ATTR: A Transformer-Based Model for Unpaired Audio-to-Audio Translation

Maharnab Saikia
Independent Researcher / Assam, India
maharnabsaikia@gmail.com

Abstract

Unpaired audio-to-audio translation aims to translate audio from a source domain to a target domain without paired training data. Cycle-Consistent Generative Adversarial Networks (CycleGAN) and Variational Autoencoders (VAE) have been used for this task, but these models suffer from difficult training and unsatisfactory results. Later, Contrastive Voice Conversion (CVC) was introduced, utilizing a contrastive learning-based approach to address these issues. However, these methods use CNN-based generators, which can capture local semantics but lacks the ability to capture long-range dependencies necessary for global semantics. In this paper, we propose ATTR, an efficient method for unpaired audio-to-audio translation that leverages the Hybrid Perception Block (HPB) and Dual Pruned Self-Attention (DPSA) along with a contrastive learning-based adversarial approach.

1 Introduction

Unpaired audio-to-audio translation aims to translate audio from a source domain to a target domain without paired training data. We focus on voice conversion, where, given a speech signal from a source speaker, the goal is to convert it to the voice of a target speaker while preserving the speech content.

Previous audio translation approaches use different training strategies, such as adversarial-based methods like Cycle-Consistent Generative Adversarial Networks (Zhu et al., 2017), Variational Autoencoders (Kingma and Welling, 2013), and contrastive learning-based adversarial approaches like Contrastive Unpaired Translation (Park et al., 2020). These training strategies were initially introduced for unpaired image-to-image translation. Since unpaired voice conversion follows a similar domain-to-domain conversion process, they were applied to this task as well, like CycleGAN-VC (Kaneko and Kameoka, 2018), VAE (Hsu et al., 2016), and CVC (Li et al., 2020). However, these methods still rely on convolutional neural network-based generators and focus primarily on training strategies. Although CNNs are effective at capturing local semantics, they lack the ability to capture long-range dependencies necessary for global semantics.

To address this issue, later works introduced the Vision Transformer into the generator, but it faced challenges related to generation difficulty and computational limitations.

Recently, ITTR (Zheng et al., 2022) was proposed to enhance the Vision Transformer, making it more effective and efficient. Inspired by its success, we propose ATTR, an audio-to-audio translation method that leverages the Hybrid Perception Block (HPB) for token mixing across different receptive fields to better utilize global semantics and Dual Pruned Self-Attention (DPSA) to significantly reduce computational complexity. Additionally, we incorporate contrastive learning for efficient one-way adversarial training.

2 Related Works

2.1 Contrastive Learning for Unpaired Translation

CUT (Contrastive Unpaired Translation) represents a significant advancement in the domain of unpaired translation. Building on the foundational work of CycleGAN, CUT introduced the use of contrastive learning to achieve high-quality translation between domains with a more efficient model architecture. By reducing the need for multiple generators and discriminators, CUT simplifies the learning process while still delivering impressive results in terms of output quality.

Later, the CycleGAN approach led to the introduction of CycleGAN-VC (Kaneko and Kameoka, 2018). Inspired by the contrastive learning ap-

proach in CUT, Contrastive Voice Conversion (CVC) (Li et al., 2020) was introduced for audio translation, achieving promising results in voice conversion tasks.

2.2 Vision Transformer

The Vision Transformer (ViT) (Dosovitskiy et al., 2020) is a pioneering model that applies the transformer architecture, originally developed for natural language processing, to the domain of image analysis. Unlike traditional convolutional neural networks (CNNs), ViT uses self-attention mechanisms to process image patches as sequences, enabling it to model long-range dependencies across an entire image. This allows ViT to develop a comprehensive global understanding of visual data, making it particularly effective for tasks requiring complex feature extraction.

