



Non-Parallel Voice Conversion for ASR Augmentation

Gary Wang, Andrew Rosenberg, Bhuvana Ramabhadran, Fadi Biadisy, Yinghui Huang, Jesse Emond, Pedro Moreno Mengibar

Google¹

{wgary, rosenberg, biadisy, bhuv, huangyinghui, emond, pedro}@google.com

Abstract

Automatic speech recognition (ASR) needs to be robust to speaker differences. Voice Conversion (VC) modifies speaker characteristics of input speech. This is an attractive feature for ASR data augmentation. In this paper, we demonstrate that voice conversion can be used as a data augmentation technique to improve ASR performance, even on LibriSpeech, which contains 2,456 speakers. For ASR augmentation, it is necessary that the VC model be robust to a wide range of input speech. This motivates the use of a non-autoregressive, non-parallel VC model, and the use of a pretrained ASR encoder within the VC model. This work suggests that despite including many speakers, speaker diversity may remain a limitation to ASR quality. Finally, interrogation of our VC performance has provided useful metrics for objective evaluation of VC quality.

Index Terms: Voice Conversion, Automatic Speech Recognition

1. Introduction

It is critical for automatic speech recognition (ASR) to be robust to speaker differences. This is typically addressed by training on speech from a wide variety of speakers. In this paper, we show that despite being trained on 1000 hours of speech, augmenting speech with diverse speaker characteristics can improve speech recognition performance.

Voice conversion (VC) models convert input speech from its source speech to some target speaker, transferring the speaker timbre and other characteristics while retaining the lexical content of the source speech. While there has been successful approaches to voice conversion in recent years (Section 2), VC performance is typically evaluated on clean, in-domain speech. That is, the same style of speech used in training the VC model is used for evaluation. We have found that this can lead to VC systems that fail to generalize to speech and speakers that are substantially different than the training material (Section 5.3).

Since ASR training data is typically more diverse than VC training data, constructing a robust VC system is essential for ASR augmentation. ASR systems have to interact with speech from a variety of recording conditions and a broad range of speakers including speakers with less common accents.

In service of this robustness we make three important VC design choices. First, we pursue a non-parallel voice conversion approach (Section 3). By not requiring parallel data for training, we can train on much more data than we could otherwise. Second, we use a non-autoregressive decoder. We found that autoregressive models were susceptible to attention failures which were catastrophic for their application as a data augmentation technique. Third, we initialize the VC encoder with a pre-trained ASR encoder. This choice substantially improves the robustness of the VC model to diverse inputs (Section 5),

while moderately limiting VC naturalness. For ASR augmentation, we find that this substantial improvement to robustness is more important than a marginal improvement to naturalness.

The main contribution of this paper are as follows:

- We show that voice conversion can be used to successfully augment speech recognition training data. We find relative word error rate (WER) improvements up to 6% when augmenting LibriSpeech which contains 2,456 speakers.
- We show that we can substantially improve the robustness of voice conversion systems to unseen corpora and different speaker characteristics by initializing the VC encoder with an ASR encoder.
- We demonstrate VC metrics that can be used for model selection prior to subjective listening tests.

2. Related Work

Non-parallel voice conversion can be broadly categorized into two major approaches, the first using Phone posteriorgram (PPG) features [1, 2]. PPGs acts as speaker-invariant representations that can be easily utilized to achieve conversion. The other main approach is with auto-encoding style training approaches, utilizing both variational auto-encoder (VAE)[3, 4, 5, 6, 7, 8] and vector-quantized VAE (VQ-VAE) [9, 10, 11, 12, 13] approaches. The auto-encoding style training approaches try to disentangle speaker information from the content of source speech, using various methods from bottlenecking to adversarial training to prevent speaker leakage in the content or speech encoder. In [7], instead of adversarial training, carefully designed bottlenecks were coupled with auto-encoding loss to train the VC system. In [13], this is achieved by VQ-VAE of the speech encoder outputs.

