HybridVC: Efficient Voice Style Conversion with Text and Audio Prompts

Xinlei Niu[1], *Jing Zhang*[1], *Charles Patrick Martin*[1]

[1]Australian National University

xinlei.niu@anu.edu.au, zjnwpu@gmail.com, charles.martin@anu.edu.au

Abstract

We introduce HybridVC, a voice conversion (VC) framework built upon a pre-trained conditional variational autoencoder (CVAE) that combines the strengths of a latent model with contrastive learning. HybridVC supports text and audio prompts, enabling more flexible voice style conversion. HybridVC models a latent distribution conditioned on speaker embeddings acquired by a pretrained speaker encoder and optimises style text embeddings to align with the speaker style information through contrastive learning in parallel. Therefore, HybridVC can be efficiently trained under limited computational resources. Our experiments demonstrate HybridVC's superior training efficiency and its capability for advanced multimodal voice style conversion. This underscores its potential for widespread applications such as user-defined personalised voice in various social media platforms. A comprehensive ablation study further validates the effectiveness of our method.

Index Terms: voice style, hybrid prompt, contrastive learning

1. Introduction

Voice conversion (VC) entails the process of modifying the vocal characteristics of a source speech to align with those of a target speaker. The key objective of VC is to alter the vocal identity, such as accent, tone, and timbre, to resemble the target speaker's voice style, while meticulously retaining the linguistic content of the source speech [1, 2, 3].

Traditional VC techniques rely on parallel training data [4, 5, 6, 7], which constrains their applicability for adapting to the voice styles of unseen speakers. To diversify the utilization of VC models, recent works focus on non-parallel any-to-any (A2A) VC models, so-called one-shot VC, that convert any speech to any voice style according to a few seconds of the target voice [8]. The key idea of A2A VC is learning disentangled feature representations of content and speaker style identity. To disentangle content and speaker information, existing works focus on methods including instance normalization [9], vector quantization [10, 11, 12], transfer learning from ASR and TTS models [13, 14, 15, 16], and adversarial training [9, 17, 18].

With the development of large language models (LLMs), leveraging natural language prompts for generating images [19], sound [20, 21, 22, 23], and speech [24, 25] has gained popularity. A notable development in VC is the introduction of voice style transfer using text prompts, as proposed by [26]. Building on this concept, [27] introduced Text-guidedVC, diverging from traditional VC techniques using natural language instructions, rather than an audio reference, to guide the VC process. This shift enhances the framework's flexibility and allows for a wider range of conversion styles by incorporating descriptors in natural language. Concurrently, PromptVC [28] proposed an A2A text prompt-based VC model, following VITS [29] structure, that employs a latent diffusion model (LDM) [30] to generate style vectors by text prompts.

We observe two limitations in Text-guidedVC [27]. Firstly, it relies on parallel training data, where effectiveness is heavily dependent on the size and quality of the dataset. This requirement can pose challenges, as accumulating extensive parallel data is often resource-intensive. Secondly, unlike conventional VC models that use audio reference, [28, 27] may struggle with achieving precise voice conversions. In practical scenarios, text descriptions may fall short in accurately capturing and conveying specific voice styles, a gap that is readily bridged when using actual target voice samples. This limitation is crucial, as it affects the framework's ability to replicate exact voice styles based solely on text guidance. Therefore, an A2A VC model with hybrid prompts can be developed to combine the strengths of flexibility of text guidance and precision of voice guidance.

VC models [31, 28, 17] based on VITS [29] framework may suffer from slow training due to decoding raw waveforms directly. A straightforward solution is to avoid the decoding step during training but maintain its advantages on performance. PromptVC [28] trains an LDM jointly with a VITS framework to learn textual information. Although [28] doesn't provide experimental setup details, jointly training an LDM and a CVAE with adversarial training requires a large volume of training data, powerful computational resources, and long training time that potentially limits its extension on practical applications. In image editing tasks, Imagic [32] proposes text embedding optimisation, which focuses on optimising text embeddings under pre-trained language models. This method offers a more flexible solution that avoids retraining the entire model, thereby providing an easier and more efficient way to extend the model through few-shot learning.

