Interspeech 2024
1-5 September 2024, Kos, Greece

Neural Codec Language Models for Disentangled and Textless Voice Conversion

Alan Baade[1], *Puyuan Peng*[1], *David Harwath*[1]

[1]The University of Texas at Austin, The United States of America

abaade@utexas.edu, pyp@utexas.edu, harwath@utexas.edu

Abstract

We introduce a method for textless any-to-any voice conversion based on the recent progress in speech synthesis driven by neural codec language models. To disentangle the speaker and linguistic information, we adapt a speaker normalizing procedure for discrete semantic units, and then generate with an autoregressive language model for greatly improved diversity. We further improve the similarity of the output audio to the target speaker's voice by leveraging classifier free guidance. We evaluate our techniques against current text to speech synthesis and voice conversion systems and compare the effectiveness of different neural codec language model pipelines. We demonstrate state-of-the-art results in accent disentanglement and speaker similarity for voice conversion with significantly less compute than existing codec language models such as VALL-E.

Index Terms: Voice Conversion, Neural Codec Lanugage Models, Textless Speech Processing

1. Introduction and Related Work

Recently, neural codec language models, which train a generative language model on discrete audio codes, have made significant breakthroughs in generative spoken language modeling [1, 2], music generation [3, 4], and zero-shot text to speech synthesis (TTS) [5, 6]. These improvements have been made in large part due to recent advances in both neural compression models [7, 8] and self-supervised speech models [9, 10]. We investigate building neural codec language models for the task of any-to-any voice conversion, which provides as input a source utterance and reference target speech with the goal of outputting an audio with the linguistic information of the source and speaker qualities of the target.

Cascaded ASR+TTS approaches for VC make models complex, expensive, can cause cascading errors, and sometimes we might want to preserve non-verbal vocalizations in the source speech such as laughter, breaths, sighs, filler words, etc. Textless voice conversion seeks to perform voice conversion without any text seen at training or inference. There are a variety of modern approaches for textless VC, such as using discrete codes for semantic disentanglement [11], pipelines with bottleneck layers [12], diffusion models [13, 14], and more recently closed-source neural codec language model approaches [15, 16]. However, we find that existing textless voice conversion models struggle to disentangle factors such as accent and prosody from the underlying phonetic and lexical content of the speech. We hypothesize this is due to the fact that textless systems by nature lack an abstracted, speaker-and-time-invariant representation of the speech such as a phonetic or text transcription.

Based on these issues, we contribute the following:

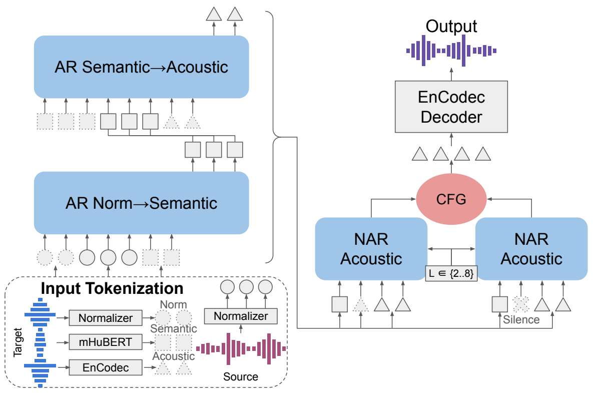

Figure 1: *Our model inference pipeline when using normalized units. Tokens containing information from the source have solid borders and tokens from the target prompting have dotted borders. Triangles represent acoustic tokens, squares represent semantic tokens, and circles represent normalized semantic tokens. The NAR section (right) is run once per quantizer layer.*

- We adapt neural codec langauge models to the task of any-to-any textless voice conversion and evaluate different model pipelines that trade off the balance between disentanglement and information loss.
- We demonstate that using semantic unit normalization, which has shown utility in textless speech to speech translation [17], is an effective way of removing entangled features such as accent from converted voices.
- We bring classifier free guidance from the diffusion model literature [18] to neural codec language models, greatly improving speaker similarity at little compute cost.

