STARGAN-VC: NON-PARALLEL MANY-TO-MANY VOICE CONVERSION USING STAR GENERATIVE ADVERSARIAL NETWORKS

Hirokazu Kameoka, Takuhiro Kaneko, Kou Tanaka, Nobukatsu Hojo

NTT Communication Science Laboratories, NTT Corporation, Japan

ABSTRACT

This paper proposes a method that allows non-parallel many-to-many voice conversion (VC) by using a variant of a generative adversarial network (GAN) called StarGAN. Our method, which we call StarGAN-VC, is noteworthy in that it (1) requires no parallel utterances, transcriptions, or time alignment procedures for speech generator training, (2) simultaneously learns many-to-many mappings across different attribute domains using a single generator network, (3) is able to generate converted speech signals quickly enough to allow real-time implementations and (4) requires only several minutes of training examples to generate reasonably realistic sounding speech. Subjective evaluation experiments on a non-parallel many-to-many speaker identity conversion task revealed that the proposed method obtained higher sound quality and speaker similarity than a state-of-the-art method based on variational autoencoding GANs.

Index Terms— Voice conversion (VC), non-parallel VC, many-to-many VC, generative adversarial networks (GANs), CycleGAN-VC, StarGAN-VC

1. INTRODUCTION

Voice conversion (VC) is a technique for converting para/non-linguistic information contained in a given utterance while preserving linguistic information. This technique can be applied to various tasks such as speaker-identity modification for text-to-speech (TTS) systems [1], speaking assistance [2, 3], speech enhancement [4–6], and pronunciation conversion [7].

One successful VC framework involves statistical methods based on Gaussian mixture models (GMMs) [8–10]. Recently, a neural network (NN)-based framework based on feed-forward deep NNs [11–13], recurrent NNs [14], and generative adversarial nets (GANs) [7], and an exemplar-based framework based on non-negative matrix factorization (NMF) [15, 16] have also proved successful. Many conventional VC methods including those mentioned above require accurately aligned parallel source and target speech data. However, in many scenarios, it is not always possible to collect parallel utterances. Even if we could collect such data, we typically need to perform time alignment procedures, which becomes relatively difficult when there is a large acoustic gap between the source and target speech. Since many frameworks are weak as regards the misalignment found with parallel data, careful pre-screening and manual correction may be required to make these frameworks work reliably. To bypass these restrictions, this paper is concerned with developing a non-parallel VC method, which requires no parallel utterances, transcriptions, or time alignment procedures.

In general, the quality and conversion effect obtained with non-parallel methods are usually limited compared with methods using parallel data due to the disadvantage related to the training condition. Thus, developing non-parallel methods with as high an audio quality and conversion effect as parallel methods can be very challenging. Recently, some attempts have been made to develop non-parallel methods [17–29]. For example, a method using automatic speech recognition (ASR) was proposed in [24]. The idea is to convert input speech under the restriction that the posterior state probability of the acoustic model of an ASR system is preserved so that the transcription of the converted speech becomes consistent with that of the input speech. Since the performance of this method depends heavily on the quality of the acoustic model of ASR, it can fail to work if ASR does not function reliably. A method using i-vectors [30], known as a feature for speaker verification, was recently proposed in [25]. Conceptually, the idea is to shift the acoustic features of input speech towards target speech in the i-vector space so that the converted speech is likely to be recognized as the target speaker by a speaker recognizer. While this method is also free from parallel data, one limitation is that it is applicable only to speaker identity conversion tasks.

Recently, a framework based on conditional variational autoencoders (CVAEs) [31, 32] was proposed in [22, 29]. As the name implies, variational autoencoders (VAEs) are a probabilistic counterpart of autoencoders (AEs), consisting of encoder and decoder networks. CVAEs [32] are an extended version of VAEs where the encoder and decoder networks can take an auxiliary variable c as an additional input. By using acoustic features as the training examples and the associated attribute labels as c, the networks learn how to convert an attribute of source speech to a target attribute according to the attribute label fed into the decoder. This CVAE-based VC approach is notable in that it is completely free from parallel data and works even with unaligned corpora. However, one well-known problem as regards VAEs is that outputs from the decoder tend to be oversmoothed. For VC applications, this can be problematic since it usually results in poor quality

buzzy-sounding speech.

One powerful framework that can potentially overcome the weakness of VAEs involves GANs [33]. GANs offer a general framework for training a generator network in such a way that it can deceive a real/fake discriminator network. While they have been found to be effective for use with image generation, in recent years they have also been employed with notable success for various speech processing tasks [7, 34–38]. We previously reported a non-parallel VC method using a GAN variant called cycle-consistent GAN (CycleGAN) [26], which was originally proposed as a method for translating images using unpaired training examples [39–41]. This method, which we call CycleGAN-VC, is designed to learn the mapping G of acoustic features from one attribute X to another attribute Y, its inverse mapping F, and a discriminator D, whose role is to distinguish the acoustic features of converted speech from those of real speech, through a training loss combining an adversarial loss and a cycle consistency loss. Although this method was shown to work reasonably well, one major limitation is that it only learns one-to-one mappings. With a lot of VC application scenarios, it is desirable to obtain many-to-many mappings. One naive way of applying CycleGAN to many-to-many VC tasks would be to train different G and F pairs for all pairs of attribute domains. However, this may be ineffective since all attribute domains are common in the sense that they represent speech and so there must be common latent features that can be shared across different domains. In practice, the number of parameters will increase quadratically with the number of attribute domains, making parameter training challenging particularly when there are a limited number of training examples in each domain.