Recent approaches have utilized pre-trained speech encoder to extract speaker agnostic representation. In [14], ASR trained to predict PPGs are used on source speech to achieve accent conversion. In [15], a trained ASR encoder is used as the speech encoder, along with explicit prosody modelling to achieve speaker and prosody in VC. In [16], accent and speaker disentanglement and control in VC was achieved in stages, where first stage is accent-agnostic ASR training, from which the trained speech encoder is used to further conduct VC training and disentangle accent and speaker.

Previous work on voice conversion and morphing augmentations to help augment ASR training include Vocal Tract Length Perturbation (VTLN) [17, 18]. More recent approaches use neural VC models to augment speech. In [19], a VC model is used to augment adult speech into children speech to help with children speech ASR. However in this work, the VC model was a Cycle-GAN[20] architecture trained on the same

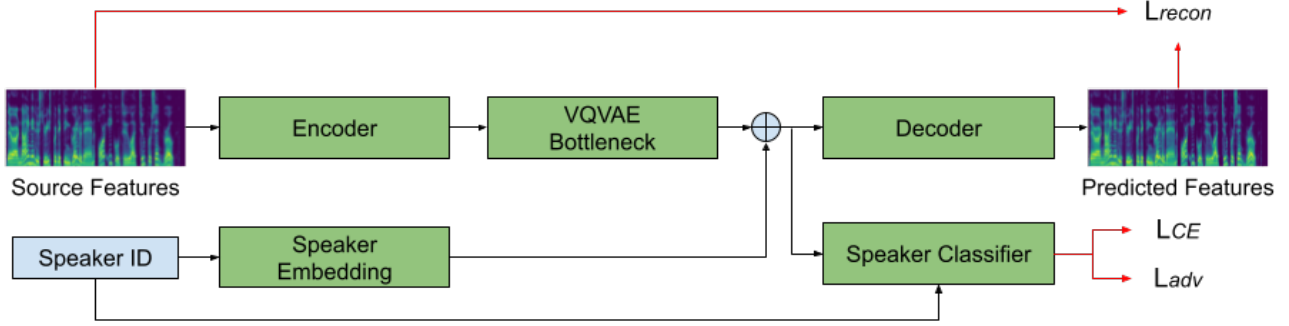


Figure 1: Schematic diagram of proposed voice conversion system.

in-domain data on which the ASR training was also conducted with. In [21], a VC system was utilized as augmentation for low resource languages. This system utilizes representations from a pretrained encoder with CPC[22] objective, instead of one trained on ASR targets. The VC model augments speech in the very low data regime.

3. Non-Parallel Voice Conversion

For our proposed VC model, we utilize an architecture consisting of an encoder, a bottleneck layer, and a non auto-regressive decoder. Figure 1 shows the architecture of the VC model.

Encoder: The VC encoder is similar to the encoder in the Conformer[23] ASR model. Specifically, the encoder consists of sub-sampling convolution layer that reduces the time dimension of the input speech by 4x. This is then followed by a stack of conformer layers (16 layers), followed by a output layer norm layer. We study two versions of encoders, the first being an conformer encoder that is trained from scratch jointly with the rest of the system. The second version being a frozen encoder, where the weights are obtained from an ASR model trained on 960 hours of Librispeech.

Speaker Embedding: Speaker information is provided in the form of one-hot vectors, which are passed through a learned speaker embedding that maps from one-hot vectors to a dense 256-dim embedding vector.

Decoder: The decoder is a non-autoregressive decoder composed of RNN layers followed by convolution layers. Speaker embedding is provided along with the encoder output to the decoder. Specifically, the dense speaker embedding vector is copied T times across time and then concatenated together with encoder outputs to be fed to the decoder. The decoder consists of 2 bi-directional LSTM layers of size 1024 followed by a stack of convolution layers. During development, we also explored an auto-regressive, attention-based decoder. However, we found that during inference even within-domain speech would sometimes show attention failures leading to extremely poor conversion [24]. While there are ways to mitigate these failures, since robustness is necessary for ASR augmentation we pursued the less error-prone, non-autoregressive approach.

