Interspeech 2024
1-5 September 2024, Kos, Greece

DiffVC+: Improving Diffusion-based Voice Conversion for Speaker Anonymization

Fan Huang[1], *Kun Zeng*[1*], *Wei Zhu*[2]

[1]School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
[2]China Mobile Internet Co., Ltd., Guangzhou, China

huangf79@mail2.sysu.edu.cn, zengkun2@mail.sysu.edu.cn, 13802885090@139.com

Abstract

The increasing risks of speech data leakage prompt growing concerns about voice privacy. This paper proposes DiffVC+, a speaker anonymization model designed to preserve speech privacy. It operates as a diffusion-based voice conversion model that suppresses identity information by converting the speaker's voice through flexible approaches. DiffVC+ comprises a self-supervised learning (SSL) content encoder that effectively extracts the source speech content, a speaker encoder and an embedding generator that both supply the target speaker embedding, and a diffusion-based decoder generating the converted speech. Furthermore, we propose DiffVC+ *light* and DiffVC+ *decoupled* for edge-side and server-side deployments, respectively. Experimental results demonstrate that our models significantly outperform the baseline in terms of the intelligibility and naturalness of the converted speech, while achieving competitive anonymization performance.

Index Terms: speaker anonymization, voice conversion, diffusion probabilistic model

1. Introduction

Speech contains much sensitive personal information like age, gender, and ethnicity [1], making it a biometric identifier in human communication or automatic speaker recognition systems [2]. However, personal speech data is vulnerable to passive disclosure on the Internet, prompting demands for the preservation of speech privacy [3]. To alleviate this issue, the VoicePrivacy initiative [4] is spearheading the development of speaker anonymization solutions that suppress speaker identity information in speech while maintaining linguistic content, paralinguistic attributes, intelligibility, and naturalness. The requirement to leave other attributes intact makes voice conversion (VC) a preferred solution [5, 6], as it aims to convert the voice of the source speaker into the target speaker's voice without modifying the content of utterances [7].

As a speech synthesis task, voice conversion has employed diverse generative models, including variational autoencoder [8, 9], generative adversarial network [10, 11], autoencoder [12, 13], sequence-to-sequence model [14, 15] and so on. Recently, a flexible and powerful generative model named diffusion probabilistic model (DPM) [16] has emerged, showcasing its ability to model complex data distributions. The DPM consists of two iterative processes: the forward diffusion injecting noise into the original data to obtain a terminal distribution, and the reverse diffusion restoring the original data from the prior distribution. It has been demonstrated that the DPM can produce promising results in waveform generation [17] and text-to-speech synthesis [18]. In the context of voice conversion, the DPM has also been employed [19, 20, 21].

Among the diffusion-based VC models, DiffVC [21] has achieved impressive performance. To perform any-to-any VC, it embraces the concept of speech disentanglement. Specifically, DiffVC employs an "average voice" encoder to extract the content representation of the source speech, referred to as the average mel-spectrogram for brevity. It uses a diffusion-based decoder to generate the converted mel-spectrogram from the average mel-spectrogram, conditioned on the speaker identity information extracted from a reference utterance. With this design, DiffVC surpasses various existing generative models in any-to-any VC. Our purpose is to adapt DiffVC to the speaker anonymization task, for which any-to-any VC is the most matching setting. However, DiffVC still has some drawbacks for robust application: (1) The average voice encoder is trained on pairs of transcript and speech, imposing requirements on datasets and restrictions on languages. (2) The average mel-spectrogram is a coarse-grained (phoneme-level) feature. Using it to represent speech inevitably discards some linguistic content and prosodic information, making the converted speech less clear and natural.