While ViT has demonstrated state-of-the-art performance on various benchmarks, particularly with large-scale datasets, other research has shown that combining CNNs with vision transformers can achieve even better results (Xiao et al., 2021). This hybrid approach leverages the strengths of both architectures: CNNs excel at capturing local features, while transformers provide global context, leading to improved performance across diverse tasks. However, it faced challenges related to generation difficulty and computational limitations. To address these issues, ITTR was later introduced, making the Vision Transformer more efficient. In our work, we employ this architecture.

2.3 Generative Adversarial Networks

Generative models have become foundational in the field of deep learning, enabling the creation of new data instances that resemble a given dataset. Among these, Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) have been particularly influential due to their ability to generate high-quality data. The adversarial framework, consisting of a generator and a discriminator, allows GANs to learn complex data distributions effectively. However, GANs often suffer from issues such as mode collapse, where the model generates limited diversity in outputs, and training instability, requiring careful tuning of hyperparameters and network architectures. We encountered instability during training, which is further discussed in the experiments section.

3 Methods

In this section, we introduce the architecture and detailed design of ATTR, including its overall structure, key building blocks, and the efficient approach for the Vision Transformer, namely the Hybrid Perception Block and Dual Pruned Self-Attention. Finally, we present the learning objectives.

3.1 Model Architecture

3.1.1 Generator

The key modification in this approach is replacing the original generator in CVC with the ITTR generator architecture. The generator consists of a ResNet (He et al., 2015), similar to CycleGAN, but with a Hybrid Perception Block at the bottleneck. The Hybrid Perception Block integrates convolution for learning local semantics and Dual Pruned Self-Attention for capturing global semantics.

In our approach, we convert audio into mel spectrograms, treating them as 2D image-like representations of audio. Mel spectrograms encode both time and frequency information in a structured visual format, making them suitable for processing with image-based architectures.

As shown in Figure 1, we use the initial convolutional stem of the CycleGAN generator to produce overlapping patch embeddings of the input image. The patch embedding layer consists of three stacked convolutional layers, and the resulting overlapping patch embeddings have dimensions of 13×13 pixels.

This is followed by nine Hybrid Perception Blocks (HPBs) stacked in the bottleneck of our generator, consistent with the generator architecture in ITTR (Zheng et al., 2022) and the original CycleGAN (Zhu et al., 2017). Lastly, the decoder of ATTR is implemented using three convolutional layers.

Hybrid Perception Block: The Hybrid Perception Block (HPB) consists of two branches designed for capturing informative features and token mixing: a local branch for extracting local features from tokens and a global branch for capturing long-range dependencies. These branches are concatenated and passed through a convolutional feedforward network.

The local branch utilizes a convolutional neural network, specifically depthwise convolution (DWConv), to reduce complexity, while the global branch employs self-attention to establish relationships between individual token pairs and their con-

