One-shot Text-aligned Virtual Instrument Generation Utilizing Diffusion Transformer

Qihui Yang*
Department of Electrical and Computer Engineering
University of California San Diego
La Jolla, CA 92093, US
qiy009@ucsd.edu

Jiahe Lei*
School of Computer and Communication Engineering
University of Science and Technology Beijing
Beijing, China
u202140450@xs.ustb.edu.cn

Qiuqiang Kong
Department of Electronic Engineering
The Chinese University of Hong Kong
Hong Kong SAR, China
qqkong@ee.cuhk.edu.hk

Abstract

Despite the success of emerging text-to-music models based on deep generative approaches in generating music clips for general audiences, they face significant limitations when applied to professional music production.

This paper introduces a one-shot Text-aligned Virtual Instrument Generation model using a Diffusion Transformer (TaVIG). The model integrates textual descriptions with the timbre information of audio clips to generate musical performances, utilizing additional musical structure features such as pitch, onset, duration, offset, and velocity. TaVIG comprises a CLAP-based text-aligned timbre extractor-encoder, a musical structure encoder for extracting MIDI information, and a disentangled representation learning module to ensure effective timbre and structure extraction. The audio synthesis process is based on a Diffusion Transformer conditioned with AdaLN. Additionally, we propose a mathematical framework to analyze timbre and structure disentanglement in MIDI-to-audio tasks.

1 Introduction

Deep learning-based music generation has garnered growing interest in the machine learning community, particularly with the advent of advanced music generation systems [1][2]. However, professional musicians often find such models impractical due to their lack of control over the intermediate generation process. Musicians frequently use virtual instruments, but these tools often come with financial constraints, long tuning periods, and limited natural performance details.

*equal contribution

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Recent research focuses on MIDI-to-audio models for neural audio synthesis, which can be categorized into synthesizer-based sound matching models, DDSP-based models [11], and spectrogram generation models. Section 2 will elaborate on related work.

Synthesizer sound matching models depend on pre-existing synthesizers, limiting the variety of timbres and struggling to emulate acoustic sounds. DDSP is promising but confines outputs to specific models like the Sinusoidal plus Noise model.

However, existing models face several challenges:

- **Structure Misalignment:** Polyphonic models may generate extra tracks or misalign instruments with the original MIDI (e.g., [31], [28]).
- **Poor Sound Quality:** Some models produce blurry or artifact-ridden sounds, such as violin audio in [28].
- **Inconsistent Timbre:** Instruments like guitars sometimes sound like other instruments, as seen in [21].
- **Limited Instrument Variety:** Models are often restricted to instruments they've been trained on.
- **Single Note Synthesis:** Some models fail to capture the complexity of full performances and only synthesize single notes [36, 35].

Our work introduces TaVIG (Text-aligned Virtual Instrument Generative model), which synthesizes musical performances from timbre and structure inputs using a Diffusion Transformer (DiT). By utilizing disentangled representation learning, TaVIG separates timbre and structure to ensure accurate note generation. Our approach allows for one-shot multi-note synthesis, with timbre controlled via audio or text prompts, ensuring flexibility and user intent fidelity.

2 Related Work

Synthesizer sound matching estimates parameters to replicate a target sound. Typically, a similar sound is provided, and the system adjusts parameters to match it. Sound2Synth [6] enhances this process by converting sounds into four representations, each processed by dedicated backbones. DDX7 [3] and DiffMoog [39] are based on Yamaha DX7 and Moog synthesizers, while Masuda and Saito [33] integrate sound matching with DDSP for subtractive synthesis.

RNNs have been applied to map MIDI to pitch and loudness via DDSP parameter estimation [15, 4, 19]. DDSP-Piano [38] utilizes Multi-Resolution Spectral Loss and DDSP for piano synthesis, and MIDI-DDSP [41] translates expressive MIDI attributes, such as articulation, into synthesis controls.

For spectrogram generation, GAN-based models like GANsynth [10], GANstrument [34], and HyperGANstrument [43] use adversarial training to match data distributions, with GANstrument and HyperGANstrument enabling timbre interpolation.

