Investigating Deep Neural Structures and their Interpretability in the Domain of Voice Conversion

Samuel J. Broughton[1], *Md Asif Jalal*[2], *Roger K. Moore*[2]

[1]School of Computing, National University of Singapore, Singapore
[2]Department of Computer Science, University of Sheffield, UK
sbrought@comp.nus.edu.sg, {majalal1,r.k.moore}@sheffield.ac.uk

Abstract

Generative Adversarial Networks (GANs) are machine learning networks based around creating synthetic data. Voice Conversion (VC) is a subset of voice translation that involves translating the paralinguistic features of a source speaker to a target speaker while preserving the linguistic information. The aim of non-parallel conditional GANs for VC is to translate an acoustic speech feature sequence from one domain to another without the use of paired data. In the study reported here, we investigated the interpretability of state-of-the-art implementations of non-parallel GANs in the domain of VC. We show that the learned representations in the repeating layers of a particular GAN architecture remain close to their original random initialised parameters, demonstrating that it is the number of repeating layers that is more responsible for the quality of the output. We also analysed the learned representations of a model trained on one particular dataset when used during transfer learning on another dataset. This also showed high levels of similarity in the repeating layers. Together, these results provide new insight into how the learned representations of deep generative networks change during learning and the importance of the number of layers, which would help build better GAN-based speech conversion models.[3]

Index Terms: voice conversion (VC), generative adversarial networks (GANs), canonical correlation analysis (CCA), SVCCA, transfer learning, non-parallel VC, multi-domain VC

1. Introduction

Deep Learning networks have been shown to exhibit superior abilities in a range of problem domains [1, 2, 3]. However, such networks are *black-box representations* in terms of their interpretability [4], and this can mitigate against informed decision making when selecting appropriate network configurations.

One problem domain, voice conversion (VC), or voice style transfer, is a technique aimed at modifying the linguistic style of speech while preserving the linguistic information contained therein [5, 6, 7]. VC can be formulated as a regression problem with the aim of building a function in which the features of a source speaker A can be mapped to a target speaker B [8, 6, 7]. Applications of VC include modifying speaker identity in text-to-speech (TTS) systems [9], aiding those with vocal disabilities [10] and generating accents for assisted language conversion in domains such as real-time language translation and device-assisted language learning [11].

Historically, methods employed to achieve VC have included mapping code books [12], Gaussian mixture models (GMMs) [8, 9, 13] and artificial neural networks (ANNs) [14, 15]. However, variations of generative adversarial networks (GANs) [16] have recently shown success in a range of different domains, such as producing convincingly real images and videos [2, 17], enhancing the quality of images [1], generating new music [18] and, of interest here, a methodology for achieving VC [19, 20, 21, 3, 22, 23].

Some of the VC methods mentioned above can be categorized as either parallel or non-parallel. Parallel VC refers to source and target speaker utterances being perfectly aligned [6, 7]. Such data can be a laborious task to collect. Furthermore, once collected, the data would need to be pre-processed with automatic time alignment which can fail, resulting in other methods of correction. However, GANs are able to learn mapping functions between data of similar domains and so mitigate the need for a parallel dataset [2]. Recent state-of-the-art non-parallel generative VC architectures include CycleGAN-VC2 [3] and StarGAN-VC2 [21]. Both make use of a gated convolutional neural network (CNN) [24], identity-mapping loss [25] and *2-1-2D CNN* architecture [3].

A major advantage of using the StarGAN [26] framework when compared to CycleGAN [2], is the ability to perform multi-domain conversion whilst only requiring a single generator. With regards to VC, the StarGAN framework allows for learned mapping functions between multiple speakers. Extending this framework, StarGAN-VC [27] and StarGAN-VC2 [21] make various modifications including updates to the training objective and alterations to the network architecture.

However, despite StarGAN-VC2 demonstrating superior VC in both objective and subjective experiments when compared to StarGAN-VC [21], there has been very little investigation into the interpretability of it's network representations - as is the case with many deep neural networks.

In this work, we conducted an evaluation of learned network representations by performing Singular Vector Canonical Correlation Analysis (SVCCA) [28] in a range of different experiments using an adaptation of the StarGAN-VC2 network. The aim was to provide insights into issues relating to the similarity of learned representations to their random initial states. This was achieved by conducting experiments with networks including frozen layers, observing how quickly networks reached their optimal representations, exploring the effects of modifying the size of networks and investigating learned network representations when trained using transfer learning.