In recent years, self-supervised learning (SSL) has achieved great success in speech processing [22, 23]. From massive unlabeled data, models can learn universal speech representations that benefit downstream tasks such as automatic speech recognition (ASR) and automatic speaker verification (ASV) [24]. Moreover, there is already evidence that SSL representations are beneficial for VC [14, 25]. In this paper, we propose DiffVC+, a speaker anonymization model that enhances DiffVC (our baseline) by incorporating SSL representations. DiffVC+ employs a pre-trained SSL model as a content encoder, obtaining finer-grained representations of speech. Simultaneously, its training does not rely on text. DiffVC+ then utilizes an ASV network as a speaker encoder, which extracts speaker embeddings as the identity information. To better fulfill the requirements of speaker anonymization, we introduce an embedding generator to generate non-existent speaker embeddings. Conditioned on the source speech content and the target speaker embedding, the decoder of DiffVC+ iteratively generates the converted mel-spectrogram from Gaussian distribution. Considering potential application scenarios, we further propose two variants, DiffVC+ *light* and DiffVC+ *decoupled*, equipped with a lightweight decoder and a novel-structure decoder, respectively. We will describe them in the following sections.

Our contributions can be summarized as follows:

(1) We propose a diffusion-based speaker anonymization model, DiffVC+, which leverages an SSL content encoder for accurate speech content capture. Then, we introduce an em-

* Corresponding author

This work is supported by China Mobile Internet Co., Ltd.

10.21437/Interspeech.2024-502

bedding generator to convert or anonymize the source speech without a reference utterance.

(2) DiffVC+ *light* uses a lightweight decoder and streaming inference for efficient edge-side deployment, and DiffVC+ *decoupled* adopts an estimator-controller decoder structure, controlling the pre-trained DPMs on servers to perform VC.

(3) Experimental results prove that our models can generate more intelligible and natural speech than the baseline. Additionally, they effectively deceive an advanced ASV model, achieving privacy preservation.

2. Methodology

As illustrated in Fig. 1, DiffVC+ follows an encoding-decoding inference procedure. Firstly, the SSL content encoder extracts content embeddings from a source utterance. Then, a target speaker embedding is obtained from the speaker encoder or the embedding generator. Subsequently, these embeddings are composed and input to the diffusion-based decoder to perform reverse diffusion, producing a converted mel-spectrogram. The remainder of this section describes each module in detail.

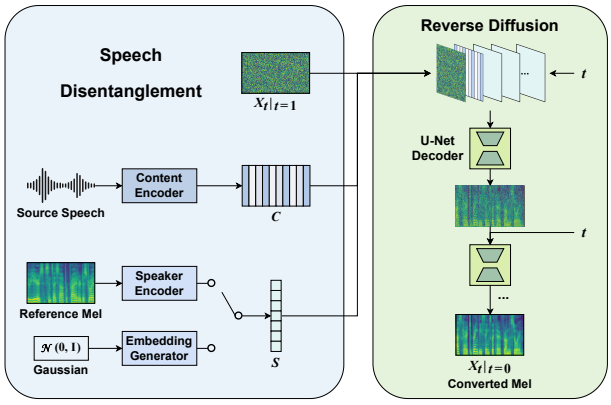

Figure 1: *The inference procedure of DiffVC+.*

2.1. Speech disentanglement

We disentangle content and speaker identity from speech, and the content of speech refers to both linguistic and paralinguistic attributes. To better represent the speaker-independent speech content, we utilize ContentVec [26] to implement the content encoder. It introduces disentangling mechanisms to remove speaker identity information while preventing the loss of content. The specific network of the content encoder has 7 temporal convolutional blocks followed by 12 transformer layers. Unlike previous works [14, 25] that use discrete "units", we directly use the continuous embeddings output by the 12th transformer layer, in our effort to preserve linguistic and paralinguistic attributes as much as possible. This choice is made because discretization still leads to some content loss [27]. Given an input speech waveform $\boldsymbol{N} = (n_1, \ldots, n_T)$, the content encoder outputs a sequence of 768-dimensional content embeddings $\boldsymbol{C} = \left(\boldsymbol{c_1}, \ldots, \boldsymbol{c_{T/\text{downsampling rate}}}\right)$ with reduced temporal resolution, where each $\boldsymbol{c_i}$ is an embedding. The embedding sequence is then upsampled to match the temporal resolution of mel-spectrograms through nearest-neighbor interpolation.