Auto-regressive models include LSTM-based mel-to-mel conversion conditioned on instrument embeddings [22] and Deep Performer [9], a Transformer leveraging performer and tempo embeddings. Nercessian et al. [36, 35] employed CLAP [40] to align text and timbre embeddings, integrating them into a timbre-generation framework.

Recent advancements in diffusion models have significantly enhanced MIDI-to-audio generation. Hawthorne et al. [13] integrated a Transformer with a diffusion model, which was further developed by Maman et al. [31] and Kim et al. [21], focusing on performance conditioning and guitar generation, respectively. Liu et al. [28] introduced a diffusion-based MIDI-to-audio system, while Demerle et al. [8] employed adversarial training to capture timbre and structure, leveraging diffusion as the generative backbone.

3 Method

In Figure , we present the envisioned four-phase framework for the MIDI-to-Audio model, comprising the following components:

1. a text-aligned timbre extractor-encoder
2. a musical structure encoder
3. a disentangled representation learning module
4. an audio synthesis model

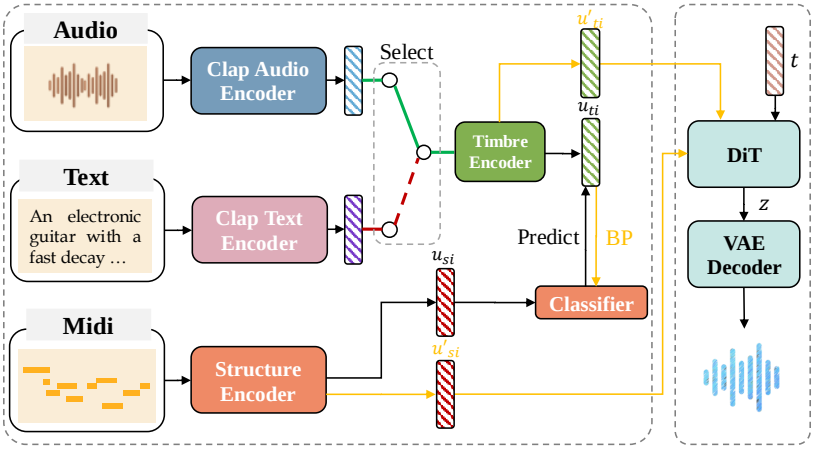

Figure 1: The overall architecture of our model. During training, either text or audio is randomly selected as the timbre prompt corresponding to the MIDI input. The timbre information $\mathbf{u}_{ti}$ is obtained through the pretrained CLAP encoder and Timbre encoder, which serves as the global condition for the DiT model. The MIDI's pianoroll is passed through the Structure encoder to obtain the structural information $\mathbf{u}_{si}$, which is used as the prepend condition for the DiT model. The DiT predicts the VAE latent $\mathbf{z}$ from noise, which is then decoded to generate audio that simultaneously contains both the timbre $\mathbf{u}_{ti}$ and the structure $\mathbf{u}_{si}$. A classifier is used to predict $\mathbf{u}_{ti}$ from the structure $\mathbf{u}_{si}$, ensuring the disentanglement between the two. The black lines show the predictor optimization process while the orange lines show the diffusion training process. BP stands for BackPropogation.

3.1 Notations

Let (Ω_t, F_t, P_t) and (Ω_s, F_s, P_s) are the probability spaces of timbre and structure random variables. Denote $\mathbf{U}_t$ and $\mathbf{U}_s$ random matrices defined in these two probability spaces that containing the timbre and structure information of the same audio clip respectively, and $\mathbf{u}_{ti}$, $\mathbf{u}_{si}$ represent related realization samples. We denote the random matrices defined in the probability space (Ω_w, F_w, P_w) representing audio clips in the form of waveform as $\mathbf{W}_i$ and its realization as $\mathbf{w}_i$. $\mathbf{z} \in \mathbb{R}^{C \times \frac{T}{c} \times \frac{F}{c}}$ is the latent that encoded by the VAE[24], where C, c, T and F represent the number of channels of the compressed latents, compression rate, length of time and frequency. $TE(\cdot)$ and $SE(\cdot)$ represent timbre encoder and structure encoder respectively. B stands for B.