A common limitation of CVAE-VC and CycleGAN-VC is that at test time the attribute of the input speech must be known. As for CVAE-VC, the source attribute label c must be fed into the encoder of the trained CVAE and with CycleGAN-VC, the source attribute domains at training and test times must be the same.

To overcome the shortcomings and limitations of CVAE-VC [22] and CycleGAN-VC [26], this paper proposes a non-parallel many-to-many VC method using a recently proposed novel GAN variant called StarGAN [42], which offers the advantages of CVAE-VC and CycleGAN-VC concurrently. Unlike CycleGAN-VC and as with CVAE-VC, our method, which we call StarGAN-VC, is capable of simultaneously learning many-to-many mappings using a single encoder-decoder type generator network G where the attributes of the generator outputs are controlled by an auxiliary input c. Unlike CVAE-VC and as with CycleGAN-VC, StarGAN-VC uses an adversarial loss for generator training to encourage the generator outputs to become indistinguishable from real speech and ensure that the mappings between each pair of attribute domains will preserve linguistic information. It is also noteworthy that unlike CVAE-VC and CycleGAN-VC, StarGAN-VC does not require any information about the attribute of the input speech at test time.

The VAE-GAN framework [43] is perhaps another natural way of overcoming the weakness of VAEs. A non-parallel VC method based on this framework has already been proposed in [23]. With this approach, an adversarial loss derived using a GAN discriminator is incorporated into the training loss to encourage the decoder outputs of a CVAE to be indistinguishable from real speech features. Although the concept is similar to our StarGAN-VC approach, we will show in Section 4 that our approach outperforms this method in terms of both the audio quality and conversion effect.

Another related technique worth noting is the vector quantized VAE (VQ-VAE) approach [27], which has performed impressively in non-parallel VC tasks. This approach is particularly notable in that it offers a novel way of overcoming the weakness of VAEs by using the WaveNet model [44], a sample-by-sample neural signal generator, to devise both the encoder and decoder of a discrete counterpart of CVAEs. The original WaveNet model is a recursive model that makes it possible to predict the distribution of a sample conditioned on the samples the generator has produced. While a faster version [45] has recently been proposed, it typically requires huge computational cost to generate a stream of samples, which can cause difficulties when implementing real-time systems. The model is also known to require a huge number of training examples to be able to generate natural-sounding speech. By contrast, our method is noteworthy in that it is able to generate signals quickly enough to allow real-time implementation and requires only several minutes of training examples to generate reasonably realistic-sounding speech.

The remainder of this paper is organized as follows. We briefly review the formulation of CycleGAN-VC in Section 2, present the idea of StarGAN-VC in Section 3 and show experimental results in Section 4.

2. CYCLEGAN VOICE CONVERSION

Since the present method is an extension of CycleGAN-VC, which we proposed previously [26], we start by briefly reviewing its formulation.

Let $\mathbf{x} \in \mathbb{R}^{Q \times N}$ and $\mathbf{y} \in \mathbb{R}^{Q \times M}$ be acoustic feature sequences of speech belonging to attribute domains X and Y, respectively, where Q is the feature dimension and N and M are the lengths of the sequences. The aim of CycleGAN-VC is to learn a mapping G that converts the attribute of $\mathbf{x}$ into Y and a mapping F that does the opposite. Now, we introduce discriminators D_X and D_Y, whose roles are to predict whether or not their inputs are the acoustic features of real speech belonging to X and Y, and define

$$\begin{aligned}\mathcal{L}_{\text{adv}}^{D_Y}(D_Y) =& -\mathbb{E}_{\mathbf{y}\sim p_Y(\mathbf{y})}[\log D_Y(\mathbf{y})] \\ & -\mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})}[\log(1 - D_Y(G(\mathbf{x})))], \quad (1)\\ \mathcal{L}_{\text{adv}}^{G}(G) =& \mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})}[\log(1 - D_Y(G(\mathbf{x})))], \quad (2)\\ \mathcal{L}_{\text{adv}}^{D_X}(D_X) =& -\mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})}[\log D_X(\mathbf{x})] \\ & -\mathbb{E}_{\mathbf{y}\sim p_Y(\mathbf{y})}[\log(1 - D_X(F(\mathbf{y})))], \quad (3)\end{aligned}$$

$$\mathcal{L}^{F}_{\text{adv}}(F) = \mathbb{E}_{\mathbf{y}\sim p_Y(\mathbf{y})}[\log(1 - D_X(F(\mathbf{y})))], \quad (4)$$

as the adversarial losses for D_Y, G, D_X and F, respectively. $\mathcal{L}^{D_Y}_{\text{adv}}(D_Y)$ and $\mathcal{L}^{D_X}_{\text{adv}}(D_X)$ measure how indistinguishable $G(\mathbf{x})$ and $F(\mathbf{y})$ are from acoustic features of real speech belonging to Y and X. Since the goal of D_X and D_Y is to correctly distinguish the converted feature sequences obtained via G and F from real speech feature sequences, D_X and D_Y attempt to minimize these losses to avoid being fooled by G and F. Conversely, since one of the goals of G and F is to generate realistic-sounding speech that is indistinguishable from real speech, G and F attempt to maximize these losses or minimize $\mathcal{L}^{G}_{\text{adv}}(G)$ and $\mathcal{L}^{F}_{\text{adv}}(F)$ to fool D_Y and D_X. It can be shown that the output distributions of G and F trained in this way will match the empirical distributions $p_Y(\mathbf{y})$ and $p_X(\mathbf{x})$. Note that since $\mathcal{L}^{G}_{\text{adv}}(G)$ and $\mathcal{L}^{F}_{\text{adv}}(F)$ are minimized when $D_Y(G(\mathbf{x})) \simeq 1$ and $D_X(F(\mathbf{y})) \simeq 1$, we can also use $-\mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})}[\log D_Y(G(\mathbf{x}))]$ and $-\mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})}[\log D_Y(G(\mathbf{x}))]$ as the adversarial losses for G and F.