In this work, we introduce HybridVC, an efficient A2A VC model that supports text prompts and audio prompts of target style voice to achieve better flexibility. HybridVC is a latent model with contrastive learning, which models the latent distribution conditioned on speaker style information and optimises the alignment of corresponding text embedding in parallel. Our experiments prove HybridVC achieves significant training efficiency under limited computational resources and achieves a flexible VC system that supports hybrid prompts. HybridVC supports small-scale training which can be easily adapted to applications such as user-defined personalised voice.

2. Methodology

2.1. HybridVC

HybridVC, shown in Figure 1, is a conditional latent model that relies on a pre-trained CVAE backbone. We denote a text-

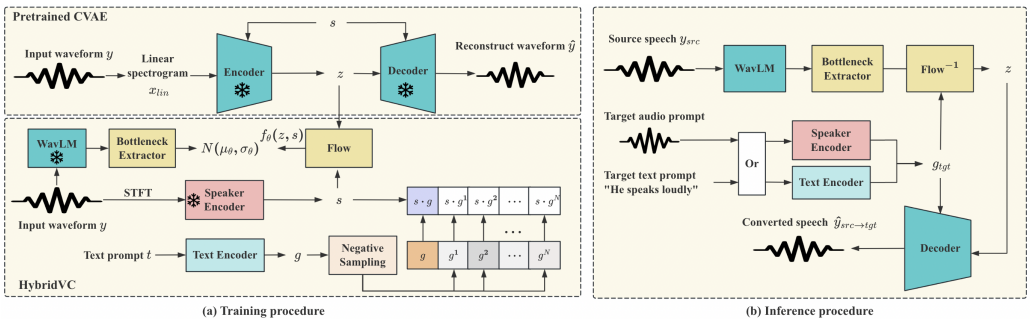

Figure 1: *Overview of HybridVC, where z represents the latent variable obtained by CVAE backbone, g is text embeddings, s is speaker style embeddings, and $\{g^1, ..., g^N\}$ are N negative samples. The snowflake symbol represents frozen parameters during training.*

speech pair (y, t), where y is an utterance and t is the corresponding style text description. Similarly to LDM [30], the first stage of HybridVC obtains a compact latent representation z from the pre-trained CVAE conditioned on speaker style information s by a pre-trained speaker encoder, as $z \sim q(z|x_{lin}, s)$. HybridVC models the distribution $p_\theta(z|y, s)$ conditioned on speaker embedding s and raw waveform y. Following [17], the architecture of HybridVC is straightforward, using a pre-trained WavLM [33] and a bottleneck extractor to disentangle text-free content information c from waveform y, a pre-trained speaker encoder to obtain speaker style information s, and a normalizing flow conditioned on s to transform the distribution to a more complex distribution. This method enables HybridVC to encapsulate the content c and speaker style information s within a compact latent space, facilitating efficient manipulation and generation of speech.

To enable HybridVC to support audio and text prompts, the model incorporates a text encoder to optimise a text embedding g through contrastive learning, as detailed in Section 2.2. The contrastive learning fine-tunes the text encoder, thereby, refining text embedding g to be well-aligned with speaker style embedding s obtained by the pre-trained speaker encoder.

CVAE backbone The CVAE backbone compresses the input waveform y into a latent variable z conditioned on a speaker-style embedding s. In our experiments, we use the posterior encoder and decoder of a pre-trained FreeVC [17] as the CVAE backbone which is augmented with GAN training.

Speaker Encoder The speaker encoder extracts the vocal identity (i.e., speaker style embedding s) from y. We follow FreeVC [17] that adopt a pre-trained verification model [34] as speaker encoder. The model is trained on a large amount of speakers. The speaker encoder embeds Mel-spectrogram of waveform y into a speaker style embedding s, as $s \in \mathbb{R}^{256}$.