Code at https://github.com/AlanBaade/DisentangledNCLM

2. Method

Our method is depicted in Figure 1. Like VALL-E we use both autoregressive (AR) models and non-autoregressive (NAR) models to trade off between inference quality and speed at different stages. However, unlike VALL-E, our model does not require a text transcript of the speech to be converted, and it leverages multiple tokenizations of the speech signal in order to separately model higher level semantic information and lower level acoustic details.

2.1. Discrete Codes

For the task of voice conversion, we want audio representations that disentangle the semantic information of the audio from speaker information. We use three main classes of codes for generation: acoustic, semantic, and normalized semantic. Acoustic tokens contain fine-grained details about the audio, and are the final output of our language model pipeline. We leverage acoustic tokens from EnCodec [8] under the same settings as VALL-E, with 8 residual vector quantizers, providing one token per quantizer at 75Hz. Semantic tokens contain time-aligned linguistic information and have been shown to increase performance for audio continuations [2] and TTS [6]. Recent and concurrent work [4, 15] has also used semantic tokens for neural codec language model voice conversion. For semantic tokens, we follow [17] and use mHuBERT.

New to neural codec language models, we use speaker normalized semantic units as the coarsest input to our models. These units are meant to contain little to no speaker information and are not time-aligned. For textless voice conversion, we use the reduced-norm units and pre-trained model from [17]. These units are generated by having a reference speaker (such as a single-speaker TTS model) repeat utterances from a variety of speakers, totaling 10h of reference speech. A CTC model is then trained to predict the mHuBERT tokenization of the reference speech given the paired speech. These predictions are run-length-encoded, resulting in the final units, henceforth “norm.”

For our TTS replications, we follow VALL-E in using phones as a coarsest (normalized semantic) input. To generate these phones, we take a self-supervised HuBERT[10] model that was pretrained and finetuned on Librispeech, following the provided recipe of CTC-based HuBERT ASR finetuning.

2.2. Autoregressive Model

The AR model is a modification of a Transformer decoder [19] that takes as input coarse-grained tokens $x = \{x_1, \ldots, x_n\}$, such as normalized semantic units, and has the task of continuing finer-grained sequence of tokens $y = \{y_1, \ldots, y_m\}$, such as mHuBERT tokens. We insert a special stop token to represent the end of the fine-grained tokens, and learn separate embeddings for each token type. Because there is no autoregressive target on x_i, we do not apply a decoder attention mask to the tokens of x, allowing information to flow bidirectionally.

2.3. Non-Autoregressive Model

When modeling ground truth audio with residual vector quantizers such as EnCodec, deeper quantized layer outputs tend to carry less-important detail than earlier code layers. To speed up inference, VALL-E introduces an NAR model that assumes independence between tokens at later quantizer layers, predicting all timesteps for a given layer in parallel. As its input, the NAR model receives semantic tokens a of the source, as well as a fixed amount (3 seconds) of EnCodec tokens from all quantizer layers, b, to represent the target speaker. Then the NAR model predicts the output of the lth quantized token layer, $l \in \{2 \ldots L\}$ of EnCodecs $c^{T \times L}$ given the previous layers, modeling $p(c_{t,l} \mid a, b, c_{<l})$. Following the NAR stage, we take all EnCodec tokens generated and feed them into EnCodec's decoder to get an output waveform. To represent multiple quantization layers at the same timestep of ground truth audio, the NAR model sums the embeddings of all available layers at that token, each layer having its own embedding projection.

To better inform NAR models about what quantizer layer to output, VALL-E uses AdaLN [20] during layer normalization. We adopt AdaLN-Zero from [21] for this injection, which has been shown to outperform AdaLN at a negligible additional compute cost.