The rest of the paper is structured as follows: Section 2 outlines the generative network architecture used, Section 3 reviews SVCCA, Section 4 describes the experimental conditions and research questions, Section 5 discusses the results, and Section 6 presents the conclusion.

[1]This work was carried out whilst at the University of Sheffield.

[3]Audio samples available at: https://samuelbroughton.github.io/interpretability-demo-2020.

2. Generative Network Architecture

The network architecture implemented for the experiments presented in this paper was based on StarGAN-VC2 [21], which allows for non-parallel many-to-many learned mappings for VC.

2.1. Training objectives

The main objective of the StarGAN framework [26] is to learn many-to-many mapping functions between multiple domains whilst only using a single generator G. StarGAN does this by conditioning itself on 'one-hot' representations of domain codes $c \in \{1, ..., N\}$, where c and N indicate the domain code the number of domains, respectively. More specifically in StarGAN-VC2, G can be formulated as the mapping function $G(x, \hat{c}) \rightarrow \hat{x}$, taking an acoustic input feature sequence $x \in \mathbb{R}^{Q \times T}$ and target domain code $\hat{c}$ to generate an acoustic output feature sequence $\hat{x}$. StarGAN-VC2 does this by making use of an adversarial loss [16], reconstruction or cycle-consistency loss [2] and identity-mapping loss [25].

Adversarial loss is used in GANs to encourage generated data, conditioned on target domain code, to be indistinguishable to that of real data [26]. Here, we implement an adversarial loss to help G generate an output closer to the real target data:

$$\begin{aligned}\mathcal{L}_{adv} &= \mathbb{E}_{(x,c)\sim P(x,c)}[\log D(x,c)] \\ &+ \mathbb{E}_{x\sim P(x),\hat{c}\sim P(\hat{c})}[\log D(G(x,\hat{c}),\hat{c})],\end{aligned} \tag{1}$$

where D is a real/fake discriminator that attempts to maximise this loss to learn the decision boundary between real and fake features. G attempts to minimize this loss by generating an output indistinguishable to the real acoustic features of target domain $\hat{c}$.

Cycle-consistency loss is used in order to guarantee that the converted output feature sequence preserves the source characteristics of input feature sequence x [2, 26]:

$$\mathcal{L}_{cyc} = \mathbb{E}_{(x,c)\sim P(x,c),\hat{c}\sim P(\hat{c})}[||x - G(G(x,\hat{c}),c)||_1]. \tag{2}$$

This cyclic constraint encourages G to reconstruct the original input feature x from the generated output $\hat{x}$ and source domain code c. This helps G to preserve the linguistic information of the speech [27].

Identity-mapping loss is employed to encourage the preservation of input feature identity within generated output data [25]:

$$\mathcal{L}_{id} = \mathbb{E}_{(x,c)\sim P(x,c)}[||G(x,c) - x||]. \tag{3}$$

Identity-mapping loss has previously been used in image-to-image translation for colour preservation [2].

The **full objective** can be summarised as follows:

$$\mathcal{L}_G = \mathcal{L}_{st\text{-}adv} + \lambda_{cyc}\mathcal{L}_{cyc} + \lambda_{id}\mathcal{L}_{id}, \tag{4}$$

$$\mathcal{L}_D = -\mathcal{L}_{st\text{-}adv}, \tag{5}$$

where λ_{cyc} and λ_{id} are hyperparameters for each term. Here, G aims to minimise the loss whilst D is trying to maximise it.

2.2. Network architectures

The fully convolutional GAN architecture used in the study reported here allows for acoustic input feature sequences of arbitrary sizes.

Generator: The input to G was an image of size $Q \times T$ of an acoustic feature sequence x, where Q and T are the feature dimension and sequence length, respectively. A *2-1-2D CNN* [3, 21] architecture was used to construct G. 2D convolutions are well suited for holding the original data structure whilst 1D convolutions work well at dynamically changing the data [3]. The implementation specifically used a gated CNN [24], which allowed for relevant features to be selected and propagated based on previous layer states. The effectiveness of a gated CNN for VC has already been confirmed in previous studies [27, 20].

Conditional domain specific style code was injected in the 1D CNN architecture by a modulation-based method [21]. Conditional instance normalisation (CIN) [29] was used to modulate parameters in a domain-specific manner, and were defined as:

$$\text{CIN}(f, \hat{c}) = \gamma_{\hat{c}}\left(\frac{f - \mu(f)}{\sigma(f)}\right) + \beta_{\hat{c}}, \tag{6}$$

where $\mu(f)$ and $\sigma(f)$ are the average and standard deviation of feature f and $\gamma_{\hat{c}}$ and $\beta_{\hat{c}}$ are domain-specific scale and bias parameters, respectively.