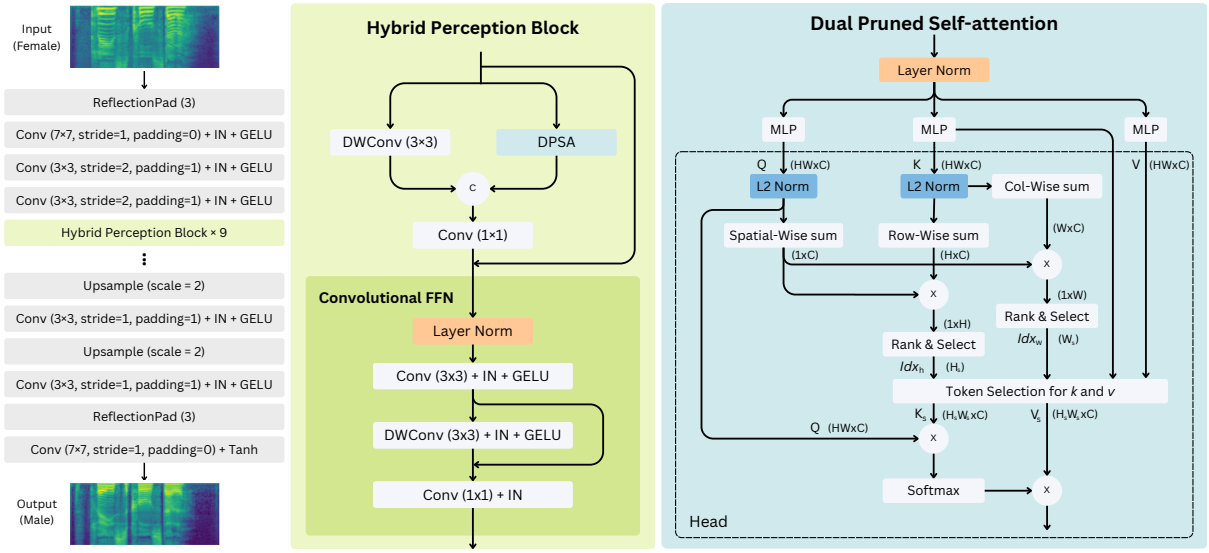

Figure 1: The ATTR architecture, consists of a ResNet on the left and a single Hybrid Perception Block (HPB) on the middle. Concatenation is denoted by 'C,' while 'IN' represents Instance Normalization and 'Conv' denotes Convolution. Dual Pruned Self-Attention (DPSA) on the right, 'L2 Norm' to indicate L2 normalization and 'X' to denote matrix multiplication. The dotted box represents operations within a single attention head. For best clarity, view the diagram zoomed in.

text. After concatenation, the tokens are processed through a convolutional MLP to integrate contextual information from both branches. Finally, they pass through a feedforward convolutional block for further token fusion. Instance normalization (Ulyanov et al., 2016) (Ulyanov et al., 2017) and GELU (Hendrycks and Gimpel, 2016) activation are applied to stabilize the flow in the convolutional feedforward network.

Dual Pruned Self-Attention: DPSA technique as shown in Figure 1, reduces computation and memory usage by focusing only on the most important tokens before calculating attention. This avoids the costly process of computing attention between every pair of tokens. Instead of analyzing each token individually, tokens are grouped into rows and columns. The contribution of each row and column to the attention is calculated, and only the top rows and columns with the highest contributions are retained. Attention is then computed only on these pruned tokens.

The token scoring is formulated as follows by grouping them into rows and columns and summing their values across each row and column:

$$Score_r = \sum_{i=1}^{N}\sum_{j=1}^{W} q_i k_{rj}^T = \left(\sum_{i=1}^{N} q_i\right)\left(\sum_{j=1}^{W} k_{rj}\right)^T, \quad r \in \{1,\ldots,H\} \quad (1)$$

$$Score_c = \sum_{i=1}^{N}\sum_{j=1}^{H} q_i k_{jc}^T = \left(\sum_{i=1}^{N} q_i\right)\left(\sum_{j=1}^{H} k_{jc}\right)^T, \quad c \in \{1,\ldots,W\} \quad (2)$$

Rows and columns are ranked based on their scores, and the top N_s rows and N_s columns are selected. N_s is a hyperparameter set to $\sqrt{H}$, where H is the height of the input grid. Tokens outside these selected rows and columns are discarded. An $ArgMaxScore$ operation is applied:

$$Index_r = ArgMaxScore(Score_r)[:N_s] \quad (3)$$

$$Index_c = ArgMaxScore(Score_c)[:N_s] \quad (4)$$

$$K_s = K[Index_r, Index_c], V_s = V[Index_r, Index_c] \quad (5)$$

Next, the pruned keys K_s and values V_s are reshaped into a smaller matrix, and attention is computed only between the queries Q and the pruned K_s and V_s. Temperature scaling is not used, as pruning naturally limits attention values, eliminating the need for the usual $\frac{1}{\sqrt{D}}$ factor.

$$Attention = Softmax(Q \cdot K_s^T) \cdot V_s \quad (6)$$

By applying L2 normalization to keys and queries, attention scores remain between -1 and 1, preventing skewed results. Since N_s is set to $\sqrt{H}$ in practice, for a single attention head, the computational complexity is reduced from $O(N^2C)$ to $O(NHC)$.