The target speaker's information for VC is represented by a speaker embedding, which can be obtained through two optional approaches. The first is through a pre-trained ASV network [28] used by the baseline, denoted as the speaker encoder. The speaker encoder extracts a 256-dimensional speaker embedding $\boldsymbol{S}$ from the mel-spectrogram of a reference utterance. The second is to use the embedding generator to generate a non-existent speaker embedding from Gaussian distribution, which we will explain in the next subsection.

2.2. Embedding generator

To handle the situation where the reference speech is limited or even not available, inspired by Yuan et al. [29], we introduce an embedding generator (EG) to generate target speaker embeddings. In particular, we pre-train a tiny variational autoencoder (VAE) on speaker embeddings extracted by the speaker encoder, and its decoder is precisely the EG. Both the encoder and decoder of this VAE are multilayer perceptrons (MLPs) with a single hidden layer containing 384 hidden units. During training, the encoding MLP maps a real speaker embedding $\boldsymbol{S}$ to a latent Gaussian distribution, outputting a mean vector $\boldsymbol{\mu}$ and a covariance vector $\boldsymbol{\sigma}$ (both 64-dimensional). Then, a latent vector $\boldsymbol{z}$ is sampled from this distribution, i.e., $\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Taking $\boldsymbol{z}$ as input, the decoding MLP produces a reconstructed speaker embedding, denoted as $\hat{\boldsymbol{S}}$. The loss function is designed as follows:

$$L = L_{\text{recon}} + \lambda L_{\text{cosdist}} + L_{\text{KL}} \tag{1}$$

$$L_{\text{recon}} = \begin{cases} \frac{1}{2}\|\boldsymbol{S} - \hat{\boldsymbol{S}}\|_1^2 & \text{if } \|\boldsymbol{S} - \hat{\boldsymbol{S}}\|_1 < 1 \\ \|\boldsymbol{S} - \hat{\boldsymbol{S}}\|_1 - \frac{1}{2} & \text{otherwise} \end{cases} \tag{2}$$

$$L_{\text{cosdist}} = 1 - \frac{\boldsymbol{S} \cdot \hat{\boldsymbol{S}}}{\|\boldsymbol{S}\|_2 \|\hat{\boldsymbol{S}}\|_2} \tag{3}$$

$$\begin{aligned} L_{\text{KL}} &= \text{KL}\left(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2) \,\|\, \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})\right) \\ &= \frac{1}{2}\sum_{i=1}^{64}\left(\mu_i^2 + \sigma_i^2 - \log(\sigma_i^2) - 1\right) \end{aligned} \tag{4}$$

where L_{recon} is the smooth L_1 loss [30] used for reconstruction, L_{cosdist} is the item preserving cosine similarity in the speaker embedding space, L_{KL} is the KL-divergence item of VAE, and λ is the weight coefficient set to 200. After training, the EG can generate non-existent speaker embeddings from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.

2.3. Diffusion probabilistic modeling

Following the baseline, we use stochastic differential equations (SDEs) to define the two processes of the DPM. The forward diffusion can be described as:

$$d\boldsymbol{X_t} = \frac{1}{2}\beta_t(\boldsymbol{\mu} - \boldsymbol{X_t})dt + \sqrt{\beta_t}d\overrightarrow{\boldsymbol{W_t}} \tag{5}$$

The reverse diffusion initiates from the terminal distribution of the forward diffusion, which can be described as:

$$d\hat{\boldsymbol{X}}_{\boldsymbol{t}} = \left(\frac{1}{2}(\boldsymbol{\mu} - \boldsymbol{X_t}) - s_\theta(\boldsymbol{X_t}, \boldsymbol{C}, \boldsymbol{S}, t)\right)\beta_t dt + \sqrt{\beta_t}d\overleftarrow{\boldsymbol{W_t}} \tag{6}$$

