Few-Step Diffusion-Based Voice Conversion Using Consistency Trajectory Models

Ryuichi Hatakeyama*, Toru Nakashika† and Takuya Takahashi‡
* The University of Electro-Communications, Tokyo
E-mail: r.hatakeyama@uec.ac.jp
† The University of Electro-Communications, Tokyo
E-mail: nakashika@uec.ac.jp
‡ The University of Electro-Communications, Tokyo
E-mail: takahashi@uec.ac.jp

***Abstract*—In this study, we propose a voice conversion method that replaces the decoder of DDPMVC with the consistency trajectory model (CTM), a form of diffusion model distillation. DDPMVC is a voice conversion (VC) framework that consists of an encoder that suppresses speaker information from the input speech through a forward diffusion process, and a decoder that generates the target speaker's mel-spectrogram via multi-step reverse diffusion. While this framework achieved high conversion accuracy, its major drawback is the long inference time due to the sequential nature of the sampling process. To address this issue, we introduce the CTM into the decoder of DDPMVC. CTM directly predicts the endpoint of diffusion inference and has demonstrated improved generation quality and faster sampling compared to conventional diffusion models. Experimental results show that the proposed method outperforms DDPMVC in voice conversion accuracy while reducing the sampling steps from thousands to just 10 steps.**

I. Introduction

Voice conversion (VC) is a technique that converts the speaker identity of a given utterance while preserving its linguistic content. VC methods are typically categorized by the type of training data: parallel data, consisting of utterances with identical content spoken by different speakers, and non-parallel data, which does not require such alignment. Depending on the conversion target, VC tasks include one-to-one, many-to-many, and any-to-many conversion, where arbitrary source speakers are converted to multiple predefined target speakers.

Recent advances in VC have been driven by deep learning frameworks such as variational autoencoders (VAEs) [1] and generative adversarial networks (GANs) [2]. Recently, diffusion-based VC methods [3], [4] emerged, demonstrating superior performance to previous deep learning-based VC approaches [5], [6]. For example, VoiceGrad [7] adopted a score-based generative model (SBM) [8] and a denoising diffusion probabilistic model (DDPM) to achieve non-parallel any-to-many voice conversion.

In non-parallel VC, it is commonly assumed that speech can be decomposed into speaker-independent (phonetic) information and speaker-dependent features. A typical approach involves extracting phonetic representations using an encoder and generating the target speech with a decoder conditioned on the target speaker label [9]. However, VoiceGrad performed sampling without an encoder and did not explicitly separate speaker information, which may have hindered conversion accuracy when the source and target speakers differed greatly.

To address this limitation, DDPMVC [10] was proposed. While adopting the sampling strategy of VoiceGrad, it introduced a rule-based diffusion process in the encoder to extract speaker-independent information. By using diffusion models in both the encoder and decoder, DDPMVC simplified the overall VC pipeline and reduced training costs. However, since the decoder still relies on iterative reverse diffusion, the inference requires multiple steps, resulting in slow generation. This undermines the advantage of a lightweight encoder, as decoding becomes the main bottleneck.

In this study, we propose a voice conversion method that integrates the consistency trajectory model (CTM) [11] into the decoder of DDPMVC to mitigate slow inference. CTM simultaneously learns the integrand (score function) and the integral (jump) of the probability flow ordinary differential equation (PF ODE), enabling direct generation of high-quality mel-spectrograms in just a few steps. Unlike conventional distillation methods, CTM achieves more accurate generation with fewer steps and does not suffer degradation as sampling steps increase. This reduces the required steps from about 1000 in conventional reverse diffusion to only a few, while preserving naturalness and speaker similarity in any-to-many VC. Since CTM was originally proposed for computer vision, applying it to speech presents challenges. In particular, the consistency loss relies on LPIPS [12] as a feature distance, which is designed for image-based similarity and is not suitable for speech. We propose to replace this with mel-cepstral distortion (MCD), a widely used similarity metric in voice conversion that captures differences in spectral envelope to assess speaker characteristics. Unlike LPIPS, MCD does not rely on a pretrained model, reducing training costs significantly. In our experiments, we distill CTM as a student model from DDPMVC as the teacher. Results show that CTM outperforms DDPMVC in conversion accuracy with only a fraction of the steps.