3.2 Text-aligned Timbre Extractor-Encoder

We leverage the CLAP model [40] to generate a standardized 512-dimensional representation for paired audio and text inputs. Pretrained on musical signals with a contrastive loss function, this model aligns audio and text embeddings, enabling either modality to serve interchangeably as input to our system. The audio encoder, E_a, uses the HTS-AT architecture [5], while the text encoder, E_t, is based on RoBERTa [29]. We further finetune this model on a custom dataset of timbre-related texts paired with descriptive audio clips.

This approach allows us to work exclusively with audio data during language model training, avoiding the need for extensive text annotations in the audio dataset. Given the high computational demand of training CLAP from scratch, we instead employ a pretrained CLAP model as a cross-modality encoder, which we finetune on our timbre text dataset. The resulting 512-dimensional vector is subsequently fed, after a linear projection, into a timbre structure following the method in [8].

3.3 Musical Structure Encoder

We use music files in MIDI format, which is a symbolic representation of music, similar to a musical score. A MIDI file consists of multiple tracks, each of which is assigned to a specific instrument. We regard the MIDI sequence S of a single track as structural information. We utilize the `pretty_midi` library to read MIDI files and extract the discrete piano roll sequence $s \in [0,1]^{128\times T}$, where 128 represents the standard MIDI pitch range from 0 to 127 (corresponding to MIDI note numbers), and T is the length of the discretized MIDI sequence.

We selected the same encoder network as in previous work[8], which is composed of multiple convolutional blocks. Each convolutional block processes the input through BatchNorm1d, SiLU activation, Conv1d layers, Dropout, and skip connections, progressively extracting temporal features. The number of channels begins at 128, corresponding to the MIDI pitch, and gradually increases through the convolutional layers until the output is a feature with u_t channels.

3.4 Audio Synthesis with Latent Diffusion Transformer

In traditional Latent Diffusion Model used in audio synthesis task, the original waveform $\mathbf{x} \in \mathbb{R}^{\mathbb{T}}$ is converted into a mel-spectrogram $\mathbf{X} \in \mathbb{R}^{T\times F}$ using short time Fourier transformation (STFT) and mel filter bank. Then the mel-spectrogram is sent into a variational autoencoder (VAE) to obtain a latent representation $\mathbf{z} \in \mathbb{R}^{C\times \frac{T}{c}\times \frac{F}{c}}$, which can also be recognized as $\mathbf{z}_0$ in the language of diffusion models.

In the forward process of diffusion models [17], which is the training phase, the model gradually add noise to the ground truth $\mathbf{z}_0$ by sampling from the following distributions:

$$q\left(\boldsymbol{z}_n \mid \boldsymbol{z}_{n-1}\right)=\mathcal{N}\left(\boldsymbol{z}_n ; \sqrt{1-\beta_n} \boldsymbol{z}_{n-1}, \beta_n \boldsymbol{I}\right)$$
$$q\left(\boldsymbol{z}_n \mid \boldsymbol{z}_0\right)=\mathcal{N}\left(\boldsymbol{z}_n ; \sqrt{\bar{\alpha}_n} \boldsymbol{z}_0,\left(1-\bar{\alpha}_n\right) \boldsymbol{\varepsilon}\right)$$

where α_i and β_i, $i \in \{1,\dots,n\}$ are parameter related to the assumption of the diffusion models, $\mathcal{N}(\mathbf{v};\mu,\mathbf{\Sigma})$ represent the probability density function of the random variable $\mathbf{v}$ which follows the distribution of $\mathcal{N}(\mu,\mathbf{\Sigma})$. $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ and the final step of the representation $\mathbf{z}_n \sim \mathcal{N}(\mathbf{0},\mathbf{I})$. In the training phase, a denoising neural network $\boldsymbol{\varepsilon}_\theta$ is trained to predict $\mathbf{z}_0$ from the noised latent $\mathbf{z}_n$ on the following diffusion loss:

$$L_{DiT}=\mathbb{E}_{\boldsymbol{z}_0,\boldsymbol{\varepsilon},n}\left\|\boldsymbol{\varepsilon}-\boldsymbol{\varepsilon}_\theta\left(\boldsymbol{z}_n, n, \mathbf{g}, \mathbf{p}\right)\right\|_2^2 \tag{1}$$

where $\mathbf{g}$ and $\mathbf{p}$ stand for global condition and prepend condition in DiT respectively.