2.4. Inference

Inference for AR models is treated as in-context-learning where the model generates a continuation of a semantic input given a few seconds of audio from the target speaker. The model takes in coarse tokens from the source speech $x_{1..T_x}$ and reference target speech $z_{1..T_z}$ as well as fine tokens from the reference speech $y_{1..T_y}$. These inputs are concatenated as $\{x, z, y\}$ and the AR model then outputs y_{T_y+1}. This process repeats until a stop token is sampled.

Inference for the NAR stage is similar to training. Fine-grained semantic units are taken from either the source audio or an earlier AR stage, first-quantizer EnCodecs are taken as output from an AR model, and the speaker conditioning is taken from the target audio EnCodecs.

2.5. Classifier Free Guidance

We borrow classifier free guidance (CFG) [18] from the diffusion model literature to dramatically improve speaker similarity between our model's output and the target speaker across all neural codec language model implementations. CFG strengthens the speaker conditioning for output generation by reweighting the outputs of a conditioned and unconditioned model. We implement CFG in our NAR models by replacing the target speaker conditioning with silence, and perform guidance in the logit domain, where for CFG parameter ω, semantic context a, enrolled target speaker codes b, tokenized silence codes b', and previously computed encodec units $c < l$ we output:

$$c_{t,l} = \arg\max_i \Big\{ (1+\omega) \log\left(p(i_t | a, b, c_{<l})\right) - \omega \log\left(p(i_t | a, b', c_{<l})\right) \Big\} \qquad (1)$$

An important note is that in diffusion models, Classifier Free Guidance doubles the compute required at inference due to having both a conditioned and unconditioned forward pass. However, because the NAR phase is light and only runs per each output codebook, CFG runs at low cost.

3. Experiments

3.1. Datasets

Like [5], we train all of our models using the unlabeled 60k hour split of LibriLight [22]. Due to storage and compute constraints, we subsample LibriLight by randomly choosing at most one hour of audio from each speaker. This leaves us with 6k hours of audio from 7439 speakers. We crop speaker files uniformly randomly into chunks between 7 and 13 seconds long, chosen to include the evaluation statistics of VALL-E (4-10 seconds plus a 3-second prompt) while keeping sequences short for compute efficiency. To get enrolled EnCodec codes for the NAR model, we randomly select a file from the same speaker and use a 3-second crop.

We primarily evaluate using LibriSpeech to directly compare our numbers with closed-source prior work such as VALL-E and Spear-TTS. Specifically, we use all 4-10 second audio files in LibriSpeech test-clean as sources and pair each source with a randomly sampled target audio from a different speaker cropped to 3 seconds. We develop all hyperparameters on LibriSpeech dev-clean. We use VCTK as an auxillary dataset frequently cited in prior work to inspect the robustness of our models. Unexplored in prior work is inspecting voice conversion across multiple accents to evaluate the extent to which features other than voice can be disentangled. We augment LibriSpeech Test-Clean with accents from the diverse EdAcc dataset [23], unseen to all models during training, using the provided timestamps to expect single-speaker spans from conversations.

3.2. Model Pipelines

We create and test several model decoding pipelines to compare the necessity of different types of tokens. As input from the source audio, we compute either norm units, phone units, or skip immediately to mHuBERT units. From there, we generate a finer token, such as mHuBERT or first quantizer encodecs using an AR model. For the NAR model we always take in first quantizer EnCodec units and semantically condition using either phones, norm, or mHuBERT tokens, which we call P-NAR, N-NAR, and M-NAR respectively.

We evaluate three main pipelines. Ours-Aligned is our output voice conversion model using mHuBERT as input, generating first layer encodec tokens autoregressively and then using M-NAR for output, in short: mHuBERT$\rightarrow$Encodec 1$\rightarrow$M-NAR. Because mHuBERT features have a fixed 50hz sample rate, Ours-Aligned has a deterministic output length and phones in the output speech are time-aligned with those in the source audio. Depicted in Figure 1, Ours-Norm = Norm$\rightarrow$mHuBERT$\rightarrow$Encodec 1$\rightarrow$M-NAR is our textless model intended to further disentangle features by using normalized semantic units. These units allow for the first non-time-aligned textless voice conversion model, and unlock now disentanglement potential. Ours-Phone = Phones$\rightarrow$ mHuBERT$\rightarrow$Encodec 1$\rightarrow$P-NAR is our implementation of a textual baseline model.

