A Pitch-Controlled End-to-End Voice Conversion System for Brazilian Portuguese

Victor P. da Costa, Sergio L. Netto, Luiz W. P. Biscainho, and Ranniery Maia

***Abstract*—Speech conversion is a technique that modifies the identity of the voice in a speech signal without changing the spoken content. Accurate pitch conversion is a requirement the best speech conversion systems must address, as this characteristic is essential to the correct identification of the target speaker. This work proposes a pitch-controlled end-to-end voice conversion model that combines state-of-the-art ideas from both speaking and singing voice conversion with a novel cost function to ensure artifact-free pitch tracking. The model is trained in Brazilian Portuguese, overcoming the lack of high-quality data by improving a large but flawed dataset with filtering operation. Our model mostly outperforms other popular open source models in both listening tests and objective measurements. In particular, on a 5-point MOS, we obtained the highest speaker similarity score (4.05), and a naturalness score of 3.48, second only to a system whose similarity score was 2.62.**

***Index Terms*—voice conversion, generative adversarial networks, Brazilian Portuguese**

I. Introduction

Voice conversion is the process of modifying a speech signal such that the perceived identity of the speaker is changed while preserving the textual content. It is a tool with many applications, both direct (e.g. as a voice acting tool) and indirect (e.g. as a way to generate diverse training data for other applications).

In traditional voice conversion, speech is transformed into an intermediate representation with useful properties, a model performs conversion in the representation space and the converted representation is transformed back into an audio signal [1] [2]. Advances in deep learning techniques have revolutionized all stages of the conversion process, producing more complex conversion models, newer intermediate representations and higher quality vocoders [3].

There are many reasons to use an intermediate representation, such as allowing the use of high-quality vocoders or isolating specific aspects such as timbre and pitch — essential for voice identification —, making conversion simpler. On the other hand, compounding errors may degrade signal quality: using a vocoder developed for natural representations in the reconstruction of converted representations may result in worse performance, as a small training error for the conversion network may be amplified into very noticeable artifacts by the synthesis process.

Our system combines a generative adversarial network-based approach for both conversion and synthesis with a pre-trained encoder to create an end-to-end network that outputs audio directly. Since there is no distinction between conversion and synthesis, gradients can be propagated throughout the framework, avoiding compounding-error artifacts and thus improving the overall quality.

One common issue in many voice conversion systems is poor pitch conversion. Pitch characteristics, specially its mean, are very important for voice identification, but most voice conversion systems only transform the pitch implicitly. Explicit pitch control is more common in singing voice conversion, and is often used to preserve the original pitch contour [4] [5] in order to avoid melody distortion. However, enforcing the original pitch may lead to poor perceived similarity between converted and target voices or poor signal quality, especially when source and target singers have different vocal registers.

In this work, we incorporate strategies from singing voice conversion to explicitly convert the pitch (operating more flexibly as we are not constrained by an original melody), greatly improving the identification of the resulting converted speech as originating from the target speaker.

We trained our model in Brazilian Portuguese, which is a language still underserved in voice conversion. We overcame the lack of large, high quality, multi-speaker datasets in Portuguese by filtering an existing dataset plagued by problems such as noise and non-standardized recording environments. These problems can even make signals from the same speaker sound like they are from different speakers.

The main contributions of this work are the following: an end-to-end voice conversion system with pitch control, adapting various techniques from the literature; a novel pitch matching loss exploiting a neural pitch extraction network; and an improved version of a large scale Brazilian Portuguese dataset.

This article is organized as follows: Section II reviews related voice-conversion works; Section III provides a description of the proposed system, both in terms of network architectures and loss functions used in its development; Section IV details aspects of system training, including the dataset, and the experiments comparing our model to other models in the literature. Finally, Section V provides a conclusion.

Victor P. da Costa (email: victor.costa@smt.ufrj.br, ORCID: 0009-0006-7024-2840), Sergio L. Netto (email: sergioln@smt.ufrj.br, ORCID: 0000-0001-7389-1463), and Luiz W. P. Biscainho (email: wagner@smt.ufrj.br, ORCID: 0000-0003-2959-6963) are with Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil.

Ranniery Maia (email: rmaia@dimap.ufrn.br, ORCID: 0000-0002-5512-3038) is with Federal University of Rio Grande do Norte (UFRN), Natal, Brazil.

This work was funded by the National Council for Scientific and Technological Development (CNPq), under Grants 311146/2021-0, and 167970/2017-8

Submission: 2024-05-09, First decision: 2024-07-02, Acceptance: 2024-08-07, Publication: 2024-08-10.

Digital Object Identifier: 10.14209/jcis.2024.13

II. Related Work

Many methods have been proposed over the years for voice conversion. Traditional methods used vocoders such as STRAIGHT [6] or WORLD [7] for feature extraction and synthesis, and conversion resorted to statistical models such as vector quantization [1], Gaussian mixture models [2] and hidden Markov models [8].

Advances in deep neural networks resulted in a rapid increase in new techniques. Some works adapted domain-agnostic generative techniques, such as generative adversarial networks (GANs) [9] [10] and variational autoencoders (VAE) [11], to perform voice conversion, while others exploited advances in other speech processing tasks, such as voice synthesis [12] or automatic speech recognition [13].

Some of the biggest advances in the field of voice conversion came from newer vocoders based in deep neural networks that could achieve higher quality than traditional ones. The first neural vocoder to achieve great results was WaveNet [14], which uses a deep convolutional network to implement an auto-regressive model of speech generation. It is able to produce high-quality samples, but the generation speed is fairly slow due to the complex network and its auto-regressive nature. Later, other works such as WaveRNN [15], Parallel WaveGAN [16], WaveGlow [17] and MelGAN [18] improved the generation speed without sacrificing quality by reducing network size and/or making the model non auto-regressive.

Newer techniques also changed the types of representation used. High-quality pre-trained vocoders made mel-spectrograms the most common representation for voice conversion. Features obtained from networks trained for automatic speech recognition (ASR), such as phonetic posteriorgrams [19] or intermediary features [4] [20], are also popular representations; since they are closer to text, and as such naturally carry less speaker information, they make conversion easier. Self supervised learning (SSL) models are a newer source for representations. These networks are trained on large scale datasets, usually with the objective of producing a disentangled representation useful for many tasks [21] [22] or focusing on specific tasks, such as content encoding [23]. All these alternative representations are not mutually exclusive: many works use, for example, SSL representations as input and mel-spectrograms as output [24].

While most works use a separate vocoder, others follow an end-to-end approach. Nachimani et al. [25], for instance, uses a WaveNet vocoder directly as the decoder of an autoencoder. NVCNet [26], FreeVC [27] and two of the best performing systems in the 2023 edition of the Singing Voice Conversion Challenge [24] use the HiFi-GAN [28] vocoder in similar ways.

In the context of pitch conversion, vocoders such as STRAIGHT or WORLD separate the audio signal into periodic or aperiodic envelopes and pitch. As such, models that use these representations usually treat pitch conversion separately [2]. As novel deep learning methods were introduced, it has become more common to use mel-spectrograms or other learned representations as the conversion domain, which has many advantages, but makes the pitch conversion implicit. Recent models, such as the SF-HiFi-GAN vocoder [29] have begun to reintroduce explicit pitch control but are not widely used yet.

