ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10889683

Learning Control of Neural Sound Effects Synthesis from Physically Inspired Models

Yisu Zong
Centre for Digital Music
Queen Mary University of London
London, United Kingdom
y.zong@qmul.ac.uk

Joshua Reiss
Centre for Digital Music
Queen Mary University of London
London, United Kingdom
joshua.reiss@qmul.ac.uk

***Abstract*—Sound effects model design commonly uses digital signal processing techniques with full control ability, but it is difficult to achieve realism within a limited number of parameters. Recently, neural sound effects synthesis methods have emerged as a promising approach for generating high-quality and realistic sounds, but the process of synthesizing the desired sound poses difficulties in terms of control. This paper presents a real-time neural synthesis model guided by a physically inspired model, enabling the generation of high-quality sounds while inheriting the control interface of the physically inspired model. We showcase the superior performance of our model in terms of sound quality and control.**

***Index Terms*—Sound effects generation, Controllable sound synthesis, Physically inspired models**

I. Introduction

Sound effects play a crucial role in the field of sound design and production. Conventionally, the predominant approaches for their utilization are based on editing recorded audio. However, as there is an increasing demand for richer sound effects, the limitations of this time-consuming and constrained method have become increasingly apparent. Alternatively, modelling the physical phenomena of sound effects could provide a large number of variations based on the control parameters. Due to the complexity of modelling an entire physical environment, *physically inspired models* are often preferred in practical implementations of procedural audio [1]. This approach utilizes fundamental digital signal processing (DSP) components to perform simplified and approximate calculations of the physical system, incorporating both perceptually and physically meaningful controls that enable real-time generation. Given the trade-off between sound realism and computational complexity, they often incorporate numerous free parameters that pose challenges for optimization. A common way to improve the model performance is by exposing more parameters but at the expense of reduced ease of control [2].

In recent years, data-driven neural sound synthesis has been the mainstream direction of academic research in sound synthesis, including Generative Adversarial Networks (GANs) [3], autoregressive models [4], autoencoders [5], diffusion models [6], and show high potentials for generating realistic sounds. However, due to the limited interpretability of neural network models, controllable neural sound effects synthesis and its control mode remain an open question. One approach involves manipulating the output randomness [7] or latent space [8] of the model to obtain variations in target sounds; however, complete control over the generation direction is still elusive. Another common strategy is leveraging category labels associated with data, such as shoe type and ground surface for footstep sounds [9] or emotions for knocking sounds [10]. This method is constrained by the availability of labelled data and often relies on discrete labels only. Furthermore, directly controlling high-level audio features extracted from data could provide an intuitive control mode, such as loudness [11], [12], pitch [11] or other timbre features [13], [14]. However, this approach may not be optimal for explicit control since it does not directly reflect the physical process underlying sound generation.

The integration of control capabilities from physically inspired models with the generative potential of neural synthesis holds promise as a sound synthesis method. Physics priors could provide reliable and structured information to neural sound synthesis, e.g., ground reaction force curve of footstep sound [15], or object interaction and resonance parameters of impact sound [16], where the control parameters are generally extracted from well-defined physical equations. On the other hand, neural synthesis with the capacity to capture intricate sound details could serve as an auxiliary component within the system, enhancing the sound quality of physically inspired models without directly optimizing their complex inner structure.

In this paper, we propose a neural sound effects synthesis system with an explicit control interface based on an example of a physically inspired explosion model. We first use synthesized sounds for training, and a latent discriminator is introduced to disentangle synthesized audio representations and the control behavior. Then, we compare two methods to perform the transfer to real sounds: supervised transfer using pseudo-label and an unsupervised transfer using CycleGAN [17]. We conduct evaluations on both audio quality and control ability to demonstrate the effectiveness of our proposed method.

II. Proposed Method

A. Explosion Model

We use a physically inspired explosion model (PM) as an example in our experiment. The design concept is derived

Authorized licensed use limited to: Queen Mary University of London. Downloaded on July 31,2025 at 07:41:52 UTC from IEEE Xplore. Restrictions apply.