As mentioned in Section 1, training G and F using only the adversarial losses does not guarantee that G or F will preserve the linguistic information of the input speech since there are infinitely many mappings that will induce the same output distributions. To further regularize these mappings, we introduce a cycle consistency loss

$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G, F) = &\mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})}[\|F(G(\mathbf{x})) - \mathbf{x}\|_1] \\ &+ \mathbb{E}_{\mathbf{y}\sim p_Y(\mathbf{y})}[\|G(F(\mathbf{y})) - \mathbf{y}\|_1], \end{aligned} \quad (5)$$

to encourage $F(G(\mathbf{x})) \simeq \mathbf{x}$ and $G(F(\mathbf{y})) \simeq \mathbf{y}$. With the same motivation, we also consider an identity mapping loss

$$\begin{aligned}\mathcal{L}_{\text{id}}(G, F) = &\mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})}[\|F(\mathbf{x}) - \mathbf{x}\|_1] \\ &+ \mathbb{E}_{\mathbf{y}\sim p_Y(\mathbf{y})}[\|G(\mathbf{y}) - \mathbf{y}\|_1], \end{aligned} \quad (6)$$

to ensure that inputs to G and F are kept unchanged when the inputs already belong to Y and X. The full objectives of CycleGAN-VC to be minimized with respect to G, F, D_X and D_Y are thus given as

$$\begin{aligned}\mathcal{I}_{G,F}(G, F) = &\mathcal{L}^{G}_{\text{adv}}(G) + \mathcal{L}^{F}_{\text{adv}}(F) \\ &+ \lambda_{\text{cyc}}\mathcal{L}_{\text{cyc}}(G, F) + \lambda_{\text{id}}\mathcal{L}_{\text{id}}(G, F), \end{aligned} \quad (7)$$

$$\mathcal{I}_D(D_X, D_Y) = \mathcal{L}^{D_X}_{\text{adv}}(D_X) + \mathcal{L}^{D_Y}_{\text{adv}}(D_Y), \quad (8)$$

where $\lambda_{\text{cyc}} \geq 0$ and $\lambda_{\text{id}} \geq 0$ are regularization parameters, which weigh the importance of the cycle consistency loss and the identity mapping loss relative to the adversarial losses.

3. STARGAN VOICE CONVERSION

While CycleGAN-VC allows the generation of natural-sounding speech when a sufficient number of training examples are available, one limitation is that it only learns one-to-one-mappings. Here, we propose using StarGAN [42] to develop a method that allows non-parallel many-to-many VC. We call the present method StarGAN-VC.

3.1. Training objectives

Let G be a generator that takes an acoustic feature sequence $\mathbf{x} \in \mathbb{R}^{Q\times N}$ with an arbitrary attribute and a target attribute label c as the inputs and generates an acoustic feature sequence $\hat{\mathbf{y}} = G(\mathbf{x}, c)$. We assume that a speech attribute comprises one or more categories, each consisting of multiple classes. We thus represent c as a concatenation of one-hot vectors, each of which is filled with 1 at the index of a class in a certain category and with 0 everywhere else. For example, if we consider speaker identities as the only attribute category, c will be represented as a single one-hot vector, where each element is associated with a different speaker. One of the goals of StarGAN-VC is to make $\hat{\mathbf{y}} = G(\mathbf{x}, c)$ as realistic as real speech features and belong to attribute c. To realize this, we introduce a real/fake discriminator D as with CycleGAN and a domain classifier C, whose role is to predict to which classes an input belongs. D is designed to produce a probability $D(\mathbf{y}, c)$ that an input $\mathbf{y}$ is a real speech feature whereas C is designed to produce class probabilities $p_C(c|\mathbf{y})$ of $\mathbf{y}$.

Adversarial Loss: First, we define

$$\begin{aligned}\mathcal{L}^{D}_{\text{adv}}(D) = &-\mathbb{E}_{c\sim p(c),\mathbf{y}\sim p(\mathbf{y}|c)}[\log D(\mathbf{y}, c)] \\ &-\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}),c\sim p(c)}[\log(1 - D(G(\mathbf{x}, c), c))], \end{aligned} \quad (9)$$

$$\mathcal{L}^{G}_{\text{adv}}(G) = -\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}),c\sim p(c)}[\log D(G(\mathbf{x}, c), c)], \quad (10)$$

as adversarial losses for discriminator D and generator G, respectively, where $\mathbf{y} \sim p(\mathbf{y}|c)$ denotes a training example of an acoustic feature sequence of real speech with attribute c and $\mathbf{x} \sim p(\mathbf{x})$ denotes that with an arbitrary attribute. $\mathcal{L}^{D}_{\text{adv}}(D)$ takes a small value when D correctly classifies $G(\mathbf{x}, c)$ and $\mathbf{y}$ as fake and real speech features whereas $\mathcal{L}^{G}_{\text{adv}}(G)$ takes a small value when G successfully deceives D so that $G(\mathbf{x}, c)$ is misclassified as real speech features by D. Thus, we would like to minimize $\mathcal{L}^{D}_{\text{adv}}(D)$ with respect to D and minimize $\mathcal{L}^{G}_{\text{adv}}(G)$ with respect to G.