3.3. Implementation Details

Like VALL-E, for both AR and NAR models we use a transformer architecture with 12 layers, an embedding dim of 1024, feed-forward dim of 4096, dropout of 0.1, sinusoidal position embeddings, and cross-entropy loss. Due to our compute budget, we train for 200k steps–1/4th of what VALL-E used, for each of our models. We warm up for 8k steps, hold for 92k steps, and linearly decay to 0 for the last 100k steps. All other hyperparameters match VALL-E. We train each model for 8 days on one 48GB Nvidia A40 using gradient accumulation. By default we use CFG during NAR decoding with $\omega = 5$. We implement our models using Fairseq [24].

During inference, our AR models with norm token inputs use a beam size of 10, like SPEAR-TTS, while phone input and mHuBERT input models use temperature sampling with $t = 0.75, t = 0.65$ respectively. For the NAR model, we use greedy sampling. During AR inference, we introduce a short period of silence at the end of the target audio to account for unnaturally cutting off speech mid-word when cropping to 3 seconds. We also find that preventing long repeating sequences of output tokens improves decoding.

4. Evaluation

We evaluate using objective and subjective metrics. We measure semantic content preservation objectively with word error rate (WER). We calculate WER using the publicly available checkpoint of HuBERT Large[1] finetuned on LibriSpeech 960h, the same model as VALL-E. We measure speaker similarity objectively using the speaker-verification model WavLM-TDNN[1] [25], the same as SPEAR-TTS. This model outputs speaker similarity (SPK) as a score between -1 and 1, 1 being most similar, and we take the mean across examples. Although SPEAR-TTS claims to use the same model as VALL-E, we find that VALL-E uses a non-publicly-available checkpoint with different results (SPEAR-TTS and VALL-E's models have 0.431 and 0.383 equal error rate respectively on Vox1-O).

For subjective evaluation, like prior work [26], we compute comparative quality scores for naturalness (CMOS) and Speaker Similarity (C-SMOS) on a 40-audio unique-speaker subset of our LibriSpeech Test-Clean objective data. We also introduce a new disentanglement metric, Dis-CMOS, which asks evaluators to rate and compare the extent to which the accent and cadence represent match target speaker. We evaluate this with the goal of disentangling speaker ID and accent, allowing them to be separately modelled and controlled. To the best of our knowledge, we are the first work to explicitly measure accent conversion for voice conversion and text-to-speech (AudioBox [27] uses an internal dataset with multiple accents, but only uses it to evaluate speaker similarity and naturalness). We restrict ourselves to only evaluate to and from American and UK-native accents to match current training datasets, and generate 40 audios. Subjectively, we do not observe meaningful out-of-domain zero-shot accent performance. We hope that future work can extend to lower-resource accents. We evaluate using MTurk and receive 5 reviews per sample.

4.1. Results

Tables 1 and 3 contain our objective evaluation. We compare against previous SOTA models in TTS and VC. For TTS we choose YourTTS, VALL-E and SPEAR-TTS. For VC we compare against TriAAN-VC [12], DiffVC [13], and FreeVC-s [28], the current state-of-the-art open source textless voice conversion models. UniAudio [16] and LM-VC [15] are recent closed-source neural codec language models that perform high quality voice conversion. Unfortunately, we find we are unable to fairly compare with the VC results from the UniAudio paper, as the provided numbers significantly outperform our replicated ground-truth different-sample-from-same-speaker baseline (Table 3). Therefore, we compare to the textful UniAudio model as a baseline instead.