979-8-3315-7206-8/25/$31.00

II. PRELIMINARY

A. Diffusion Models

The denoising diffusion probabilistic model (DDPM) [4] consists of two processes: a forward diffusion process that gradually adds noise to a data sample $\boldsymbol{x}_0$, and a reverse diffusion process that removes the added noise. By treating the noisy samples as latent variables, DDPM performs maximum likelihood estimation in a latent variable modeling framework, generating data by sequentially sampling from the reverse diffusion process. Song *et al.* [13] extended discrete-time DDPMs into a continuous-time formulation using stochastic differential equations (SDEs). The diffusion SDE is represented as:

$$d\boldsymbol{x}_t = \boldsymbol{\mu}(\boldsymbol{x}_t, t)dt + \sigma(t)d\boldsymbol{w}_t, \tag{1}$$

where $t \in [0, T]$ with $T > 0$, and $\boldsymbol{\mu}(\cdot, \cdot)$, $\sigma(t)$, $\boldsymbol{w}_t$ and $d\boldsymbol{w}_t$ denote the drift coefficient, the diffusion coefficient, a standard Wiener process (Brownian motion) and a Gaussian random variable with mean 0 and variance τ over an infinitesimal time interval τ, respectively. This SDE can be converted into an ordinary differential equation, i.e., probability flow ODE (PF ODE), as:

$$d\boldsymbol{x}_t = \left[\boldsymbol{\mu}(\boldsymbol{x}_t, t) - \frac{1}{2}\sigma(t)^2 \nabla \log p_t(\boldsymbol{x}_t)\right] dt, \tag{2}$$

where $\nabla \log p_t(\boldsymbol{x}_t)$ is the score function. Song *et al.* proposed the variance preserving (VP) SDE, where they used $\boldsymbol{\mu}(\boldsymbol{x}, t) = -\frac{1}{2}\beta(t)\boldsymbol{x}$ and $\sigma(t) = \sqrt{\beta(t)}$ [13]. Karras *et al.* then proposed an alternative setting with $\boldsymbol{\mu}(\boldsymbol{x}, t) = 0$ and $\sigma(t) = \sqrt{2t}$ [14], which is the primary setting used in this study. Under this formulation, the following empirical PF ODE can be derived:

$$\frac{d\boldsymbol{x}_t}{dt} = -t\nabla \log p_t(\boldsymbol{x}_t) = \frac{\boldsymbol{x}_t - \mathbb{E}_{p_{t0}(\boldsymbol{x}|\boldsymbol{x}_t)}[\boldsymbol{x}|\boldsymbol{x}_t]}{t}. \tag{3}$$

The term $\mathbb{E}_{p_{t0}(\boldsymbol{x}|\boldsymbol{x}_t)}[\boldsymbol{x}|\boldsymbol{x}_t] = \boldsymbol{x}_t + t\nabla \log p_t(\boldsymbol{x}_t)$ denotes the denoiser function, which serves as an alternative representation for the score function. In this case, $p_t(\boldsymbol{x}) = p_{\text{data}}(\boldsymbol{x}) \otimes \mathcal{N}(\mathbf{0}, t^2\mathbf{I})$, where $p_{\text{data}}(\boldsymbol{x})$, $\otimes$, and $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denote the data distribution, the convolution operation, and Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, respectively. The denoiser $\mathbb{E}[\boldsymbol{x}|\boldsymbol{x}_t]$ is approximated by a neural network $D_{\boldsymbol{\phi}}$ with parameters $\boldsymbol{\phi}$ trained to minimize the denoising score matching (DSM) loss:

$$\mathbb{E}_{\boldsymbol{x}_0, p_{0t}(\boldsymbol{x}|\boldsymbol{x}_t)}[||\boldsymbol{x}_0 - D_{\boldsymbol{\phi}}(\boldsymbol{x}_t, t)||_2^2], \tag{4}$$

where $p_{0t}(\boldsymbol{x}|\boldsymbol{x}_t)$ denotes the transition probability from time 0 to t starting from $\boldsymbol{x}_0$. When using an approximated denoiser, the empirical PF ODE becomes:

$$\frac{d\boldsymbol{x}_t}{dt} = \frac{\boldsymbol{x}_t - D_{\boldsymbol{\phi}}(\boldsymbol{x}_t, t)}{t}. \tag{5}$$

Sampling in diffusion models corresponds to solving this PF ODE, which is equivalent to computing the following integral:

$$\int_T^0 \frac{d\boldsymbol{x}_t}{dt}dt = \int_T^0 \frac{\boldsymbol{x}_t - D_{\boldsymbol{\phi}}(\boldsymbol{x}_t, t)}{t}dt \quad \Longleftrightarrow$$
$$\boldsymbol{x}_0 = \boldsymbol{x}_T + \int_T^0 \frac{\boldsymbol{x}_t - D_{\boldsymbol{\phi}}(\boldsymbol{x}_t, t)}{t}dt. \tag{6}$$