Text Encoder The text encoder extracts stylistic features from text prompt t. We fine-tune a pre-trained text encoder to refine the text embeddings g, ensuring they are closely aligned with the corresponding speaker style information s. The text encoder contains a RoBERTa [35] model from a pre-trained CLAP [36] that encodes a text prompt t into an embedding $g \in \mathbb{R}^{512}$, and a linear layer further compress g to a lower feature dimension $g \in \mathbb{R}^{256}$ to discard irrelevant textual information.

2.2. PromptSpeech + Text Embedding Optimisation

We use the PromptSpeech dataset [37], which pairs over 26,000 real speech samples from LibriTTS [38] with free-text style prompts involving categories of gender, pitch, energy, and speed. However, text style prompts in PromptSpeech do not guarantee comprehensive coverage of information across these four categories. For instance, a prompt like "he said loudly" specifies gender and energy but not pitch or speed. This inconsistency presents a challenge to ensure comprehensive representations of all stylistic attributes, particularly when attempting to precisely map these free-form textual prompts to a predefined set of labels for speaker embedding s. To address this, we augment the data with N negative samples on text embeddings g extracted by the text encoder. Inspired by [39, 40], we propose a negative sampling method for text embeddings g that randomly shuffle g into N negative samples $G' = \{g^1, ..., g^N\}$ as in Figure 2. Given the ground truth text embedding g, we construct G' by dividing g into K randomly ordered chunks. As illustrated in Figure 3, HybridVC fine-tunes the text encoder through contrastive learning to optimise text embedding.

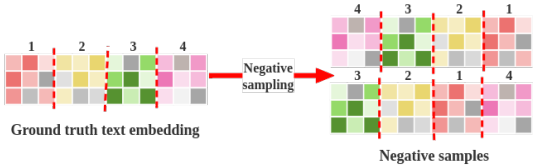

Figure 2: *Illustration of negative sampling for text embedding.*

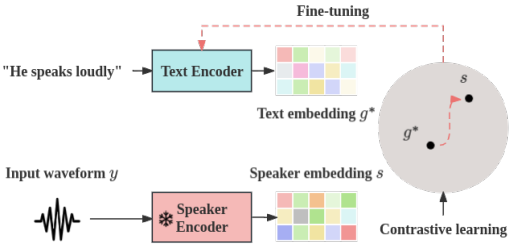

Figure 3: *Illustration of optimising the text embedding g^*.*

2.3. Training

As in Figure 1(a), the loss function of HybridVC contains two parts: a KL-divergence between the latent distributions of HybridVC and CVAE, and a contrastive loss between text embedding with negative samples $\{g, G'\}$ and speaker embedding s. Following [29], the KL loss $\mathcal{L}_{kl}$ between a Gaussian and normalising flow optimises the latent distribution of HybridVC,

$$\mathcal{L}_{kl} = \log q(z|x_{lin}, s) - \log p_\theta(z|y, s) \tag{1}$$

where f is the normalising flow, x_{lin} is the linear spectrogram,

$$p_\theta(z|y, s) = N(f_\theta(z, s); \mu_\theta, \sigma_\theta)|\frac{\partial f_\theta(z, s)}{\partial z}| \tag{2}$$

$$q(z|x_{lin}, s) = N(z; \mu_q, \sigma_q) \tag{3}$$

The contrastive loss $\mathcal{L}_{contrastive}$ is used to optimise the text encoder aligning the text embedding g with the speaker embedding s. Given a set of negative sample $G' = \{g^1, ..., g^N\}$, the

Table 1: *Model comparison measured on unseen source speeches with audio prompts from 9 unseen speakers.*