Explicit pitch conditioning is an important aspect of singing voice conversion. One of the first deep learning models with explicit pitch conditioning was PitchNet [30], which uses additional costs to make a pitch-independent embedding, which is concatenated with a pitch-embedding stage and subsequently decoded. Nachimani et al. [25] use an artificial sinusoidal signal as input for their network together with linguistic and loudness representations, while Liu et al. [4] use downsampled versions of a similar signal to condition the network at different scales.

Some methods attempt to improve pitch conversion without explicit pitch conditioning. Speechsplit2 [31] and FreeVC [27] both use data augmentation to assist their internal representations in becoming speaker independent. The former uses traditional techniques to generate signals with modified pitch and spectral envelope, while the latter employs frequency shifted versions of the signals by stretching and compressing its mel-spectrogram and synthesizing the result. In both works, the conversion network recovers the original speech from the corrupted version, thus learning to ignore the source's pitch and spectral content.

Our model combines a WavLM [21] encoder to extract features, a modified version of the generator from HiFi-GAN [28] as a decoder, and a pitch encoder inspired by FastSCV [4]. We improve overall quality and pitch tracking by incorporating a novel pitch specific loss using a neural pitch estimator [32], as well as other losses adapted from the literature [23] [33].

III. Description of the Proposed Model

The proposed model is a generative adversarial network, and as such is composed of two sub-networks: a generator $G(\mathbf{s}, \mathbf{c}, \mathbf{e})$ that converts the audio signal $\mathbf{s}$ from its original speaker into the target speaker represented by $\mathbf{c}$, conditioned on a pitch excitation represented by $\mathbf{e}$; and a discriminator $D(\mathbf{s}, \mathbf{c})$ that evaluates the signal. Fig. 1 shows an overview of the system.

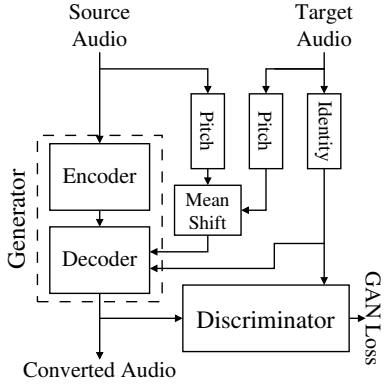


Fig. 1. Overview of the system, showing inputs & outputs and the general structure around a simplified view of the generator.

The system performs many-to-many conversions, meaning that it can convert from and to any speaker in the training set, with the same generator for all conversion pairs, receiving the target speaker as an input to perform the necessary conditioning. The discriminator receives as input both the audio and the target speaker's identity and decides if it is a natural signal from the target speaker or one produced by the generator. These networks are trained adversarially: the discriminator tries to classify real signals as real and produced by the correct speaker, rejecting generated signals that either sound too unnatural or have the wrong identity; in turn, the generator tries to deceive the discriminator. Sections III-A and III-B describe the structure of the generator and discriminator, respectively, while Section III-C describes the cost functions used in their training process.

A. Generator

The generator is an autoencoder where the encoder embeds the audio signal into a latent space and the decoder transforms the embedding, together with target speaker and target pitch conditioning, back to an audio signal.

1) Decoder: The decoder structure alternates upsampling layers and stacks of residual block, following the same approach employed in the HiFi-GAN generator [28]. The upsample layer is a single transposed convolution and roughly expands the previous level of computation, increasing the time resolution and reducing the number of channels, whereas the residual blocks refines the signal while keeping both the time resolution and number of channels constant. Fig. 2 shows an overview of the decoder.

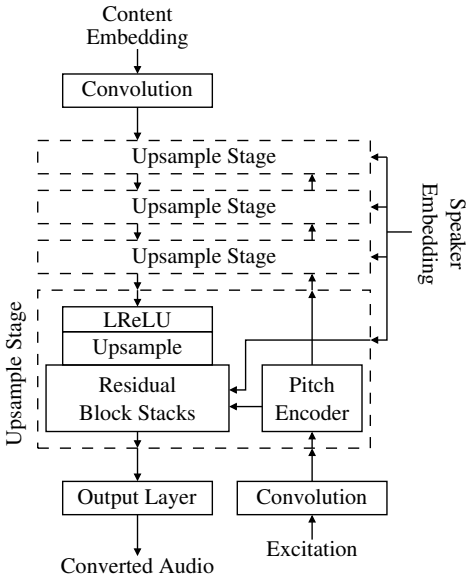


Fig. 2. Overview of the decoder, showing the internal structure of each stage and how the content, identity, and pitch inputs are processed.

Each residual block has a residual connection that adds its input to its output, and is composed of a dilated convolution and a convolution with kernel of length one, both preceded by leaky ReLU activations. Blocks are organised into parallel stacks, with the dilation of the convolution starting at one (no dilation) and increasing throughout the stack, improving the perceptive field of the stack without significantly increasing its number of parameters. Each stack uses a different kernel size to help the network learn structures of different lengths. Dilation and kernel size configurations follow [28].

Information on the target speaker and the desired pitch is fed into the generator between the convolutions of each residual block through an affine transformation similar to Perez et al. [34]. In that manner, channel-wise concatenated pitch and speaker embeddings are transformed into per-channel and per-time step scaling and translation via a pair of convolutional layers with a leaky ReLU activation in between, which are respectively multiplied and added to the feature map. Mathematically, the transformation is implemented as

$$\hat{\mathbf{h}}^{(i)} = \mathbf{h}^{(i)} \odot \boldsymbol{\sigma}_c^{(i)} + \boldsymbol{\mu}_c^{(i)}, \tag{1}$$

where, $\mathbf{h}^{(i)}$ is the feature map, $\boldsymbol{\sigma}_c^{(i)}$ and $\boldsymbol{\mu}_c^{(i)}$ are speaker- and pitch-dependent scale and translation, respectively, all and $\odot$ is the point-wise product. This transformation allows the conditioning vectors to modulate the synthesis more effectively than concatenating the vector to the maps [9]. The transformation is not accompanied by instance normalization as done in Chou et al. [35] or Kaneko et al. [9], to avoid loss of information and artifact production during audio generation [18].

Speaker embeddings are obtained, according to the speaker ID, from a learnable codebook and are the same for all time steps. When concatenating, they are simply repeated to match the pitch embedding time dimension. Details on the pitch vectors are explained in the next item.

A final convolutional layer reduces the number of channels to one and a hyperbolic tangent activation produces the normalized audio signal as an output between -1 and 1.

2) Pitch encoder: While the generator with only the speaker conditioning is able to produce signals with a spectral envelope close to that of the target speaker, it struggles to produce the same mean pitch as the target, thus greatly impairing the perceived identity of the converted signal. In particular, the model has a tendency to sustain, to halve or to double the original pitch. Therefore, in order to improve the pitch control, we employ techniques inspired by singing voice conversion and use the desired pitch as an input to the network [4].