We find that our models obtain SOTA results against available models for voice conversion in SPK on LibriSpeech. Our aligned textless model and textual phone model also outperform all models but FreeVC-s in WER, which has a very low speaker similarity. Ours-Norm has a significantly higher WER than other models, but outperforms our VALL-E reproduction (Ours-VALL-E) on equal data and compute. An increase in WER is expected because disentangling almost necessarily requires losing more information from the source. We leave creating different less lossy speaker normalized unit types to future work, with [29] being a promising direction. We notice that Ours-Phone significantly outperforms both VALL-E and our replication, directly demonstrating the importance of adding intermediate semantic units during training. Our models are effective across multiple datasets, outperforming TriAAN-VC on VCTK on speakers not seen during training, where Ours Norm shows a significantly smaller WER gap. We only compare against TriAAN-VC because the work has published evaluation splits and due the potential of overlapping train-test between other models on VCTK.

Table 2 contains subjective evaluations. We see striking results in terms of the performance of our normalized model

[1]github.com/microsoft/UniSpeech/tree/main/downstreams

improving accent disentanglement, matching or outperforming our TTS phone-based model which contains little-to-no source speaker accent information. This is followed by our time-aligned model, which still manages to significantly disentangle accent compared to prior work. We find that our models subjectively perform at par with voice conversion systems in terms of naturalness and similarity, with confidence intervals for CMOS and S-CMOS tighter than similar works.

Table 1: *Objective Evaluation on LibriSpeech Test-Clean. We use provided open-source implementations and weights for all models except VALL-E and SPEAR-TTS. No Test-Clean speakers are observed during training for any model.*

Type	Model	WER (%) ↓	SPK ↑
	Oracle Encodec	2.4	0.90
TTS	YourTTS [30]	8.5	0.46
	SPEAR-TTS [6]	-	0.56
	VALL-E [5]	5.9	0.58*
	Replication	14.1	0.46
	UniAudio-1B [16]	2.0*	**0.71**
VC	DiffVC [13]	7.5	0.33
	FreeVC-s [28]	**3.1**	0.20
	TriAAN-VC [12]	5.8	0.28
Ours	Ours-Aligned (VC)	5.4	0.58
	Ours-Norm (VC)	12.6	**0.61**
	Ours-Phone (TTS)	5.1	0.62

Table 2: *Subjective evaluation with 95% confidence intervals. CMOS and S-CMOS are evaluated on Librispeech Test-Clean. Dis-CMOS is evaluated on a mix of unseen speakers from the US and UK from EdAcc and LibriSpeech Test-Clean. Bold is best, overlapping confidence intervals are underlined.*

Model	CMOS	S-CMOS	Dis-CMOS
Diff-VC	<u>0.07</u> ±0.09	**0.10** ±0.09	-0.30 ±0.12
Free-VC-s	**0.12** ±0.09	-0.10 ±0.09	-0.51 ±0.12
TriAAN-VC	-0.06 ±0.08	-0.07 ±0.08	-0.12 ±0.13
Ours Aligned	-0.12 ±0.08	<u>0.01</u> ±0.09	0.12 ±0.14
Ours Norm	<u>0.02</u> ±0.08	<u>0.00</u> ±0.08	**0.52** ±0.15
Ours Phone	<u>-0.03</u> ±0.09	<u>-0.01</u> ±0.09	<u>0.30</u> ±0.14

Table 4 shows our evaluation of several different model pipelines for the purpose of better understanding where cascaded error comes from, without CFG. In our ground truth (GT) Encodec 1 experiments, we perform resynthesis to the same speaker because first quantizer encodec units contain large amounts of speaker information. We notice that when using GT mHuBERT units, M-NAR improves SPK during resynthesis but harms SPK when performing VC, implying the NAR model has picked up on speaker information within semantic units during training. This highlights the benefits of using normalized units, although adding CFG closes the gap significantly as seen in 1. We also notice that our Phones→Encodec 1→P-NAR model (VALL-E Replication in 1), a replication of VALL-E except for changes described in our methods, performs poorly compared to VALL-E. This demonstrates the importance of additional data and train time. Meanwhile, inserting semantic mHuBERT units as an additional step dramatically improves results across the board. Although this result has been implicitly seen across different works [5, 6], to the best of our knowledge this is the first experiment to explicitly control for other factors.