VQVAE Bottleneck: A Vector Quantized Variational Auto Encoding (VQVAE) Bottleneck [25] is used to quantize encoder outputs into discrete codebook entries. This restricts information through the VC system during auto-encoding training. The VQVAE bottleneck typically contain 3 loss terms, the reconstruction loss, codebook loss and commitment loss.

$$L_{codebook} = \|sg[z_e(x)] - e\|_2^2 \quad (1)$$

$$L_{commit} = \beta \|z_e(x) - sg[e]\|_2^2 \quad (2)$$

Reconstruction loss is computed on the decoder, thus we simply add codebook and commitment loss from VQVAE to the overall loss objective during training. We fix VQVAE codebook size to 128 and group size to be 2.

Adversarial Speaker Classification: To disentangle speaker information, we apply adversarial speaker classification loss on the VQVAE codebook outputs to further ensure no speaker leakage. Given encoder outputs $m_{enc}(x)$ and speaker labels $l_{speaker}$, we combine reverse gradient operation R_{grad} along with softmax cross entropy loss as follows:

$$L_{adv} = CE(R_{grad}(m_{enc}(x)), l_{speaker}) \quad (3)$$

In effect, the training will try to maximize L_{adv} , which in effect will remove as much speaker information from the VQVAE output as possible. We scale adversarial gradient by weight, and find that this adversarial gradient weight term requires careful tuning, too small and it does nothing, too large and it overwhelms the reconstruction loss, see section 5.5 for details.

Objective Function We train on non-parallel data, and only utilize speech features and speaker embedding for the loss computation. To enable the VC models to be used as ASR data augmentation, we train our VC model to predict ASR log-mel features directly. During VC training, the same log-mel features are used as input and target. We use Huber loss for reconstruction loss. The model is jointly optimized by minimizing L_{total} , with all loss weights set to 1.0.

$$L_{recon} = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & \text{if } |(y - \hat{y})| < \delta \\ \delta((y - \hat{y}) - \frac{1}{2}\delta) & \text{otherwise} \end{cases} \quad (4)$$

$$L_{total} = L_{recon} + \gamma L_{codebook} + \epsilon L_{commit} + \eta L_{adv} \quad (5)$$

Frozen ASR Encoder In addition to training the full VC model (Figure 1) from scratch, we replace the encoder network with the encoder of an ASR model. ASR models implicitly normalize out speaker differences. ASR outputs are not (explicitly) dependent on speaker characteristics so they are less likely to be retained in the internal representations of the ASR network. We use an ASR encoder as the VC encoder, and freeze its weights, training only the decoder, VQVAE and speaker embedding. By freezing the encoder, it retains the representations learned during ASR training which leads to improved robustness.

4. Data

VC Corpora: Three TTS corpora were used for training the VC model. The first **LS** is the publicly available LibriSpeech Corpus containing 960 Hours of speech with 2,456 speakers [26]. The second **EN-VC** is an in-house corpus containing professional English speakers from 6 English locales, including United States (en/us), Britain (en/gb), India (en/in), Singapore (en/sg), Nigeria (en/ng) and Australia (en/au). This corpus contains 58 speakers, totaling around 400 hours of studio quality speech intended for Text-to-speech (TTS) training and research and hence was recorded at 48KHz. The third corpus **SV-VC** is, similar to the second, an in-house Swedish (sv/se) dataset collected for TTS training and research, which contains 21 hours of across 6 speakers. The data distribution between speakers is skewed, with speaker hours ranging from 0.5 hours to 40 hours. All datasets are downsampled to 16kHz, and we use 80-dim mel spectrum as features, with a frame size of 25ms and a 10ms frame shift.