Compared to other diffusion models, DiTs shows its edge on effectively capturing data dependencies and producing high-quality results. Recent studies highlight DiTs' impressive image generation capabilities, especially through techniques like Stable Diffusion 3, and their success in video generation, demonstrated by the Sora framework [30]. As [27] had shown, Diffusion Transformer (DiT) shows great capacity of preserving the quality of the generated audio.

The technical details about encoder, decoder and the DiT block is coverd in A.2.

3.5 Disentangled Representation Learning Module

We assume that $\mathbf{U}_t$ and $\mathbf{U}_s$ are not correlated, i.e. $\text{cov}(\mathbf{U}_t, \mathbf{U}_s) = 0$. This assumption is reasonable since correct timbre shift in music performing would not change the the pitch of each note, and vice versa. Therefore, the audio clip $\mathbf{W}_i$ can be formulated into a function of two uncorrelated random matrices:

$$\mathbf{W}_i = f(\mathbf{U}_t, \mathbf{U}_s) \tag{2}$$

where $f : (\Omega_t, F_t, P_t) \times (\Omega_s, F_s, P_s) \mapsto (\Omega_w, F_w, P_w)$.

Ideally , we wish our text-aligned timbre extractor-encoder and structure encoder can serve as the following two functions:

$$g_t(\mathbf{W}_i) = g_t(f(\mathbf{U}_t, \mathbf{U}_s)) = \mathbf{U}_t, \tag{3}$$
$$g_s(\mathbf{W}_i) = g_s(f(\mathbf{U}_t, \mathbf{U}_s)) = \mathbf{U}_s \tag{4}$$

If $TE = g_t$ and $SE = g_s$, then the information of timbre and musical structure is completely disentangled. Our timbre encoder TE and structure encoder SE should serve an under-optimized g_t and g_s:

$$TE(\mathbf{w}_i) = TE(f(\mathbf{u}_{ti}, \mathbf{u}_{si})) = \mathbf{u}_{ti}, \tag{5}$$

$$SE(\mathbf{w}_i) = SE(f(\mathbf{u}_{ti}, \mathbf{u}_{si})) = \mathbf{u}_{si} \tag{6}$$

Therefore, our goal in this disentangled representation learning module is to minimize the correlation of embedding provided by TE and SE. Inspired by [8], we optimize a predictor function $h(\cdot)$ trying to recover samples of $\mathbf{u}_{ti}$ only using the information in $\mathbf{u}_{si}$, $i \in \{1, \ldots, \mathrm{B}\}$ and use it as an negative guidance, i.e. preventing $h(\cdot)$ to correctly predict $\mathbf{u}_{ti}$ from $\mathbf{u}_{si}$:

$$L_{pred} = -\mathbb{E}_{\mathbf{U}_t, \mathbf{U}_s} \left[\|\mathbf{U}_t - h(\mathbf{U}_s)\|\right] \tag{7}$$

$$= -\Sigma_{i=1}^{B} \left[\|\mathbf{u}_{ti} - h(\mathbf{u}_{si})\|\right] / \mathrm{B} \tag{8}$$

Overall Architecture In odd number training steps, we fix the TE and SE to optimize the predictor as well as training the DiT. On the contrary, we fix the predictor instead and train TE, SE and the DiT in even number training steps.

$$L = L_{DiT} + \lambda L_{pred} \tag{9}$$

where λ is a hyper parameter controlling learning emphasis on the disentangled representation learning.

4 Experiments

4.1 Datasets

We adopt two synthetic dataset: Slakh2100 and Nysnth for evaluation. For details about the datasets, see A.1.

4.2 Results

Specific training settings are given in A.3. For the VAE, we adopted the architecture and training parameters from Stable Audio Open[12], using a sampling rate of 44,100 Hz but converting from stereo to mono. We trained the model for 50k steps to ensure that the VAE can faithfully reconstruct the instrument sounds.