Generation processes in diffusion models are generally categorized into two types: score-based sampling [8], [14]–[16] and distillation-based sampling [17], [18]. Score-based sampling solves the ODE using numerical integration over discretized time steps, which typically leads to long generation times. Improving sample quality under a low number of function evaluations (NFE) remains a challenge. Distillation-based sampling, in contrast, trains a neural network to directly approximate the above integral, enabling single-step generation. However, as NFE increases, the sample quality tends to degrade.

B. consistency trajectory model

The CTM [11] serves as a bridge between score-based models (SBMs) and distilled models by simultaneously estimating both the integrand and the integral of the PF ODE. CTM enables jump estimation over arbitrary time intervals along the PF ODE trajectory, supporting both infinitesimal jumps (as in score functions) and long-range transitions (as in integral over any time horizon). This makes CTM capable of flexible anytime-to-anytime generation. Consequently, the solution to (6) can be formulated as:

$$G(\boldsymbol{x}_t, t, s) = \frac{s}{t}\boldsymbol{x}_t + \left(1 - \frac{s}{t}\right) g(\boldsymbol{x}_t, t, s) \tag{7}$$
$$g(\boldsymbol{x}_t, t, s) = \boldsymbol{x}_t + \frac{t}{t-s}\int_s^t \left(\frac{\boldsymbol{x}_u - \mathbb{E}[\boldsymbol{x}|\boldsymbol{x}_u]}{u}\right) du,$$

where $G(\boldsymbol{x}_t, t, s)$ denotes the solution of the PF ODE from initial time t to terminal time $s \leq t$. To stabilize training, G is formulated as a linear interpolation between $\boldsymbol{x}_t$ and the function g.

Let θ denote the parameters of the student model. To match the neural network prediction $G_{\boldsymbol{\theta}}$ (neural jump) with the true jump G, we train $G_{\boldsymbol{\theta}}$ by comparing it with the output of a pretrained numerical solver $\text{Solver}(\boldsymbol{x}_t, t, s; \boldsymbol{\phi})$, which solves the PF ODE defined in (5) and serves as the teacher model. As shown in Eq. (8), the jump prediction comprises the teacher and the student: the teacher solves the PF ODE over the interval (u, t) and jumps to the output of the stop-gradient student.

$$G_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, s) \approx G_{\text{sg}(\boldsymbol{\theta})}\left(\text{Solver}(\boldsymbol{x}_t, t, u; \boldsymbol{\phi}), u, s\right) \tag{8}$$

Here, $\text{sg}(\boldsymbol{\theta})$ denotes the stop-gradient operator, typically implemented via exponential moving average (EMA) for stabilization.

To quantify the mismatch between the student prediction $G_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, s)$ and the teacher's reference output $G_{\text{sg}(\boldsymbol{\theta})}\left(\text{Solver}(\boldsymbol{x}_t, t, u; \boldsymbol{\phi}), u, s\right)$, CTM uses a feature distance $d(\cdot, \cdot)$ computed in the clean data space. Specifically, both predictions at time s are mapped to time 0 via the stop-gradient student $G_{\text{sg}(\boldsymbol{\theta})}(\cdot, s, 0)$. Let the student-estimated sample be: $\boldsymbol{x}_{\text{est}}(\boldsymbol{x}_t, t, s) := G_{\text{sg}(\boldsymbol{\theta})}\left(G_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, s), s, 0\right)$ and the target output from the teacher be: $\boldsymbol{x}_{\text{tgt}}(\boldsymbol{x}_t, t, u, s) := G_{\text{sg}(\boldsymbol{\theta})}\left(G_{\text{sg}(\boldsymbol{\theta})}\left(\text{Solver}(\boldsymbol{x}_t, t, u; \boldsymbol{\phi}), u, s\right), s, 0\right)$. The CTM loss is defined as:

$$\mathcal{L}_{\text{CTM}}(\boldsymbol{\theta}; \boldsymbol{\phi}) := \mathbb{E}_{t\in[0,T], s\in[0,t], u\in[s,t), \boldsymbol{x}_0, \boldsymbol{x}_t|\boldsymbol{x}_0}$$
$$\left[d\left(\boldsymbol{x}_{\text{tgt}}(\boldsymbol{x}_t, t, u, s),\ \boldsymbol{x}_{\text{est}}(\boldsymbol{x}_t, t, s)\right)\right]. \tag{9}$$

