LEAP: A DIFFUSION TRANSFORMER FRAMEWORK FOR LEVEL-ADAPTIVE POP-TO-PIANO GENERATION

Emilio Picard, Loïs Guerci, Julien Verstraete, Louis Gosselin
Department of Computer Science
IRCAM, Sorbonne University & Telecom Paris
1 Place Igor Stravinsky, 75004 Paris
{picard, guerci, verstraete, gosselin}@ircam.fr

ABSTRACT

Piano covers of pop music are enjoyed by many people. While previous approaches have demonstrated the feasibility of this task, they often struggle to capture the subtle expressiveness and diversity required for realistic and musically convincing arrangements. Moreover, existing works do not provide explicit control over the complexity level of the generated piano covers, which limits their applicability for educational contexts. Conditional music arrangement thus remains a significant and relatively underexplored challenge. In this paper, we introduce LEAP, a Latent Diffusion Transformer Model designed to generate controllable piano roll covers conditioned on pop music waveforms. To the best of our knowledge, this is the first model to generate piano covers directly from pop music while explicitly controlling the difficulty level of the arrangement. We show that `LEAP`, trained on only a fraction of the dataset used by the baseline method, is capable of producing convincing piano covers with varying degrees of arrangement complexity. The code used to produce our results is available at `https://github.com/emilio-pcrd/leap-ml`, and a web demo associated with the project can be accessed at `https://emilio-pcrd.github.io/leap-studio/`.

1 INTRODUCTION

Piano covers of pop music are a widely enjoyed form of music, used for entertainment and music education purposes, serving as a bridge between complex polyphonic audio and solo instrumentation. Unlike simple transcription, which aims for note-for-note accuracy, a piano cover requires an arrangement: a creative adaptation that captures the melody, harmony, and rhythmic essence of a full mix (vocals, drums, accompaniment) and translates it into a cohesive, playable piano performance. This task demands not only robust music information retrieval (MIR) capabilities but also a high degree of generative creativity to maintain musicality under the constraints of a single instrument.

To create a piano cover as a human, the task can be really challenging, as it is necessary to adapt the whole polyphonic music piece into a single music track that can be played for piano performances. This necessitate a lot of music knowledge and audible qualities to transfer ideas to a new piece. Automating this process remains a fundamental challenge. Early approaches relied heavily on pipeline systems that explicitly extracted features, such as melody and chord progressions, before mapping them to piano textures using statistical methods Ariga et al. (2017); Takamori et al. (2019); Shibata et al. (2021). While interpretable, these pipelines often fail to capture the holistic mood or expressive dynamics of the original audio. The advent of deep learning introduced end-to-end models, most notably Pop2Piano Choi & Lee (2023), which leverages the Transformer autoregressive paradigm widely adopted in symbolic music generation Huang et al. (2018); Gardner et al. (2022).

However the autoregressive modeling of discrete symbolic tokens presents inherent limitations for high-fidelity arrangement. First, discrete tokenization (e.g., MIDI-like events) necessitates the quantization of continuous performance attributes such as velocity and micro-timing, limiting expressive nuance. Second, autoregressive models are prone to error propagation and often struggle with long-term structural consistency, a critical requirement for arranging full-length songs.

To overcome these limitations, we propose a paradigm shift from discrete sequence modeling to continuous diffusion. Drawing inspiration from the success of Latent Diffusion Models (LDMs) in image Rombach et al. (2022) and audio synthesis Liu et al. (2023), and recent advances in symbolic music diffusion Mittal et al. (2021) (although final generation is in discrete representation), we hypothesize that the complex interplay of pitch, duration, and velocity is better modeled in a continuous, high-dimensional manifold rather than a discrete codebook.

In this work, we introduce a Level-Adaptative Expressive Audio-to-Piano Arrangement (`LEAP`) model, a novel Pop-to-Piano framework that integrates a ResNet-based piano roll variational autoencoder (VAE) with a diffusion transformer (DiT) Peebles & Xie (2023). Our method learns a compact, continuous representation of piano performance geometry and employs a diffusion model to generate these latents conditioned directly on the raw pop audio waveform. Furthermore, we introduce a controllable complexity parameter, allowing users to explicitly guide the density and difficulty of the generated arrangement.

Our contributions are summarized as follows:

- We present the first framework to apply Latent Diffusion Models (LDMs) specifically to the Audio-to-Symbolic arrangement task. We demonstrate that operating in a continuous piano-roll latent space yields superior expressivity compared to discrete token baselines.
- We adapt the DiT architecture to condition on raw audio embeddings (via MERT Li et al. (2024)), enabling the model to implicitly learn the mapping from spectral features to symbolic piano textures without explicit feature extraction.
- Controllable Arrangement Complexity: We introduce a density-based conditioning mechanism that allows users to modulate the arrangement style—ranging from sparse, simple accompaniment to complex, virtuosic performance, addressing a key limitation in previous arrangement models.