The 1D repeating blocks were not residual because the use of skip connections was reported to result in partial conversion [30].

Real/Fake Discriminator: A 2D gated CNN [24] was used for the architecture of the real/fake discriminator D, which has been formulated as a projection discriminator [31], as seen in StarGAN-VC2 [21]. D outputs a sequence of probabilities, calculating how close the input acoustic feature sequence x is to domain c.

3. SVCCA on Deep Neural Representations

Singular Vector Canonical Correlation Analysis (SVCCA) is an extension of Canonical Correlation Analysis (CCA), a method used in statistics to measure the similarity of two vectors formed by some *underlying process* [34, 35, 36]. In the case of deep neural networks, these are the "neuron activation vectors" formed from training on a particular dataset [36, 28]. A single neuron activation vector is the output of a single neuron of a layer in a network. Combining the outputs of all neurons for a particular layer in a network results in a set of multidimensional output [36, 28]. Subsequently, CCA can be used to compare the similarity between two layers of the same network, similar networks using layers of same/differing dimensionality, or a given layer at different stages of training [36]. Given subspaces $X = \{x_1, ..., x_T\}$ and $Y = \{y_1, ..., y_N\}$, maximal correlation between X and Y is calculated by finding bases w, s for two matrices such that the subspaces are projected onto these bases

$$\frac{w^T \sum_{XY} s}{\sqrt{w^T \sum_{XX} w}\sqrt{s^T \sum_{YY} s}} \tag{7}$$

where $\sum_{XX}, \sum_{XY}, \sum_{YY}$ are the covariance and cross-covariance. SVCCA is an extension to CCA that involves a preprocessing step [36, 28]. The authors of [28] explain that SVCCA takes the same inputs as CCA, for example two layers of a neural network l_1 and l_2 that each contain a set of neuron activation vectors. SVCCA then factorises the vectors by computing Singular Value Decomposition (SVD) over each layer to obtain subspaces $l_1' \subset l_1$ and $l_2' \subset l_2$. These subspaces contain the most important variance directions, which can account for 99% of the variance in input layers l_1 and l_2 [28]. CCA is then performed on l_1' and l_2' to return the correlation coefficients, providing a measure of similarity of the two layers.

4. Experiments

Datasets: To evaluate our methods, we made use of the Device and Produced Speech Dataset [37], as seen in the multi-speaker VC task in the Voice Conversion Challenge 2018 (VCC2018) [7] and the English Multi-speaker Corpus for CSTR Voice

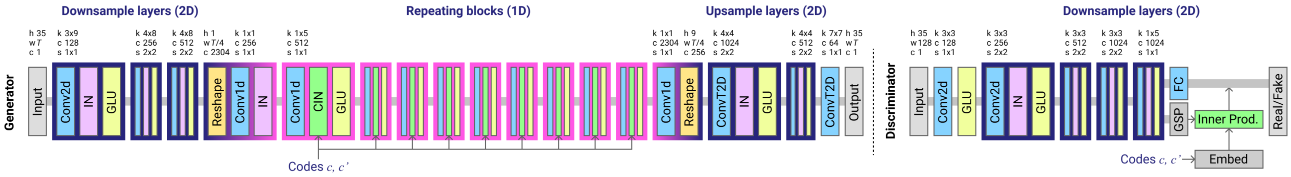

Figure 1: *Network architectures of the fully convolutional [32] generator and discriminator based on StarGAN-VC2 [21]. In the input, output and reshape layers 'h', 'w' and 'c' represent the height, width and channel number respectively. In the Conv2d, Conv1D and ConvT2D convolution layers, 'k', 'c' and 's' represent the kernel size, channel number and stride, respectively. 'IN', 'GLU', 'GSP' and 'FC' denote instance normalisation [33], gated linear unit [24], global sum pooling and fully connected layers, respectively.*

Cloning Toolkit (CSTR VCTK) [38]. We used a subset of both datasets in all experiments except during transfer learning where the initial model was trained using the CSTR VCTK dataset.

In both datasets four speakers were selected covering all inter- and intra-gender conversions. In the CSTR VCTK dataset we selected speakers labelled `p262`, `p229`, `p272` and `p232`; speakers `p262` and `p229` are female, and speakers `p272` and `p232` are male. The data from VCC2018 mimicked the data used to test StarGAN-VC2 [21], whereby `VCC2SF1` and `VCC2SF2` are female speakers, and `VCC2SM1` and `VCC2SM2` are male speakers.