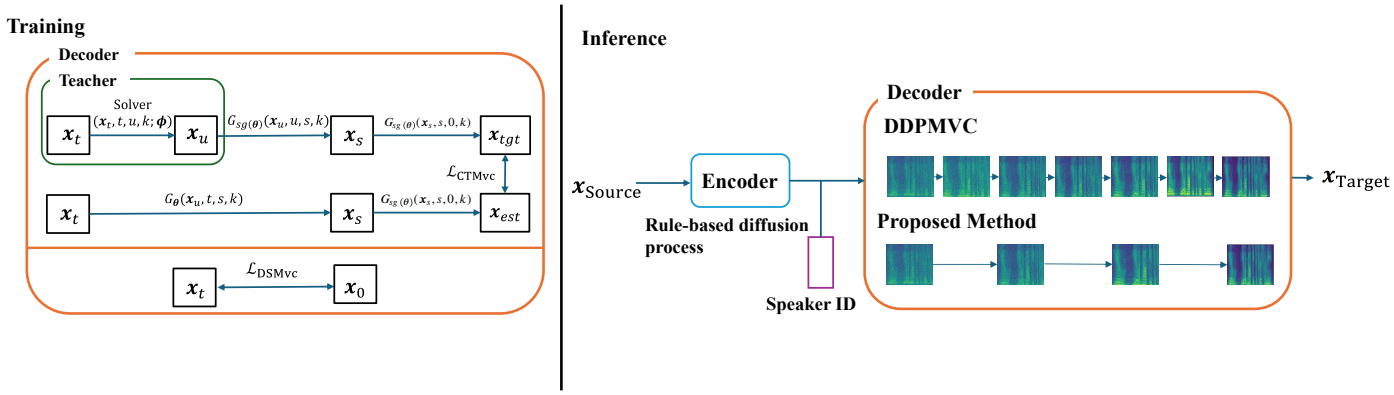

Fig. 1: Overview of the proposed method

Minimizing this loss encourages the learned neural jump $G_{\boldsymbol{\theta}}$ to approximate the empirical jump obtained by solving the PF ODE.

C. Conventional Method:DDPMVC

DDPMVC [10] is a voice conversion (VC) method inspired by the VoiceGrad[7] framework. It incorporated a rule-based diffusion process as the encoder (referred to as the diffusion encoder) and used reverse diffusion sampling as the decoder. VoiceGrad is a diffusion-based VC model built upon denoising diffusion probabilistic models (DDPMs). It is trainable with non-parallel data and supports any-to-many voice conversion. A key characteristic of VoiceGrad is that it does not use an encoder. As a result, the source speech during sampling contains both phonetic and speaker-specific information, which can lead to residual source speaker characteristics in the converted output. To address this issue, DDPMVC introduced a rule-based diffusion process as an encoder to suppress speaker information in the input speech while preserving phonetic content.

The diffusion noise added by the encoder is designed to perturb fine-grained structures such as speaker identity, without significantly degrading the phonetic information. Given a source log-mel spectrogram, the encoder applies the rule-based diffusion process to inject controlled levels of noise. This helps retain content information and prevent undesired changes in phonetic content during decoding.

The diffusion encoder uses the noise schedule of a variance-preserving SDE (VPSDE), along with the following forward and reverse diffusion processes shown in (10) and (11), respectively:

$$d\boldsymbol{x} = -\frac{1}{2}\beta(t)\boldsymbol{x}dt + \sqrt{\beta(t)}d\boldsymbol{w}, \tag{10}$$

$$d\boldsymbol{x} = \left[-\frac{1}{2}\beta(t)\boldsymbol{x} - \beta(t)\nabla\boldsymbol{x}\log p_t(\boldsymbol{x})\right]dt + \sqrt{\beta(t)}d\boldsymbol{w}. \tag{11}$$

The training objective is defined as:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\,\mathbb{E}_{k,t}\left\{\lambda(t)\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t,k) - \nabla_{\boldsymbol{x}_t}\log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)\|_2^2\right]\right\}, \tag{12}$$

where $\lambda(t)$, $k \sim p(k)$, and $\boldsymbol{x}_0 \sim p(\boldsymbol{x}|k)$ denote the weighting coefficient at time t, the speaker index, and the clean log-mel spectrogram, respectively. The noise-injected, speaker-independent representation, i.e., $\boldsymbol{x}_1$, is then processed by the decoder, which performs reverse diffusion sampling based on the VPSDE to generate the target speaker's log-mel spectrogram.

