WaveTransfer: A Flexible End-to-end Multi-instrument Timbre Transfer with Diffusion

Teysir Baoueb, Xiaoyu Bie, Hicham Janati, Gael Richard

2024 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 22–25, 2024, LONDON, UK

WAVETRANSFER: A FLEXIBLE END-TO-END MULTI-INSTRUMENT TIMBRE TRANSFER WITH DIFFUSION

Teysir Baoueb, Xiaoyu Bie, Hicham Janati, Gaël Richard

LTCI, Télécom Paris, IP Paris, France

ABSTRACT

As diffusion-based deep generative models gain prevalence, researchers are actively investigating their potential applications across various domains, including music synthesis and style alteration. Within this work, we are interested in timbre transfer, a process that involves seamlessly altering the instrumental characteristics of musical pieces while preserving essential musical elements. This paper introduces WaveTransfer, an end-to-end diffusion model designed for timbre transfer. We specifically employ the bilateral denoising diffusion model (BDDM) for noise scheduling search. Our model is capable of conducting timbre transfer between audio mixtures as well as individual instruments. Notably, it exhibits versatility in that it accommodates multiple types of timbre transfer between unique instrument pairs in a single model, eliminating the need for separate model training for each pairing. Furthermore, unlike recent works limited to 16 kHz, WaveTransfer can be trained at various sampling rates, including the industry-standard 44.1 kHz, a feature of particular interest to the music community.

***Index Terms*—** Multi-instrumental timbre transfer, diffusion models, music transformation, generative AI

1. INTRODUCTION

In recent years, there has been a growing interest in the manipulation and transformation of audio signals, particularly in the realm of music [1–3]. One intriguing area of exploration within this domain is timbre transfer, a process that involves altering the tonal characteristics of musical sounds while preserving their content including fundamental pitch and temporal structure. Timbre is often described as the unique quality or color of a sound or 'that attribute of auditory sensation in terms of which a listener can judge that two steady-state complex tones having the same loudness and pitch; are dissimilar' [4]. It plays a crucial role in shaping our perception and emotional response to music. Timbre transfer can be considered as a more focused and well-defined objective than musical style transfer which usually involves not only timbre transfer (e.g., change of instrumentation) but also rhythmic and other high-level musical knowledge transfer (as in [5] for music style transfer for symbolic music).

A large variety of approaches has already been proposed for timbre transfer. Several methods rely on the modeling capabilities of autoencoders or Generative Adversarial Networks (GAN) to obtain a disentangled latent space suitable for timbre transfer. For instance, popular architectures include WaveNets autoencoders [6], Variational AutoEncoders (VAE) [7–9] or GANs [10]. In [11], an interesting hierarchical approach is proposed using source-filtering networks, which reconstruct the transferred signal at increasing resolution.

More recently, the potential of diffusion models for high-quality audio synthesis has opened a new path for diffusion-based timbre transfer. For instance, in [12], Transplayer utilizes a two-phase approach where the initial timbre transformation operated at the Constant Q Transform (CQT) representation level using an autoencoder architecture is further converted to audio waveform employing an audio-synthesis diffusion-based model. In [13], optimal transport principles are jointly exploited with diffusion modeling and successfully applied to the many-to-many timbre transfer task.

Other recent models for timbre transfer of particular interest for this work include the Music-STAR [14] and DiffTransfer [15] systems. Music-STAR [14] is built upon the WaveNet autoencoder [16] with a universal encoder and individual decoders corresponding to each of the target domains. In Difftransfer [15], the timbre transfer is carried out by means of Denoising Diffusion Implicit models [17]. This model was shown to be particularly efficient for both single- and multi-instrument timbre transfer and at the state of the art for the task on the Starnet dataset [18], also considered in this work.

In this paper, we introduce *WaveTransfer*, a novel end-to-end diffusion model designed for timbre transfer. If our model shares some common concepts with the DiffTransfer model of [15], it is capable of conducting timbre transfer between audio mixtures as well as individual instruments in a single global model eliminating the need for separate model training for each specific timbre transfer task. Another important property of our model is that it directly generates the audio waveform without needing to rely on an external vocoder. Finally, our model can operate at any sampling rate extending all previous works that are limited to a rather low 16 kHz.

The paper will be structured as follows: Section 2 presents background work on denoising diffusion probabilistic models and bilateral denoising diffusion models. Section 3 introduces our timbre transfer method. Section 4 describes our experimental procedures, and Section 5 discusses the results. Finally, Section 6 concludes with insights, a summary of our contributions, and suggestions for future research. Audio files and code are provided on our demo page: https://wavetransfer.github.io/.

2. BACKGROUND

2.1. Denoising diffusion probabilistic models (DDPM)

Denoising diffusion probabilistic models (DDPM) [19] represent a class of generative models characterized by a dual-process framework, the **forward** and **backward** processes.