Here, $\boldsymbol{X_t}$ is the mel-spectrogram during the diffusion processes, where $t \in [0, 1]$. β_t is a noise schedule function following a linear schedule $\beta_t = \beta_0 + t(\beta_1 - \beta_0)$, where β_0 and β_1 are set to 0.05 and 20.0 respectively. $\overrightarrow{\boldsymbol{W_t}}$ and $\overleftarrow{\boldsymbol{W_t}}$ are two independent Wiener processes that simulate injecting noise. Assuming the original data $\boldsymbol{X_0} \sim p(\boldsymbol{X_0})$, the forward diffusion transforms this data distribution into $\boldsymbol{X_1} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I})$, where $\boldsymbol{\mu}$ is the mean of Gaussian prior. In the baseline, $\boldsymbol{\mu}$ is the encoded average mel-spectrogram, and we modify it to $\boldsymbol{0}$ to align with

our framework. s_θ is the decoder with parameters θ, which estimates the gradient of the log probability density with respect to data, i.e., the "score". The inputs of s_θ are from the same utterance during training.

The forward SDE (5) has fixed parameters, we can sample $\boldsymbol{X_t}$ by:

$$p(\boldsymbol{X_t}|\boldsymbol{X_0}) = \mathcal{N}\left(e^{-\frac{1}{2}\int_0^t \beta_r\, dr}\boldsymbol{X_0}, \left(1 - e^{-\int_0^t \beta_r\, dr}\right)\boldsymbol{I}\right) \quad (7)$$

To restore $\boldsymbol{X_0}$ from $\boldsymbol{X_1}$, the decoder solves the reverse SDE (6) starting from $\boldsymbol{X_1}$, and it is trained to minimize the weighted L_2 loss during the reverse diffusion:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \int_0^1 \lambda_t \mathbb{E}\|s_\theta(\boldsymbol{X_t}, \boldsymbol{C}, \boldsymbol{S}, t) - \nabla \log p(\boldsymbol{X_t}|\boldsymbol{X_0})\|_2^2\, dt \quad (8)$$

where $\lambda_t = 1 - e^{-\int_0^t \beta_r\, dr}$, $\nabla \log p(\boldsymbol{X_t}|\boldsymbol{X_0})$ is the score of distribution (7). The computation details of solving the SDE (6) can be found in the baseline's original paper.

2.4. Decoders and variants

As mentioned previously, we propose three models with different decoders: DiffVC+, DiffVC+ *light*, and DiffVC+ *decoupled*. Each of them will be explained individually.

2.4.1. DiffVC+

Our basic model, DiffVC+, utilizes the U-Net used by the baseline as the decoder. As illustrated in Fig. 2(a), the decoder contains 3 downsampling blocks, 1 middle block, 2 upsampling blocks, and 1 final convolutional block. These blocks, except the final one, are mainly built by residual blocks (ResBlocks, see Fig. 2(b)) and linear attention layers (LA layers, see Fig. 2(c)). Feature maps at three resolutions, $F \times T$, $F/2 \times T/2$, and $F/4 \times T/4$, are involved in the network. As a classic technique, skip connections concatenate the output of LA layers with the input of upsampling blocks at the same resolution. The decoder takes the concatenation of $\boldsymbol{X_t}$, $\boldsymbol{C}$, and 128 condition channels as input. The interpolated $\boldsymbol{C}$ is first transformed into an 80-dimensional embedding sequence through a 1D convolutional layer. The condition channels are obtained through the following procedure. Firstly, the timestep t is encoded into a sinusoidal positional encoding that is then transformed into a 256-dimensional timestep embedding through an MLP. The concatenation of the timestep embedding and $\boldsymbol{S}$ is further transformed by another MLP to get a 128-dimensional condition embedding, which is broadcast as the 128 condition channels. Moreover, the timestep embedding is also fed into each ResBlock.