We first extract the fundamental frequencies from the source signal and a randomly selected signal from the target speaker. The desired converted pitch is then determined as

$$f_{\text{conv}} = f_{\text{src}} \frac{\overline{f_{\text{tgt}}}}{\overline{f_{\text{src}}}}, \tag{2}$$

where $\overline{f_{\text{src}}}$ and $\overline{f_{\text{tgt}}}$ are the average pitches of the source and target signals, respectively. Instead of using the pitch directly as the input of the network, we follow Liu et al. [4] and use the desired pitch to obtain an excitation signal, which is then fed to the network to produce better results [36]. To obtain the excitation signal, the pitch is interpolated to the same dimensions as the source signal, thus providing the instantaneous frequency, which in turn is integrated into the instantaneous phase. The excitation signal e_t is then the sine

of the instantaneous phase for voiced frames and random noise for the unvoiced ones:

$$e_t = \begin{cases} \alpha \sin\left(\sum_{k=0}^{t} 2\pi \frac{f_k}{f_s} t + \theta\right) + n_t, & \text{if voiced} \\ \frac{\alpha}{3\sigma} n_t, & \text{if unvoiced,} \end{cases} \quad (3)$$

where $n_t \sim \mathcal{N}(0, \sigma^2)$, θ is a random initial phase and f_s is the sampling frequency. Gain α controls the energy of the excitation signal.

The excitation signal travels backwards through the decoder, as shown in Fig. 2, starting with the same time dimensions as the output and being downsampled to produce pitch embeddings at each internal scale of the decoder. The downsampling is performed by a residual block inspired by Liu et al. [4] but slightly less complex, which we found did not affect the quality of the pitch conditioning. Fig. 3 shows the structure of the pitch encoder.

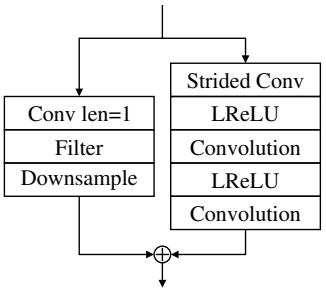


Fig. 3. Overview of the pitch encoder, showing the main and residual branches.

The signal passes through two branches, one responsible for the processing, and one residual, that changes only the number of channels and the sampling rate. The main branch is composed of a strided convolution that adjusts the scale and the number of channels, followed by two pairs of ReLU activations and convolutions. In the residual branch, the signal also passes through a convolutional layer of kernel size one, to adjust the number of channels, a non-learnable filter and a downsample, to adjust the sample rate. Unlike Liu et al., we use a selective non-learnable filter in the residual branch to prevent aliasing of the excitation signal. The two parts are then added together.

3) Encoder: We use two types of encoder, one based on a pre-trained self-supervised learning (SSL) model, which attains better results, and a purely convolutional model, which is faster and does not depend on a pre-trained model that may not be available for low-resource languages. Fig. 4 illustrates the SSL-based content encoder.

The chosen pre-trained encoder was the WavLM [21], which transforms an audio signal into a sequence of vectors that can be used in many tasks. The model was trained on a large scale multilingual audio dataset that includes Portuguese. The WavLM representation encodes various aspects of the audio signal, such as content, identity, noise, etc. To isolate the content information, the embedding is passed through a convolutional layer, which greatly reduces the number of

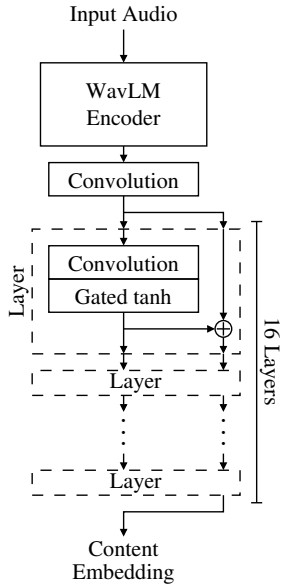


Fig. 4. Overview of the SSL-based content encoder, showing the pre-trained WavLM encoder and additional layers to refine the result

channels, and a series of convolutional layers with skip-connections, similarly to [27].

The purely convolutional encoder is the dual of the decoder. It uses a similar structure of alternating layers that change the resolution and the number of channels with residual blocks to refine the result, but with strided convolutional layers performing downsampling instead of transposed convolutional layers performing upsampling.

Neither encoder type uses speaker or pitch conditioning, and as such they are both speaker independent. In both encoders we also use an additional cost to help the network remove as much speaker information as possible, as detailed in Section III-C.

All convolutions in the generator are one-dimensional and weight normalized [37].

B. Discriminator

We use a multi-scale discriminator inspired by MelGAN [18]. A multi-scale discriminator is an ensemble of discriminators operating at different sample rates, that is, each discriminator receives as input a signal downsampled by a certain factor. Specifically, each discriminator D_i, $i = 1, 2, \ldots, N$, of the ensemble has its input downsampled by a factor 2^{i-1}. This allows the ensemble as a whole to analyze both long time windows and wide frequency bands. The discriminators have an identical architecture and do not share parameters. Other discriminator architectures to address the complexities of audio signals include the multi-period discriminator from HiFi-GAN [28] or the sub-band discriminator from Avocodo [38]. We found that using additional discriminators did not improve the results enough to justify the increased complexity.

Each discriminator has the task of identifying whether a signal is real of fake and if it is spoken by the correct speaker. The output of each discriminator is one independent binary

classification per speaker in the dataset, with a positive value representing a signal that is both real and spoken by that speaker. During training only the output corresponding to the target speaker is considered. We do not use other speakers because this often impairs the similarity by inducing the generator to produce signals that are distant from decision boundaries rather than more similar to the target [39]. The speaker dependent branching only happens in the last layer of each discriminator, allowing the network to share relevant information among the speaker classification tasks.

The structure of each discriminator is largely based on MelGAN, except for the output layers. Each discriminator consists of a series convolutional layers interleaved with Leaky ReLu activations, and each convolutional layer is weight normalized [37]. An input layer is followed by four layers of strided convolutional layers with relatively large kernel sizes, each downsampling the signal by a factor of four and increasing the number of channels. One additional layer and an output layer with a number of dimensions equal to the number of speakers produce the output per speaker. The output of the corresponding speaker is then selected as the final output. The output is not time-averaged, but rather presented as a sequence of results, one per window, as in PatchGAN [40].

Also following MelGAN, we use three scales in the multi-scale discriminator, and the downsampling operation in each scale is preceded by a moving average of length four. Fig. 5 shows the structure of the discriminator.