This work was funded by the European Union (ERC, HI-Audio, 101052978). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

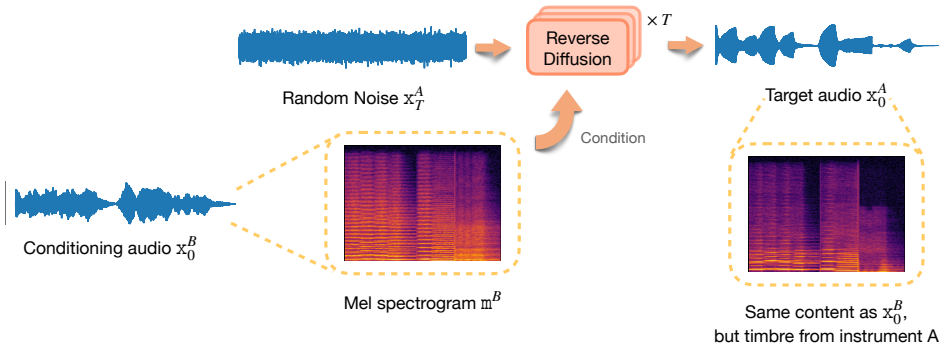

Fig. 1: Timbre transfer using diffusion models. The objective is to generate a target audio $\mathbf{x}_0^A$ from a random noise $\mathbf{x}_T^A$ and a conditioning audio $\mathbf{x}_0^B$, where $\mathbf{x}_0^A$ has the same content as $\mathbf{x}_0^B$ but is played with a different instrument.

In mathematical terms, consider $\mathbf{x}_0$ as a datum drawn from the distribution $q(\mathbf{x}_0)$ of a specified dataset. In the **forward** process, $\mathbf{x}_0$ is gradually perturbed by a Gaussian noise in T steps, resulting in a sequence of noisy samples $\mathbf{x}_1, \ldots, \mathbf{x}_T$. Given a noise schedule $\{\beta_t\}_{t=1}^T$, the forward diffusion process can be formulated as:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}). \tag{1}$$

In a more concise manner, $\mathbf{x}_t$ can be sampled at any time step t using the closed-form expression:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \tag{2}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$. When T is sufficiently large, $\mathbf{x}_T$ is equivalent to an isotropic Gaussian distribution.

If we can **reverse** the above process, we can generate new data from a Gaussian noise. However, directly computing the conditional distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is not feasible, therefore we seek to learn a model $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$ that approximates the true distribution. The parameters $\boldsymbol{\theta}$ can be optimized by minimizing the Kullback-Leibler divergence between the two distributions, $\mathrm{KL}(p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)||q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0))$. Since the reverse process is tractable conditioned on $\mathbf{x}_0$, we can obtain the analytical expression of $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ during training. In DDPM [19], the optimization is further simplified to the minimization of the noise estimation:

$$\mathcal{L}_{\boldsymbol{\theta}} = \min_{\boldsymbol{\theta}} \mathbb{E}\left[\|\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) - \boldsymbol{\epsilon}\|_2^2\right], \tag{3}$$

where t is the diffusion step randomly sampled from $[1, T]$, $\boldsymbol{\epsilon}$ is sampled from a normal distribution and $\mathbf{x}_t$ can be easily obtained via Eq. 2. During inference, we can iteratively sample the data from $\mathbf{x}_T$ to $\mathbf{x}_0$ via[1]:

$$\mathbf{x}_{t-1} = \mathcal{N}\left(\mathbf{x}_{t-1}; \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right), \sigma_t^2 I\right) \tag{4}$$

2.2. Bilateral denoising diffusion models (BDDM)

Given the premise of a sufficiently large value for T, executing the entire reverse process using Eq 4 is computationally expensive. To circumvent this, Lam et al. introduced bilateral denoising diffusion models (BDDM) [20], an approach for judiciously determining an appropriate noise schedule with a length set to be within or match a specified maximum number of inference iterations.

[1] In DDPM [19], both $\sigma_t^2 = \beta_t$ and $\sigma_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$ had similar results.

More specifically, besides a diffusion model, an additional neural network, called the schedule network, is trained to select a noise schedule used for sampling at inference time. The training is done by minimizing the following objective function:

$$\mathcal{L}_\phi^{(t)} = \frac{1}{2(1-\hat{\beta}_t(\phi)-\bar{\alpha}_t)}\left\|\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} - \frac{\hat{\beta}_t(\phi)}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\theta^\star}(\mathbf{x}_t, t)\right\|_2^2 + \frac{1}{4}\log\frac{1-\bar{\alpha}_t}{\hat{\beta}_t(\phi)} + \frac{D}{2}\left(\frac{\hat{\beta}_t(\phi)}{1-\bar{\alpha}_t} - 1\right), \tag{5}$$

where ϕ denotes the parameters of the schedule network and $\theta^\star$ represents well-optimized parameters of $\boldsymbol{\epsilon}_\theta$. Here $\hat{\beta}_t(\phi)$ denotes the noise scale at time t which is obtained through the neural network of parameters ϕ by considering the previous noise scale $\hat{\beta}_{t+1}$ and the current noisy input $\mathbf{x}_t$.