In conventional diffusion-based VC methods [19], [20], separate modules are introduced in the encoder to remove speaker information, and only phonetic content is fed into the diffusion-based decoder. In contrast, DDPMVC adopted a unified architecture where both the encoder and decoder are formulated as diffusion models. This allows for joint optimization across the entire pipeline and results in improved conversion accuracy compared to VoiceGrad. However, despite the simplicity of the encoder design—which consists solely of the noise injection process—the overall generation time remains long due to the sequential nature of diffusion sampling.

III. Proposed Method

This study aims to improve the inference speed and generation quality of DDPMVC by focusing on naturalness and speaker similarity. To this end, we propose replacing the decoder of DDPMVC with a CTM-based decoder adapted for voice conversion. Since CTM can estimate arbitrary-length jumps along the PF ODE in a single step, it is expected to generate high-quality mel-spectrograms with significantly fewer steps than the reverse diffusion sampling used in DDPMVC. While various distillation methods have been proposed for diffusion models, they tend to degrade in sample quality as the NFE increases. Moreover, their stochastic sampling can lead to semantic inconsistencies in the generated outputs. Therefore, this study focuses on the CTM. In our approach, DDPMVC serves as the teacher model to distill trajectory consistency into CTM.

For the encoder, we adopt the diffusion encoder used in DDPMVC, based on a VPSDE diffusion process. The noise schedule is set to $\beta(t) \in [0.1, 2]$. Under this schedule, the intelligibility of the noised speech was evaluated using the short-time objective intelligibility (STOI) [21], yielding an

average score of approximately 0.52, indicating that the added noise is weak enough to preserve most of the phonetic content.

In the decoder, conditional generation using speaker labels is introduced, which requires modifications to the CTM loss formulation. Letting k denote the speaker label, the neural jump is defined as:

$$G_{\boldsymbol{\theta}}(x_t, t, s, k) = \frac{s}{t} x_t + \left(1 - \frac{s}{t}\right) g_{\boldsymbol{\theta}}(x_t, t, s, k). \tag{13}$$

Accordingly, the consistency loss for training the neural jump is defined as:

$$\mathcal{L}_{\text{CTMvc}} = \mathbb{E}_{k,t,s,u,\boldsymbol{x}_0,\boldsymbol{x}_t|\boldsymbol{x}_0}\Big[\, d(\boldsymbol{x}_{\text{tgt}}, \boldsymbol{x}_{\text{est}})\Big], \tag{14}$$

where $\boldsymbol{x}_{\text{est}} := G_{\text{sg}(\boldsymbol{\theta})}\left(G_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, s, k), s, 0, k\right)$ and $\boldsymbol{x}_{\text{tgt}} := G_{\text{sg}(\boldsymbol{\theta})}\left(G_{\text{sg}(\boldsymbol{\theta})}\left(\text{Solver}(\boldsymbol{x}_t, t, u, k; \boldsymbol{\phi}), u, s, k\right), s, 0, k\right)$. The parameter $\boldsymbol{\phi}$ corresponds to the DDPMVC teacher model. While CTM [11] originally employed learned perceptual image patch similarity (LPIPS) [12] as the feature distance $d(\cdot,\cdot)$, LPIPS computes perceptual similarity based on features extracted from intermediate layers of convolutional networks pretrained on computer vision tasks. This enables it to capture visual perceptual similarity more effectively than conventional losses such as L2 or L1, and it has shown superior performance in training CTM. However, LPIPS is specifically designed for image-based applications and is not directly applicable to speech synthesis tasks. Therefore, we adopt mel-cepstral distortion (MCD) as the distance metric, which is widely used in the speech domain to evaluate auditory perceptual similarity. MCD effectively captures spectral envelope differences crucial for voice conversion quality assessment and provides computational efficiency by requiring no pretrained models.

In the original CTM [11], both denoising score matching (DSM) loss and adversarial loss were used as auxiliary objectives. In this study, we adopt only the DSM loss. We exclude adversarial loss since our experiments showed that it tended to degrade generation quality, likely due to difficulties in identifying a suitable discriminator. A similar issue was reported in SoundCTM [22]. The DSM loss is defined as:

$$\mathcal{L}_{\text{DSMvc}}(\boldsymbol{\theta}) = \mathbb{E}_{k,t,\boldsymbol{x}_0,\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\|\boldsymbol{x}_0 - g_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, t, k)\|_2^2\right]. \tag{15}$$

The overall training loss is given by:

$$\mathcal{L}(\boldsymbol{\theta}) := \mathcal{L}_{\text{CTMvc}} + \lambda_{\text{DSMvc}} \mathcal{L}_{\text{DSMvc}}, \tag{16}$$

where λ_{DSMvc} is an adaptive weight determined by gradient norm balancing [23].