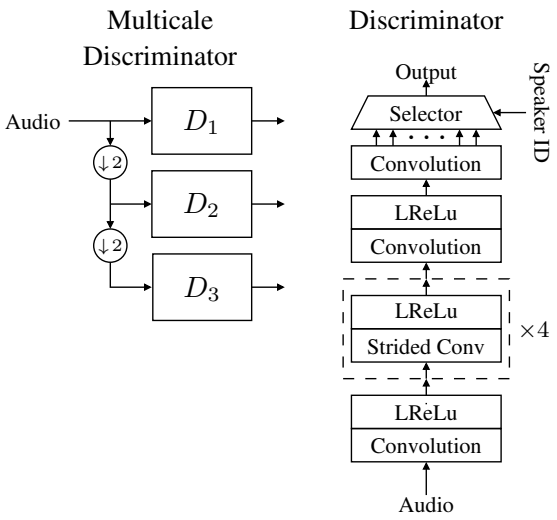


Fig. 5. Structure of the discriminator. Left: Ensemble of discriminators, each operating at a different sample rate. Right: Block diagram of one discriminator.

C. Loss Function

1) Discrimination Loss: The generator and the discriminators are trained adversarially with the Least Squares GAN [41] formulation of the adversarial loss, defined as

$$\mathcal{L}^D_{\text{adv}}(G,D) = \mathbb{E}_{p(\mathbf{s})}[(D(\mathbf{s})[i_t]-1)^2] + \mathbb{E}_{p(\mathbf{s},\mathbf{t})}[(D(G(\mathbf{s},\mathbf{c},\mathbf{e}))[i_t])^2] \quad (4)$$

for the discriminator, and

$$\mathcal{L}^G_{\text{adv}}(G,D) = \mathbb{E}_{p(\mathbf{s},\mathbf{t})}[(D(G(\mathbf{s},\mathbf{c},\mathbf{e}))[i_t]-1)^2] \quad (5)$$

for the generator, where i_t is the index of the output corresponding to the target speaker and $\mathbf{t}$ is a signal from the target speaker from which we obtain its associated embedding $\mathbf{c}$ and an excitation $\mathbf{e}$ according to equation (3).

The discriminator uses only the output corresponding to the target speaker, with the others ignored; therefore, its only criteria is how close the evaluated signal is to a real signal from the target speaker. The discriminator also is not explicitly trained to reject real signals from wrong speakers. The losses for each discriminator in the ensemble are independent.

2) Pitch loss: To train the pitch conditioning and guarantee that the network uses the information on pitch contained in the excitation signal, we introduce a novel pitch loss that compares the pitch contour of the converted signal to the ideal contour obtained from source and target pitches by exploiting a pre-trained pitch estimation network.

To obtain pitch estimates of the source, target and converted signals we use the CREPE neural pitch estimator [32]. The output of CREPE is a distribution over frequencies, representing the probability of the pitch being in the frequency range represented by a particular bin. The frequency of the bins is given in cents (a logarithmic scale). From this distribution, the pitch, as well as a voiced/unvoiced decision, can be obtained.

We use CREPE distributions as a basis for the pitch loss. The target converted pitch is obtained by equation (2) and we obtain the target converted distribution by shifting the distribution obtained from the source signal by a proportional amount. Because the frequency axis of the distribution is in cents, a logarithmic scale, right-shifting the distribution by $(\text{¢}_{\text{tgt}} - \text{¢}_{\text{src}})$, with ¢ being a frequency in cents, is equivalent to multiplying the pitch by $f_{\text{tgt}}/f_{\text{src}}$. We then convert the signal conditioned on the target converted pitch and obtain its corresponding pitch distribution. Both distributions are compared via L^2-norm.

Mathematically, the loss is given by

$$\mathcal{L}_{\text{f0}}(G) = \mathbb{E}_{p(\mathbf{s},\mathbf{t})}\left[\|C(\mathbf{s})[\text{¢}+\text{¢}_{\text{tgt}}-\text{¢}_{\text{src}}] - C(G(\mathbf{s},\mathbf{c},\mathbf{e}))[\text{¢}]\|_2\right], \quad (6)$$

where C is the CREPE pitch estimator.

The CREPE-based loss is a perceptual loss [42] in that it uses a pre-trained network to obtain high level information, but unlike most models, CREPE's output has a physical interpretation. CREPE and similar networks have been used to condition networks, both as input [43] and to provide losses [5], but to our knowledge, only for reconstruction, where the desired pitch is the input pitch. Our approach is more flexible, allowing one to obtain losses for converted signals when there is no ground truth.

3) Content embedding loss: To obtain a content embedding that is speaker independent, we use an approach similar to [23]. We compare the content embedding of the source signal with the content embedding of a modified version of the signal with changed identity and the same content.

To change the identity of the signal we apply the transformations used in [44]: the signal has its pitch and formants shifted by a random amount and then passes through a filter bank with random gain on each band.

The two content embeddings are compared with a contrastive loss defined as

$$\mathcal{L}_{\text{cont}}(E) = \mathbb{E}_{p(\mathbf{s})}\left[\sum_{t=0}^{T} \frac{\exp(\text{CS}(E(\mathbf{s}_t), E(\tilde{\mathbf{s}}_t)))}{\sum_{\tau\in\mathcal{T}_t}\exp(\text{CS}(E(\mathbf{s}_t), E(\mathbf{s}_\tau)))}\right], \tag{7}$$

where $\tilde{\mathbf{s}}$ is the speaker-modified version of the signal, E is the encoder of generator G, $\mathcal{T}_t$ is a set of time indices that includes time t plus a number of randomly selected time steps, and CS is the cosine similarity.

The contrastive loss minimizes the distance between vectors from each embedding at the same time stamps, where their content is the same, and maximizes the distance between vectors from the same embedding at different time stamps, where their content is likely different.

4) Reverse conversion loss: A converted signal contains the same content as the source signal, including time alignment, but different speakers. Training a model to convert such signal back to its original speaker is the ideal case for voice conversion, in which we have the exact desired signal to compare the result to.

We use this reverse transformation as an additional loss, comparing its result to the original signal. We found that, as long as the model is already able to convert signals decently, using this loss improves quality. As such, we exclude this loss from the training at the start, introducing it after the model has trained for a while.

To compare the signals we use both a multi-scale mel-spectral distance, the L^1 distance between the mel-spectrum of each signal, using a range of spectral resolutions, and a feature matching loss [45] [18], the L^1 distance between the feature maps of the discriminators when each signal is presented.

The multi-scale mel-spectral distance is

$$\mathcal{L}_{\text{spec}}(G, D) = \mathbb{E}_{p(\mathbf{s})}\left[\sum_{n\in N} \|\mathcal{S}_n(\mathbf{s}) - \mathcal{S}_n(\mathbf{G}(\tilde{\mathbf{s}}, \mathbf{c}, \mathbf{e}))\|_1\right], \tag{8}$$

where $\mathcal{S}_n(\mathbf{x})$ is the mel-spectrum of $\mathbf{x}$ obtained from a short-time Fourier transform with n frequency bins and proportional window and hop sizes and $N = \{512, 1024, 2048\}$.

The feature matching loss is defined as

$$\mathcal{L}_{\text{feat}}(G, D) = \mathbb{E}_{p(\mathbf{s})}\left[\sum_{k=1}^{K}\sum_{l=1}^{L} \frac{\|D_k^{(l)}(\mathbf{s}) - D_k^{(l)}(G(\tilde{\mathbf{s}}, \mathbf{c}, \mathbf{e}))\|_1}{N_l}\right], \tag{9}$$

where $D_k^{(l)}$ is the l-th feature map of the discriminator at scale k. In both losses, both $\mathbf{c}$ and $\mathbf{e}$ are obtained from the source signal, while $\tilde{\mathbf{s}}$ is the signal previously transformed into an arbitrary speaker.