Model	Training Config.	Dataset	WER ↓	CER ↓	F_0-PCC ↑	SSIM ↑	FAD↓	FD ↓	NISQA ↑
FreeVC	Nvidia RTX3090	VCTK	13.56%	4.77%	0.719	0.781	1.15	0.71	4.51
FreeVC ⋆	2 Nvidia RTX3090 (12 days)	VCTK	**13.16**%	**4.60**%	**0.720**	0.771	0.89	0.71	4.34
YourTTS-VC	Nvidia TESLA V100 64G	VCTK	28.50%	12.45%	0.637	0.662	3.99	1.37	3.88
HybridVC ⋄	2 Nvidia RTX3090 (15 hours)	VCTK	14.01%	4.85%	0.716	**0.783**	0.87	**0.70**	4.51
HybridVC ⋄	2 Nvidia RTX3090 (17 hours)	PromptSpeech	18.18%	6.15%	0.701	0.771	**0.74**	0.81	4.51
HybridVC	2 Nvidia RTX3090 (25 hours)	PromptSpeech	17.54%	6.17%	0.712	0.775	0.79	0.75	**4.52**

contrastive loss is defined as

$$\mathcal{L}_{contrastive} = \log \frac{exp(cos(g, s)/\tau)}{\sum_{g_i \in \{g, G'\}} exp(cos(g_i, s)/\tau)} \quad (4)$$

where τ is a learnable temperature parameter.

The overall loss function $\mathcal{L}$ in HybridVC is defined as

$$\mathcal{L} = (1 - \alpha)\mathcal{L}_{kl} + \alpha\mathcal{L}_{contrastive} \quad (5)$$

where α is a weight-scaling hyper-parameter parameter.

2.4. Inference

As in Figure 1(b), given a source speech y_{src}, and a prompt embedding g_{tgt} obtained from either the speaker encoder or the text encoder, the inference phase of HybridVC is

$$\hat{y}_{src \to tgt} = Decoder(z, g_{tgt}) \quad (6)$$

where $z \sim p_\theta(z|y_{src}, g_{tgt})$.

3. Experiments and Results

Datasets We conduct experiments on VCTK [41] and PromptSpeech [37] datasets. PromptSpeech includes over 26000 real utterances from LibriTTS [38] with corresponding text style prompts, it is used for training HybridVC only. For VCTK, we follow the train/test separation as in [17] which involves 107 speakers in total, 314 utterances for validation, 1070 utterances for testing and the rest for training[1]. VCTK is used to train HybridVC for a model comparison experiment. The VCTK testing set is used to verify models' performance as source speeches.

Experimental setups All the audio samples are downsampled to 16kHz. The short-term Fourier transform (STFT) with a 1280 window size and 25% overlapping is performed to extract linear and 80-bin Mel-spectrograms. Following [17], we use SR-based data augmentation during training, where the resize ratio r ranges from 0.85 to 1.15. A HiFi-GAN v1 vocoder [42] is used for the data augmentation. The negative sample size N is 10 and the number of shuffle chunks K is 8. We use the posterior encoder and decoder of a FreeVC [17] trained on the VCTK training set as the CVAE backbone. All experiments were performed on 2 Nvidia RTX3090 GPUs. The model is trained with up to 80k training steps, where the batch size is 256 and initial learning rate is 2e-4. The RoBERTa model is initialized by weights from a CLAP checkpoint [2].

Models for comparison Since HybridVC is proposed to improve the flexibility and training efficiency compared to the VITS-like VC models, we select the following models for comparison[3]: 1) FreeVC [17], an official checkpoint trained with SR data argumentation and a pre-trained speaker encoder on the VCTK training set, 2) FreeVC⋆: a retrained 900k steps FreeVC with SR data argumentation and a pre-trained speaker encoder on the VCTK training set under the official training configuration[1], 3) YourTTS-VC [31][4], officially released checkpoint on VCTK trained with pre-trained speaker encoder.

Two versions of the proposed models are tested: 1) HybridVC⋄, the proposed model trained with KL loss $\mathcal{L}_{kl}$, and 2) HybridVC, the proposed model trained with the total loss $\mathcal{L}$.