Domain Classification Loss: Next, we define

$$\mathcal{L}^{C}_{\text{cls}}(C) = -\mathbb{E}_{c\sim p(c),\mathbf{y}\sim p(\mathbf{y}|c)}[\log p_C(c|\mathbf{y})], \quad (11)$$

$$\mathcal{L}^{G}_{\text{cls}}(G) = -\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}),c\sim p(c)}[\log p_C(c|G(\mathbf{x}, c))], \quad (12)$$

as domain classification losses for classifier C and generator G. $\mathcal{L}^{C}_{\text{cls}}(C)$ and $\mathcal{L}^{G}_{\text{cls}}(G)$ take small values when C correctly classifies $\mathbf{y} \sim p(\mathbf{y}|c)$ and $G(\mathbf{x}, c)$ as belonging to attribute c. Thus, we would like to minimize $\mathcal{L}^{C}_{\text{cls}}(C)$ with respect to C and $\mathcal{L}^{G}_{\text{cls}}(G)$ with respect to G.

Cycle Consistency Loss: Training G, D and C using only the losses presented above does not guarantee that G will preserve the linguistic information of input speech. To encourage $G(\mathbf{x}, c)$ to be a bijection, we introduce a cycle consistency loss to be minimized

$$\begin{aligned}&\mathcal{L}_{\text{cyc}}(G) \\ &= \mathbb{E}_{c'\sim p(c),\mathbf{x}\sim p(\mathbf{x}|c'),c\sim p(c)}[\|G(G(\mathbf{x}, c), c') - \mathbf{x}\|_\rho], \end{aligned} \quad (13)$$

where $\mathbf{x} \sim p(\mathbf{x}|c')$ denotes a training example of an acoustic feature sequence of real speech with attribute c' and ρ is a

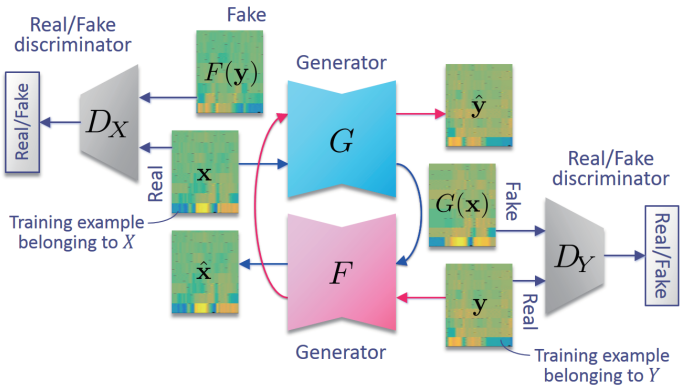

Fig. 1. Concept of CycleGAN training.

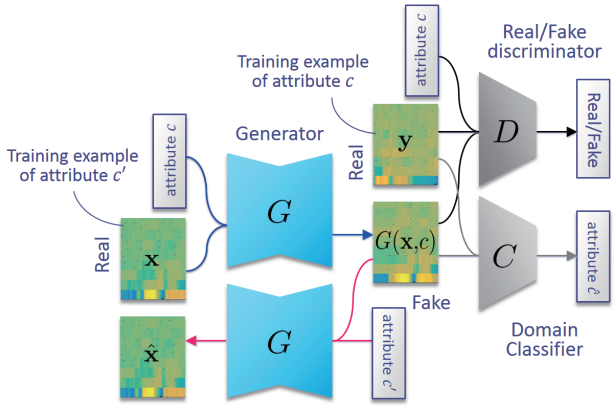

Fig. 2. Concept of StarGAN training.

positive constant. We also consider an identity mapping loss

$$\mathcal{L}_{\mathrm{id}}(G) = \mathbb{E}_{c' \sim p(c), \mathbf{x} \sim p(\mathbf{x}|c')}[\|G(\mathbf{x}, c') - \mathbf{x}\|_\rho], \tag{14}$$

to ensure that an input into G will remain unchanged when the input already belongs to the target attribute c'.

To summarize, the full objectives of StarGAN-VC to be minimized with respect to G, D and C are given as

$$\begin{aligned}\mathcal{I}_G(G) =& \mathcal{L}^G_{\mathrm{adv}}(G) + \lambda_{\mathrm{cls}}\mathcal{L}^G_{\mathrm{cls}}(G) \\ &+ \lambda_{\mathrm{cyc}}\mathcal{L}_{\mathrm{cyc}}(G) + \lambda_{\mathrm{id}}\mathcal{L}_{\mathrm{id}}(G), \qquad (15)\\ \mathcal{I}_D(D) =& \mathcal{L}^D_{\mathrm{adv}}(D), \qquad (16)\\ \mathcal{I}_C(C) =& \mathcal{L}^C_{\mathrm{cls}}(C), \qquad (17)\end{aligned}$$

respectively, where $\lambda_{\mathrm{cls}} \geq 0$, $\lambda_{\mathrm{cyc}} \geq 0$ and $\lambda_{\mathrm{id}} \geq 0$ are regularization parameters, which weigh the importance of the domain classification loss, the cycle consistency loss and the identity mapping loss relative to the adversarial losses.