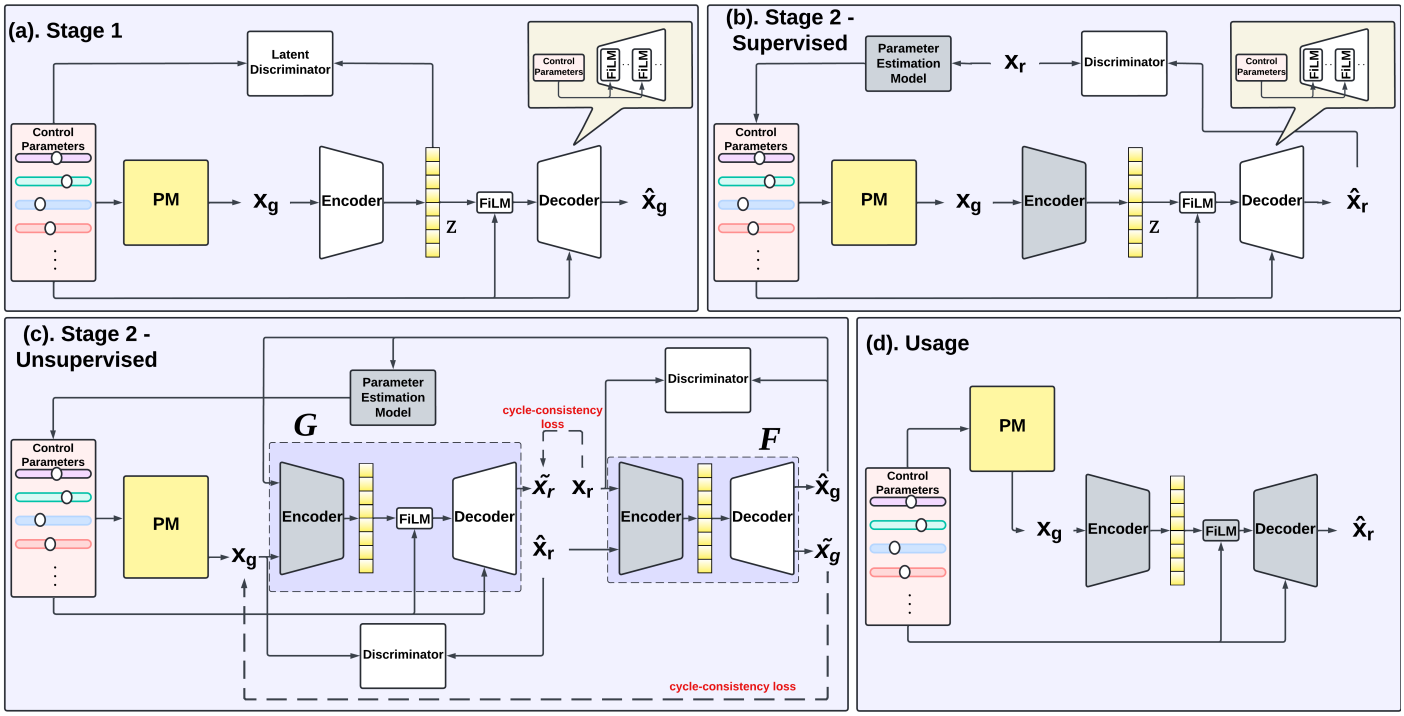

Fig. 1. Flow diagram of proposed methods. Grey boxes represent frozen networks. (a). Representation learning stage for synthesized sounds x_g by the PM, achieving disentangled control facilitated by the latent discriminator. (b). Supervised transfer from x_g to real-world sounds x_r using pseudo-parameters obtained by a pre-trained parameter estimation model. (c). Unsupervised transfer from x_g to x_r by CycleGAN. (d). Utilization of the proposed model. Control parameters and their corresponding x_g as inputs of the model to obtain x_r.

from the implementation by Andy Farnell [18], and the model can satisfy real-time queries[1]. The explosion sound is dominated by three main parts: rumble amount, air amount and dust amount, with eight continuous value control parameters: "Rumble", "Rumble Decay", "Air", "Air Decay", "Dust", "Dust Decay", "Time Separation", and "Grit Amount". Further details of this PM can be found in [2], [18].

B. Overall Architecture

Considering the quality of synthesis and its applicability in real-time scenarios, we adopt a similar variational autoencoder (VAE) architecture to that of RAVE [5]. It uses Pseudo Quadrature Mirror filters (PQMF) [19] to decompose the sound into multiple downsampled sub-signals, enabling real-time synthesis speed. The encoder is a convolutional downsampling network, and the decoder first uses convolutional upsampling layers and residual blocks, then waveform, loudness, and noise synthesizer networks are employed to process the signal.