The reverse conversion loss is the weighted sum

$$\mathcal{L}_{\text{rev}} = \lambda_{\text{spec}}\mathcal{L}_{\text{spec}} + \lambda_{\text{feat}}\mathcal{L}_{\text{feat}}. \tag{10}$$

This loss is similar to the cyclic conversion loss used in previous works [46], but with the important distinction that the gradient is not propagated beyond the converted signal. This prevents the network from 'cheating' by coordinating the forward and backward conversions to reduce the loss without actually improving the conversion.

5) Identity loss: We also use the identity loss to regularize the training and help enforce coherence between the input and output signals. The identity loss is the distance between a signal and itself converted with its own speaker as target, i.e. the signal reconstructed.

To compare the signals we employ the same losses used in Section III-C4, but with the original signal $\mathbf{s}$ instead of a converted signal $\tilde{\mathbf{s}}$.

This loss guarantees that the network is able to reconstruct signals. Because the other aspects of the system stimulating the encoded signal to be as speaker independent as possible, reconstructing the signal greatly overlaps with conversion. We also found that the identity loss is specially important at the beginning of training.

6) Global loss function: The global loss function is the weighted sum of all individual loss components:

$$\mathcal{L}_{\text{cyc}} = \mathcal{L}_{\text{adv}} + \lambda_{\text{cont}}\mathcal{L}_{\text{cont}} + \lambda_{\text{rev}}\mathcal{L}_{\text{rev}} + \lambda_{\text{idt}}\mathcal{L}_{\text{idt}} + \lambda_{\text{f0}}\mathcal{L}_{\text{f0}}, \tag{11}$$

where $\lambda_{\cdot}$ are the corresponding weights with respect to $\mathcal{L}_{\text{adv}}$.

These weights are hyper-parameters and affect the final result in a variety of ways. For example, the identity loss is important for the training to converge into intelligible audio, but if their weights are too high the network can learn to only reconstruct the input signal instead of performing conversion, since that is the best way to reduce those losses.

IV. Experimental Results

A. Dataset

One of the challenges of developing a Portuguese language voice conversion system is the lack of large, high-quality datasets. In this work we use the CETUC dataset [47], a dataset that is large and balanced, both in terms of frequency of phonemes and samples per speaker, but contains defects that can degrade the quality of generated voices.

The CETUC dataset contains recordings of a set of 1000 phonetically balanced sentences [48] uttered by 100 different speakers, lasting a few seconds each, totaling 145 hours of speech. Sampling rate is 16 kHz, but the overall quality of the recordings varies: the dataset contains both clean, relatively high-quality audio and audio degraded by various defects, such as noise, high frequency noise, ambient reverberation, etc. Even between different recordings of the same person there are inconsistencies that can make them appear, in extreme cases, to come from different people.

To improve intra-speaker consistency, we filtered the dataset. For each speaker, we extracted the speaker embedding of each signal using Resemblyzer [49] and clustered the embeddings using Gaussian mixture models. Because the dataset was recorded in different sessions, each speaker has a small number of signal groups, usually one to three, that share the same environmental and recording characteristics, and are therefore easily clustered. Based on cluster visualization and a few metrics, the most representative cluster for each speaker was manually selected to be part of the filtered dataset, in an attempt to balance signal quality and number of samples[1].

[1]The filtered dataset can be found in https://github.com/vicpc00/filtered_cetuc_dataset.

The new dataset is composed of 75.5 total hours of speech, i.e. 52% of the original. We use approximately 90% of the dataset for training, with the rest being used for test and validation. We also use a subset of 16 speakers, 8 male and 8 female, speaking the same 15 phrases for objective tests, with a smaller subset of 8 speakers being used for listening tests. For the tests, each signal is converted to each of the 16 speakers (including self transformation), totaling 187 hours of audio, which our model takes approximately 27 minutes to generate.

B. Training details

When training the conversion network from scratch, the model had trouble converging. We observed that at the beginning of the training, the conversion process mainly introduced noise into the learning process, making convergence difficult. For this reason, we decided to perform training in two stages: in the first stage the network performs only reconstruction, that is, the target speaker is always the same as the source speaker; in the second stage the network is trained for the speaker conversion itself.

In the first training stage, we give more weight to the identity loss and use neither pitch loss nor reverse conversion loss, forcing the network to simply learn how to produce audio. In the second stage, all loss functions are used, with the reverse conversion loss introduced only after a number of epochs. Hence, the network converts each signal to randomly selected speakers without the original speaker being excluded from the selection.

The first step is the longer of the two, due to the fact that synthesis is generally a more difficult task. The second step does not necessarily have to use the same choices of relative weights or other hyper-parameters such as the learning rate, which do not change the network structure. Since the final result is also more responsive to hyper-parameter choices in the second step, the separation also makes experimentation and hyper-parameter tuning less time-consuming and computationally expensive.

We found that even when using an additional cost function to make the latent space more speaker independent, it was never fully independent and our network still benefited from conversion-specific training.

Our code, detailed training instructions and hyper-parameter configuration can be found at https://github.com/vicpc00/td-vc-gan. Our model was trained on a single NVIDIA GeForce RTX 3090 for a total of 110 epochs, or approximately 355 hours.

C. Experiments

We performed listening tests to evaluate the quality of conversion: two versions of our model, with and without the pre-trained encoder, were compared with FreeVC [27] and YourTTS [50] — both trained with the same dataset as our model, following the configuration recommended in their repositories[2].

[2]Examples available at vicpc00.github.io/td-vc-gan.

The test was carried out by 20 listeners in a controlled environment. Each listener performed a series of evaluations in which they heard a converted signal and a natural reference signal from the target speaker uttering a different sentence, and were asked to rate the former, on a scale of one to five, for naturalness and similarity. As naturalness they are expected to assess the signal regarding defects such as noise, lack of intelligibility, and distortions; as similarity they are asked to assess how close the speaker of the sentence under test is to the reference speaker. Each listener evaluated 16 signals per system in a random order. Signals were balanced such that each speaker was used as source and target the same number of times per listener, with the frequency of conversion pair being balanced over the course of multiple listeners.