3. PROPOSED METHOD

This section presents our approach for timbre transfer. Given a pair of tracks played with distinct instruments, our objective is to transfer the timbre from instrument B (the conditioning instrument) to instrument A (the target instrument), while maintaining the content from the conditioning instrument. As shown in Fig. 1, our model takes the mel spectrogram $\boldsymbol{m}^B$ from the conditioning instrument B as input, then applies an iterative diffusion process to generate a waveform $\boldsymbol{x}_0^A$ with the timbre traits of the target instrument A. The objective of our model is to maximize the likelihood of the conditional distribution $q(\boldsymbol{x}_0^A|\boldsymbol{m}^B)$.

3.1. Training procedure

The training process follows the principles of DDPM [19], and is depicted in Figure 2. We start with an initial audio signal from the target instrument $\mathbf{x}_0^A \sim q_A(\mathbf{x}_0)$. Using the nice property from Eq. 2, we can easily compute its perturbation $\mathbf{x}_t^A$ at diffusion step t. We then consider the corresponding audio $\mathbf{x}_0^B$ that has the same content as $\mathbf{x}_0^A$ but is played by a different instrument. Taking the mel spectrogram $\mathbf{m}^B$ of $\mathbf{x}_0^B$ as an additional condition, the model learns to predict the noise introduced to $\mathbf{x}_0^A$, leveraging the performance information encapsulated within $\mathbf{m}^B$.

Similar to WaveGrad [21], we consider a continuous noise level $\sqrt{\bar{\alpha}}$ as conditioning provided to the neural network to serve the role of the time index t, where the sampling process for $\sqrt{\bar{\alpha}}$ involves utilizing a training noise schedule with length T. The training objective

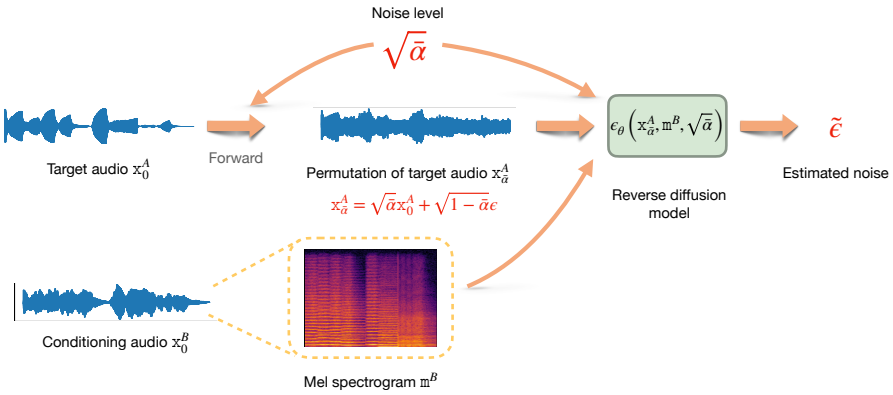

Fig. 2: Training process of WaveTransfer

can thus be modified from Eq. 3 as:

$$\mathcal{L}_{\boldsymbol{\theta}} = \min_{\boldsymbol{\theta}} \mathbb{E}\left[\left\|\boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}}\mathbf{x}_0^A + \sqrt{1-\bar{\alpha}}\boldsymbol{\epsilon}, \mathbf{m}^B, \sqrt{\bar{\alpha}}\right) - \boldsymbol{\epsilon}\right\|_1\right], \quad (6)$$

As previously mentioned, running inference with the T-long training noise schedule is computationally expensive. In WaveGrad [21], Chen et al. proposed utilizing a grid search approach to select a shorter noise schedule. However, the search might take over a day for as few as 6 iterations on 1 NVIDIA Tesla P40 GPU, as observed by Lam et al. [20]. Therefore, we opt to train a schedule network using the BDDM approach, as outlined in Section 2, subsequent to training the timbre transfer neural network model.

3.2. Model architecture

The architecture of the timbre transfer neural network is similar to WaveGrad [21], featuring a series of upsampling blocks to expand the temporal dimension of the conditioning mel spectrogram $\mathbf{m}^B$ into the time domain. Conversely, downsampling blocks reduce the temporal dimension of the noisy audio input. Both pathways leverage a feature-wise linear modulation (FiLM) [22] module to integrate the information gleaned from upsampling and downsampling processes synergistically.

The schedule network has a GALR (globally attentive locally recurrent) network architecture [23]. Within each GALR block, there exist two distinct modeling perspectives. The initial perspective focuses on recurrently modeling the local structures present in input signals, while the subsequent perspective is dedicated to capturing global dependencies through the utilization of the multi-head self-attention mechanism.