Our purpose is to synthesize real sounds x_r based on a set of continuous control parameters of the PM θ_{x_g}. To accomplish this, it is essential to acquire information about the controls of the PM and transfer them to real-world data. Therefore, we propose a two-stage training process: the first stage involves learning the latent representation and continuous control of PM for reconstruction, while the second stage focuses on translating generated sounds into real sounds.

[1]https://nemisindo.com/models/explosion

1) Learning Representation and Disentangled Control: In the first stage, we train the VAE to reconstruct the generated sound by the PM along with its control parameters. For reconstruction, we aim to optimize a multi-resolution spectral loss [11] $\mathcal{L}_s(x_g, \hat{x}_g)$,

$$\mathcal{L}_s(x_g, \hat{x}_g) = \sum_{i \in N} (\|S(x_g)_i - S(\hat{x}_g)_i\|_1 + \|\log S(x_g) - \log S(\hat{x}_g)_i\|_1) \quad (1)$$

where $S(\cdot)$ is the magnitude spectrogram, and N is a set of Fast Fourier Transform sizes. The corresponding training objective $\mathcal{L}_{\text{vae}}$ is derived from the Evidence Lower Bound (ELBO) [20] as in RAVE [5].

To achieve disentangled control, the parameters should serve as additional inputs to the decoder. Instead of directly concatenating them with the latent vector z, we utilize feature-wise linear modulation (FiLM) [21] layers to inject the control information into the latent vector and decoder residual blocks.

However, the encoder may have already acquired sufficient information for reconstruction just from the data, leading to the decoder disregarding the control information. To address this issue, we employ a latent discriminator [22] that compels the encoder E to learn a representation without any control information. This discriminator D takes z as input and aims to output accurate control parameters θ_{x_g}, while the encoder aims to remove relevant information accordingly.

The original latent discriminator [22] was designed for handling binary values; and in [14], [23], a multivariate dis-

Authorized licensed use limited to: Queen Mary University of London. Downloaded on July 31,2025 at 07:41:52 UTC from IEEE Xplore. Restrictions apply.

criminator was introduced to handle real values by partitioning the control parameter range into multiple equal segments and predicting the correct segment. In our case, our discriminator directly outputs the probability distributions for all the segments, so the loss of discriminator loss $\mathcal{L}(D;E)$ and its corresponding generator loss for the encoder $\mathcal{L}(E;D)$ are

$$\mathcal{L}(D;E) = -\mathbb{E}[log(p(\theta_{x_g}|z))] \tag{2}$$

$$\mathcal{L}(E;D) = -\mathbb{E}[log(1 - p(\theta_{x_g}|z))] \tag{3}$$

The total loss for our model at this stage is

$$\mathcal{L} = \mathcal{L}_{\text{vae}} + \mathcal{L}(E;D) \tag{4}$$

This stage is depicted in Figure 1(a). We train the model until the convergence of this loss, and subsequently freeze the encoder. Previous studies [5], [14] have incorporated an additional adversarial fine-tuning stage to enhance sound quality. However, in our case, we observed that satisfactory performance was achieved after stage 1, rendering adversarial training unnecessary.

2) Transfer to Real Sound: After completing stage 1 training, our model can be considered as a neural proxy of the PM model with disentangled controls. In the second stage, we further enhance sound realism by training our decoder on real sound data. The primary challenge in this transfer lies in the absence of ground-truth control parameters for real sounds due to misalignment between control parameters and any data labels. This discrepancy arises from our utilization of simulation-based control parameters rather than relying solely on strict physics. To address this issue, we experiment with both supervised and unsupervised learning modes: supervised learning using pseudo-label and unsupervised learning using CycleGAN [17].

Supervised learning using Pseudo-Label: This stage is shown in Figure 1(b). To train our model in a supervised manner, paired data of x_r and x_g is required. We utilize a set of pseudo-parameters for x_r obtained through a sound matching task, i.e., estimating the PM control parameters of the best-matched generated sound for the real one. Following the approach in [2], we train an end-to-end parameter estimation network. It should be noted that this method necessitates a differentiable implementation of the PM; therefore, applying it to non-differentiable models may require alternative derivative-free optimization methods, such as genetic algorithms. The pseudo-parameters θ_{x_r} and their corresponding x_g are used as input to our model, with the reconstruction loss $\mathcal{L}_s(x_r, \hat{x}_r)$ defined in equation (1).