Table 3: *VCTK Objective Evaluation. We evaluate on 400 pairs of samples selected from the TriAAN-VC dev and test set, making all speakers unseen.*

Model	WER (%) ↓	SPK ↑
Ground Truth	4.7	0.63
Ours Aligned	**12.7**	0.44
Ours Norm	17.7	0.44
Ours Phone	13.8	**0.47**
TriAAN-VC	15.6	0.31

Table 4: *Ablations on different model pipelines, using notation defined in 3.2. Top: we resynthesize speech from GT first-quantizer-EnCodecs with each NAR model. Middle: we evaluate cascaded error in voice conversion using GT mHuBERT tokens as a semantic input. Bottom: We evaluate conversion from normalized units with different pipelines.*

Pipeline Types (no CFG)	WER	SPK
GT Encodec 1→P-NAR	2.9	0.67
GT Encodec 1→N-NAR	4.5	0.66
GT Encodec 1→M-NAR	3.3	0.68
GT mHuBERT→Encodec 1→P-NAR	3.9	0.49
GT mHuBERT→Encodec 1→N-NAR	6.2	0.48
GT mHuBERT→Encodec 1→M-NAR	5.0	0.43
Norm→Encodec 1→N-NAR	19.9	0.43
Phones→Encodec 1→P-NAR	14.1	0.46
Norm→mHuBERT→Encodec 1→M-NAR	12.6	0.46
Phones→mHuBERT→Encodec 1→P-NAR	4.5	0.48

Table 5 shows the importance of using CFG to improve speaker similarity output. We see that adding classifier free guidance dramatically raises speaker similarity at the cost of a slight increase in WER, similar to how CFG trades off Inception Score and FID in diffusion models [18].

Table 5: *CFG Ablation on LibriSpeech test-clean.*

Model	WER	SPK	WER+cfg	SPK+cfg
Ours Aligned	4.99	0.428	5.39	0.576
Ours Norm	12.60	0.461	12.61	0.607
Ours Phone	**4.55**	0.48	5.12	**0.615**

5. Conclusion

We introduce a novel neural codec language model that obtains state-of-the-art results in voice conversion. We also find that adding unit normalization and CFG allows us to more expressively navigate the tradeoffs between faithfulness to the source content and target audio. In future work we plan to investigate ways of better disentangling speaker, accent, and prosodic information so that they may be independently modified for more controllable voice conversion.

6. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. **2238605**

7. References

[1] K. Lakhotia, E. Kharitonov, W.-N. Hsu, Y. Adi, A. Polyak, B. Bolte, T.-A. Nguyen, J. Copet, A. Baevski, A. Mohamed, and E. Dupoux, "On generative spoken language modeling from raw audio," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1336–1354, 2021. [Online]. Available: https://aclanthology.org/2021.tacl-1.79

[2] D. V. E. K. Z. Borsos, R. Marinier *et al.*, "Audiolm: a language modeling approach to audio generation," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[3] C. Donahue, A. Caillon, A. R. E. Manilow, P. Esling *et al.*, "Singsong: Generating musical accompaniments from singing," 2023.

[4] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," 2023.

[5] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen *et al.*, "Neural codec language models are zero-shot text to speech synthesizers," 2023.

[6] E. Kharitonov, D. Vincent, Z. Borsos, R. Marinier, S. Girgin, M. S. O. Pietquin, M. Tagliasacchi, and N. Zeghidour, "Speak, read and prompt: High-fidelity text-to-speech with minimal supervision," 2023.

[7] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 495–507, 2021.

[8] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

[9] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[10] W.-N. Hsu, B. Bolte, Y.-H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. PP, pp. 1–1, 10 2021.