2.4.2. DiffVC+ light

In some scenarios, speaker anonymization needs to be done before the speech data is uploaded to the Internet to further protect privacy or reduce latency. Thus, speaker anonymization models that can be deployed on edge devices are desirable and necessary. To enhance the applicability of our model on edge devices, inspired by Chen et al. [31], we propose a variant using a lightweight decoder, DiffVC+ *light*. Specifically, we retain the U-Net structure of the original decoder and replace regular convolutional layers with depth-wise separable convolutional layers to reduce parameters and computational complexity. As shown in Fig. 2(a)-(c), we mark a "$\star$" symbol on the right side of the replaced layers. In this way, we reduce the number of parameters from 117.17M to 26.38M and the training time by about one-third. Additionally, we implement streaming inference for DiffVC+ *light* to reduce memory usage and user-perceived latency. The entire $\boldsymbol{C}$ is partitioned into fixed-length chunks, which are sequentially fed into the decoder to yield mel-spectrograms. To mitigate the effects of chunking, we set slight overlaps between chunks and remove overlapping mel-spectrogram frames in the generated results.

2.4.3. DiffVC+ decoupled

Given the potential of SSL representations in full-stack speech processing [23], it is reasonable to anticipate the deployment of large pre-trained "content-to-speech" DPMs on servers. In this case, a speaker anonymization model can operate as a plug-in module that controls the generation process of pre-trained DPMs using speaker embeddings instead of being built from scratch. Thus, we propose DiffVC+ *decoupled* with a novel-structure decoder for potential server-side deployment. As illustrated in Fig. 2(d), the decoder follows a ControlNet [32] -style design, consisting of an estimator pre-trained for content-to-speech reconstruction and a controller that injects speaker conditions into the estimator. The estimator closely resembles DiffVC+'s decoder but excludes speaker embeddings from its input. The controller is a copy of the encoding blocks of the estimator, and each block of it is connected to the estimator using a zero-initialized 1×1 convolutional layer (zero convolutional layer), preventing noise injection during early training. The controller is initialized by parameters of the pre-trained estimator, which is frozen to preserve generation capability. The condition speaker embedding is transformed by an MLP and a zero convolutional layer to get a 128-dimensional embedding, which is broadcast and added to the condition channels. In this way, the controller gradually learns to control the estimator using speaker embeddings. Notably, DiffVC+ *decoupled* can perform speaker anonymization through unconditional generation. We can input the content embeddings of the source speech into the estimator, and it will implicitly sample a target speaker.

3. Experiment

3.1. Setup

We evaluate our models on VCTK [33] corpus, which includes 109 speakers reading about 400 English sentences. We hold out 5 males and 5 females as unseen speakers during training, along with 5 unseen sentences. All recordings are resampled to 22.05 kHz and used to compute 80-dimensional mel-spectrograms (window size 1024, hop size 256). HiFi-GAN [34] is used to invert mel-spectrograms to waveforms. We employ a released ContentVec[1] as the content encoder, which is pre-trained on LibriSpeech [35] corpus. For the baseline, we use its official average voice encoder[2] pre-trained on LibriTTS [36] corpus and best input type with extra information from mel-spectrograms of the target speaker. We use the speaker encoder and HiFi-GAN checkpoints provided by the baseline[2]. DiffVC+, DiffVC+ *light*, and the baseline are trained for 100 epochs with a batch size of 16 on the same dataset split, using an Adam optimizer with a learning rate of 1e-4. As for DiffVC+ *decoupled*, the controller is only trained for 50 epochs, and the estimator is pre-trained on LibriTTS for 50 epochs. The EG is trained on speaker embeddings extracted from 20 random utterances of each speaker in LibriTTS for 50 epochs, using an