Given our limited training dataset size, we introduce a small random perturbation δ to θ_{x_r} during input processing to obtain $\tilde{x_r}$, aiming to minimize the loss function $\mathcal{L}_s(x_r, \tilde{x_r})$. Our assumption is that close parameters would generate similar sounds. Additionally, adversarial training is incorporated into this stage using MelGAN's discriminator [24], where both its training objective $\mathcal{L}_{adv}$ and the discriminator feature map loss $\mathcal{L}_{FM}$ are employed. The total loss for this stage can be expressed as

$$\mathcal{L} = \mathcal{L}_s(x_r, \hat{x}_r) + \mathcal{L}_s(x_r, \tilde{x}_r) + \mathcal{L}_{adv} + \mathcal{L}_{FM} \tag{5}$$

Unsupervised learning using CycleGAN: CycleGAN [17] offers an unsupervised approach for image-to-image translation without the need for paired data, and has been successfully applied to sound-to-sound tasks [25]–[27]. Following the method of CycleGAN, as illustrated in Figure 1(c), we train our model G using GAN framework to map input x_g onto x_r, simultaneously training another original RAVE model F to map input x_r onto x_g. The translation cycle should ensure the *cycle-consistency*: mapping back from output space brings back to the original input space, i.e. $F(G(x_g, \theta_{x_g})) = x_g$, and similarly for the reverse cycle: $G(F(x_r), \theta_{\hat{x}_g}) = x_r$, where $\theta_{\hat{x}_g}$ is obtained by a parameter estimation network [2] exclusively trained on x_g. The cycle-consistency loss serves as the training objective:

$$\mathcal{L}_{cycle} = \|F(G(x_g, \theta_{x_g})) - x_g\|_1 + \|G(F(x_r), \theta_{\hat{x}_g}) - x_r\|_1 \tag{6}$$

We employ the identical GAN training objective as in the above supervised learning approach, thereby the total loss is

$$\mathcal{L} = \mathcal{L}_{cycle} + \mathcal{L}_{adv}(G) + \mathcal{L}_{adv}(F) \tag{7}$$

III. Experiments

A. Data

For stage 1 training, we generated a dataset of 20,000 samples by randomly varying the parameter settings within the predefined range of the PM. All sounds are 3 seconds at the sample rate of 24 kHz.

For the subsequent real data transfer stage, we curated 76 high-quality real explosion sound samples from Pro Sound Effects[2] and BBC Sound Effects[3]. Our PM model is designed to generate a single explosion without including any environmental reflections or interactive effects such as glass shattering or secondary impacts caused by the explosion. Therefore, our data collection process adheres to the standard of avoiding obvious echoes and other interactive effects. Additionally, all collected samples were trimmed or zero-padded to maintain a consistent duration of 3 seconds at the sample rate of 24 kHz.

B. Baselines

To evaluate audio quality, we compared our supervised learning method (Supervised), CycleGAN method (Unsupervised), the original PM, and an enhanced version of the PM (PM-24params) [2], wherein 24 parameters within PM were directly exposed.

C. Evaluation Metrics

We evaluated our model's generated sound quality and control ability. For sound quality evaluation, we adopted Fréchet Audio Distance (FAD) [28], Maximum Mean Discrepancy (MMD) [29], and mel-cepstral distortion (MCD) [30].

To assess the control capability, we employed Spearman's rank correlation coefficient to evaluate the relationship between high-level audio features in the original PM outputs

[2]https://www.prosoundeffects.com/hybrid-library/

[3]https://sound-effects.bbcrewind.co.uk/

Authorized licensed use limited to: Queen Mary University of London. Downloaded on July 31,2025 at 07:41:52 UTC from IEEE Xplore. Restrictions apply.

and our model outputs with identical control parameters. Specifically, we selected *Boominess, Brightness, Roughness*, and *Depth* from the Audio Commons project[4] as our target audio features of interest.