[11] A. Polyak, Y. Adi, J. Copet, E. Kharitonov, K. Lakhotia, W.-N. Hsu, A. Mohamed, and E. Dupoux, "Speech Resynthesis from Discrete Disentangled Self-Supervised Representations," in *Proc. Interspeech 2021*, 2021, pp. 3615–3619.

[12] H. Park, S. W. Yang, Seok, J. Kim, W. Shin, and S. Han, "Triaan-vc: Triple adaptive attention normalization for any-to-any voice conversion," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[13] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, M. S. Kudinov, and J. Wei, "Diffusion-based voice conversion with fast maximum likelihood sampling scheme," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=8c50f-DoWAu

[14] H.-Y. Choi, S.-H. Lee, and S.-W. Lee, "Dddm-vc: Decoupled denoising diffusion models with disentangled representation and prior mixup for verified robust voice conversion," *arXiv preprint arXiv:2305.15816*, 2023.

[15] Wang, Zhichao, Chen, Yuanzhe, Xie, Lei, Tian, Qiao, Wang, and Yuping, "Lm-vc: Zero-shot voice conversion via speech generation based on language models," *arXiv preprint arXiv:2306.10521*, 2023.

[16] Yang, Dongchao, Tian, Jinchuan, Tan, Xu, Huang, Rongjie, Liu, Songxiang, Chang, Xuankai, Shi, Jiatong, Zhao, Sheng, Bian, Jiang, Wu, Xixin *et al.*, "Uniaudio: An audio foundation model toward universal audio generation," *arXiv preprint arXiv:2310.00704*, 2023.

[17] A. Lee, H. Gong, P. Duquenne, H. Schwenk, P.-J. Chen, C. Wang, S. Popuri, Y. Adi, J. Pino, J. Gu, and W.-N. Hsu, "Textless speech-to-speech translation on real data," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 860–872. [Online]. Available: https://aclanthology.org/2022.naacl-main.63

[18] J. Ho and T. Salimans, "Classifier-free diffusion guidance," in *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. [Online]. Available: https://openreview.net/forum?id=qw8AKxfYbI

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[20] J. Xu, X. Sun, Z. Zhang, G. Zhao, and J. Lin, "Understanding and improving layer normalization," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[21] W. Peebles and S. Xie, "Scalable diffusion models with transformers," *arXiv preprint arXiv:2212.09748*, 2022.

[22] J. Kahn, M. Riviere, W. Zheng, E. Kharitonov, Q. Xu, P. Mazare, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux, "Libri-light: A benchmark for asr with limited or no supervision," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, May 2020. [Online]. Available: http://dx.doi.org/10.1109/ICASSP40776.2020.9052942

[23] R. Sanabria, N. Bogoychev, N. M. andrea Carmantini, O. Klejch, and P. Bell, "The edinburgh international accents of english corpus: Towards the democratization of english asr," *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257901049

[24] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[25] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[26] K. Shen, Z. Ju, X. Tan, Y. Liu, Y. Leng, L. He, T. Qin, S. Zhao, and J. Bian, "Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers," in *International Conference on Learning Representations*, 2024.

[27] A. Vyas, B. Shi, M. L. andros Tjandra, Y.-C. Wu, B. Guo, J. Zhang, X. Zhang, R. Adkins, W. Ngan, J. Wang, I. Cruz, B. Akula, A. Akinyemi, B. Ellis, R. Moritz, Y. Yungster, A. Rakotoarison, L. Tan, C. Summers, C. Wood, J. Lane, M. Williamson, and W.-N. Hsu, "Audiobox: Unified audio generation with natural language prompts," 2023.

[28] J. Li, W. Tu, and L. Xiao, "Freevc: Towards high-quality text-free one-shot voice conversion," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[29] 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," 2022.

[30] E. Casanova, J. Weber, C. Shulby, A. Junior, E. Gölge, and M. Ponti, "Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone," in *International Conference on Machine Learning*. PMLR, 2022, pp. 2709–2720.