For each experiment, $4 \times 3 = 12$ source-and-target pair mappings were learnt for each single model trained on both datasets. All the recordings for both datasets were downsampled to 22.05 kHz. 36 Mel-cepstral coefficients (MCEPs) were extracted from each recording. The logarithmic fundamental frequency ($\log F_0$) and aperiodicities (APs) were extracted every 5 ms using the WORLD vocoder [39].

Conversion process: The conversion process mimicked that of StarGAN-VC [27] and StarGAN-VC2 [21], by not using any form of post filtering [40, 41] or powerful vocoding [42, 43] and just focusing on MCEP conversion. As in previous studies, the WORLD vocoder [39] was used to synthesise speech, directly taking APs and converting the $\log F_0$ using a logarithm Gaussian normalised transformation [44].

Network implementations: Figure 1 presents the network architectures for G and D, influenced by StarGAN-VC2 [21] and CycleGAN-VC2 [3][4]. The networks were initially trained for 2.5×10^5 batch iterations on both datasets. During transfer learning, optimal models trained on the CSTR VCTK dataset were selected and trained for an extra 1.5×10^5 batch iterations on the VCC2018 dataset. During the training of the networks for all experiments, the states for G and D were saved at every 1×10^4 batch iterations. All networks were trained using the 'Adam' optimizer [45] with a momentum term β_1 set to 0.5. The batch size was set to 8 and we randomly cropped segments of 512 frames from randomly selected sentences. Learning rates for G and D were both set to 0.0001, $\lambda_{cyc} = 10$ and $\lambda_{id} = 5$. $\mathcal{L}_{id}$ was only used for the first 10^4 iterations.

Experimental investigation: Experiments were conducted in order to provide insights into questions relating to the interpretability of the trained networks. Experiment 1 addressed the issue as to how similar the learned representations of the optimally trained network are to its random initialisation. Experiment 2 addressed the question of how similar the learned representations of networks trained via transfer learning on a new dataset are to their previously optimal representations when trained on the original dataset. Experiment 3 addressed the issue as to how similar the learned representations of networks with various frozen repeating layers are. Experiment 4 addressed the question of how the quality of the output feature sequence changes with networks of a differing number of repeating layers.

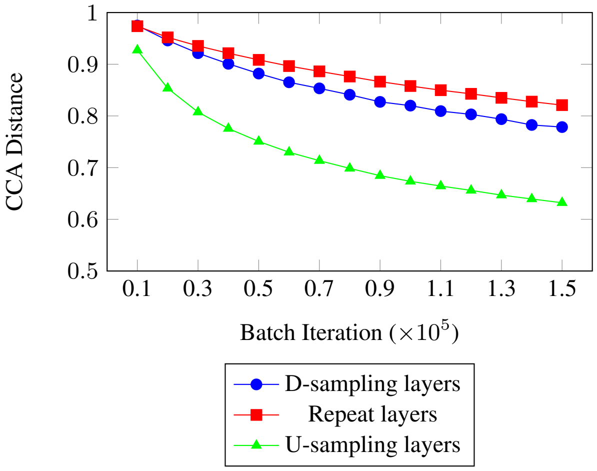

Figure 2: *Average CCA distance between the downsampling, repeating and upsampling portions of the trained network and its random initial states (trained on the CSTR VCTK dataset).*

5. Results and Discussions

Experiment 1: To assess how close the optimally trained network's learned parameters were to their random initialisations, SVCCA was used to compare networks at 0 and their optimal number of batch iterations. The number of optimal batch iterations for networks trained on the CSTR VCTK and VCC2018 datasets were found to be approximately 1.5×10^5 and 1.2×10^5, respectively.

Figures 2 and 3 show the CCA distance between the learned parametric representations of layers in the network at different stages of training and their random initialisation. Both figures show a greater correlation of similarity in the learned network representations of the repeating 1D CNN layers (R1-R9) and their random initial states when compared to the less similar downsampling and upsampling portions of the network. Both figures show networks trained using the CSTR VCTK dataset however, similar results were observed with the VCC2018 dataset.