3.3. Inference procedure

Given a noise schedule of length N, during inference, the model is provided with the mel spectrogram $\boldsymbol{m}^B$ from the conditioning instrument B along with random noise $\mathbf{x}_N \sim \mathcal{N}(\mathbf{x}_N; \mathbf{0}, \mathbf{I})$, as illustrated in Figure 3. The model approximates the added noise at each iteration. The estimated noise at step $n \in [\![1, N]\!]$ is then used to generate $\mathbf{x}_{n-1}$. Finally, this iterative algorithm produces an audio signal with the same content as $\boldsymbol{m}^B$ but played with instrument A. Similar to Eq. 4, this procedure can be described by the following equation for DDPMs:

$$\mathbf{x}_{n-1} = \frac{1}{\sqrt{\alpha_n}}\left(\mathbf{x}_n - \frac{1-\alpha_n}{\sqrt{1-\bar{\alpha}_n}}\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_n, \mathbf{m}^B, \sqrt{\bar{\alpha}_n}\right)\right) + \sigma_n \boldsymbol{z}, \quad (7)$$

where $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{z}; \mathbf{0}, \mathbf{I})$ for $n > 1$, $\boldsymbol{z} = \mathbf{0}$ for $n = 1$ and $\sigma_n^2 = \frac{1-\bar{\alpha}_{n-1}}{1-\bar{\alpha}_n}\beta_n$.

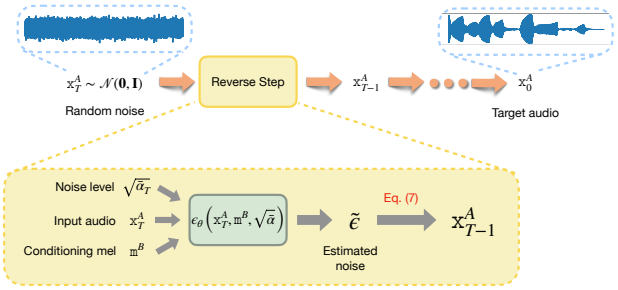

Fig. 3: Inference process of WaveTransfer

4. EXPERIMENTS

In this section, we outline the experiments we conducted and the evaluation protocol employed.

4.1. Dataset and preprocessing

The StarNet dataset [18] consists of 104 classical music compositions sampled at 44.1 kHz. Each composition comprises six distinct tracks:

- 2 Mixture tracks: clarinet-vibraphone, strings-piano
- 4 Individual stem tracks: clarinet, vibraphone, strings, piano

There is a correspondence between the content of the clarinet and strings tracks on one side and the piano and vibraphone tracks on the other.

The provided test set contains 10 compositions, encompassing both classical and modern music pieces, with 6 tracks each for stems and mixtures, resulting in a total of 60 tracks.

In this work, we adopt the same preprocessing steps as in [14], which involves detecting and removing intervals where one or both instruments are silent, and converting to mono for training on 44.1 kHz tracks. For model training on 16 kHz, we employ the reduced StarNet dataset, achieved through downsampling the preprocessed dataset. As the entire length of the track is not considered during training, but rather a random segment is extracted from each, we offline fragment the tracks into 5-second excerpts for both training sets at each sampling rate to expedite loading time.

For validation purposes, we reserve 1 composition ($\sim$ 100 seconds $\times 6$) from the training set and utilize the remainder for training.

4.2. Training setup

Four models were trained on the StarNet dataset, each serving different purposes.

The first two models perform all six timbre transfer types (clarinet $\leftrightarrow$ strings, piano $\leftrightarrow$ vibraphone, (clarinet + vibraphone) $\leftrightarrow$ (piano + strings)). These models, denoted as $\text{WT}^{16}_{\text{global}}$ and $\text{WT}^{44}_{\text{global}}$, were trained with sampling rates of 16 kHz and 44.1 kHz, respectively.

Furthermore, in order to evaluate the model's capability to execute timbre transfer between mixtures without the requirement of individual tracks, two additional models, $\text{WT}^{16}_{\text{mix}}$ (16 kHz) and $\text{WT}^{44}_{\text{mix}}$ (44.1 kHz), were trained exclusively on a subset of the training data containing only the two mixture tracks for each performance.

All models were trained on 1 A100 GPU for 1 M steps with a learning rate of $2 \cdot 10^{-4}$ and a batch size of 32. We employ 128-dimensional log-mel spectrograms calculated using a Hann window of 1200 size, a hop length of 300, and a 2048-point FFT. We extract 66 time frames from each training sample.

For each of the previous models, we train a schedule network for 10000 steps on 1 V100 GPU. We use 1 GALR block with 128 hidden dimensions, a window length of 8 and a segment size of 64.