Evaluation metrics We use nine objective evaluation metrics to verify the proposed method in diverse aspects including voice similarity, content error rates, speech naturalness, audio quality, and style prompt consistency. We calculate *word error rate* (WER) and *character error rate* (CER) between source and converted voice by an ASR model [5]. We verify the voice similarity by *Pearson correlation coefficient of fundamental frequency* (F_0-PCC) between source and converted voice, *structural similarity index measure* (SSIM) between target and converted voice. F_0-PCC measures the similarity of pitch contours between converted voices and source voices. We then make use of *speech quality and naturalness assessment* (NISQA) [43], *Fréchet audio distance* (FAD) [21] and *Fréchet distance* (FD) [21] to verify the naturalness of converted voice and audio quality. The NISQA is a deep learning framework to predict speech quality and naturalness [43], and FAD/FD measures audio quality. We calculate the cosine similarity (COS) between text embeddings extracted by the text encoder and audio embeddings of converted speeches extracted by the speaker encoder to verify the voice style consistency. Referring to [27], we calculate the classification accuracy between converted voice and source voice given single-factor text prompts. *Sample audios demonstration of HybridVC are provided in supplement.*

3.1. Speech Intelligibility, Naturalness and Audio Quality

Following experimental design in [17], we randomly select 9 unseen speakers as target voices and the test set of VCKT as source speeches. We aim to verify whether HybridVC improves training efficiency and maintains similar performance on audio prompt voice style conversion compared to baseline models, which only support audio prompt voice conversion. We train HybridVC⋄ and FreeVC⋆ on the VCTK training set under the same configuration. Table 1 shows that HybridVC can achieve competitive performance on speech intelligibility, naturalness, and audio quality with only 15 hours of training on limited computational resources. Extending the experiment to include training on the PromptSpeech dataset, HybridVC maintains overall performance without noticeable degradation, despite the backbone CVAE only being pre-trained on the VCTK training set. This underscores the robustness of HybridVC and highlights its capability to adapt to new datasets with minimal training resources and time, further explaining the advantage of

[1] https://github.com/OlaWod/FreeVC

[2] https://github.com/LAION-AI/CLAP

[3] Since the closest methods mentioned previously, Text-guidedVC [27] and PromptVC [28], do not release their implementation code and training datasets, we cannot use them for direct comparison.

[4] https://github.com/Edresson/YourTTS

[5] https://huggingface.co/facebook/hubert-large-ls960-ft

Table 2: *Consistency test for audio prompts and text prompts*

	F_0-PCC ↑	FAD ↓	SSIM ↑	COS ↑
Text prompts	0.69	0.79	\	0.68
Audio prompts	0.71	0.76	0.78	\

Table 3: *Accurcy of converted speech by given text prompts*

Text prompt	Higher pitch	Higher volume	Lower pitch	Lower volume
Accuracy	89.8 %	91.1%	60.5%	58.9%

HybridVC on extensions of practical application.

3.2. Consistency Test

We then conduct a consistency test between converted voice and style prompts for both audio and text prompts. We randomly select 9 unseen speakers as audio prompts and the VCTK testing set as source speeches to test consistency between converted voices and style audio prompts. To test the text-style voice conversion consistency, we randomly selected 10 speakers (5 female and 5 male) including 5 utterances for each speaker as source speech and 5 style text prompts from the PromptSpeech. As in Table 2, HybridVC effectively maintains the prosody of source speech and audio quality but accurately converts the voice characteristics given audio and text prompts. Text prompts typically offer less information compared to audio prompts, resulting in a slight drop in F_0-PCC and FAD values. The COS between converted voice and text prompt, and SSIM between converted voice and audio prompt indicate the style of the converted voice is consistent with the provided prompts.