[1] https://ibm.box.com/s/z1wgl1stco8ffooyatzdwsqn2psd9lrr

[2] https://github.com/huawei-noah/Speech-Backbones

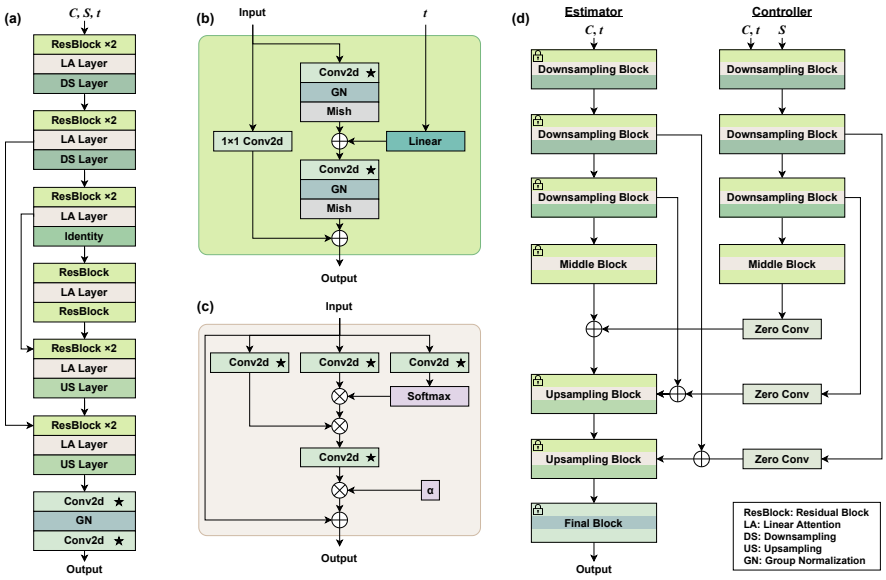

Figure 2: *(a) The decoder of DiffVC+ or DiffVC+ light. (b) ResBlock. (c) LA layer. (d) The decoder of DiffVC+ decoupled.*

Adam optimizer with a learning rate of 1e-3 and a batch size of 128.

For testing, we only consider unseen-to-unseen VC. We combine 10 unseen speakers and 5 unseen sentences to obtain 450 "source-target-sentence" triplets. We use the maximum likelihood sampling scheme proposed by the baseline for 30-step reverse diffusion. For DiffVC+ *light*, streaming inference is also performed. Then, we use the EG to generate a target speaker embedding for each speaker and synthesize 50 utterances. Finally, we use DiffVC+ *decoupled* to generate 50 more utterances unconditionally. Some samples are available online[3].

3.2. Objective evaluation

We evaluate all generated speech using the following metrics:

(1) Word error rate (WER). We utilize Whisper [37] to conduct ASR tests on the converted speech and compute WER with reference to both transcripts and the source speech.

(2) Pitch correlation coefficient (ρ^{F_0}). Considering prosody as the main paralinguistic attribute of speech, we use CREPE [38] to extract pitch contours and compute Pearson correlation coefficients between the converted and source speech.

(3) Speaker anonymization performance. We use ECAPA-TDNN [39] to conduct ASV tests. False rejection rate (FRR) and equal error rate (EER) are calculated, and EER is obtained by mixing the source speech with the converted speech of a single target speaker. Additionally, we compute the cosine distance (see Eq. (3)) between the converted and source speech, along with the cosine similarity between the converted and target speech, using the speaker encoder to extract embeddings.

(4) Automatic mean opinion score (AMOS). We use UTMOS [40] to predict MOSs of all generated speech, complementing the above metrics.

All evaluation results are shown in Table 1. Our models achieve much lower WERs than the baseline, significantly improving the intelligibility of the generated speech. The greater pitch correlation coefficients prove that our models can better preserve paralinguistic attributes and naturalness. The FRR and EER results indicate that our models can effectively deceive the advanced ASV model and protect speakers' privacy. As the baseline discards part of speech content, it reaches the greater cosine distance but lower cosine similarity. The AMOSs show that our models can generate speech with higher quality. When referring to the EG-generated embeddings, the converted speech is slightly less clear but further impairs the ASV model. Note that DiffVC+ *light* produces comparable results by performing streaming inference, and so does DiffVC+ *decoupled* through unconditional generation. Generally, our models can achieve competitive speaker anonymization performance without sacrificing the intelligibility and naturalness of speech, which can be realized through various means.