The extreme similarity observed at D1 can be seen as a fundamental trait of these networks. During pre-experiments,

[4] StarGAN-VC2 source code available at: `https://github.com/SamuelBroughton/StarGAN-Voice-Conversion-2`

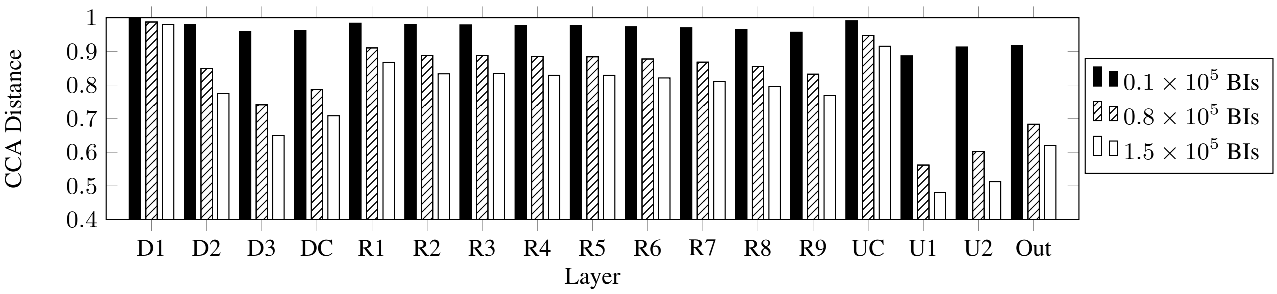

Figure 3: *CCA distance between each layer of a network at different stages of training and its random initial states, where "BI" denotes batch iteration trained on the CSTR VCTK dataset.*

this trait was also seen in training StarGAN-VC [27]. We removed the GLU of the first downsampling layer to check if this was preventing the first convolution from learning as much as it could. However, the same trait was still observed.

The results show that the parameters learned for network representations in these optimally trained networks remain close to their initial random states, especially in the repeating 1D CNN layers.

Experiment 2: The optimal model trained on the CSTR VCTK dataset in experiment 1 was used as the initial state for training during transfer learning on the VCC2018 dataset. Similar to experiment 1, the repeating 1D portion of the network remained closer to its random initialisations. As expected, this gave rise to a better output quality of samples converted using this transfer learning model when compared with the original model trained on the same dataset.

Experiment 3: The motivation of this experiment is to find inter-dependencies between the layers over time in the training process. By freezing corresponding layers to their initial random state, we aim to find the layers' learning dynamics. The random initial state of the models trained in experiment 1 were used to train networks with various frozen layers in the repeating portion of the network. A total of four were evaluated with various frozen layers, the first of which froze R2, R3 and R4. The second network froze R5 and R6, and the third network froze R7 and R8. The similarity of these networks was then calculated against the optimally trained model from experiment 1. The repeating 1D layers again showed a high degree of similarity in their learned network representations and were extremely similar in terms of their acoustic output.

Experiment 4: This experiment's motivation is to analyse the extent of repeating convolution 1D layers in an audio generation. The random initial state of the models trained in experiment 1 were used to train networks with differing numbers of repeating 1D layers. Four models were trained with 3, 6, 12 and 15 repeating layers in addition to the previously trained model from experiment 1, which had 9 repeating layers. It was observed that, in general, the audio quality of the models using 3, 6 and 9 repeating layers sounded better than the models using 12 and 15 layers. However, each model included at least one instance of having a worse quality of output than their counterparts for various different source-target pairs.

It was also observed that, as the number of repeating layers increased, the modification of speaker identity was more pronounced. In other words, the output from models with a greater number of repeating layers had clearer accents than the output of those with fewer repeating layers. However, at some points the modification of speaker identity was so pronounced that the intelligibility of the audio began to deteriorate. Also, as the amount of repeating layers of the network increased, so the overall level of noise increased.

6. Conclusions

In the research reported here, we provide new insights into the interpretability of Generative Adversarial Networks (GANs) for Voice Conversion (VC). Using a network architecture based on StarGAN-VC2 [21], we conducted an investigation into the learned representations of the network over a range of different experimental conditions. The results showed that there is at least one local optimum that lies close to the random initial states of the network. It was also found that it is the number of repeating layers in the network architecture that has a noticeable effect on the quality of the output speech. In general, as the number of repeating layers in the network increased, so too did the noise and certain aspects of speaker identity became more pronounced. Future work will involve looking more into the importance of network depth in GANs for VC.

7. References

[1] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[2] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[3] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Cyclegan-vc2: Improved cyclegan-based non-parallel voice conversion," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6820–6824.

[4] Q.-s. Zhang and S.-C. Zhu, "Visual interpretability for deep learning: a survey," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 1, pp. 27–39, 2018.