3.2. Conversion process

As an acoustic feature vector, we use mel-cepstral coefficients computed from a spectral envelope obtained using WORLD [46]. After training G, we can convert the acoustic feature sequence $\mathbf{x}$ of an input utterance with

$$\hat{\mathbf{y}} = G(\mathbf{x}, c), \tag{18}$$

where c denotes the target attribute label. A naïve way of obtaining a time-domain signal is simply to use $\hat{\mathbf{y}}$ to reconstruct a signal with a vocoder. Instead of directly using $\hat{\mathbf{y}}$, we can also use the reconstructed feature sequence

$$\hat{\mathbf{y}}' = G(\mathbf{x}, c'), \tag{19}$$

to obtain a time-domain signal if the attribute c' of the input speech is known. By using $\hat{\mathbf{y}}$ and $\hat{\mathbf{y}}'$, we can obtain a sequence of spectral gain functions. Once we obtain the spectral gain functions, we can reconstruct a time-domain signal by multiplying the spectral envelope of input speech by the spectral gain function frame-by-frame and resynthesizing the signal using a vocoder.

3.3. Network architectures

One of the key features of our approach including [7, 26] is that we consider a generator that takes an acoustic feature sequence instead of a single-frame acoustic feature as an input and outputs an acoustic feature sequence of the same length. This allows us to obtain conversion rules that capture time dependencies. While RNN-based architectures are a natural choice for modeling time series data, we use a convolutional neural network (CNN)-based architecture to design G as detailed below. The generator G consists of encoder and decoder networks where only the decoder network takes an auxiliary input c. We also design D and C to take acoustic feature sequences as inputs and generate sequences of probabilities.

Generator: Here, we treat an acoustic feature sequence $\mathbf{x}$ as an image of size $Q \times N$ with 1 channel and use 2D CNNs to construct G, as they are suitable for parallel computations. Specifically, we use a gated CNN [47], which was originally introduced to model word sequences for language modeling and was shown to outperform long short-term memory (LSTM) language models trained in a similar setting. We previously applied gated CNN architectures for voice conversion [7, 26] and audio source separation [48], and their effectiveness has already been confirmed. In the encoder part, the output of the l-th hidden layer is described as a linear projection modulated by an output gate

$$\mathbf{h}_l = (\mathbf{W}_l * \mathbf{h}_{l-1} + \mathbf{b}_l) \odot \sigma(\mathbf{V}_l * \mathbf{h}_{l-1} + \mathbf{d}_l), \tag{20}$$

where $\mathbf{W}_l \in \mathbb{R}^{D_l \times D_{l-1} \times Q_l \times N_l}$, $\mathbf{b}_l \in \mathbb{R}^{D_l}$, $\mathbf{V}_l \in \mathbb{R}^{D_l \times D_{l-1} \times Q_l \times N_l}$ and $\mathbf{d}_l \in \mathbb{R}^{D_l}$ are the generator network parameters to be trained, and σ denotes the elementwise sigmoid function. Similar to LSTMs, the output gate multiplies each element of $\mathbf{W}_l * \mathbf{h}_{l-1} + \mathbf{b}_l$ and control what information should be propagated through the hierarchy of layers. This gating mechanism is called Gated Linear Units (GLU). In the decoder part, the output of the l-th hidden layer is given by

$$\begin{aligned}\mathbf{h}'_{l-1} &= [\mathbf{h}_{l-1}; \mathbf{c}_{l-1}], \qquad (21)\\ \mathbf{h}_l &= (\mathbf{W}_l * \mathbf{h}'_{l-1} + \mathbf{b}_l) \odot \sigma(\mathbf{V}_l * \mathbf{h}'_{l-1} + \mathbf{d}_l), \qquad (22)\end{aligned}$$

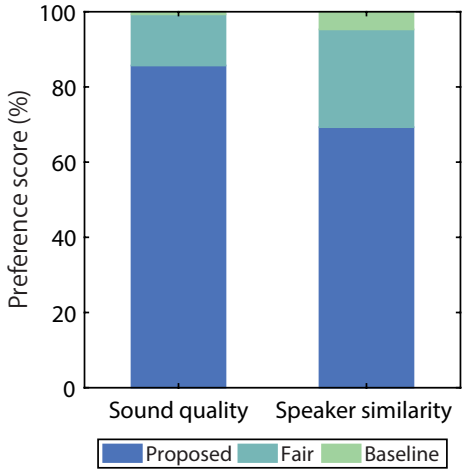

Fig. 3. Results of the AB test for sound quality and the ABX test for speaker similarity.

where $[\mathbf{h}_l; \mathbf{c}_l]$ means the concatenation of $\mathbf{h}_l$ and $\mathbf{c}_l$ along the channel dimension, and $\mathbf{c}_l$ is a 3D array consisting of a Q_l-by-N_l tiling of copies of c in the feature and time dimensions. The input into the 1st layer of G is $\mathbf{h}_0 = \mathbf{x}$ and the output of the final layer is given as a regular linear projection

$$\mathbf{h}'_{L-1} = [\mathbf{h}_{L-1}; \mathbf{c}_{L-1}], \tag{23}$$

$$\hat{\mathbf{y}} = \mathbf{W}_L * \mathbf{h}'_{L-1} + \mathbf{b}_L. \tag{24}$$

It should be noted that the entire architecture is fully convolutional with no fully-connected layers, which allows us to take an entire sequence with an arbitrary length as an input and convert the entire sequence.

Real/Fake Discriminator: We leverage the idea of PatchGANs [49] to devise a real/fake discriminator D, which classifies whether local segments of an input feature sequence are real or fake. More specifically, we devise D using a gated CNN, which takes an acoustic feature sequence $\mathbf{y}$ and an attribute label c as inputs and produces a sequence of probabilities that measures how likely each segment of $\mathbf{y}$ is to be real speech features of attribute c. The output of the l-th layer of D is given in the same way as (21) and (22) and the final output $D(\mathbf{y}, c)$ is given by the product of all these probabilities. See Section 4 for more details.