2 BACKGROUND

Our proposed architecture integrates a Variational Autoencoder (VAE) for learning a structured latent representation of symbolic music and a Diffusion Transformers model (DiT) Peebles & Xie (2023) based on Denoising Diffusion Probabilistic Model (DDPM) Ho et al. (2020) operating in this latent space, conditioned on the source audio and a complexity scalar. This section reviews the fundamental components and their formulation.
In this work, we denote x as the piano roll image of size $(\text{num\_notes}, \text{context\_length})$, and z as a random latent variable.

2.1 VARIATIONAL AUTOENCODER (VAE) FOR PIANO ROLL ENCODING

A *Variational Autoencoder* (VAE) is employed to model and learn the unknown data distribution $p^*(x)$ of piano rolls $x \in \mathbb{R}^{N \times T}$, enabling both reconstruction and generation of novel samples. The VAE introduces a latent variable z and defines a joint distribution $p_\theta(x, z) = p_\theta(z)\, p_\theta(x|z)$, where $p_\theta(z)$ is a simple prior (e.g., a standard Gaussian) and $p_\theta(x|z)$ acts as a stochastic decoder. To handle the intractability of the true posterior $p_\theta(z|x)$, an approximate variational posterior $q_\phi(z|x)$ is introduced. The VAE is trained by maximizing the *Evidence Lower Bound* (ELBO):

$$\mathcal{L}^{\text{VAE}}_{\phi,\theta}(x) = \underbrace{\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]}_{\text{reconstruction term}} - \underbrace{\mathcal{D}_{\text{KL}}[q_\phi(z|x)\|p_\theta(z)]}_{\text{latent regularization term}},$$

where the first term encourages accurate reconstruction of the input, and the second term regularizes the latent space by aligning $q_\phi(z|x)$ with the prior $p_\theta(z)$. This formulation ensures a well-structured latent space, facilitating downstream tasks such as diffusion-based generation Kingma & Welling (2022); Mittal et al. (2021)

In practice, the VAE is trained by minimizing a composite loss function that combines reconstruction, harmonic coherence, and latent space regularization:

$$\mathcal{L}^{\text{Final}}_{\text{VAE}} = \mathcal{L}_{\text{MSE}} + \lambda_c \cdot \mathcal{L}_{\text{Chroma}} + \lambda_{kl} \cdot \mathcal{D}_{\text{KL}}[q_\phi(z|x)\|p(z)].$$

where the weighted chroma loss specifically regularize the harmonic content. The Chroma feature representation captures the pitch class profile (C, C#, D, ...) regardless of octave. This term addresses an observed failure mode where initial VAE training relying on $\mathcal{L}_{MSE}$ and $\mathcal{D}_{KL}$, produced arrangements with bloated notes, meaning there were several notes active simultaneously, leading to harmonic incoherence. By penalizing reconstruction errors in the chroma domain, we force the VAE to prioritize musically coherent harmonic structures. Furthermore, we employ the standard MSE loss as the primary reconstruction term.

The KL divergence term $\mathcal{D}_{\mathrm{KL}}[q_\phi(z|x)\|p(z)]$ is weighted minimally to avoid posterior collapse while maintaining a structured latent space for downstream tasks.

2.2 Latent Diffusion with Controlled Difficulty

We build upon denoising diffusion probabilistic models (DDPMs) Ho et al. (2020) and operate *entirely in the latent space* of the VAE. Let $z_0 \sim q_\phi(z \mid x)$ denote a latent representation produced by the VAE encoder. Diffusion is applied to z_0, yielding a scalable and semantically structured generative process.

Forward process. The forward diffusion process is defined as a fixed Markov chain that gradually corrupts z_0 into Gaussian noise:

$$q(z_t \mid z_{t-1}) = \mathcal{N}\Big(z_t; \sqrt{1-\beta_t}\, z_{t-1}, \beta_t \mathbf{I}\Big)\,, \tag{1}$$

where $\{\beta_t\}_{t=1}^T$ is a predefined variance schedule. Defining $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$, one can sample z_t at an arbitrary timestep directly from z_0:

$$z_t = \sqrt{\bar{\alpha}_t}\, z_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}). \tag{2}$$