Table 1: *Evaluation results (streaming: streaming inference, uncond: unconditional generation).*

	WER (ref=txt/src)	ρ^{F_0}	FRR/EER	cosine sim/dist	AMOS
source	2.57/0.00	1.00	0.00/0.02	1.00/0.00	4.15
baseline	56.68/55.91	0.26	**0.99/0.37**	0.79/**0.37**	3.33
DiffVC+	**2.77/2.04**	0.69	**0.99**/0.30	0.79/0.35	3.87
_ *light*	3.23/2.36	**0.73**	0.96/0.30	**0.82**/0.33	3.92
_ _ streaming	3.06/2.40	0.66	0.97/0.31	0.71/0.34	3.63
_ *decoupled*	2.88/2.07	0.72	0.96/0.28	0.80/0.32	**3.98**
(w/ the EG)					
DiffVC+	**2.95/2.41**	0.68	**1.00**/0.28	- /**0.35**	3.93
_ *light*	3.21/2.61	**0.76**	0.98/0.30	- /0.32	4.02
_ *decoupled*	3.03/2.42	0.68	0.98/**0.33**	- /0.34	**4.04**
_ _ uncond	3.63/2.79	0.66	0.94/ -	- /0.32	3.90

4. Conclusion

In this paper, we propose DiffVC+, a diffusion-based speaker anonymization model that improves the baseline by introducing the SSL content encoder. It uses the EG to convert speech without reference utterances. We carefully design two variants, DiffVC+ *light* and DiffVC+ *decoupled*, advancing the deployment of our model on edge devices and servers. Preliminary evaluations demonstrate that our models effectively anonymize speech while preserving linguistic and paralinguistic attributes.

[3]https://huangf79.github.io/diffvc-plus-demo

5. References

[1] A. Nautsch, C. Jasserand, E. Kindt, M. Todisco, I. Trancoso, and N. Evans, "The gdpr & speech data: Reflections of legal and technology communities, first steps towards a common understanding," in *Interspeech 2019*, 2019, pp. 3695–3699.

[2] J. H. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[3] A. Nautsch, A. Jiménez, A. Treiber, J. Kolberg, C. Jasserand, E. Kindt, H. Delgado, M. Todisco, M. A. Hmani, A. Mtibaa *et al.*, "Preserving privacy in speaker and speech characterisation," *Computer Speech & Language*, vol. 58, pp. 441–480, 2019.

[4] N. Tomashenko, B. M. L. Srivastava, X. Wang, E. Vincent, A. Nautsch, J. Yamagishi, N. Evans, J. Patino, J.-F. Bonastre, P.-G. Noé, and M. Todisco, "Introducing the VoicePrivacy Initiative," in *Interspeech 2020*, 2020, pp. 1693–1697.

[5] B. M. L. Srivastava, N. Vauquier, M. Sahidullah, A. Bellet, M. Tommasi, and E. Vincent, "Evaluating voice conversion-based privacy protection against informed attackers," in *ICASSP 2020*. IEEE, 2020, pp. 2802–2806.

[6] G. P. Prajapati, D. K. Singh, P. P. Amin, and H. A. Patil, "Voice privacy through x-vector and cyclegan-based anonymization," in *Interspeech 2021*, 2021, pp. 1684–1688.

[7] B. Sisman, J. Yamagishi, S. King, and H. Li, "An overview of voice conversion and its challenges: From statistical modeling to deep learning," *TASLP*, vol. 29, pp. 132–157, 2020.

[8] 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 *APSIPA 2016*. IEEE, 2016, pp. 1–6.

[9] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Acvae-vc: Non-parallel voice conversion with auxiliary classifier variational autoencoder," *TASLP*, vol. 27, no. 9, pp. 1432–1443, 2019.

[10] T. Kaneko and H. Kameoka, "Cyclegan-vc: Non-parallel voice conversion using cycle-consistent adversarial networks," in *EUSIPCO 2018*. IEEE, 2018, pp. 2100–2104.

[11] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Stargan-vc: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *SLT 2018*. IEEE, 2018, pp. 266–273.

[12] 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.