3.1.2 Discriminator

For the discriminator, the PatchGAN architecture is used, as described in the original Pix2Pix paper (Isola et al., 2016). The PatchGAN discriminator evaluates the realism of overlapping image patches rather than the entire image. This allows the discriminator to focus on high-frequency details. Since we treat spectrograms as 2D images, this helps to improve the sharpness and overall quality of the generated spectrograms.

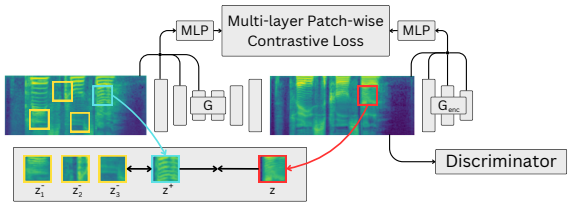

Figure 2: Patchwise contrastive learning is used for one-sided translation, A generated output patch should be closer to its corresponding input patch compared to other random patches. To achieve this, we use a multi-layer patch-wise contrastive loss, which maximizes mutual information between corresponding input and output patches.

3.2 Loss Functions

Adversarial Loss: The adversarial loss (Goodfellow et al., 2014), denoted as L_{GAN}, encourages the generator to produce images that are indistinguishable from those in the target domain. The loss is formulated as follows:

$$L_{GAN}(G, D, X, Y) = \mathbb{E}_{y \sim Y} \log D(y) + \mathbb{E}_{x \sim X} \log(1 - D(G(x))) \quad (7)$$

where G is the generator, D is the discriminator and X and Y are the input and target domains, respectively.

Patchwise Contrastive Loss: A key component of the CUT framework. It encourages the corresponding patches between the input and output images to share similar features, thereby enhancing the consistency of the image translation. For each selected layer l in the generator's encoder. The PatchNCE loss is given by:

$$l(v, v^+, v^-) = -\log \left(\frac{\exp(\frac{v \cdot v^+}{\tau})}{\exp(\frac{v \cdot v^+}{\tau}) + \sum_{n=1}^{N} \exp(\frac{v \cdot v_n^-}{\tau})} \right) \quad (8)$$

$$L_{PatchNCE}(G, H, X) = \mathbb{E}_{x \sim X} \sum_{l=1}^{L} \sum_{s=1}^{S_l} l(\hat{z}_l^s, z_l^s, z_l^{S \setminus s}) \quad (9)$$

In this equation, $\hat{z}_l^s$ represents the features of the output patches, z_l^s represents the corresponding input patches, and $z_l^{S \setminus s}$ represents negative patches from the same input image. Corresponding patches from the input and output features are pulled closer, while non-corresponding patches from the same input image are pushed apart, as illustrated in Figure 2.

Final Objective: The final loss is a combination of the adversarial loss and the PatchNCE loss $L_{PatchNCE}(G, H, X)$. There is also a PatchNCE loss $L_{PatchNCE}(G, H, Y)$, which acts as an identity loss, as explained in the CUT paper (Park et al., 2020). The overall loss function is expressed as:

$$L_{total} = L_{GAN}(G, D, X, Y) + \lambda_X L_{PatchNCE}(G, H, X) + \lambda_Y L_{PatchNCE}(G, H, Y) \quad (10)$$

The weights λ_X and λ_Y are hyperparameters that control the contribution of each loss term in PatchNCE loss $L_{PatchNCE}(G, H, X)$ and $L_{PatchNCE}(G, H, Y)$. When using identity loss, the values are chosen as $\lambda_X = 1$ and $\lambda_Y = 1$. When identity loss is not used, the values are set as $\lambda_X = 10$ and $\lambda_Y = 0$.