Domain Classifier: We also devise a domain classifier C using a gated CNN, which takes an acoustic feature sequence $\mathbf{y}$ and produces a sequence of class probability distributions that measures how likely each segment of $\mathbf{y}$ is to belong to attribute c. The output of the l-th layer of C is given in the same way as (20) and the final output $p_C(c|\mathbf{y})$ is given by the product of all these distributions. See Section 4 for more details.

4. SUBJECTIVE EVALUATION

To confirm the performance of the proposed method, we conducted subjective evaluation experiments on a non-parallel many-to-many speaker identity conversion task. We used the Voice Conversion Challenge (VCC) 2018 dataset [50], which consists of recordings of six female and six male US English speakers. We used a subset of speakers for training and evaluation. Specifically, we selected two female speakers, 'VCC2SF1' and 'VCC2SF2', and two male speakers, 'VCC2SM1' and 'VCC2SM2'. Thus, c is represented as a four-dimensional one-hot vector and there were twelve different combinations of source and target speakers in total. The audio files for each speaker were manually segmented into 116 short sentences (about 7 minutes) where 81 and 35 sentences (about 5 and 2 minutes) were provided as training and evaluation sets, respectively. All the speech signals were sampled at 22050 Hz. For each utterance, a spectral envelope, a logarithmic fundamental frequency (log F_0), and aperiodicities (APs) were extracted every 5 ms using the WORLD analyzer [46]. 36 mel-cepstral coefficients (MCCs) were then extracted from each spectral envelope. The F_0 contours were converted using the logarithm Gaussian normalized transformation described in [51]. The aperiodicities were used directly without modification. The network configuration is shown in detail in Fig. 4. The signals of the converted speech were obtained using the method described in 3.2.

We chose the VAEGAN-based approach [23] as a comparison for our experiments. Although we would have liked to exactly replicate the implementation of this method, we made our own design choices owing to missing details of the network configuration and hyperparameters. We conducted an AB test to compare the sound quality of the converted speech samples and an ABX test to compare the similarity to target speaker of the converted speech samples, where "A" and "B" were converted speech samples obtained with the proposed and baseline methods and "X" was a real speech sample of a target speaker. With these listening tests, "A" and "B" were presented in random orders to eliminate bias in the order of stimuli. Eight listeners participated in our listening tests. For the AB test for sound quality, each listener was presented {"A","B"} $\times$ 20 utterances, and for the ABX test for speaker similarity, each listener was presented {"A","B","X"} $\times$ 24 utterances. Each listener was then asked to select "A", "B" or "fair" for each utterance. The results are shown in Fig. 3. As the results show, the proposed method significantly outperformed the baseline method in terms of both sound quality and speaker similarity. Fig. 5 shows an example of the MCC sequences of source, reconstructed, and converted speech. Audio samples are provided at `http://www.kecl.ntt.co.jp/people/kameoka.hirokazu/Demos/stargan-vc/`.

5. CONCLUSION

This paper proposed a method that allows non-parallel many-to-many VC by using a novel GAN variant called StarGAN. Our method, which we call StarGAN-VC, is noteworthy in that it (1) requires no parallel utterances, transcriptions, or time alignment procedures for speech generator training, (2)

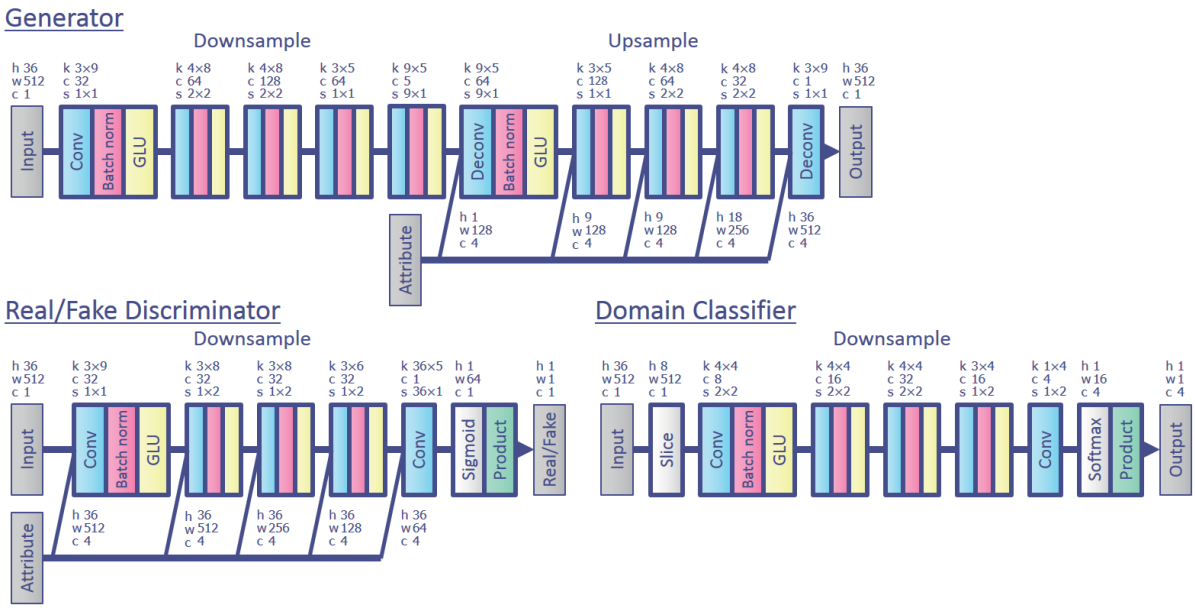

Fig. 4. Network architectures of generator G, real/fake discriminator D and domain classifier C. Here, the inputs and outputs of G, D and C are interpreted as images, where "h", "w" and "c" denote the height, width and channel number, respectively. "Conv", "Batch norm", "GLU", "Deconv" "Sigmoid", "Softmax" and "Product" denote convolution, batch normalization, gated linear unit, transposed convolution, sigmoid, softmax, and product pooling layers, respectively. "k", "c" and "s" denote the kernel size, output channel number and stride size of a convolution layer, respectively. Note that all the networks are fully convolutional with no fully connected layers, thus allowing inputs to have arbitrary sizes.