We then explore the congruence between source and converted voices using single-factor text prompts. Figure 4 provides an example of a visualization comparison of a converted voice by a single-factor text prompt. In Table 3, we calculate the accuracy percentage between the absolute average value of the converted voices and the source voices according to the text prompts. For instance, we count the number of converted voices with higher pitch under the "higher pitch" text prompt by comparing the average pitch value between converted voices and source voices. Despite these prompts containing only single style factors, HybridVC successfully adapts voices to match the specified style text prompts. However, we observe that HybridVC demonstrates less sensitivity to text prompts that include words such as "lower", leading to a future improvement.

3.3. Ablation Study

Latent model We conduct an ablation study to verify whether the proposed latent model improves training efficiency while maintaining a similar performance as the baseline [17]. As shown in Table 1, HybridVC$\diamond$ has great training efficiency, achieving competitive performance in only 15 hours of training. Adding the negative sampling method to HybridVC increases

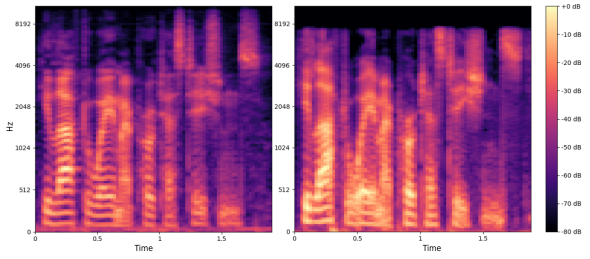

Figure 4: *Visualization of source speech (left) and converted speech with text prompt "he speaks loudly" (right).*

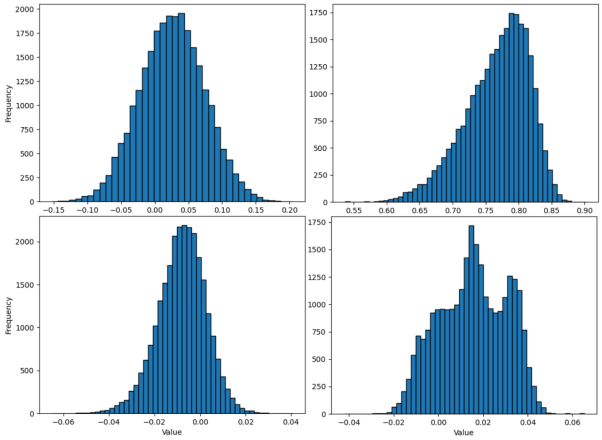

Figure 5: *Cosine similarity distribution before text embedding optimisation (left-top); after fine-tuned with proposed negative sampling method (right-top); after fine-tuned with negative samples within the batch (left-bottom); after fine-tuned with negative samples with manually classified labels (right-bottom).*

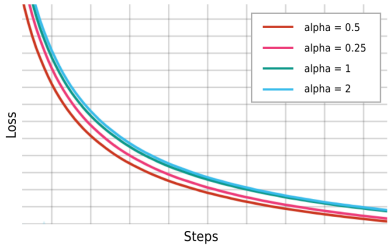

Figure 6: *Convergence analysis plot against* α.

the training time while maintaining HybridVC's performance. This presents a reasonable trade-off between reducing training efficiency and enabling support for flexible hybrid prompts.

Negative sampling method To verify the proposed negative sampling method, we train the HybridVC augmented with: 1) construct negative samples within a training batch as in [36]; 2) classify text embeddings into 54 labels across four categories in PromptSpeech. Figure 5 shows the cosine similarity between text embedding and speaker embedding before and after optimisation with different negative samples. Only HybridVC trained with the proposed negative sampling method effectively refines text embeddings well-aligned to speaker embeddings.

Hyper-parameter α We conduct convergence analysis against α values. Figure 6 shows that the loss convergence is not sensitive to α values although smaller α values (greater weighting on $\mathcal{L}_{kl}$) lead to slightly faster convergence.