4.3. Metrics for evaluation

For objective evaluation, we employ the following metrics:

- **Fréchet Audio Distance (FAD)** [24] is a reference-free metric designed for evaluating audio quality. It utilizes a pretrained audio model to generate embedding statistics for the set of produced tracks. These statistics are compared with those from a database of clean music by computing multivariate Gaussian distributions for each set of embeddings. The FAD score is then determined by calculating the Fréchet distance between these distributions. Smaller FAD scores indicate higher audio quality. We use the following models to compute the embeddings: the VGGish model [25] (16 kHz, 44.1 kHz), PANN [26] (16 kHz) and CLAP [27] (44.1 kHz)[2].
- **Perceptual Evaluation of Audio Quality (PEAQ)** [28,29] is a standardized method used for evaluating the perceived audio quality of audio signals. It aims to quantify the difference between an original audio signal and a degraded version of that signal. It is composed of two scores:
  - **Objective Difference Grade (ODG):** This metric quantifies the perceived quality difference between the original and generated signals. It assigns a score from -4 to 0, where higher values indicate better quality.
  - **Distortion Index (DI):** This index measures the level of distortion introduced in the generated signal. Lower values signify greater distortion.
- **ViSQOL (Virtual Speech Quality Objective Listener)** [30] stands as a signal-based metric for full-reference assessment. Initially crafted to mirror human perception of speech quality, it relies on a spectro-temporal measure to gauge similarity between reference and test speech signals at 16 kHz. Subsequently, its scope expanded to encompass music signals at a 48 kHz sampling rate. When employing ViSQOL for evaluation, we upsample both the ground-truth and generated signals.

[2]Since VGGish and PANN are trained on 16 kHz and CLAP is trained on 48 kHz, when we test waveforms at 44.1 kHz, we downsample them to 16 kHz to compute VGGish embeddings and upsample to 48 kHz to compute the CLAP embeddings.

4.4. Inference noise schedules

During inference, we adopt WaveGrad's 6-iteration noise schedule (WG-6). Additionally, given that DiffTransfer utilizes 20 iterations for its reverse process, we investigate noise schedule searching with BDDM, setting a maximum of 20 iterations. During the search, the network generates noise schedules of length $n \leq 20$. The BDDM approach necessitates employing a metric to determine the optimal noise schedule, evaluated based on its performance according to this metric on a validation set.

Given the slow computation of PEAQ and the potential inaccuracies in ViSQOL's assessment of short signals and its requirement of upsampling, we have decided to employ FAD alongside VGGish embeddings for this task, even though this choice is not flawless. One drawback lies in the necessity to compute the metric on a sufficiently large set of generated signals, which inevitably slows down the process compared to utilizing a rapid full-reference metric on as few as 1 sample as specified in [20]. We denote the selected optimal noise schedule by BDDM-n.

5. RESULTS

Hereafter, we present the results for timbre transfer, starting with the global models, which are capable of performing timbre transfer between individual stems and mixtures. Subsequently, we delve into the results concentrating on mixture timbre transfer, encompassing both global models and mixture-specific models.

5.1. Inference conducted with global models

In this subsection, we conduct the timbre transfer process across all 6 possible transformations. To achieve this, we utilize both $\text{WT}^{16}_{\text{global}}$ and $\text{WT}^{44}_{\text{global}}$ on the 6 tracks of the 10 performances in the test set, resulting in a total of 60 tracks. The results are presented in Tables 1 and 2.

Table 1: FAD results ($\downarrow$) on the test set (60 tracks) using 16 kHz and 44.1 kHz sampling rates and different embeddings

SR	Model	VGGish	PANN	CLAP
16	$\text{WT}^{16}_{\text{global}}$ with BDDM-20	$\mathbf{4.17}$	$3.67 \cdot 10^{-3}$	-
	$\text{WT}^{16}_{\text{global}}$ with WG-6	4.38	$\mathbf{3.59 \cdot 10^{-3}}$	-
44.1	$\text{WT}^{44}_{\text{global}}$ with BDDM-19	$\mathbf{4.89}$	-	$\mathbf{0.51}$
	$\text{WT}^{44}_{\text{global}}$ with WG-6	5.52	-	0.56

Table 2: ViSQOL and PEAQ results (mean $\pm$ standard deviation) on the test set (60 tracks) using 16 kHz and 44.1 kHz sampling rates

SR	Model	ViSQOL ($\uparrow$)	ODG ($\uparrow$)	DI ($\uparrow$)
16	$\text{WT}^{16}_{\text{global}}$ with BDDM-20	$\mathbf{3.17} \pm 0.48$	-2.22 ± 0.02	-0.34 ± 0.03
	$\text{WT}^{16}_{\text{global}}$ with WG-6	3.13 ± 0.53	-2.22 ± 0.02	-0.34 ± 0.03
44.1	$\text{WT}^{44}_{\text{global}}$ with BDDM-19	$\mathbf{4.23} \pm 0.46$	-2.23 ± 0.03	-0.37 ± 0.05
	$\text{WT}^{44}_{\text{global}}$ with WG-6	4.18 ± 0.50	-2.23 ± 0.03	-0.36 ± 0.05

We observe that employing the noise schedule derived from BDDM led to superior outcomes in terms of FAD scores with VGGish embeddings. This outcome aligns with expectations, as the selection of this particular noise schedule was predicated on its performance with respect to that metric. For the remaining embeddings, PANN shows comparable results, while CLAP exhibits a slight improvement with the BDDM approach.