ASR Corpora: For ASR, three training datasets were used in this work. The first is **LS** the public LibriSpeech[26] corpus. The other two datasets are in-house data sets comprising of anonymized short utterances representative of Google’s voice search (VS) traffic in two languages, English **EN-ASR** and Swedish **SV-ASR**. While the choice of English allows us to analyze two very different domains, namely, audio books and voice search with similar amounts of training data(1000 hours), the choice of Swedish(6500 hours) allows us to explore some transfer effects. While the **LS** corpus used in VC and ASR training is identical, the in-house VC (**EN-VC** and **SV-VC**) and ASR corpora (**EN-ASR** and **SV-ASR**) are completely disjoint.

5. Experiments & Results

5.1. VC Model Training

The VC model as described in Section 3 is trained with constant learning rate of $1e-4$ for 800k steps with Adam optimizer. Note that the VC model is trained on studio quality professional speech and has not seen any ASR training data. When using a frozen ASR encoder, the encoder is initialized from a LibriSpeech trained ASR encoder. For the LibriSpeech ASR experiments, the VC model is trained on Librispeech dataset. For in-house datasets **EN-ASR** and **SV-ASR**, we train VC models on our respective internal TTS datasets (Section 4).

For **LS-ASR** experiments, the VC model is trained on Librispeech data. For **EN-ASR** experiments, the VC model is trained on our internal en-us TTS data. For **SV-ASR** experiments, we train VC model on both our internal US and Swedish TTS datasets to have access to more speakers. Note, that the SV VC model is initialized with the LibriSpeech ASR encoder. This investigates the use of an ASR encoder as a proxy for a language agnostic representation for VC.

5.2. VC for ASR Augmentation

When using VC as an ASR augmentation technique, we follow [27, 28] for ASR training. Two views of the same data is presented to the ASR model, original and VC converted speech. Additionally a decoder consistency term is introduced to force model prediction to be consistent for the two views. We apply SpecAugment to both views. During VC inference, we uniform randomly sample speaker ids from the pool of **VC training** speakers whether **LS**, **EN-VC** or **SV-VC** to generate augmented versions of the ASR training data. No speaker information from

the ASR training data is used, nor is it available for **EN-ASR** or **SV-ASR**. The VC model is frozen during ASR training and used purely as an augmentation technique.

5.3. VC Quality Analysis

For analyzing VC quality, we train two models on **EN-VC**, one using a frozen ASR encoder (**ASR Encoder**) and one from scratch (**VC Encoder**). We train a separate WaveRNN neural vocoder for each VC model that maps from predicted log-mel features to 16kHz waveform. We analyze conversion quality with 3 measures, Mean Opinion Score Naturalness (MOS), Speaker Similarity, and Word Error Rate (WER) scored with in-house ASR system as a proxy for intelligibility. MOS and Similarity are rated on a 5 point Likert scale. For all evaluations, we convert from 5 example utterances from 20 source speakers to 5 target speakers, totaling 500 utterances.

Table 1: *MOS naturalness, Speaker Similarity and conversion word-error-rate for In-Domain, In-Locale Conversion*

Method	MOS	Similarity	WER
Groundtruth	4.347 ± 0.049	-	10.9
VC Encoder	4.001 ± 0.069	4.227 ± 0.059	14.4
ASR Encoder	3.611 ± 0.075	4.419 ± 0.051	16.0

In-Domain Conversion Our first analysis evaluates in-domain conversion, where source speakers and target speakers come from the same US English locale and corpus (**EN-VC**) (Table 5.3). When comparing VC using a trained encoder to the Frozen ASR encoder, we find a higher MOS and lower WER from the converted speech, but we find lower speaker similarity. This confirms the hypothesis that the frozen ASR encoder reliably eliminates speaker information from its output representation.

Cross-Locale Conversion To evaluate cross-local conversion, the source speakers consists of 5 non-US English locales (see Section 4), converting to 5 US English speakers. Since ASR training and evaluation data often includes a variety of accents, this evaluates how robust our VC system will be to differently accented speech.