For sampling, we use γ-sampling [11]. Finally, the converted acoustic features are passed through a neural vocoder to synthesize waveform audio. An overview of the full conversion pipeline is illustrated in Fig. 1.

IV. Experiments

A. Experimental settings

We conducted experiments using the CMU Arctic speech corpus [24], which contains recordings of multiple English speakers. Four speakers were selected for training: two male speakers (`aew`, `ksp`) and two female speakers (`clb`, `lnh`). For each speaker, 1000 utterances were used for training, 100 for validation, and 32 for testing. Additionally, to evaluate the performance of voice conversion on unseen speakers, 32 utterances from each of four unseen speakers—two male (`bdl`, `rms`) and two female (`slt`, `jmk`)—were used for testing. All audio signals were downsampled to 16kHz and converted into 80-dimensional log-mel spectrograms, which were used as acoustic features. The spectrograms $x_{d,m}$, where \$d\$ and \$m\$ denote the mel filterbank channel and the frame index, respectively, were normalized as $x_{d,m} \leftarrow \frac{x_{d,m} - \zeta_d}{\xi_d}$ using mean ζ_d and standard deviation ξ_d computed over the training set.

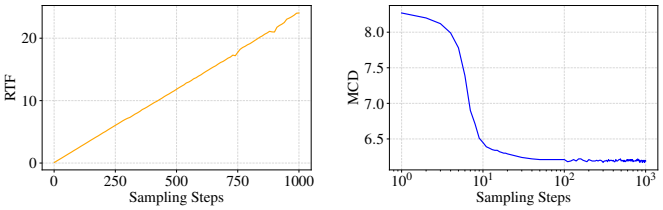

Fig. 2: RTF and MCD versus the number of sampling steps

We adopted DDPMVC as the baseline model. It also serves as the teacher model for training the proposed method, referred to as CTM-DDPMVC. DDPMVC employs a diffusion model based on a variance-preserving SDE (VP-SDE) with 1000 timesteps and a noise schedule of $\beta(t) \in [0.1, 2]$. For sampling, we used the Predictor-Corrector (PC) method [13]. For the proposed CTM-DDPMVC, we used a CTM with 1000 timesteps. The training setup followed the training details provided in [11], with $\sigma_{\min} = 0.002$ and $\sigma_{\max} = 4.0$ as modified hyperparameters. Sampling was performed using γ-sampling with $\gamma = 0$. HiFi-GAN [25] was used as the neural vocoder for waveform synthesis.

We used three objective evaluation metrics: mel-cepstral distortion (MCD), UTMOS [26], and real-time factor (RTF). UTMOS is a machine learning-based speech quality metric that estimates naturalness on a scale from 1.0 to 5.0. RTF was computed as RTF $= \Delta t / t_{\text{playback}}$, where Δt is the processing time from the input mel spectrogram of the source speaker to the converted spectrogram, and t_{playback} is the duration of the synthesized audio.

B. Analysis of Sampling Steps Effects

We conducted an experiment to determine an appropriate number of sampling steps for γ-sampling in CTM-DDPMVC. Prior studies [11], [22] have shown that increasing the number of sampling steps generally leads to improved generation accuracy. However, this improvement comes at the cost of longer generation time, suggesting a trade-off between fidelity and efficiency. To investigate this trade-off, we measured both RTF and MCD for various sampling steps settings. The number of sampling steps was varied from 1 to 1000. For steps 1 through 20, the interval was set to 1, and for steps beyond 20, we increased the sampling steps in increments of 10.