IV. Results

A. Audio Quality

We compared the sound matching results of PM (i.e. inputs of Supervised), PM-24params, and reconstruction quality of *Supervised*. Also, we are interested in the overall audio quality with random parameters, since it could represent our model's reliability in generating realistic sounds. We compared quality with random parameters of PM, Supervised and Unsupervised across FAD and MMD as introduced in Section III-C. Moreover, due to the limited size of our dataset, the pseudo-parameters derived from real data fail to encompass the entire parameter range. Consequently, we also evaluated the audio quality with random interpolated parameters within the pseudo parameter range of Supervised (Supervised (interpolation)). The results are shown in Table I, and we encourage readers to access our demo website[5] for subjective evaluation.

The Supervised method shows a major improvement compared with PM and PM-24params, indicating the neural network's efficiency in improving the design of a physically inspired model. For random control parameters, we observe that the Supervised (interpolation) has a pretty good performance, but this performance cannot extrapolate to the full parameter range, and the audio quality is even worse than PM. The Unsupervised method shows a stable performance in the entire parameter range but is slightly worse than the Supervised (interpolation). This is consistent with the expected since it can explore the parameter space more freely during training.

B. Controls

We compared the correlation between PM and Supervised, Supervised (interpolation), and Unsupervised, using randomly selected values for all control parameters with 100 samples. The results are shown in Table II. Additionally, we investigated the correlation when changing a single parameter while keeping all other parameters fixed. In this scenario, we present the results for Supervised (interpolation) in Table III and Unsupervised in Table IV.

The Supervised (interpolation) method shows the highest correlation with PM, yet it still lacks the ability to extrapolate across the entire parameter range. It demonstrates a significant positive correlation in terms of Roughness and Depth, while encountering challenges in capturing the characteristics of Boominess and Brightness. Similar trends are observed for the Unsupervised method, although there is an overall decrease compared to the Supervised (interpolation) approach. For single-parameter control, Supervised (interpolation) shows comparable performance to the overall correlations across all parameters. However, it is challenging for Unsupervised methods to replicate the same level of single-parameter control ability as PM. Without explicit labels, unsupervised learning has difficulty capturing the detailed relationships between individual parameters and specific sound characteristics, resulting in less effective control compared to the supervised approach.

[4]https://audiocommons.github.io/

[5]https://zys711.github.io/NeuralPM/

TABLE I
Audio Quality Results

	FAD	MMD	MCD
PM	29.29	119.54	1.20
PM-24params	17.26	65.97	1.05
Supervised	**5.21**	**22.77**	**0.60**
PM (random)	30.95	163.49	-
Supervised (interpolation)	**8.87**	**58.35**	-
Supervised (random)	37.76	190.09	-
Unsupervised (random)	12.71	95.54	-

TABLE II
All parameters Control Correlations

	Boominess	Brightness	Roughness	Depth
Supervised (interpolation)	**0.70**	**0.66**	**0.86**	**0.95**
Supervised (random)	0.03	0.18	0.52	0.40
Unsupervised (random)	0.16	0.33	0.64	0.91

TABLE III
(Supervised(interpolation)) Single-Parameter control Correlations

	Boominess	Brightness	Roughness	Depth
Rumble	0.96	0.80	0.71	0.88
Rumble Decay	0.75	0.44	0.71	0.73
Air	0.71	0.47	0.85	0.94
Air Decay	0.46	0.38	0.70	0.95
Dust	0.42	0.35	0.70	0.95
Dust Decay	0.44	0.34	0.70	0.90
Time Separation	0.38	0.52	0.59	0.88
Grit Amount	0.34	0.67	0.60	0.87

TABLE IV
(Unsupervised) Single-Parameter control Correlations

	Boominess	Brightness	Roughness	Depth
Rumble	-0.08	-0.04	0.78	0.93
Rumble Decay	-0.23	-0.15	0.88	0.27
Air	0.44	0.85	-0.81	-0.51
Air Decay	0.91	0.94	-0.80	0.80
Dust	0.54	0.60	-0.55	-0.76
Dust Decay	-0.37	-0.31	0.81	0.90
Time Separation	-0.38	0.21	0.82	0.91
Grit Amount	0.44	0.28	0.35	0.86

V. Conclusion

We presented a real-time neural sound effects synthesis system that combines intuitive control from physically inspired models with the high-quality output of neural networks. The supervised method excels in terms of both quality and control within interpolated parameters, while the unsupervised method consistently delivers high audio quality performance across the entire parameter space at the expense of sacrificing fine-grained control. This integration of physically inspired models and neural networks offers a promising solution for achieving both control and realism in sound model design.