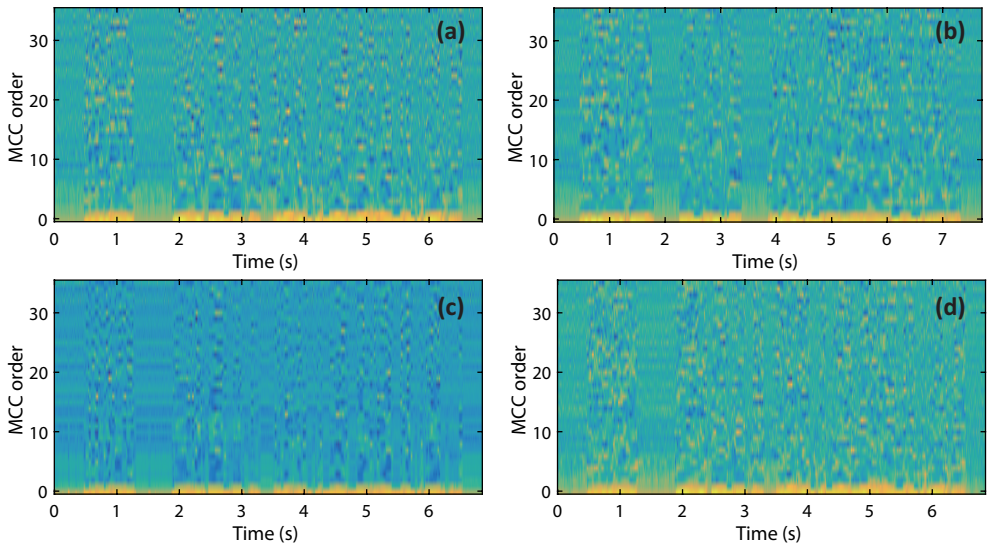

Fig. 5. Examples of acoustic feature sequences of (a) source speech, (c) converted speech obtained with the baseline method and (d) converted speech obtained with the proposed method, along with an acoustic feature sequence of (b) the target speaker uttering the same sentence.

simultaneously learns many-to-many mappings across different voice attribute domains using a single generator network, (3) is able to generate signals of converted speech quickly enough to allow real-time implementations and (4) requires only several minutes of training examples to generate reasonably realistic sounding speech. Subjective evaluation experiments on a non-parallel many-to-many speaker identity conversion task revealed that the proposed method obtained higher sound quality and speaker similarity than a baseline method based on a VAE-GAN approach. A preprint version of this paper is provided at [52].

6. REFERENCES

[1] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *Proc. ICASSP*, 1998, pp. 285–288.

[2] A. B. Kain, J.-P. Hosom, X. Niu, J. P. van Santen, M. Fried-Oken, and J. Staehely, "Improving the intelligibility of dysarthric speech," *Speech Commun.*, vol. 49, no. 9, pp. 743–759, 2007.

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

[4] Z. Inanoglu and S. Young, "Data-driven emotion conversion in spoken English," *Speech Commun.*, vol. 51, no. 3, pp. 268–283, 2009.

[5] O. Türk and M. Schröder, "Evaluation of expressive speech synthesis with voice conversion and copy resynthesis techniques," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 5, pp. 965–973, 2010.

[6] T. Toda, M. Nakagiri, and K. Shikano, "Statistical voice conversion techniques for body-conducted unvoiced speech enhancement," *IEEE Trans. Audio Speech Lang. Process.*, vol. 20, no. 9, pp. 2505–2517, 2012.

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

[8] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. Speech Audio Process.*, vol. 6, no. 2, pp. 131–142, 1998.

[9] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximumlikelihood estimation of spectral parameter trajectory," *IEEE Trans. Audio Speech Lang. Process.*, vol. 15, no. 8, pp. 2222–2235, 2007.

[10] E. Helander, T. Virtanen, J. Nurminen, and M. Gabbouj, "Voice conversion using partial least squares regression," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 5, pp. 912–921, 2010.

[11] S. Desai, A. W. Black, B. Yegnanarayana, and K. Prahallad, "Spectral mapping using artificial neural networks for voice conversion," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 5, pp. 954–964, 2010.

[12] S. H. Mohammadi and A. Kain, "Voice conversion using deep neural networks with speaker-independent pre-training," in *Proc. SLT*, 2014, pp. 19–23.

[13] Y. Saito, S. Takamichi, and H. Saruwatari, "Voice conversion using input-to-output highway networks," *IEICE Trans Inf. Syst.*, vol. E100-D, no. 8, pp. 1925–1928, 2017.

[14] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," in *Proc. ICASSP*, 2015, pp. 4869–4873.

[15] R. Takashima, T. Takiguchi, and Y. Ariki, "Exemplar-based voice conversion using sparse representation in noisy environments," *IEICE Trans. Inf. Syst.*, vol. E96-A, no. 10, pp. 1946–1953, 2013.

[16] Z. Wu, T. Virtanen, E. S. Chng, and H. Li, "Exemplar-based sparse representation with residual compensation for voice conversion," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 22, no. 10, pp. 1506–1521, 2014.