4 Experiments

4.1 Unsupervised Audio-to-Audio Translate

Dataset: Experiments are conducted on the VCTK corpus (Yamagishi et al., 2016), which contains 44 hours of speech data uttered by 109 native English speakers with various accents. For one-to-one voice conversion, speech data was collected from two different speakers. Fifty speech samples from the source speaker were excluded from training for evaluation.

For many-to-one voice conversion, data from 100 speakers was used, treating them as a single source domain, while 9 speakers were excluded for evaluation on unseen source speakers. Additionally, 50 unique speech samples were set aside for evaluation in the many-to-one setting.

Training Strategy: To ensure a fair comparison, we adopt the same training strategy as CVC. From the dataset, we extracted speech data and used only 2-second segments. No padding was applied, as described in CVC, since it can lead to model collapse.

For the vocoder, Parallel WaveGAN (Yamamoto et al., 2019) was used to synthesize waveform signals. The speech data, originally in 48 kHz, was downsampled to 24 kHz. We extracted 80-band mel spectrograms and followed the default settings of Parallel WaveGAN. The window size, FFT size, and hop size were set to 1024, 1024, and 256, respectively. The frequency range was set with fmin = 80 Hz and fmax = 7600 Hz.

We used the Adam (Kingma and Ba, 2014) optimizer with a learning rate of 2e-4 for the first 850 epochs, which was then linearly reduced to 0 over the next 150 epochs, for a total of 1000 epochs. Training was conducted on a single GPU with a batch size of 1. Since our architecture differs, we used layers [0, 4, 7, 10, 14] from the encoder of our generator to calculate PatchNCE loss.

Evaluation: We used a voice encoder system to measure the similarity between the generated fake-translated voice and the real voice in the target domain. Specifically, we used Resemblyzer, an open-source speaker verification system that derives high-level voice representations using a deep learning model.

Given a speech sample, the Resemblyzer generates an embedding vector. Then, cosine similarity is computed between the generated embedding and the target embedding, producing a numerical score between 0 and 1, where 0 indicates completely different speakers and 1 signifies the same speaker.

Baseline: We compared ATTR with previous unpaired audio-to-audio translation methods, as shown in Table 1. It achieved comparable or better voice similarity between the generated fake voice and the real voice. The baseline methods include VAE (Hsu et al., 2016), CycleGAN-VC3 (Kaneko et al., 2020), CVC (Li et al., 2020), and CNEG-VC (Prihasto et al., 2023).

Additionally, Table 2 presents the MACs (Multiply-Accumulate Operations) and parameters of the generator. Although ATTR is not a lightweight architecture, it achieved better results with lower complexity than previous models.

Method	One to One (M-F)	One to One (F-M)	Many to One (M-F)	Many to One (F-M)	Many (unseen) to One (M-F)	Many (unseen) to One (F-M)
VAE	0.805	0.874	0.803	0.845	-	-
CycleGAN	0.925	0.951	0.926	0.968	0.910	0.935
CVC (CUT)	0.929	0.952	0.937	0.974	0.925	0.945
CNEG-VC	0.934	0.963	0.945	**0.976**	0.937	**0.957**
ATTR (CUT)	**0.963**	**0.973**	**0.946**	0.967	**0.950**	0.949

Table 1: Comparison of voice similarity scores for different methods. Higher values indicate better performance.

Method	MACs(G)	Params(G)
VAE	1.1	1.1
CycleGAN-VC	12.8	11.4
CVC	12.8	11.4
CNEG-VC	12.8	11.4
ATTR	10.2	8.5

Table 2: Statistics of generator MACs and parameters for each method.

While CVC used replication padding without output normalization to mitigate model collapse, they stated that using reflection padding or zero padding led to model collapse.