Condition	Dataset	$\text{FAD}_{\text{Vggish}} \downarrow$	$\text{FAD}_{\text{Encodec}} \downarrow$	Transcription F1 $\uparrow$	Clap Score $\uparrow$
$\text{Clap}_{\text{audio}}$	Slakh	0.51	2.23	0.47	0.090
$\text{Clap}_{\text{text}}$	Slakh	0.25	2.28	0.53	0.135
$\text{Clap}_{\text{audio}}$	Nsynth	0.52	2.34	0.48	0.150
$\text{Clap}_{\text{text}}$	Nsynth	0.29	1.78	0.54	0.152
Ground Truth	-	0.00	1.00	1.00	-

Table 1: Experimental results on audio quality and pitch accuracy. Metrics are computed separately for each evaluation dataset.

5 Conclusions

In this paper, we introduced TaVIG, a one-shot Text-aligned Virtual Instrument Generation model using a Diffusion Transformer. We evaluated TaVIG on both the Slakh and NSynth datasets, using transcription F1 scores to assess the alignment of generated music with the provided MIDI, and FAD scores to measure audio quality.

This work provides a step forward in bridging the need of music production industry and machine learning technology. Future work will focus on refining the disentanglement of centain timbres and structures, improving the diversity of generated timbres but maintaining the loyalty to the text prompts, and extending the model's ability to handle polyphonic and complex instrumentations in real-world scenarios.

References

[1] Suno ai. `https://suno.com/blog/v3`.

[2] Udio. `http://www.udio.com`.

[3] Franco Caspe, Andrew McPherson, and Mark Sandler. Ddx7: Differentiable fm synthesis of musical instrument sounds. In *Ismir 2022 Hybrid Conference*, 2022.

[4] Rodrigo Castellon, Chris Donahue, and Percy Liang. Towards realistic midi instrument synthesizers. In *NeurIPS Workshop on Machine Learning for Creativity and Design*, 2020.

[5] Ke Chen, Xingjian Du, Bilei Zhu, Zejun Ma, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 646–650. IEEE, 2022.

[6] Zui Chen, Yansen Jing, Shengcheng Yuan, Yifei Xu, Jian Wu, and Hang Zhao. Sound2synth: Interpreting sound via fm synthesizer parameters estimation. *arXiv preprint arXiv:2205.03043*, 2022.

[7] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

[8] Nils Demerlé, Philippe Esling, Guillaume Doras, and David Genova. Combining audio control and style transfer using latent diffusion. *arXiv preprint arXiv:2408.00196*, 2024.

[9] Hao-Wen Dong, Cong Zhou, Taylor Berg-Kirkpatrick, and Julian McAuley. Deep performer: Score-to-audio music performance synthesis. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 951–955. IEEE, 2022.

[10] Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. Gansynth: Adversarial neural audio synthesis. In *International Conference on Learning Representations*, 2018.

[11] Jesse Engel, Lamtharn (Hanoi) Hantrakul, Chenjie Gu, and Adam Roberts. Ddsp: Differentiable digital signal processing. In *International Conference on Learning Representations*, 2020.

[12] Zach Evans, Julian D Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open. *arXiv preprint arXiv:2407.14358*, 2024.

[13] Curtis Hawthorne, Ian Simon, Adam Roberts, Neil Zeghidour, Joshua Gardner, Ethan Manilow, and Jesse Engel. Multi-instrument music synthesis with spectrogram diffusion. In *Ismir 2022 Hybrid Conference*, 2022.

[14] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. Enabling factorized piano music modeling and generation with the MAESTRO dataset. In *International Conference on Learning Representations*, 2019.

[15] Ben Hayes, Jordie Shier, György Fazekas, Andrew McPherson, and Charalampos Saitis. A review of differentiable digital signal processing for music and speech synthesis. *Frontiers in Signal Processing*, 3:1284100, 2024.

[16] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE, 2017.

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[18] Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. In *International Conference on Machine Learning*, pages 13916–13932. PMLR, 2023.

[19] Nicolas Jonason. The control-synthesis approach for making expressive and controllable neural music synthesizers, 2020.

[20] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fr\'echet audio distance: A metric for evaluating music enhancement algorithms. *arXiv preprint arXiv:1812.08466*, 2018.