Transitioning to full-reference metrics, we notice minimal variation in results between noise schedules, with the BDDM approach prevailing in ViSQOL but displaying almost no difference in PEAQ.

5.2. Inference conducted only on mixture tracks

In contradistinction to Models $\text{WT}^{16}_{\text{mix}}$ and $\text{WT}^{44}_{\text{mix}}$, DiffTransfer and Music-STAR train a single model for each type of transformation: one model for (piano + vibraphone) $\rightarrow$ (clarinet + vibraphone) and another one for (clarinet + vibraphone) $\rightarrow$ (piano + vibraphone).

To ensure consistency with the evaluation protocols of DiffTransfer and Music-STAR in [15], we exclusively utilize the mixture tracks from each performance within the test set (2 per performance, totaling 20 tracks). In addition to using $\text{WT}^{16}_{\text{mix}}$ and $\text{WT}^{44}_{\text{mix}}$, which are tailored explicitly for mixture-to-mixture timbre transfer, we incorporate models $\text{WT}^{16}_{\text{global}}$ and $\text{WT}^{44}_{\text{global}}$, where we focus only on evaluation with mixture tracks. The results [3] are showcased in Tables 3 and 4.

Table 3: FAD results (↓) on the mixture tracks in the test set (20 tracks) using 16 kHz and 44.1 kHz sampling rates and different embeddings

	Model	VGGish	PANN	CLAP
16 kHz	DiffTransfer [15]	$\mathbf{4.37}$	$\mathbf{2.3 \cdot 10^{-3}}$	-
	Music-STAR [14]	8.93	$3.3 \cdot 10^{-3}$	-
	$\text{WT}^{16}_{\text{mix}}$ with WG-6	6.10	$3.90 \cdot 10^{-3}$	-
	$\text{WT}^{16}_{\text{mix}}$ with BDDM-20	5.60	$3.42 \cdot 10^{-3}$	-
	$\text{WT}^{16}_{\text{global}}$ with WG-6	6.34	$3.68 \cdot 10^{-3}$	-
	$\text{WT}^{16}_{\text{global}}$ with BDDM-20	6.01	$3.75 \cdot 10^{-3}$	-
44.1 kHz	$\text{WT}^{44}_{\text{mix}}$ with WG-6	7.30	-	0.67
	$\text{WT}^{44}_{\text{mix}}$ with BDDM-20	6.74	-	$\mathbf{0.63}$
	$\text{WT}^{44}_{\text{global}}$ with WG-6	7.42	-	0.73
	$\text{WT}^{44}_{\text{global}}$ with BDDM-19	$\mathbf{6.45}$	-	0.67

Table 4: ViSQOL and PEAQ results on the mixture tracks in the test set (20 tracks) using 16 kHz and 44.1 kHz sampling rates

	Model	ViSQOL (↑)	ODG (↑)	DI (↑)
16 kHz	DiffTransfer [15]	$\mathbf{3.28} \pm 0.42$	$\mathbf{-2.20} \pm 0.05$	$\mathbf{-0.32} \pm 0.07$
	Music-STAR [14]	2.43 ± 0.29	-2.24 ± 0.07	-0.37 ± 0.11
	$\text{WT}^{16}_{\text{mix}}$ with WG-6	3.02 ± 0.32	-2.22 ± 0.02	-0.34 ± 0.03
	$\text{WT}^{16}_{\text{mix}}$ with BDDM-20	3.11 ± 0.31	-2.23 ± 0.03	-0.35 ± 0.04
	$\text{WT}^{16}_{\text{global}}$ with WG-6	2.86 ± 0.33	-2.22 ± 0.03	-0.34 ± 0.03
	$\text{WT}^{16}_{\text{global}}$ with BDDM-20	2.99 ± 0.30	-2.22 ± 0.03	-0.34 ± 0.03
44.1 kHz	$\text{WT}^{44}_{\text{mix}}$ with WG-6	3.98 ± 0.58	-2.24 ± 0.03	-0.37 ± 0.05
	$\text{WT}^{44}_{\text{mix}}$ with BDDM-20	2.82 ± 0.83	-2.25 ± 0.04	-0.40 ± 0.08
	$\text{WT}^{44}_{\text{global}}$ with WG-6	3.76 ± 0.71	-2.24 ± 0.03	-0.37 ± 0.05
	$\text{WT}^{44}_{\text{global}}$ with BDDM-19	$\mathbf{4.06} \pm 0.54$	-2.24 ± 0.04	-0.38 ± 0.06

[3] The results reported for DiffTransfer and Music-STAR are extracted from [15]. They were computed as follows: performing the timbre transfer task with each model: the (clarinet + vibraphone) $\rightarrow$ (piano + vibraphone) model and the (piano + vibraphone) $\rightarrow$ (clarinet + vibraphone) model, then running FAD on all generated mixture tracks.