During training, we applied different padding strategies across the entire generator. However, we still observed occasional loss spikes, indicating that the issue was not related to padding. To address this, we experimented with various techniques, such as gradient clipping. Ultimately, we found that using a Tanh activation function at the output resolved instability, even when using reflection padding, as shown in Figure 3. Despite this improvement, modal collapse remains a persistent challenge in many-to-one scenarios.

We applied min-max normalization to scale the values to the range [0,1] and then adjusted the range to [-1,1] by transforming the normalized values.

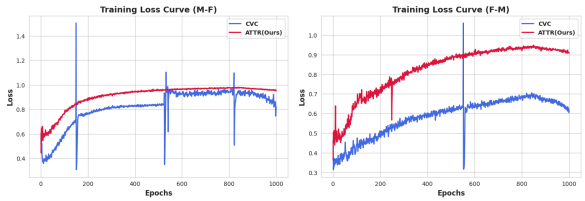

Figure 3: The loss curve from generator training is shown, with CVC represented in royal blue and our model, ATTR, in crimson. The left graph corresponds to male-to-female (one-to-one) conversion, while the right graph represents female-to-male (one-to-one) conversion.

4.2 Ablation Study

First, we removed each individual branch from HPB separately. Removing either the local perception branch or the DPSA branch lowered the voice similarity score, suggesting that both DPSA for global context and depth-wise convolution for local features play important roles in performance. Next, we removed token-wise L2 normalization for Q and K, which was used to eliminate the negative impact of peaked token vectors before the softmax. This led to a significant drop in the score. The experimental results for the ablation study are shown in Table 3.

Configuration	One-to-One (M-F)	One-to-One (F-M)
Ours	**0.959**	**0.970**
w/o DPSA	0.945	0.961
w/o Local Perception	0.936	**0.970**
w/o L2 Norm	0.937	0.948

Table 3: Experimental results for ablation study. Higher scores are shown in bold.

5 Conclusion

In this paper, we proposed an efficient generator architecture combined with a contrastive learning method for unpaired audio-to-audio translation. Our approach captures local features using CNNs and global context using Transformers in unpaired voice conversion. Experimental results have shown that our method achieved better voice similarity scores compared to baseline methods.

Limitations

However, modal collapse persists in many-to-one audio translations. While Tanh activation stabilizes training by constraining outputs, residual issues remain. Future work could explore adaptive thresholds or spectral diversity losses to fully address these challenges.

References

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv*.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. *NIPS*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv*.

Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang. 2016. Voice conversion from non-parallel corpora using variational auto-encoder. *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2016. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Takuhiro Kaneko and Hirokazu Kameoka. 2018. Cyclegan-vc: Non-parallel voice conversion using cycle-consistent adversarial networks. *2018 26th European Signal Processing Conference (EUSIPCO)*.

Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo. 2020. Cyclegan-vc3: Examining and improving cyclegan-vcs for mel-spectrogram conversion. In *Interspeech*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *CoRR*.

Tingle Li, Yichen Liu, Chenxu Hu, and Hang Zhao. 2020. Cvc: Contrastive learning for non-parallel voice conversion. In *Interspeech*.

Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. 2020. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*.

Bima Prihasto, Yi-Xing Lin, Phuong Thi Le, Chien-Lin Huang, and Jia-Ching Wang. 2023. Cneg-vc: Contrastive learning using hard negative example in non-parallel voice conversion. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2016. Instance normalization: The missing ingredient for fast stylization. *arXiv*.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2017. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. 2021. Early convolutions help transformers see better. In *Neural Information Processing Systems*.

Yamagishi, Junichi, Veaux, Christophe, MacDonald, and Kirsten. 2016. Superseded - cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. *CSTR*.

Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. 2019. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Wanfeng Zheng, Qiang Li, Guoxin Zhang, Pengfei Wan, and Zhongyuan Wang. 2022. Ittr: Unpaired image-to-image translation with transformers. *arXiv*.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)*.