[13] K. Qian, Z. Jin, M. Hasegawa-Johnson, and G. J. Mysore, "F0-consistent many-to-many non-parallel voice conversion via conditional autoencoder," in *ICASSP 2020*. IEEE, 2020, pp. 6284–6288.

[14] W.-C. Huang, Y.-C. Wu, and T. Hayashi, "Any-to-one sequence-to-sequence voice conversion using self-supervised discrete speech representations," in *ICASSP 2021*. IEEE, 2021, pp. 5944–5948.

[15] 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," *TASLP*, vol. 29, pp. 1717–1728, 2021.

[16] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *ICML*. PMLR, 2015, pp. 2256–2265.

[17] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "Wavegrad: Estimating gradients for waveform generation," in *ICLR*, 2021.

[18] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov, "Grad-tts: A diffusion probabilistic model for text-to-speech," in *ICML*. PMLR, 2021, pp. 8599–8608.

[19] H. Kameoka, T. Kaneko, K. Tanaka, N. Hojo, and S. Seki, "Voicegrad: Non-parallel any-to-many voice conversion with annealed langevin dynamics," *TASLP*, vol. 32, pp. 2213–2226, 2024.

[20] S. Liu, Y. Cao, D. Su, and H. Meng, "Diffsvc: A diffusion probabilistic model for singing voice conversion," in *ASRU 2021*. IEEE, 2021, pp. 741–748.

[21] 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 *ICLR*, 2022.

[22] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *TASLP*, vol. 29, pp. 3451–3460, 2021.

[23] 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," *JSTSP*, vol. 16, no. 6, pp. 1505–1518, 2022.

[24] S. wen Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin *et al.*, "Superb: Speech processing universal performance benchmark," in *Interspeech 2021*, 2021, pp. 1194–1198.

[25] M. Chen and Z. Duan, "Controlvc: Zero-shot voice conversion with time-varying controls on pitch and speed," in *Interspeech 2023*, 2023, pp. 2098–2102.

[26] K. Qian, Y. Zhang, H. Gao, J. Ni, C.-I. Lai, D. Cox, M. Hasegawa-Johnson, and S. Chang, "Contentvec: An improved self-supervised speech representation by disentangling speakers," in *ICML*. PMLR, 2022, pp. 18 003–18 017.

[27] B. van Niekerk, M.-A. Carbonneau, J. Zaïdi, M. Baas, H. Seuté, and H. Kamper, "A comparison of discrete and soft speech units for improved voice conversion," in *ICASSP 2022*, 2022, pp. 6562–6566.

[28] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. Lopez Moreno, Y. Wu *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," *NeurIPS*, vol. 31, 2018.

[29] R. Yuan, Y. Wu, J. Li, and J. Kim, "Deid-vc: Speaker de-identification via zero-shot pseudo voice conversion," in *Interspeech 2022*, 2022, pp. 2593–2597.

[30] R. Girshick, "Fast r-cnn," in *ICCV*, 2015, pp. 1440–1448.

[31] J. Chen, X. Song, Z. Peng, B. Zhang, F. Pan, and Z. Wu, "Lightgrad: Lightweight diffusion probabilistic model for text-to-speech," in *ICASSP 2023*. IEEE, 2023, pp. 1–5.

[32] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *ICCV*, 2023, pp. 3836–3847.

[33] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, "Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2017.

[34] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *NeurIPS*, vol. 33, pp. 17 022–17 033, 2020.

[35] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP 2015*. IEEE, 2015, pp. 5206–5210.

[36] 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," in *Interspeech 2019*, 2019, pp. 1526–1530.

[37] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *ICML*. PMLR, 2023, pp. 28 492–28 518.

[38] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," in *ICASSP 2018*. IEEE, 2018, pp. 161–165.

[39] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," in *Interspeech 2020*, 2020, pp. 3830–3834.

[40] T. Saeki, D. Xin, W. Nakata, T. Koriyama, S. Takamichi, and H. Saruwatari, "Utmos: Utokyo-sarulab system for voicemos challenge 2022," in *Interspeech 2022*, 2022, pp. 4521–4525.