We also computed a series of objective measures: the mean log-f0 difference (MF0D), defined as $\sum_{(f_{\mathrm{conv}},f_{\mathrm{tgt}})} \left|\overline{\log f_{\mathrm{conv}}} - \overline{\log f_{\mathrm{tgt}}}\right| /N$, where $(f_{\mathrm{conv}}, f_{\mathrm{tgt}})$ is a pair of pitch contours extracted from corresponding utterances; the mel-cepstral distortion (MCD), defined as $\sum_{(\mathbf{C}_{\mathrm{conv}},\mathbf{C}_{\mathrm{tgt}})} \left\|\mathbf{C}_{\mathrm{conv}} - \mathbf{C}_{\mathrm{tgt}}\right\|^2 /N$, where $(\mathbf{C}_{\mathrm{conv}}, \mathbf{C}_{\mathrm{tgt}})$ is a pair of mel-cepstral representations extracted from corresponding utterances and aligned via dynamic time warping; the speaker embedding similarity (SES), defined as $\sum_{(\mathbf{R}_{\mathrm{conv}},\mathbf{R}_{\mathrm{tgt}})} \mathrm{CS}(\mathbf{R}_{\mathrm{conv}}, \mathbf{R}_{\mathrm{tgt}})/N$, where $(\mathbf{R}_{\mathrm{conv}}, \mathbf{R}_{\mathrm{tgt}})$ is a pair of speaker embeddings extracted from corresponding utterances with Resemblyzer [49] and $\mathrm{CS}()$ is the cosine similarity; and the word error rate (WER) of the Whisper [51] automatic speech recognizer when analyzing the converted signals.

Results are shown in Tab. I: the proposed model obtained the highest similarity score and the second highest naturalness score, losing only to FreeVC, which obtained the worse similarity score. Our model also outperformed or tied with the best model in all objective measures, except for word error rate, where once more it only lost to FreeVC. FreeVC's high naturalness score and low word error rate must be related to its poor similarity performance, since it results of not changing the source speaker enough. More dramatic changes between distant speakers tent to be harder and introduce more artifacts.

A very likely reason for FreeVC's poor similarity performance can be found in the mean log-f0 difference, where it obtained the worse scores, indicating that it often failed to convert the pitch properly, which greatly affects the perceived identity. YourTTS, on the other hand, obtained an MF0D comparable to our proposed method, even without pitch conditioning, because during training it aligns its content embedding with a text based representation, which is naturally pitch independent. This forces the model to ignore the source pitch, but is an additional requirement to train the network. Our method obtained the best pitch performance, with the choice of encoder not affecting the result.

Compared to our main proposed method, the version with a purely convolutional encoder attained worse results, reinforcing why most recent works in voice conversion utilize pre-trained networks as encoders [24]. The non-SSL model did obtain results comparable to YourTTS, with better naturalness and worse similarity. For underserved languages, where neither a pre-trained SSL model nor the resources to train one might

TABLE I
RESULTS OF LISTENING TESTS AND OBJECTIVE MEASURES. MEASURES: NATURALNESS MEAN OPINION SCORE (NAT MOS), SIMILARITY MOS (SIM MOS), MEAN LOG F0 DISTANCE (MF0D), MEL-CEPSTRAL DISTORTION (MCD), SPEAKER EMBEDDING SIMILARITY (SES), AND WORD ERROR RATE (WER)

Model	Nat MOS	Sim MOS	MF0D	MCD	SES	WER
Proposed	3.48 ± 0.14	4.05 ± 0.13	0.133 ± 0.004	1.334 ± 0.006	0.875 ± 0.001	0.057
Proposed (non-SSL)	2.93 ± 0.14	3.20 ± 0.16	0.131 ± 0.004	1.393 ± 0.005	0.847 ± 0.001	0.115
YourTTS	2.78 ± 0.13	3.67 ± 0.13	0.141 ± 0.004	1.347 ± 0.006	0.875 ± 0.001	0.177
FreeVC	3.95 ± 0.12	2.62 ± 0.17	0.227 ± 0.005	1.406 ± 0.007	0.826 ± 0.001	0.050

be available, this model may still be an adequate solution.

TABLE II
ABLATION STUDY FOR THE PITCH COST

Model	MF0D	WER
Base model	0.138 ± 0.004	0.290
No pitch loss	0.132 ± 0.004	0.657

To test the effectiveness of the pitch cost on the results, we trained a pair of models, with and without it. We trained a model with the convolutional encoder under the same conditions, except for the pitch cost during the second stage of training. The reverse conversion loss was not used in this training. Results are presented in Tab. II, where it can be seen that the lack of the pitch cost did not affect the pitch distance, but affected the word error rate. Even without the pitch cost, other costs, such as the adversarial cost, can indirectly guide the pitch conversion, as they are speaker dependent. While the excitation input was enough to guide the pitch conversion in many signals, in many others it led to distinctive artifacts that degraded the converted signal quality. Such artifacts[3] tend to occur near the transition between voiced and unvoiced frames; we speculate that they are due to the excitation signal not being correctly mixed with the rest of the signal during the transitions.

V. CONCLUSION

In this work, we proposed a pitch-controlled end-to-end system for voice conversion, as well as a cost function to effectively train it. We train our network on a pre-existing Brazilian Portuguese language dataset, overcoming problems in the dataset with a filtering scheme. Our network outperformed other public available models in listening tests and several objective measures, obtaining the best overall results when considering both the similarity and naturalness scores. Its similarity score of 4.05 was the highest among evaluated models and its naturalness score of 3.48 was the second highest, second only to FreeVC, which achieved a naturalness score of 3.95 but a similarity score of only 2.62. This result highlights the importance of precise pitch tracking: in our pitch difference metric, FreeVC obtained 0.227, versus our method's 0.133, which surely contributed to our better similarity performance. An ablation study of our novel pitch tracking cost confirms its importance for the quality of generated signals.

Our tests demonstrate the usefulness of pre-trained encoders for voice conversion. We used WavLM, which is designed to be a general purpose model. Some models, such as ContentVec [23], are specifically designed for voice conversion, but publicly available implementations are often only trained in English. Training or fine-tuning one such model for Portuguese might further improve our model's performance. Another natural extension is incorporating one of many techniques to allow conversion of speakers not seen during training (i.e. any-to-any conversion), such as speaker encoders.

[3]Examples can be heard in the example page.

REFERENCES

[1] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, vol. 1, New York, NY, USA, April 1988, pp. 655–658, doi: 10.1109/ICASSP.1988.196671.

[2] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Speech Audio Process.*, vol. 6, no. 2, pp. 131–142, March 1998, doi: 10.1109/89.661472.

[3] B. Sisman, J. Yamagishi, S. King, and H. Li, "An overview of voice conversion and its challenges: From statistical modeling to deep learning," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 132–157, 2021, doi: 10.1109/TASLP.2020.3038524.

[4] S. Liu, Y. Cao, N. Hu, D. Su, and H. Meng, "FastSVC: Fast cross-domain singing voice conversion with feature-wise linear modulation," in *IEEE Int. Conf. Multimedia Expo (ICME)*, Virtual, July 2021, doi: 10.1109/ICME51207.2021.9428161.

[5] A. Polyak, L. Wolf, Y. Adi, and Y. Taigman, "Unsupervised cross-domain singing voice conversion," in *Proc. Interspeech 2020*, Shanghai, China, October 2020, pp. 801–805, doi: 10.21437/Interspeech.2020-1862.

[6] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Commun.*, vol. 27, no. 3–4, pp. 187–207, April 1999, doi: 10.1016/S0167-6393(98)00085-5.

[7] 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. 99, no. 7, pp. 1877–1884, July 2016, doi: 10.1587/transinf.2015EDP7457.