Table 2 shows that both models result in less natural conversion (MOS) though the ASR Encoder mitigates this regression. Moreover, the ASR Encoder shows better robust speaker conversion (Similarity) and better WER.

Table 2: *MOS naturalness, Speaker Similarity and conversion word-error-rate for Cross-Locale Conversion*

Method	MOS	Similarity	WER
Groundtruth	4.347 ± 0.049	-	12.0
VC Encoder	3.344 ± 0.094	2.629 ± 0.086	39.4
ASR Encoder	3.281 ± 0.088	3.494 ± 0.098	34.7

Out-Of-Corpus Conversion For out-of-corpus conversion, the source speech consists of 20 random speakers from LibriTTS test-other data. The VC training data is **EN-VC**. Table 3 shows that when encountering out-of-domain input speech, the

VC Encoder system attains worse MOS score and much higher conversion WER as compared to the ASR Encoder system. We note that while the VC system was trained on **EN-VC**, the ASR Encoder has only seen LibriSpeech training utterances.

Table 3: *MOS naturalness, Speaker Similarity and conversion word-error-rate for Out-of-Corpus Conversion*

Method	MOS	Similarity	WER
Groundtruth	4.347 ± 0.049	-	10.4
VC Encoder	2.336 ± 0.087	2.549 ± 0.08	49.2
ASR Encoder	2.982 ± 0.088	4.068 ± 0.071	21.0

These three experiments demonstrate that the use of a pre-trained, frozen ASR encoder in VC leads to substantial improvements to VC robustness to accent and recording condition.

5.4. VC Augmentation & ASR Training

We now present our results of using VC as general ASR augmentation strategy across 3 datasets (Section 4): **LS**, **EN-ASR** and **SV-ASR**. Due to the low quality conversion of out-of-locale, and out-of-corpus sampled, all VC systems across 3 experiments use a LibriSpeech-trained **ASR encoder** as the VC encoder.

For **LibriSpeech** experiments, we use a conformer RNN transducer model as the ASR model. The ASR model consists of 16 layers of conformer layers in the encoder, each with conformer dimension of 16 and 4 attention heads. The decoder consists of a RNN transducer with RNN dimension of size 320. The system is trained for up to 120k steps with Adam optimizer, with annealed learning rate to $5e-4$. VC augmentation is described in section 5.2. Table 4 shows that VC augmentation improves WER by 0.2 and 0.1 absolute on test-clean and test-other. The LibriSpeech training data contains speech from 2,456 speakers, and still the augmentation provided by VC within a consistent data augmentation framework [27] is able to provide improved performance.

Table 4: *VC Augmentation Results on Librispeech*

Method	LS test	LS test-other
Baseline	3.0	6.8
+ VC Augmentation	2.8	6.7

For **EN-ASR** experiments, the ASR system consists of a Conformer [23] RNN transducer model of approximately 600 Million parameters. The ASR encoder consists of 24 layers of conformer layers each with 1024 dimension and 8 attention heads. The decoder consists of a RNN transducer with 2 Bi-directional LSTM layers each with 2048 dimension. The encoder has been pre-trained via self-supervised learning on LibriLight data along with unspoken text injection via tts as laid out in [29]. The baseline system is fine-tuned on **EN-ASR**.

The ASR model used in the Swedish ASR (**SV-ASR**) experiment is a Conformer transducer model of approximately 120 Million parameters. The encoder consists of 12 Conformer[23] layers, each with conformer dimension of 512 encoder dimension and 8 attention heads. The decoder consists of an embedding based decoder [30], with embedding dimension of 640 dimension. The model is trained with random initialization with

Adam optimizer for 300k steps, with a learning rate of $5e-4$. VC augmentation is utilized in the set up as described in section 5.2. The VC model is trained on both **EN-VC** and **SV-VC** data. When initializing the VC encoder with an ASR Encoder, this is the same LibriSpeech-trained ASR Encoder used in the **EN-ASR** models. This addresses the question of whether an out-of-language ASR model can be used within a VC model. This is akin to performing PPG Voice Conversion with a phone recognition model trained on a different language.