4. Conclusion

We propose HybridVC, a flexible VC model supporting audio and text prompts that can be efficiently trained under limited computational resources. HybridVC is a latent model that models the latent distribution of a pre-trained CVAE backbone conditioned on speaker embeddings, and refines the text embedding to better align with information of speaker style in parallel. Our experiments show that HybridVC can achieve great training efficiency while maintaining robustness, flexibility, and performance compared to baseline models. Our model could be extended easily to applications, such as personalised voice on social media. However, as the first VC model supporting hybrid text/audio prompts, one limitation is that the optimised text embedding appears to be less sensitive to text prompts including words such as "lower". Future work could improve text-speaker embedding alignments by techniques such as prompt tuning.

5. References

[1] S. H. Mohammadi and A. Kain, "An overview of voice conversion systems," *Speech Communication*, vol. 88, pp. 65–82, 2017.

[2] K. Zhou, B. Sisman, R. Liu, and H. Li, "Seen and unseen emotional style transfer for voice conversion with a new emotional speech dataset," in *ICASSP*. IEEE, 2021, pp. 920–924.

[3] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *ICML*. PMLR, 2018, pp. 5180–5189.

[4] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE TASLP*, vol. 15, no. 8, pp. 2222–2235, 2007.

[5] S. Desai, E. V. Raghavendra, B. Yegnanarayana, A. W. Black, and K. Prahallad, "Voice conversion using artificial neural networks," in *ICASSP*. IEEE, 2009, pp. 3893–3896.

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

[7] J.-X. Zhang, Z.-H. Ling, L.-J. Liu, Y. Jiang, and L.-R. Dai, "Sequence-to-sequence acoustic modeling for voice conversion," *IEEE/ACM TASLP*, 2019.

[8] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "Autovc: Zero-shot voice style transfer with only autoencoder loss," in *ICML*. PMLR, 2019, pp. 5210–5219.

[9] H. J. Park, S. W. Yang, J. S. Kim, W. Shin, and S. W. Han, "Triaan-vc: Triple adaptive attention normalization for any-to-any voice conversion," in *ICASSP*. IEEE, 2023, pp. 1–5.

[10] D.-Y. Wu and H.-y. Lee, "One-shot voice conversion by vector quantization," in *ICASSP*. IEEE, 2020, pp. 7734–7738.

[11] D.-Y. Wu, Y.-H. Chen, and H.-Y. Lee, "Vqvc+: One-shot voice conversion by vector quantization and u-net architecture," *arXiv preprint arXiv:2006.04154*, 2020.

[12] D. Wang, L. Deng, Y. T. Yeung, X. Chen, X. Liu, and H. Meng, "Vqmivc: Vector quantization and mutual information-based unsupervised speech representation disentanglement for one-shot voice conversion," *arXiv preprint arXiv:2106.10132*, 2021.

[13] X. Zhang, Y. Gu, H. Chen, Z. Fang, L. Zou, L. Xue, and Z. Wu, "Leveraging content-based features from multiple acoustic models for singing voice conversion," *arXiv preprint arXiv:2310.11160*, 2023.

[14] Y. A. Li, C. Han, and N. Mesgarani, "Styletts-vc: One-shot voice conversion by knowledge transfer from style-based tts models," in *IEEE SLT*. IEEE, 2023, pp. 920–927.

[15] S. Hussain, P. Neekhara, J. Huang, J. Li, and B. Ginsburg, "Ace-vc: Adaptive and controllable voice conversion using explicitly disentangled self-supervised speech representations," in *ICASSP*. IEEE, 2023, pp. 1–5.

[16] J. Rekimoto, "Wesper: Zero-shot and realtime whisper to normal voice conversion for whisper-based speech interactions," in *CHI*, 2023, pp. 1–12.

[17] J. Li, W. Tu, and L. Xiao, "Freevc: Towards high-quality text-free one-shot voice conversion," in *ICASSP*. IEEE, 2023, pp. 1–5.

[18] W. Li and T.-J. Wei, "Asgan-vc: One-shot voice conversion with additional style embedding and generative adversarial networks," in *APSIPA ASC*. IEEE, 2022.