Once more, the noise schedule selected by BDDM demonstrates superior performance in FAD metrics when paired with VGGish and CLAP embeddings. However, the findings regarding PANN embeddings remain inconclusive, as the use of the BDDM-generated noise schedule sometimes leads to either improvement or deterioration. Additionally, enhancements in quality are evident with ViSQOL, yet outcomes with PEAQ lack decisiveness.

Comparing $\text{WT}^{16}_{\text{global}}$ with $\text{WT}^{16}_{\text{mix}}$ on one end, and $\text{WT}^{44}_{\text{global}}$ alongside $\text{WT}^{44}_{\text{mix}}$ on the other, showcases the efficacy of the approach without stems, eliminating the need for individual tracks featuring single instruments during timbre transfer within mixture compositions.

Compared to the baseline models, WaveTransfer surpasses Music-STAR across all metrics except FAD when utilizing PANN embeddings. Notably, WaveTransfer demonstrates smaller standard deviations, indicating a more consistent and stable generation process. Additionally, WaveTransfer performances approach those of DiffTransfer while our models employ a single model trained for both timbre transformations, contrasting with the need for separate models in DiffTransfer.

A subjective evaluation was performed using a MUSHRA test [31]. The results, available on our demo page, clearly demonstrate the superiority of our model over the baseline models. The discrepancy with the objective FAD scores can be attributed to the fact that FAD scores do not consistently align with human perception. This misalignment has been noted in previous studies [32, 33], where the choice of embedding significantly influences the results.

5.3. Model complexity

The WaveTransfer model has 15.92 M parameters.

Concerning the time complexity, the WaveTransfer models trained at 16 kHz generate at speeds of $\times 36.21$ times faster than real-time with a 6-iteration noise schedule and $\times 9.65$ with a 20-iteration schedule. Comparatively, WaveTransfer models trained at 44.1 kHz demonstrate speeds of $\times 14.05$ (6-iteration noise schedule), $\times 3.93$ (19-iteration noise schedule), and $\times 3.72$ (20-iteration noise schedule) times faster than real-time.

6. CONCLUSION & FUTURE WORK

In this work, we introduced WaveTransfer, an end-to-end diffusion-based model designed for timbre transfer across both monophonic and polyphonic music, leveraging multi-instrument training. We effectively demonstrated the versatility and efficacy of our model by showcasing its performance across various sampling rates. Additionally, we incorporated the BDDM approach to enhance noise selection efficiency. By carefully choosing a fitting metric for noise schedule selection with BDDM, or by delving into alternative methods for determining inference noise schedules, we believe that there is ample room for enhancing the outcomes even further.

A constraint in our current methodology is the necessity for transferred timbre pairs to be disjoint. For example, if piano $\leftrightarrow$ vibraphone is a designated pair in the dataset, no other pair should involve either the piano or vibraphone. To address this limitation, our forthcoming efforts will focus on broadening the model's capabilities to encompass a wider array of instruments. This involves conditioning the network on instrument embeddings, enabling it to facilitate any-to-any timbre transfer.

7. REFERENCES

[1] Gino Brunner, Yuyi Wang, Roger Wattenhofer, and Sumu Zhao, "Symbolic music genre transfer with cyclegan," *Proc. ICTAI*, 2018.

[2] Kristy Choi, Curtis Hawthorne, Ian Simon, Monica Dinculescu, and Jesse Engel, "Encoding musical style with transformer autoencoders," in *Proc. ICML*, 2020.

[3] Rui Guo, Ivor Simpson, Chris Kiefer, Thor Magnusson, and Dorien Herremans, "Musiac: An extensible generative framework for music infilling applications with multi-level control," in *Proc. EvoMUSART*, 2022.

[4] Zachary Wallmark, "The Meaning of Timbre," in *Nothing but Noise: Timbre and Musical Meaning at the Edge*. Oxford University Press, 2022.

[5] Ondřej Cífka, Umut Şimşekli, and Gaël Richard, "Groove2groove: One-shot music style transfer with supervision from synthetic data," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, 2020.

[6] Noam Mor, Lior Wolf, Adam Polyak, and Yaniv Taigman, "Autoencoder-based music translation," in *Proc. ICLR*, 2019.