Table 5 shows VC augmentation to be an effective augmentation technique for **EN-ASR** fine-tuning, improving WER from 6.7% to 6.3%, while the improvement is smaller for **SV-ASR** at only 1%.

Table 5: *VC Augmentation Results on EN-ASR and SV-ASR*

Method	EN-ASR WER	SV-ASR WER
Baseline	6.7	13.7
+ VC Augmentation	6.3	13.6

5.5. VC Training Quality Metrics

Tuning weights for adversarial losses can be tricky, too small, they do nothing, too large, the main loss (here reconstruction) is overwhelmed. During our VC modelling development, we found two measures to be useful for tracking VC training performance. Table 5.5 shows VC training runs with different adversarial gradient weights. Also provided is the speaker classification accuracy, VQVAE codebook perplexity and the reconstruction loss. Based on these three measures, we identify good candidate VC models as those with low speaker accuracy, high VQVAE perplexity and low reconstruction loss. While not narrowly predictive of VC performance, these three measures were able to identify models that were performing VC well enough to be evaluated by subjective tests.

Table 6: *The effect of adversarial weight on metrics available during VC training.*

Speaker Adversarial Gradient Weight	Speaker Acc	VQVAE Perplexity	Reconstruction Loss
0.0	80%	110	0.038
0.1	12%	105	0.045
0.5	11%	80	0.070
1.0	10%	60	0.085

6. Conclusion

We demonstrate that even when trained on more than two thousand speakers, LibriSpeech ASR performance can be improved by Voice Conversion augmentation. We find similar ($\sim 6\%$) relative gains on an in-house English ASR task, but smaller improvements ($\sim 1\%$) on a comparable Swedish task. It is necessary for VC models to be robust to noisy inputs and diverse speakers to achieve these result. We show that using a pretrained ASR Encoder to serve as the encoder in a non-autoregressive, non-parallel VC model is a successful approach to promoting the necessary VC robustness.