TABLE I: Comparison of Voice Conversion Performance Using MCD, UTMOS, and RTF

Type	Sampling Steps	MCD					UTMOS	RTF
		M2M	F2F	M2F	F2M	All		
DDPMVC,$T = 50$	50	6.66 ± 0.05	6.14 ± 0.07	6.84 ± 0.05	6.59 ± 0.04	6.58 ± 0.03	2.73 ± 0.03	2.35 ± 0.24
DDPMVC, $T = 1000$	50	8.09 ± 0.04	7.91 ± 0.06	8.50 ± 0.05	8.01 ± 0.04	8.15 ± 0.03	1.44 ± 0.01	1.95 ± 0.19
	1000	6.50 ± 0.06	5.99 ± 0.07	6.62 ± 0.06	6.48 ± 0.05	6.42 ± 0.03	$\mathbf{3.17 \pm 0.03}$	25.17 ± 2.63
CTM-DDPMVC	10	6.23 ± 0.05	6.27 ± 0.05	6.52 ± 0.04	6.30 ± 0.04	6.34 ± 0.02	2.14 ± 0.03	$\mathbf{0.23 \pm 0.05}$
	20	6.12 ± 0.05	6.00 ± 0.05	6.31 ± 0.04	6.19 ± 0.04	6.17 ± 0.02	2.58 ± 0.03	0.46 ± 0.09
	30	6.10 ± 0.05	5.94 ± 0.05	6.26 ± 0.04	6.18 ± 0.04	6.13 ± 0.02	2.66 ± 0.03	0.68 ± 0.13
	40	6.10 ± 0.05	5.92 ± 0.05	6.25 ± 0.04	6.16 ± 0.05	6.12 ± 0.03	2.70 ± 0.03	0.90 ± 0.12
	50	$\mathbf{6.09 \pm 0.05}$	$\mathbf{5.90 \pm 0.05}$	$\mathbf{6.23 \pm 0.04}$	$\mathbf{6.16 \pm 0.05}$	$\mathbf{6.11 \pm 0.03}$	2.73 ± 0.03	1.12 ± 0.13

The experimental results are shown in Fig. 2. The RTF increases proportionally with the number of sampling steps. Fig. 2 presents the results for MCD. Note that the horizontal axis is plotted on a logarithmic scale. We observe a sharp drop in MCD within the first 10 steps, indicating that generation quality improves rapidly in the early stages. Although the 1 step generation fails under our experimental conditions, the model exhibits consistent improvement as the number of steps increases, demonstrating the effectiveness of CTM. However, the MCD stabilizes around 50 steps, and no significant improvement is observed up to 1000 steps. Based on these findings, we used five sampling step settings—10, 20, 30, 40, and 50—in subsequent experiments to evaluate performance under varying inference-efficiency conditions.

C. Result

We compare the conversion performance between the baseline DDPMVC and the proposed CTM-DDPMVC. The evaluation metrics used are similarity (MCD), naturalness (UTMOS), and generation speed (RTF). DDPMVC was evaluated with 50 and 1000 sampling steps. For 50 steps, DDPMVC trained with both 50 and 1000 diffusion timesteps is reported. CTM-DDPMVC is evaluated with sampling steps of 10, 20, 30, 40, and 50.

Fig. I presents the results. In terms of MCD, CTM-DDPMVC consistently outperforms DDPMVC across all sampling steps and conversion pairs. DDPMVC with $T = 1000$ showed noticeable degradation at 50 sampling steps due to artifacts and noise. In contrast, CTM-DDPMVC distilled from the same $T = 1000$ model achieved the best MCD score at 50 steps. At 50 steps, CTM-DDPMVC achieves comparable UTMOS scores to DDPMVC. The results also indicate that UTMOS improves as the number of sampling steps increases. One possible reason for the relatively lower UTMOS is that the use of MCD as a feature distance emphasizes spectral envelope similarity during distillation, which may result in the loss of phonetic content. A potential workaround is to adopt an L2 distance such as mean squared error (MSE), which considers the entire mel-spectrogram and may help preserve phonetic structures. However, we found that training with MSE led to lower performance in both MCD and UTMOS compared to using MCD. Therefore, we conclude that using MCD as the feature distance is the most effective choice under the current setup. In terms of RTF, CTM-DDPMVC achieved a score of only 0.44 ± 0.05 at 10 sampling steps while outperforming DDPMVC trained and sampled with 1000 steps in terms of MCD. This result indicates that CTM-DDPMVC not only significantly improves inference speed compared to the teacher model, DDPMVC, but also achieves better conversion accuracy.

V. Conclusion

In this paper, we proposed a voice conversion method that replaces the diffusion-based decoder in DDPMVC with a consistency trajectory model (CTM). Experimental results demonstrated that the proposed method significantly improves generation speed while achieving comparable or superior performance in speaker similarity compared to the baseline. Furthermore, we observed that increasing the number of sampling steps up to a certain point leads to improved performance, enabling a trade-off between speed and accuracy depending on the application requirements. However, in terms of naturalness, there remains a performance gap between CTM-DDPMVC and the teacher model DDPMVC with 1000 sampling steps. In future work, we plan to improve naturalness by incorporating adversarial training with an appropriate discriminator and exploring alternative feature distances for distillation.