[17] L.-H. Chen, Z.-H. Ling, L.-J. Liu, and L.-R. Dai, "Voice conversion using deep neural networks with layer-wise generative training," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 22, no. 12, pp. 1859–1872, 2014.

[18] T. Nakashika, T. Takiguchi, and Y. Ariki, "Voice conversion based on speaker-dependent restricted Boltzmann machines," *IEICE Trans. Inf. Syst.*, vol. 97, no. 6, pp. 1403–1410, 2014.

[19] T. Nakashika, T. Takiguchi, and Y. Ariki, "High-order sequence modeling using speaker-dependent recurrent temporal restricted Boltzmann machines for voice conversion," in *Proc. Interspeech*, 2014, pp. 2278–2282.

[20] T. Nakashika, T. Takiguchi, and Y. Ariki, "Parallel-data-free, many-to-many voice conversion using an adaptive restricted Boltzmann machine," in *Proc. MLSP*, 2015.

[21] M. Blaauw and J. Bonada, "Modeling and transforming speech using variational autoencoders," in *Proc. Interspeech*, 2016, pp. 1770–1774.

[22] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," in *Proc. APSIPA*, 2016, pp. 1–6.

[23] 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," in *Proc. Interspeech*, 2017, pp. 3364–3368.

[24] F.-L. Xie, F. K. Soong, and H. Li, "A KL divergence and DNN-based approach to voice conversion without parallel training sentences," in *Proc. Interspeech*, 2016, pp. 287–291.

[25] T. Kinnunen, L. Juvela, P. Alku, and J. Yamagishi, "Non-parallel voice conversion using i-vector PLDA: Towards unifying speaker verification and transformation," in *Proc. ICASSP*, 2017, pp. 5535–5539.

[26] T. Kaneko and H. Kameoka, "Parallel-data-free voice conversion using cycle-consistent adversarial networks," *arXiv:1711.11293 [stat.ML]*, Nov. 2017.

[27] A. van den Oord and O. Vinyals, "Neural discrete representation learning," in *Adv. NIPS*, 2017, pp. 6309–6318.

[28] T. Hashimoto, H. Uchida, D. Saito, and N. Minematsu, "Parallel-data-free many-to-many voice conversion based on dnn integrated with eigenspace using a non-parallel speech corpus," in *Proc. Interspeech*, 2017.

[29] Y. Saito, Y. Ijima, K. Nishida, and S. Takamichi, "Non-parallel voice conversion using variational autoencoders conditioned by phonetic posteriorgrams and d-vectors," in *Proc. ICASSP*, 2018, pp. 5274–5278.

[30] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio Speech Lang. Process.*, vol. 19, no. 4, pp. 788–798, 2011.

[31] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. ICLR*, 2014.

[32] D. P. Kingma and D. J. Rezendey, S. Mohamedy, and M. Welling, "Semi-supervised learning with deep generative models," in *Adv. NIPS*, 2014, pp. 3581–3589.

[33] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Adv. NIPS*, 2014, pp. 2672–2680.

[34] T. Kaneko, H. Kameoka, N. Hojo, Y. Ijima, K. Hiramatsu, and K. Kashino, "Generative adversarial network-based postfilter for statistical parametric speech synthesis," in *Proc. ICASSP*, 2017, pp. 4910–4914.

[35] Y. Saito, S. Takamichi, and H. Saruwatari, "Statistical parametric speech synthesis incorporating generative adversarial networks," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 26, no. 1, pp. 84–96, Jan. 2018.

[36] S. Pascual, A. Bonafonte, and J. Serrá, "SEGAN: Speech enhancement generative adversarial network," *arXiv:1703.09452 [cs.LG]*, Mar. 2017.

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

[38] K. Oyamada, H. Kameoka, T. Kaneko, K. Tanaka, N. Hojo, and H. Ando, "Generative adversarial network-based approach to signal reconstruction from magnitude spectrograms," *arXiv:1804.02181 [eess.SP]*, Apr. 2018.

[39] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. ICCV*, 2017, pp. 2223–2232.

[40] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *Proc. ICML*, 2017, pp. 1857–1865.

[41] Z. Yi, H. Zhang, P. Tan, and M. Gong, "DualGAN: Unsupervised dual learning for image-to-image translation," in *Proc. ICCV*, 2017, pp. 2849–2857.

[42] 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," *arXiv:1711.09020 [cs.CV]*, Nov. 2017.

[43] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," *arXiv:1512.09300 [cs.LG]*, Dec. 2015.

[44] A. van den 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:1609.03499 [cs.SD]*, Sept. 2016.

[45] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. C. Cobo, F. Stimberg, N. Casagrande, D. G., S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis, "Parallel WaveNet: Fast high-fidelity speech synthesis," *arXiv:1711.10433 [cs.LG]*, Nov. 2017.

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

[47] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. ICML*, 2017, pp. 933–941.

[48] L. Li and H. Kameoka, "Deep clustering with gated convolutional networks," in *Proc. ICASSP*, 2018, pp. 16–20.

[49] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. CVPR*, 2017.

[50] 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:1804.04262 [eess.AS]*, Apr. 2018.

[51] K. Liu, J. Zhang, and Y. Yan, "High quality voice conversion through phoneme-based linear mapping functions with STRAIGHT for mandarin," in *Proc. FSKD*, 2007, pp. 410–414.

[52] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "StarGAN-VC: Non-parallel many-to-many voice conversion with star generative adversarial networks," *arXiv:1806.02169 [cs.SD]*, June 2018.