Reverse process and training. The generative model learns the reverse diffusion process

$$p_\theta(z_{t-1} \mid z_t) = \mathcal{N}(z_{t-1}; \mu_\theta(z_t, t), \Sigma_\theta(z_t, t))\,, \tag{3}$$

which approximates the true posterior $q(z_{t-1} \mid z_t, z_0)$. Following standard practice, the mean μ_θ is reparameterized via a neural network ϵ_θ that predicts the injected noise. This yields the simplified training objective

$$\mathcal{L}_{\text{simple}}(\theta) = \mathbb{E}_{t, z_0, \epsilon}\left[\|\epsilon - \epsilon_\theta(z_t, t)\|_2^2\right]. \tag{4}$$

When learning a timestep-dependent covariance Σ_θ, we follow Nichol & Dhariwal (2021) and optimize ϵ_θ using $\mathcal{L}_{\text{simple}}$, while training Σ_θ with the full variational lower bound.

At inference time, generation proceeds by sampling $z_T \sim \mathcal{N}(0, \mathbf{I})$ and iteratively applying

$$z_{t-1} \sim p_\theta(z_{t-1} \mid z_t), \quad t = T, \ldots, 1. \tag{5}$$

The final latent z_0 is decoded into a piano arrangement using the VAE decoder.

Diffusion Transformer backbone. We parameterize the denoising network using a Diffusion Transformer (DiT). At each timestep, the noisy latent tensor $z_t \in \mathbb{R}^{1\times 12\times 128}$ is patchified into a sequence of tokens and processed by a stack of Transformer blocks. We adopt the AdaLN-Zero architecture, which injects conditioning information via adaptive layer normalization, enabling stable training and effective control.

Conditioning and classifier-free guidance. The reverse process is conditioned on auxiliary information c, yielding $p_\theta(z_{t-1} \mid z_t, c)$. Conditioning consists of (i) an audio embedding extracted from the input waveform using a pretrained encoder, and (ii) a discrete arrangement difficulty label $c \in \{0, 1, 2, 3\}$. The difficulty label is computed from the number of notes in a piano-roll segment and reflects increasing arrangement density.

We employ classifier-free guidance (CFG) to enforce conditioning without an explicit classifier. During training, the conditioning variable c is randomly dropped and replaced with a learned null embedding $\varnothing$. At inference time, guidance is applied by modifying the predicted noise:

$$\hat{\epsilon}_\theta(z_t, c) = \epsilon_\theta(z_t, \varnothing) + s\big(\epsilon_\theta(z_t, c) - \epsilon_\theta(z_t, \varnothing)\big), \tag{6}$$

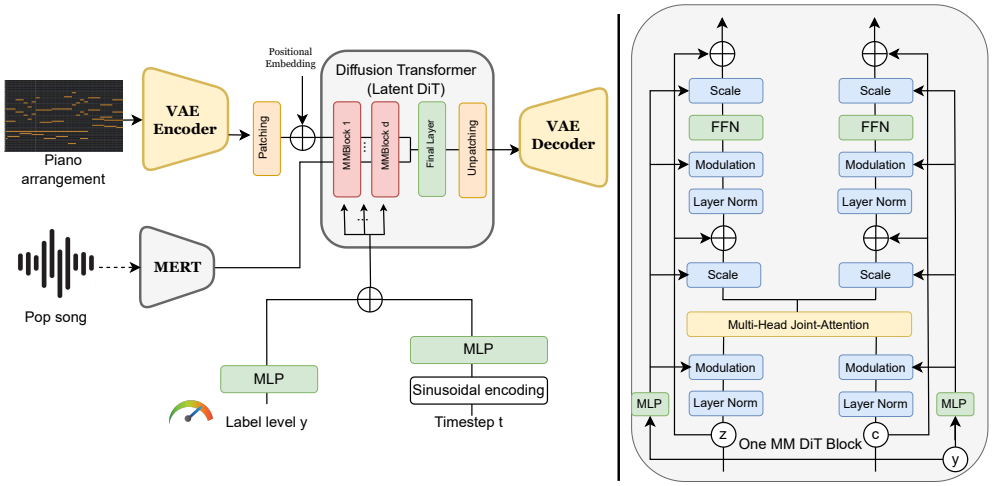

Figure 1: Our model architecture. *Left*. The input latent inside the latent space of the VAE is decomposed into patches and processed by several DiT blocks. We add embeddings of difficulty-level, timestep and wav songs trough each Multi-modal DiT blocks. *right.* Details about one multi-modal DiT block. Green flags are multi-linear layers. Blue flags are normalization, scaling and modulation steps. Orange flags refer to the patching and unpatching of the latent piano rolls for being processed by the DiT latent model. Best viewed zoomed in.

where $s \geq 1$ is the guidance scale. Increasing s biases sampling toward latents that better satisfy the desired difficulty level while remaining consistent with the audio conditioning.