[7] Adrien Bitton, Philippe Esling, and Axel Chemla-Romeu-Santos, "Modulated variational auto-encoders for many-to-many musical timbre transfer," *arXiv preprint arXiv:1810.00222*, 2018.

[8] Ondřej Cífka, Alexey Ozerov, Umut Şimşekli, and Gaël Richard, "Self-supervised vq-vae for one-shot music style transfer," in *Proc. ICASSP*, 2021.

[9] Yin-Jyun Luo, Kat Agres, and Dorien Herremans, "Learning disentangled representations of timbre and pitch for musical instrument sounds using gaussian mixture variational autoencoders," in *Proc. ISMIR*, 2019.

[10] Sicong Huang, Qiyang Li, Cem Anil, Xuchan Bao, Sageev Oore, and Roger B. Grosse, "Timbretron: A wavenet(cycleGAN(CQT(audio))) pipeline for musical timbre transfer," in *Proc. ICLR*, 2019.

[11] Michael Michelashvili and Lior Wolf, "Hierarchical timbre-painting and articulation generation," in *Proc. ISMIR*, 2020.

[12] Yuxuan Wu, Yifan He, Xinlu Liu, Yi Wang, and Roger B. Dannenberg, "Transplayer: Timbre style transfer with flexible timbre control," in *Proc. ICASSP*, 2023.

[13] Vadim Popov, Amantur Amatov, Mikhail A. Kudinov, Vladimir Gogoryan, Tasnima Sadekova, and Ivan Vovk, "Optimal transport in diffusion modeling for conversion tasks in audio domain," *Proc. ICASSP*, 2023.

[14] Mahshid Alinoori and Vassilios Tzerpos, "Music-star: a style translation system for audio-based re-instrumentation," in *Proc. ISMIR*, 2022.

[15] Luca Comanducci, Fabio Antonacci, and Augusto Sarti, "Timbre transfer using image-to-image denoising diffusion implicit models," in *Proc. ISMIR*, 2023.

[16] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan, "Neural audio synthesis of musical notes with WaveNet autoencoders," in *Proc. ICML*, 2017.

[17] Jiaming Song, Chenlin Meng, and Stefano Ermon, "Denoising diffusion implicit models," in *Proc. ICLR*, 2021.

[18] Mahshid Alinoori and Vassilios Tzerpos, "Starnet," Available at https://doi.org/10.5281/zenodo. 6917099, August 2022.

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel, "Denoising diffusion probabilistic models," in *Proc. NeurIPS*, 2020.

[20] Max W. Y. Lam, Jun Wang, Dan Su, and Dong Yu, "BDDM: Bilateral denoising diffusion models for fast and high-quality speech synthesis," in *Proc. ICLR*, 2022.

[21] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan, "Wavegrad: Estimating gradients for waveform generation," in *Proc. ICLR*, 2021.

[22] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proc. AAAI*, 2017.

[23] Max W. Y. Lam, Jun Wang, Dan Su, and Dong Yu, "Effective low-cost time-domain audio separation using globally attentive locally recurrent networks," in *Proc. SLT*, 2021.

[24] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi, "Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms," in *Proc. Interspeech*, 2019.

[25] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin Wilson, "Cnn architectures for large-scale audio classification," in *Proc. ICASSP*, 2017.

[26] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, 2019.

[27] Yusong Wu, K. Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," *Proc. ICASSP*, 2022.

[28] Thilo Thiede, William Treurniet, Roland Bitto, Chris Schmidmer, Thomas Sporer, John Beerends, Catherine Colomes, Michael Keyhl, Gerhard Stoll, Karlheinz Brandenburg, and Bernhard Feiten, "Peaq—the itu standard for objective measurement of perceived audio quality," *J. Audio Eng. Soc.*, 2000.

[29] Ashvala Vinay and Alexander Lerch, "Aquatk: An audio quality assessment toolkit," *arXiv preprint arXiv:2311.10113*, 2023.

[30] Michael Chinen, Felicia S. C. Lim, Jan Skoglund, Nikita Gureev, Feargus O'Gorman, and Andrew Hines, "Visqol v3: An open source production ready objective speech and audio metric," in *Proc. QoMEX*, 2020.

[31] Michael Schoeffler, Sarah Bartoschek, Fabian-Robert Stöter, Marlene Roess, Susanne Westphal, Bernd Edler, and Juergen Herre, "webmushra — a comprehensive framework for web-based listening tests," *J. Open Res. Softw.*, 2018.

[32] Azalea Gui, Hannes Gamper, Sebastian Braun, and Dimitra Emmanouilidou, "Adapting frechet audio distance for generative music evaluation," in *Proc. ICASSP*, 2024.

[33] Modan Tailleur, Junwon Lee, Mathieu Lagrange, Keunwoo Choi, Laurie M Heller, and et al., "Correlation of fréchet audio distance with human perception of environmental audio is embedding dependent," in *Proc. EUSIPCO*, 2024.