[19] S. Gu, D. Chen, J. Bao, F. Wen, B. Zhang, D. Chen, L. Yuan, and B. Guo, "Vector quantized diffusion model for text-to-image synthesis," in *CVPR*. IEEE/CVF, 2022, pp. 10 696–10 706.

[20] D. Yang, J. Yu, H. Wang, W. Wang, C. Weng, Y. Zou, and D. Yu, "Diffsound: Discrete diffusion model for text-to-sound generation," *IEEE/ACM TASLP*, 2023.

[21] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "Audioldm: Text-to-audio generation with latent diffusion models," *arXiv preprint arXiv:2301.12503*, 2023.

[22] R. Huang, J. Huang, D. Yang, Y. Ren, L. Liu, M. Li, Z. Ye, J. Liu, X. Yin, and Z. Zhao, "Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models," *arXiv preprint arXiv:2301.12661*, 2023.

[23] X. Niu, J. Zhang, C. Walder, and C. P. Martin, "Soundlocd: An efficient conditional discrete contrastive latent diffusion model for text-to-sound generation."

[24] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li *et al.*, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023.

[25] D. Yang, S. Liu, R. Huang, G. Lei, C. Weng, H. Meng, and D. Yu, "Instructtts: Modelling expressive tts in discrete latent space with natural language style prompt," *arXiv preprint:2301.13662*, 2023.

[26] E. Gölge and K. Davis, Feb 2022. [Online]. Available: https://archive.is/MotSP#selection-187.0-187.26

[27] C.-Y. Kuan, C. A. Li, T.-Y. Hsu, T.-Y. Lin, H.-L. Chung, K.-W. Chang, S.-y. Chang, and H.-y. Lee, "Towards general-purpose text-instruction-guided voice conversion," *arXiv preprint arXiv:2309.14324*, 2023.

[28] J. Yao, Y. Yang, Y. Lei, Z. Ning, Y. Hu, Y. Pan, J. Yin, H. Zhou, H. Lu, and L. Xie, "Promptvc: Flexible stylistic voice conversion in latent space driven by natural language prompts," *arXiv preprint arXiv:2309.09262*, 2023.

[29] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *ICML*.

[30] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *CVPR 2022*.

[31] 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 *ICML 2022*.

[32] B. Kawar, S. Zada, O. Lang, O. Tov, H. Chang, T. Dekel, I. Mosseri, and M. Irani, "Imagic: Text-based real image editing with diffusion models," in *CVPR*. IEEE/CVF, 2023.

[33] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, 2022.

[34] S. Liu, Y. Cao, D. Wang, X. Wu, X. Liu, and H. Meng, "Any-to-many voice conversion with location-relative sequence-to-sequence modeling," *IEEE/ACM TASPL 2021*.

[35] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *ICLR*, 2020.

[36] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *ICASSP*. IEEE, 2023.

[37] Z. Guo, Y. Leng, Y. Wu, S. Zhao, and X. Tan, "Prompttts: Controllable text-to-speech with text descriptions," in *ICASSP 2023*.

[38] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "Libritts: A corpus derived from librispeech for text-to-speech," *arXiv preprint arXiv:1904.02882*, 2019.

[39] Y. Zhu, Y. Wu, K. Olszewski, J. Ren, S. Tulyakov, and Y. Yan, "Discrete contrastive diffusion for cross-modal music and image generation," in *ICLR*, 2022.

[40] T. Lüddecke and A. Ecker, "Image segmentation using text and image prompts," in *CVPR*. IEEE/CVF, 2022, pp. 7086–7096.

[41] J. Yamagishi, C. Veaux, K. MacDonald *et al.*, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92)," 2019.

[42] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Neurips 2020*.

[43] G. Mittag, B. Naderi, A. Chehadi, and S. Möller, "Nisqa: A deep cnn-self-attention model for multidimensional speech quality prediction with crowdsourced datasets," *InterSpeech*, 2021.