[8] Y. Qiao, D. Saito, and N. Minematsu, "HMM-based sequence-to-frame mapping for voice conversion," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Dallas, TX, USA, March 2010, pp. 4830–4833, doi: 10.1109/ICASSP.2010.5495141.

[9] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "CycleGAN-VC3: Examining and improving cyclegan-vcs for mel-spectrogram conversion," in *Proc. Interspeech 2020*, Shanghai, China, October 2020, pp. 2017–2021, doi: 10.21437/Interspeech.2020-2280.

[10] 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*, Brno, Czechia, September 2021, pp. 1349–1353, doi: 10.21437/Interspeech.2021-319.

[11] P. L. Tobing, Y.-C. Wu, T. Hayashi, K. Kobayashi, and T. Toda, "Non-parallel voice conversion with cyclic variational autoencoder," in *Proc. Interspeech 2019*, Graz, Austria, September 2019, pp. 674–678, doi: 10.21437/Interspeech.2019-2307.

[12] W.-C. Huang, T. Hayashi, Y.-C. Wu, H. Kameoka, and T. Toda, "Voice transformer network: Sequence-to-sequence voice conversion using transformer with text-to-speech pretraining," in *Proc. Interspeech 2020*, Shanghai, China, October 2020, pp. 4676–4680, doi: 10.21437/Interspeech.2020-1066.

[13] S. H. Mohammadi and T. Kim, "One-shot voice conversion with disentangled representations by leveraging phonetic posteriorgrams," in *Proc. Interspeech 2019*, Graz, Austria, September 2019, pp. 704–708, doi: 10.21437/Interspeech.2019-1798.

[14] 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," in *Arxiv*, 2016, [Online]. Available: https://arxiv.org/abs/1609.03499, doi: 10.48550/arXiv.1609.03499.

[15] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, vol. 80, Stockholm, Sweden, July 2018, pp. 2410–2419, [Online]. Available: https://proceedings.mlr.press/v80/kalchbrenner18a.html.

[16] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Barcelona, Spain, May 2020, pp. 6199–6203, doi: 10.1109/ICASSP40776.2020.9053795.

[17] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, May 2019, pp. 3617–3621, doi: 10.1109/ICASSP.2019.8683143.

[18] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "MelGAN: Generative adversarial networks for conditional waveform synthesis," in *Advances Neural Inf. Process. Syst. 32 (NeurIPS)*, Vancouver, Canada, December 2019, p. 14910–14921, [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/6804c9bca0a615bdb9374d00a9fcba59-Paper.pdf.

[19] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *IEEE Int. Conf. Multimedia Expo (ICME)*, Seattle, USA, July 2016, pp. 1–6, doi: 10.1109/ICME.2016.7552917.

[20] A. Polyak, L. Wolf, and Y. Taigman, "TTS skins: Speaker conversion via ASR," in *Proc. Interspeech 2020*, Shanghai, China, October 2020, pp. 786–790, doi: 10.21437/Interspeech.2020-1416.

[21] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 6, pp. 1505–1518, 2022, doi: 10.1109/JSTSP.2022.3188113.

[22] W.-N. Hsu, Y.-H. H. Tsai, B. Bolte, R. Salakhutdinov, and A. Mohamed, "Hubert: How much can a bad teacher benefit asr pre-training?" in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Toronto, ON, Canada, 2021, pp. 6533–6537, doi: 10.1109/ICASSP39728.2021.9414460.

[23] K. Qian, Y. Zhang, H. Gao, J. Ni, C.-I. Lai, D. Cox, M. Hasegawa-Johnson, and S. Chang, "ContentVec: An improved self-supervised speech representation by disentangling speakers," in *Proc. 39th Int. Conf. Mach. Learn. (ICML)*, vol. 162. Baltimore, MD, USA: PMLR, July 2022, pp. 18 003–18 017, [Online]. Available: https://proceedings.mlr.press/v162/qian22b.html.

[24] W.-C. Huang, L. P. Violeta, S. Liu, J. Shi, and T. Toda, "The singing voice conversion challenge 2023," in *IEEE Autom. Speech Recognit. Understanding Workshop (ASRU)*, Taipei, Taiwan, 2023, pp. 1–8, doi: 10.1109/ASRU57964.2023.10389671.

[25] E. Nachmani and L. Wolf, "Unsupervised singing voice conversion," in *Proc. Interspeech 2019*, Graz, Austria, September 2019, pp. 2583–2587, doi: 10.21437/Interspeech.2019-1761.

[26] B. Nguyen and F. Cardinaux, "NVC-Net: End-to-end adversarial voice conversion," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP))*, Singapore, Singapore, 2022, pp. 7012–7016, doi: 10.1109/ICASSP43922.2022.9747020.

[27] J. Li, W. Tu, and L. Xiao, "Freevc: Towards high-quality text-free one-shot voice conversion," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Rhodes Island, Greece, 2023, pp. 1–5, doi: 10.1109/ICASSP49357.2023.10095191.

[28] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Advances Neural Inf. Process. Syst. 33 (NeurIPS)*, vol. 33, 2020, pp. 17 022–17 033, [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/c5d736809766d46260d816d8dbc9eb44-Paper.pdf.

[29] R. Yoneyama, Y.-C. Wu, and T. Toda, "Source-filter HiFi-GAN: Fast and pitch controllable high-fidelity neural vocoder," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Rhodes Island, Greece, 2023, pp. 1–5, doi: 10.1109/ICASSP49357.2023.10095298.

[30] C. Deng, C. Yu, H. Lu, C. Weng, and D. Yu, "Pitchnet: Unsupervised singing voice conversion with pitch adversarial network," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Barcelona, Spain, 2020, pp. 7749–7753, doi: 10.1109/ICASSP40776.2020.9054199.

[31] C. Ho Chan, K. Qian, Y. Zhang, and M. Hasegawa-Johnson, "Speech-Split2.0: Unsupervised speech disentanglement for voice conversion without tuning autoencoder bottlenecks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Singapore, Singapore, 2022, pp. 6332–6336, doi: 10.1109/ICASSP43922.2022.9747763.

[32] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "CREPE: A convolutional representation for pitch estimation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Calgary, Canada, April 2018, pp. 161–165, doi: 10.1109/ICASSP.2018.8461329.

[33] P. Neekhara, S. Hussain, R. Valle, B. Ginsburg, R. Ranjan, S. Dubnov, F. Koushanfar, and J. McAuley, "SelfVC: Voice conversion with iterative refinement using self transformations," 2023, doi: 10.48550/arXiv.2310.09653.

[34] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville, "FiLM: Visual reasoning with a general conditioning layer," in *Proc. AAAI Conf. Artif. Intell., 32(1)*, New Orleans, USA, Feb. 2018, doi: 10.1609/aaai.v32i1.11671.

[35] J. chieh] Chou and H.-Y. Lee, "One-shot voice conversion by separating speaker and content representations with instance normalization," in *Proc. Interspeech 2019*, September 2019, pp. 664–668, doi: 10.21437/Interspeech.2019-2663.