7. References

- [1] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, "Phonetic posteriograms for many-to-one voice conversion without parallel data training," in *2016 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2016, pp. 1–6.
- [2] J.-X. Zhang, Z.-H. Ling, and L.-R. Dai, "Non-parallel sequence-to-sequence voice conversion with disentangled linguistic and speaker representations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 540–552, 2019.
- [3] 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 *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2016, pp. 1–6.
- [4] W.-C. Huang, H. Luo, H.-T. Hwang, C.-C. Lo, Y.-H. Peng, Y. Tsao, and H.-M. Wang, "Unsupervised representation disentanglement using cross domain features and adversarial learning in variational autoencoder based voice conversion," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 4, pp. 468–479, 2020.
- [5] P. L. Tobing, Y.-C. Wu, T. Hayashi, K. Kobayashi, and T. Toda, "Non-parallel voice conversion with cyclic variational autoencoder," *arXiv preprint arXiv:1907.10185*, 2019.
- [6] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Acvae-vc: Non-parallel voice conversion with auxiliary classifier variational autoencoder," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 9, pp. 1432–1443, 2019.
- [7] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "Autovc: Zero-shot voice style transfer with only autoencoder loss," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5210–5219.
- [8] K. Qian, Y. Zhang, S. Chang, M. Hasegawa-Johnson, and D. Cox, "Unsupervised speech decomposition via triple information bottleneck," in *International Conference on Machine Learning*. PMLR, 2020, pp. 7836–7846.
- [9] S. Ding and R. Gutierrez-Osuna, "Group latent embedding for vector quantized variational autoencoder in non-parallel voice conversion," in *INTERSPEECH*, 2019, pp. 724–728.
- [10] B. van Niekerk, L. Nortje, and H. Kamper, "Vector-quantized neural networks for acoustic unit discovery in the zerospeech 2020 challenge," *arXiv preprint arXiv:2005.09409*, 2020.
- [11] T. V. Ho and M. Akagi, "Non-parallel voice conversion based on hierarchical latent embedding vector quantized variational autoencoder," 2020.
- [12] H. Zhang, "The neteasegames system for voice conversion challenge 2020 with vector-quantization variational autoencoder and wavenet," *arXiv preprint arXiv:2010.07630*, 2020.
- [13] K. Kobayashi, W.-C. Huang, Y.-C. Wu, P. L. Tobing, T. Hayashi, and T. Toda, "crank: An open-source software for nonparallel voice conversion based on vector-quantized variational autoencoder," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5934–5938.
- [14] G. Zhao, S. Sosaat, J. Levis, E. Chukharev-Hudilainen, and R. Gutierrez-Osuna, "Accent conversion using phonetic posteriograms," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5314–5318.
- [15] Z. Wang, X. Zhou, F. Yang, T. Li, H. Du, L. Xie, W. Gan, H. Chen, and H. Li, "Enriching source style transfer in recognition-synthesis based non-parallel voice conversion," *arXiv preprint arXiv:2106.08741*, 2021.
- [16] Z. Wang, W. Ge, X. Wang, S. Yang, W. Gan, H. Chen, H. Li, L. Xie, and X. Li, "Accent and speaker disentanglement in many-to-many voice conversion," in *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2021, pp. 1–5.
- [17] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 9, pp. 1469–1477, 2015.
- [18] J. R. Bellegarda, P. V. de Souza, A. Nadas, D. Nahamoo, M. A. Picheny, and L. R. Bahl, "The metamorphic algorithm: A speaker mapping approach to data augmentation," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 3, pp. 413–420, 1994.
- [19] S. Shahnawazuddin, N. Adiga, K. Kumar, A. Poddar, and W. Ahmad, "Voice conversion based data augmentation to improve children's speech recognition in limited data scenario." in *Interspeech*, 2020, pp. 4382–4386.
- [20] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.
- [21] M. Baas and H. Kamper, "Voice conversion can improve asr in very low-resource settings," *arXiv preprint arXiv:2111.02674*, 2021.
- [22] T. A. Nguyen, M. de Seyssel, P. Rozé, M. Rivière, E. Kharitonov, A. Baevski, E. Dunbar, and E. Dupoux, "The zero resource speech benchmark 2021: Metrics and baselines for unsupervised spoken language modeling," *arXiv preprint arXiv:2011.11588*, 2020.
- [23] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.
- [24] J. Shen, Y. Jia, M. Chrzanowski, Y. Zhang, I. Elias, H. Zen, and Y. Wu, "Non-attentive tacotron: Robust and controllable neural tts synthesis including unsupervised duration modeling," 2020. [Online]. Available: <https://arxiv.org/abs/2010.04301>
- [25] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.
- [26] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.
- [27] G. Wang, A. Rosenberg, Z. Chen, Y. Zhang, B. Ramabhadran, Y. Wu, and P. Moreno, "Improving speech recognition using consistent predictions on synthesized speech," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7029–7033.
- [28] Z. Chen, A. Rosenberg, Y. Zhang, G. Wang, B. Ramabhadran, and P. J. Moreno, "Improving speech recognition using gan-based speech synthesis and contrastive unspoken text selection." in *INTERSPEECH*, 2020, pp. 556–560.
- [29] Z. Chen, Y. Zhang, A. Rosenberg, B. Ramabhadran, G. Wang, and P. Moreno, "Injecting text in self-supervised speech pretraining," *arXiv preprint arXiv:2108.12226*, 2021.
- [30] M. Ghodsi, X. Liu, J. Apfel, R. Cabrera, and E. Weinstein, "Rnn-transducer with stateless prediction network," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7049–7053.