[21] Hounsu Kim, Soonbeom Choi, and Juhan Nam. Expressive acoustic guitar sound synthesis with an instrument-specific input representation and diffusion outpainting. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7620–7624. IEEE, 2024.

[22] Jong Wook Kim, Rachel Bittner, Aparna Kumar, and Juan Pablo Bello. Neural music synthesis for flexible timbre control. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 176–180. IEEE, 2019.

[23] Jong Wook Kim, Justin Salamon, Peter Li, and Juan Pablo Bello. Crepe: A convolutional representation for pitch estimation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 161–165. IEEE, 2018.

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[25] Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved rvqgan. *Advances in Neural Information Processing Systems*, 36, 2024.

[26] Bochen Li, Xinzhao Liu, Karthik Dinesh, Zhiyao Duan, and Gaurav Sharma. Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications. *Trans. Multi.*, 21(2):522–535, feb 2019.

[27] Chang Li, Ruoyu Wang, Lijuan Liu, Jun Du, Yixuan Sun, Zilu Guo, Zhenrong Zhang, and Yuan Jiang. Qa-mdt: Quality-aware masked diffusion transformer for enhanced music generation, 2024.

[28] Kaiyang Liu, Wendong Gan, and Chenchen Yuan. Maid: A conditional diffusion model for long music audio inpainting. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[29] Yinhan Liu. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[30] Xudong Lu, Aojun Zhou, Ziyi Lin, Qi Liu, Yuhui Xu, Renrui Zhang, Yafei Wen, Shuai Ren, Peng Gao, Junchi Yan, and Hongsheng Li. Terdit: Ternary diffusion models with transformers, 2024.

[31] Ben Maman, Johannes Zeitler, Meinard Müller, and Amit H Bermano. Performance conditioning for diffusion-based multi-instrument music synthesis. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5045–5049. IEEE, 2024.

[32] Ethan Manilow, Gordon Wichern, Prem Seetharaman, and Jonathan Le Roux. Cutting music source separation some Slakh: A dataset to study the impact of training data quality and quantity. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019.

[33] Naotake Masuda and Daisuke Saito. Synthesizer sound matching with differentiable dsp. In *ISMIR*, pages 428–434, 2021.

[34] Gaku Narita, Junichi Shimizu, and Taketo Akama. Ganstrument: Adversarial instrument sound synthesis with pitch-invariant instance conditioning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[35] Shahan Nercessian, Johannes Imort, Ninon Devis, and Frederik Blang. Generating sample-based musical instruments using neural audio codec language models. *arXiv preprint arXiv:2407.15641*, 2024.

[36] Shahan Nercessian, Johannes Imort, and Native Instruments. Instrumentgen: Generating sample-based musical instruments from text. In *Neural Information Processing Systems Workshop on Machine Learning for Audio*, 2023.

[37] Colin Raffel, Brian McFee, Eric J Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, Daniel PW Ellis, and C Colin Raffel. Mir_eval: A transparent implementation of common mir metrics. In *ISMIR*, volume 10, page 2014, 2014.

[38] Lenny Renault, Rémi Mignot, and Axel Roebel. Differentiable piano model for midi-to-audio performance synthesis. In *25th International Conference on Digital Audio Effects (DAFx20in22)*, 2022.

[39] Noy Uzrad, Oren Barkan, Almog Elharar, Shlomi Shvartzman, Moshe Laufer, Lior Wolf, and Noam Koenigstein. Diffmoog: a differentiable modular synthesizer for sound matching. *arXiv preprint arXiv:2401.12570*, 2024.

[40] Yusong Wu, Ke 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. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[41] Yusong Wu, Ethan Manilow, Yi Deng, Rigel Swavely, Kyle Kastner, Tim Cooijmans, Aaron Courville, Cheng-Zhi Anna Huang, and Jesse Engel. Midi-ddsp: Detailed control of musical performance via hierarchical modeling. In *International Conference on Learning Representations*, 2022.

[42] Qingyang Xi, Rachel M Bittner, Johan Pauwels, Xuzhou Ye, and Juan P Bello. Guitarset: A dataset for guitar transcription. In *19th International Society for Music Information Retrieval Conference, ISMIR 2018*, pages 453–460. International Society for Music Information Retrieval, 2018.

