F. Villavicencio, M. Wester, Z. Wu, and J. Yamagishi, "The voice conversion challenge 2016," in *Proc. INTERSPEECH*, 2016, pp. 1632–1636.

[19] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z. Ling, "The voice conversion challenge 2018: promoting development of parallel and nonparallel methods," in *Proc. Odyssey 2018*, 2018, pp. 195–2026.

[20] Z. Yi, W.-C. Huang, X. Tian, J. Yamagishi, R. K. Das, T. Kinnunen, Z.-H. Ling, and T. Toda, "Voice Conversion Challenge 2020 — Intra-lingual semi-parallel and cross-lingual voice conversion —," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, 2020, pp. 80–98. [Online]. Available: http://dx.doi.org/10.21437/VCC$_B C$.2020 $-$ 14

[21] R. K. Das, T. Kinnunen, W.-C. Huang, Z.-H. Ling, J. Yamagishi, Z. Yi, X. Tian, and T. Toda, "Predictions of Subjective Ratings and Spoofing Assessments of Voice Conversion Challenge 2020 Submissions," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, 2020, pp. 99–120. [Online]. Available: http://dx.doi.org/10.21437/VCC$_B C$.2020 $-$ 15

[22] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "SpeechBrain: A general-purpose speech toolkit," 2021, arXiv:2106.04624.

[23] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, "Spoken language recognition using x-vectors," pp. 105–111, 2018.