Authorized licensed use limited to: Queen Mary University of London. Downloaded on July 31,2025 at 07:41:52 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] D. Menexopoulos, P. Pestana, and J. Reiss, "The state of the art in procedural audio," *Journal of the Audio Engineering Society*, no. 12, pp. 826–848, 2023.

[2] Y. Zong, N. Garcia-Sihuay, and J. Reiss, "A machine learning method to evaluate and improve sound effects synthesis model design," in *Audio Engineering Society Conference: AES International Audio for Games Conference*, 2024.

[3] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," in *International Conference on Learning Representations*, 2018.

[4] A. v. d. Oord *et al.*, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[5] A. Caillon and P. Esling, "Rave: A variational autoencoder for fast and high-quality neural audio synthesis," *arXiv preprint arXiv:2111.05011*, 2021.

[6] Y. Chung, J. Lee, and J. Nam, "T-foley: A controllable waveform-domain diffusion model for temporal-event-guided foley sound synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 6820–6824.

[7] S. Andreu and M. V. Aylagas, "Neural synthesis of sound effects using flow-based deep generative models," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 18, no. 1, 2022, pp. 2–9.

[8] L. Wyse, P. Kamath, and C. Gupta, "Sound model factory: An integrated system architecture for generative audio modelling," in *International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar)*, 2022, pp. 308–322.

[9] M. Comunità, H. Phan, and J. D. Reiss, "Neural synthesis of footsteps sound effects with generative adversarial networks," in *Audio Engineering Society Convention 152*, 2022.

[10] A. Barahona-Rios and S. Pauletto, "Synthesising knocking sound effects using conditional wavegan," in *SMC Sound and Music Computing Conference*, 2020.

[11] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "Ddsp: Differentiable digital signal processing," in *International Conference on Learning Representations*, 2019.

[12] A. Barahona-Ríos and T. Collins, "Noisebandnet: controllable time-varying neural synthesis of sound effects using filterbanks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 1573–1585, 2024.

[13] J. Nistal, S. Lattner, and G. Richard, "Drumgan: Synthesis of drum sounds with timbral feature conditioning using generative adversarial networks," in *International Society for Music Information Retrieval Conference*, 2020.

[14] N. Devis, N. Demerlé, S. Nabi, D. Genova, and P. Esling, "Continuous descriptor-based control for deep audio synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[15] D. Serrano and M. Cartwright, "A general framework for learning procedural audio models of environmental sounds," *arXiv preprint arXiv:2303.02396*, 2023.

[16] P. Kamath, C. Gupta, L. Wyse, and S. Nanayakkara, "Example-based framework for perceptually guided audio texture generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[17] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[18] A. Farnell, *Designing sound*. Mit Press, 2010.

[19] C. Yu *et al.*, "Durian: Duration informed attention network for multimodal synthesis," *arXiv preprint arXiv:1909.01700*, 2019.

[20] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations*, 2014.

[21] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[22] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer, and M. Ranzato, "Fader networks: Manipulating images by sliding attributes," *Advances in neural information processing systems*, vol. 30, 2017.

[23] L. Kawai, P. Esling, and T. Harada, "Attributes-aware deep music transformation." in *International Society for Music Information Retrieval Conference*, 2020, pp. 670–677.

[24] K. Kumar *et al.*, "Melgan: Generative adversarial networks for conditional waveform synthesis," *Advances in neural information processing systems*, vol. 32, 2019.

[25] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Cyclegan-vc2: Improved cyclegan-based non-parallel voice conversion," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6820–6824.

[26] S. Huang, Q. Li, C. Anil, X. Bao, S. Oore, and R. B. Grosse, "Timbretron: A wavenet (cyclegan (cqt (audio))) pipeline for musical timbre transfer," in *International Conference on Learning Representations*, 2019.

[27] J. Yang, T. Cinquin, and G. Sörös, "Unsupervised musical timbre transfer for notification sounds," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 3735–3739.

[28] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms," in *Interspeech*, 2019.

[29] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

[30] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, vol. 1, 1993, pp. 125–128.

Authorized licensed use limited to: Queen Mary University of London. Downloaded on July 31,2025 at 07:41:52 UTC from IEEE Xplore. Restrictions apply.