[43] Zhe Zhang and Taketo Akama. Hyperganstrument: Instrument sound synthesis and editing with pitch-invariant hypernetworks. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6640–6644. IEEE, 2024.

A Appendix

A.1 Dataset

To ensure the model encounters a wide variety of timbres, our approach to dataset selection focuses on including as many timbres as possible, each with detailed text descriptions.

Synthetic Data To ensure the variety of timbre, synthesized data are needed.

- Slakh2100_flac_redux: The Synthesized Lakh Dataset (SLAKH) [32] is a multi-track audio and MIDI dataset for music source separation and multi-instrument transcription. Created from the Lakh MIDI Dataset v0.1 using high-quality virtual instruments, it features 2100 tracks with aligned MIDI, synthesized from 187 patches across 34 instrument classes. For this study, we focus on 400 hours of non-percussive instrument stems.
- The NSynth dataset is a large, structured collection of 300,000 isolated musical notes from 1,000 diverse instruments, with detailed labels for pitch, velocity, instrument type, and acoustic qualities.

Real Data To capture natural performance nuances, we focus on datasets with real audio recordings:

- **Maestro** [14]: A dataset of approximately 200 hours of paired MIDI-audio piano recordings from classical piano competitions, annotated with composer and piece information.
- **Guitarset** [42]: A 6-hour collection of live guitar performances, featuring both solos and accompaniment across various genres and playing styles.
- **URMP** [26]: A dataset of pieces performed by various classical instruments, including brass, woodwinds, and strings. We use approximately 4 hours of monophonic instrumental recordings.

We construct timbre descriptions using the template: A {INSTRUMENT SOURCE} {INSTRUMENT FAMILY} WITH {SOUND QUALITY} PLAYING IN {STYLE}. For example:

- **Instrument Source**: electric, acoustic, synthesized, etc.
- **Instrument Family**: piano, guitar, bass, etc.
- **Sound Quality**: decay, delay, reverb, etc.
- **Style**: Jazz, Flamenco, Blues, etc.

If specific information is unavailable, the corresponding description is omitted.

A.2 Technical Details

The encoder and decoder in our VAE follow the same architecture as DAC[25]. DAC is a neural codec that offers high compression and high fidelity; however, we modify its bottleneck by replacing RVQ with VAE.. Before the conditional embeddings $\mathbf{g}$ and $\mathbf{p}$ are fed into the DiT block, Patchify is conducted to convert the 2D latent representations encoded by the VQ-VAE into an one-dimensional representation. The denoising network ε_θ is formulated as a DiT block with adaLN-Zero conditoning mechanism. The adaLN-Zero mechanism can be decomposed into two main parts, Adaptive Layer Normalization (AdaLN) and the adaZero-Block. The core idea of AdaLN is to use $\mathbf{g}$ and $\mathbf{p}$ to learn two normalization parameters β and γ (not sure), which is obtained

by adding up time slice features t and conditional features β. In addition, DiT also has a regression scaling parameter after each residual connection α. While adaZero-Block initialize some training parameters as zeros to accelerate the training process.

A.3 Training Settings

For the DiT model, we configured a 12-layer transformer with a latent dimension of 768 and trained it for 50k steps using a learning rate of 1e-5. We evaluated the model's performance under different CLAP conditions and across different validation datasets. Since the text descriptions in Nsynth are more detailed and the timbre information is more complete, we also compared the results with those from the synthetic dataset Slakh. For audio quality evaluation, we use the Frechet Audio Distance (FAD) [20] to measure how well the generated audio aligns with the dataset distribution for both reconstruction and transfer tasks. We compute FAD using embeddings from VGGish [16] and Encodec [7].To assess whether the generated audio aligns with the structural information provided by the MIDI, we use CREPE [23] to extract pitch and compute the Onset F1 score using mir_eval[37].We employed CLAP score to evaluate the similarity[40] [18]between the generated audio and its corresponding audio or text prompt. A higher CLAP score indicates stronger alignment of the generated audio with the specified prompt.