[36] X. Wang, S. Takaki, and J. Yamagishi, "Neural source-filter-based waveform model for statistical parametric speech synthesis," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, May 2019, pp. 5916–5920, doi: 10.1109/ICASSP.2019.8682298.

[37] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Advances Neural Inf. Process. Syst. 29 (NeurIPS)*, December 2016, p. 901–909, [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/ed265bc903a5a097f61d3ec064d96d2e-Paper.pdf.

[38] T. Bak, J. Lee, H. Bae, J. Yang, J.-S. Bae, and Y.-S. Joo, "Avocodo: Generative adversarial network for artifact-free vocoder," in *Proc. AAAI Conf. Artif. Intell., 37(11)*, vol. 37, no. 11, Jun. 2023, pp. 12 562–12 570, doi: 10.1609/aaai.v37i11.26479.

[39] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "StarGAN-VC2: Rethinking conditional methods for stargan-based voice conversion," in *Proc. Interspeech 2019*, Graz, Austria, September 2019, pp. 679–683, doi: 10.21437/Interspeech.2019-2236.

[40] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, USA, July 2017, pp. 5967–5976, doi: 10.1109/CVPR.2017.632.

[41] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. on Comput. Vis. (ICCV)*, Venice, Italy, October 2017, pp. 2813–2821, doi: 10.1109/ICCV.2017.304.

[42] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Amsterdam, The Netherlands: Springer International Publishing, October 2016, pp. 694–711, doi: 10.1007/978-3-319-46475-6_43.

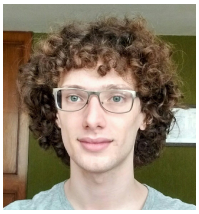
[43] J. Engel, L. H. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," in *8th Int. Conf. on Learn. Representations, (ICLR)*, Addis Ababa, Ethiopia, April 2020, [Online]. Available: https://openreview.net/forum?id=B1x1ma4tDr.

[44] H.-S. Choi, J. Lee, W. Kim, J. Lee, H. Heo, and K. Lee, "Neural analysis and synthesis: Reconstructing speech from self-supervised representations," in *Advances Neural Inf. Process. Syst. 34 (NeurIPS)*, vol. 34, 2021, pp. 16 251–16 265, [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/87682805257e619d49b8e0dfdc14affa-Paper.pdf.

[45] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. 33rd Int. Conf. Mach. Learn. (ICML)*, vol. 48, New York, USA, June 2016, pp. 1558–1566, [Online]. Available: https://proceedings.mlr.press/v48/larsen16.html.

[46] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. on Comput. Vis. (ICCV)*, Venice, Italy, October 2017, pp. 2242–2251, doi: 10.1109/ICCV.2017.244.

[47] V. F. S. Alencar and A. Alcaim, "LSF and LPC – Derived features for large vocabulary distributed continuous speech recognition in Brazilian Portuguese," in *42nd Asilomar Conf. Signals Syst. Comput.*, Pacific Grove, CA, USA, 2008, pp. 1237–1241, doi: 10.1109/ACSSC.2008.5074614.
[48] R. J. Cirigliano and C. Monteiro, "Um conjunto de 1000 frases foneticamente balanceadas para o português brasileiro obtido utilizando a abordagem de algoritmos genéticos," in *22nd Brazilian Telecommun. Symp. (SBrT)*, Campinas, Brazil, September 2005, doi: 10.14209/sbrt.2005.544.
[49] G. Louppe, "Resemblyzer," 2019, [Online]. Available: https://github.com/resemble-ai/Resemblyzer.
[50] E. Casanova, J. Weber, C. D. Shulby, A. C. Junior, E. Gölge, and M. A. Ponti, "YourTTS: Towards zero-shot multi-speaker TTS and zero-shot voice conversion for everyone," in *Proc. 39th Int. Conf. Mach. Learn. (ICML)*, vol. 162. Baltimore, MD, USA: PMLR, July 2022, pp. 2709–2720, [Online]. Available: https://proceedings.mlr.press/v162/casanova22a.html.
[51] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc. 40th Int. Conf. Mach. Learn. (ICML)*, vol. 202. Honolulu, HI, USA: PMLR, JulY 2023, pp. 28 492–28 518, [Online]. Available: https://proceedings.mlr.press/v202/radford23a.html.



Victor P. da Costa recieved a B.Sc. in Eletronic and Computing Engineering (2015) and a M.Sc. in Electric Engineering (2017), both from Universidade Federal do Rio de Janeiro (UFRJ), Brazil. He is currently pursuing the D.Sc. degree in electrical engineering at COPPE/UFRJ. His interests are machine learning and digital signal processing, particularly audio processing.

Sergio L. Netto holds BSc (Federal University of Rio de Janeiro, 1991), MSc (COPPE/Federal University of Rio de Janeiro, 1992), and PhD (University of Victoria, Canada, 1996) degrees, all in Electrical Engineering. He is currently a full professor at the Federal University of Rio de Janeiro. He is a coauthor of Digital Signal Processing: System Analysis and Design (Cambridge University Press, 2nd ed., 2010) and Variational Methods for Machine Learning with Applications to Deep Networks, (Springer, 2021). His main teaching and research interests include speech processing, information theory, and applied artificial intelligence.

Luiz W. P. Biscainho was born in Rio de Janeiro, Brazil, in 1962. He received the Electronic Engineering degree (magna cum laude) from the EE (now Poli) at Universidade Federal do Rio de Janeiro (UFRJ), Brazil, in 1985, and the M.Sc. and D.Sc. degrees in Electrical Engineering from the COPPE at UFRJ in 1990 and 2000, respectively. Having worked in the telecommunications industry between 1985 and 1993, Dr. Biscainho is now Associate Professor at the Department of Electronic and Computer Engineering (DEL) of Poli and the Electrical Engineering Program (PEE) of COPPE, at UFRJ. His research area is digital audio processing. He is currently a member of the IEEE (Institute of Electrical and Electronics Engineers), the AES (Audio Engineering Society), the SBrT (Brazilian Telecommunications Society), and the SBC (Brazilian Computer Society).

Ranniery Maia received the B.Sc. degree in Electrical Engineering from Federal University of Rio Grande do Norte (1998), M.Sc. degree in Electrical Engineering from Federal University of Rio de Janeiro (2000) and D.Eng. degree in Engineering and Computer Science from Nagoya Institute of Technology (2006). From 2006 to 2009 he was a Research Scientist at NICT/ATR Spoken Language Communication Labs, Kyoto, Japan. From 2009 to 2016 he was a Research Engineer at Toshiba Research Europe Limited, Cambridge, UK. From 2018 to 2022 he was a Consultant on Machine Learning and Text-to-Speech at DeepZen Limited, London, United Kingdom, and a Visiting Researcher at Federal University of Santa Catarina, Florianopolis, Brazil. Currently he is an Assistant Professor at Federal University of Rio Grande do Norte, Natal, Brazil. His interests are artificial intelligence, machine learning, deep learning, speech synthesis, speech recognition and voice conversion.