Acknowledgment

This research was partly funded by JSPS Grants-in-Aid for Scientific Research 24H00715.

References

[1] D. P. Kingma and M. Welling, *Auto-encoding variational bayes*, 2022. arXiv: 1312.6114 [stat.ML].

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27, Curran Associates, Inc., 2014.

[3] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach and D. Blei, Eds., ser. Proceedings of Machine Learning Research, vol. 37, Lille, France: PMLR, Jul. 2015, pp. 2256–2265.

[4] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 6840–6851.

[5] Y.-H. Chen, D.-Y. Wu, T.-H. Wu, and H.-y. Lee, "Again-vc: A one-shot voice conversion using activation guidance and adaptive instance normalization," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, pp. 5954–5958.

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

[7] H. Kameoka, T. Kaneko, K. Tanaka, N. Hojo, and S. Seki, *Voicegrad: Non-parallel any-to-many voice conversion with annealed langevin dynamics*, 2024. arXiv: 2010.02977 [cs.SD]. [Online]. Available: https://arxiv.org/abs/2010.02977.

[8] Y. Song and S. Ermon, *Generative modeling by estimating gradients of the data distribution*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., 2019.

[9] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from unaligned corpora using variational autoencoding Wasserstein generative adversarial networks," *Interspeech*, pp. 5289–5293, 2017.

[10] R. Hatakeyama, K. Okuda, and T. Nakashika, "Ddp-mvc: Non-parallel any-to-many voice conversion using diffusion encoder," in *2024 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2024, pp. 1–6. DOI: 10.1109/APSIPAASC63619.2025.10849015.

[11] D. Kim, C.-H. Lai, W.-H. Liao, *et al.*, *Consistency trajectory models: Learning probability flow ode trajectory of diffusion*, 2024. arXiv: 2310.02279 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2310.02279.

[12] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, 2018, p. 586.

[13] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, *Score-based generative modeling through stochastic differential equations*, 2021. arXiv: 2011.13456 [cs.LG].

[14] T. Karras, M. Aittala, T. Aila, and S. Laine, *Elucidating the design space of diffusion-based generative models*, 2022. arXiv: 2206.00364 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2206.00364.

[15] Y. Song and S. Ermon, *Improved techniques for training score-based generative models*, 2020. arXiv: 2006.09011 [cs.LG].

[16] J. Song, C. Meng, and S. Ermon, *Denoising diffusion implicit models*, 2022. arXiv: 2010.02502 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2010.02502.

[17] T. Salimans and J. Ho, *Progressive distillation for fast sampling of diffusion models*, 2022. arXiv: 2202.00512 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2202.00512.

[18] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, *Consistency models*, 2023. arXiv: 2303.01469 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2303.01469.

[19] S. Liu, Y. Cao, D. Su, and H. Meng, "DiffSVC: A diffusion probabilistic model for singing voice conversion," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 741–748. DOI: 10.1109/ASRU51503.2021.9688219.

[20] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, M. Kudinov, and J. Wei, *Diffusion-based voice conversion with fast maximum likelihood sampling scheme*, 2022. arXiv: 2109.13821 [cs.SD].

[21] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 4214–4217. DOI: 10.1109/ICASSP.2010.5495701.

[22] K. Saito, D. Kim, T. Shibuya, *et al.*, *Soundctm: Unifying score-based and consistency models for full-band text-to-sound generation*, 2025. arXiv: 2405.18503 [cs.SD]. [Online]. Available: https://arxiv.org/abs/2405.18503.

[23] P. Esser, R. Rombach, and B. Ommer, *Taming transformers for high-resolution image synthesis*, 2020. arXiv: 2012.09841 [cs.CV].

[24] J. Kominek and A. W. Black, "The CMU arctic speech databases," in *Fifth ISCA workshop on speech synthesis*, 2004.

[25] J. Kong, J. Kim, and J. Bae, *Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis*, 2020. arXiv: 2010.05646 [cs.SD]. [Online]. Available: https://arxiv.org/abs/2010.05646.

[26] T. Saeki, D. Xin, W. Nakata, T. Koriyama, S. Takamichi, and H. Saruwatari, "Utmos: Utokyo-sarulab system for voicemos challenge 2022," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2022, 2022, pp. 4521–4525.