[5] S. H. Mohammadi and A. Kain, "An overview of voice conversion systems," *Speech Communication*, vol. 88, pp. 65–82, 2017.

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

[7] 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," *arXiv preprint arXiv:1804.04262*, 2018.

[8] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Transactions on speech and audio processing*, vol. 6, no. 2, pp. 131–142, 1998.

[9] H. Kawanami, Y. Iwami, T. Toda, H. Saruwatari, and K. Shikano, "Gmm-based voice conversion applied to emotional speech synthesis," in *Eighth European Conference on Speech Communication and Technology*, 2003.

[10] K. Nakamura, T. Toda, H. Saruwatari, and K. Shikano, "Speaking-aid systems using gmm-based voice conversion for electrolaryngeal speech," *Speech Communication*, vol. 54, no. 1, pp. 134–146, 2012.

[11] D. Felps, H. Bortfeld, and R. Gutierrez-Osuna, "Foreign accent conversion in computer assisted pronunciation training," *Speech communication*, vol. 51, no. 10, pp. 920–932, 2009.

[12] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," *Journal of the Acoustical Society of Japan (E)*, vol. 11, no. 2, pp. 71–76, 1990.

[13] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[14] S. Desai, E. V. Raghavendra, B. Yegnanarayana, A. W. Black, and K. Prahallad, "Voice conversion using artificial neural networks," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 3893–3896.

[15] S. H. Mohammadi and A. Kain, "Voice conversion using deep neural networks with speaker-independent pre-training," in *2014 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2014, pp. 19–23.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[17] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410.

[18] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[19] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks," *arXiv preprint arXiv:1704.00849*, 2017.

[20] T. Kaneko, H. Kameoka, K. Hiramatsu, and K. Kashino, "Sequence-to-sequence voice conversion with similarity metric learned using generative adversarial networks." in *INTERSPEECH*, vol. 2017, 2017, pp. 1283–1287.

[21] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Stargan-vc2: Rethinking conditional methods for stargan-based voice conversion," *arXiv preprint arXiv:1907.12279*, 2019.

[22] D. Paul, Y. Pantazis, and Y. Stylianou, "Non-parallel voice conversion using weighted generative adversarial networks," *Proc. Interspeech 2019*, pp. 659–663, 2019.

[23] H.-Y. Lee and Y. Tsao, "Generative adversarial network and its applications to speech processing and natural language processing," Speech Processing Laboratory, National Taiwan University. http://speech.ee.ntu.edu.tw/~tlkagk/GAN_3hour.pdf, (Accessed: November 19, 2019).

[24] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 933–941.

[25] Y. Taigman, A. Polyak, and L. Wolf, "Unsupervised cross-domain image generation," *arXiv preprint arXiv:1611.02200*, 2016.

[26] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8789–8797.

[27] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Stargan-vc: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 266–273.

[28] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein, "Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability," in *Advances in Neural Information Processing Systems*, 2017, pp. 6076–6085.

[29] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," *arXiv preprint arXiv:1610.07629*, 2016.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[31] T. Miyato and M. Koyama, "cgans with projection discriminator," *arXiv preprint arXiv:1802.05637*, 2018.

[32] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[33] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[34] H. Hotelling, "Relations between two sets of variates," in *Breakthroughs in statistics*. Springer, 1992, pp. 162–190.

[35] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

[36] A. Morcos, M. Raghu, and S. Bengio, "Insights on representational similarity in neural networks with canonical correlation," in *Advances in Neural Information Processing Systems*, 2018, pp. 5727–5736.

[37] G. 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.

[38] C. Veaux, J. Yamagishi, and K. MacDonald, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit, [sound]," Datashare, Edinburgh. https://doi.org/10.7488/ds/1994, (Accessed: December 16, 2019).

[39] 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.

[40] T. Kaneko, H. Kameoka, N. Hojo, Y. Ijima, K. Hiramatsu, and K. Kashino, "Generative adversarial network-based postfilter for statistical parametric speech synthesis," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4910–4914.

[41] T. Kaneko, S. Takaki, H. Kameoka, and J. Yamagishi, "Generative adversarial network-based postfilter for stft spectrograms." in *INTERSPEECH*, 2017, pp. 3389–3393.

[42] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[43] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-dependent wavenet vocoder." in *Interspeech*, vol. 2017, 2017, pp. 1118–1122.

[44] K. Liu, J. Zhang, and Y. Yan, "High quality voice conversion through phoneme-based linear mapping functions with straight for mandarin," in *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, vol. 4. IEEE, 2007, pp. 410–414.

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.