This design yields a controllable latent diffusion model that captures long-range temporal and harmonic structure while providing explicit control over arrangement complexity.

2.3 Architectures

In this subsection, we describe the architectural choices of our system, consisting of a convolutional variational autoencoder (VAE) for latent representation learning and a multimodal Diffusion Transformer (DiT) for conditional generation in latent space. Figure 1 summarizes the architecture of the model.

VAE. We employ a convolutional-attention VAE to encode piano-roll representations into a compact latent space suitable for diffusion modeling. The encoder follows a hierarchical ResNet-style architecture, multiple residual blocks. Channel widths increase progressively, with spatial downsampling applied between levels, reaching a downsampling factor of 8. A bottleneck stage consisting of residual blocks and a self-attention layer captures long-range temporal and harmonic dependencies before projecting features into the latent space. The decoder mirrors the encoder structure with symmetric residual blocks and upsampling operations, reconstructing the piano roll from the latent representation.

Latent diffusion architecture. Diffusion is performed in the latent space using a multimodal Diffusion Transformer, inspired by DiT Peebles & Xie (2023) and Stable Diffusion 3 Esser et al. (2024). The denoising network operates on latent tensors with a patch size of 2, which are flattened and linearly projected into a sequence of tokens with hidden dimension 768. A learnable positional embedding is added to preserve spatial structure. The model consists of 8 stacked Transformer blocks, each using 8 attention heads, and follows an AdaLN-Zero conditioning scheme to incorporate both diffusion timesteps and class-label conditioning, which we utilized to control the cover difficulty level, incorporated via classifier-free guidance.

To incorporate audio conditioning, pretrained audio features are projected into the same hidden space (768) and processed jointly with latent tokens through the MM-DiT blocks. Each Transformer

block performs joint attention over concatenated latent and audio tokens. Conditioning information modulates both attention and feed-forward sublayers through adaptive layer normalization and gated residual connections.

During inference, both audio and difficulty-level condition are entered as user's inputs. They are first processed by the latent DiT model for few timesteps, and then decoded through the trained VAE decoder.

3 EXPERIMENTAL SETUP

We investigate the efficacy of the Diffusion Transformer (DiT) architecture operating within the continuous latent space of a specialized Piano Roll VAE. By conditioning on raw audio embeddings and complexity labels, we study the model's ability to generate high-fidelity, structure-aware piano arrangements. Examples can be checked in this website: `https://emilio-pcrd.github.io/leap-studio/`.

3.1 DATASET

As our proposed architecture builds upon and extends the Pop2Piano framework, we adopt the same data preprocessing pipeline described in their work. In our experiments, due to computational constraints, we use a subset of this curated dataset consisting of 100 aligned pop–piano song pairs. We also filter out data pairs that exhibit differences in musical progression or mismatched keys between the pop audio and its corresponding piano transcription. We compute the Melody Chroma Accuracy (MCA), introduced in Pop2Piano, to discard pairs with an MCA score lower than 0.15, as well as pairs with an audio length mismatch exceeding 20%. The MCA is calculated between the pitch contour of the vocal signal extracted from the audio and the top line of the MIDI. After this filtering process, we retain a total of 96 aligned audio–MIDI pairs (300 minutes of audio), which are used to train both our VAE and diffusion models.

As proposed in Choi & Lee (2023) paper, we segment both audio and MIDI data into fixed-length, overlapping chunks to facilitate efficient training and batching. Specifically, the audio signals are resampled at 22,050 Hz and sliced into overlapping chunks of 10.24 seconds, with a stride of 5 seconds between consecutive segments. The corresponding MIDI files are segmented using the same temporal boundaries to preserve alignment. Each audio chunk is then encoded using a pretrained MERT audio encoder, described in Li et al. (2024). The aligned MIDI chunks are converted into piano roll representations of shape 128×1024, where the pitch dimension corresponds to MIDI note numbers and the temporal resolution is set to 100 frames per second.

3.2 TRAINING

We train conditional latent DiT models on paired Pop-Audio and Piano-Cover data chunks. The input data consists of piano rolls with dimensions 128×1024. We initialize the final linear layer with zeros and use standard ViT weight initialization techniques for the remaining layers. All models are trained using the AdamW optimizer. Unlike the constant learning rate schedule used in the original DiT implementation (Peebles & Xie (2023)), we find that a dynamic schedule aids convergence for multimodal audio tasks. We employ a *Cosine Annealing* learning rate schedule with a base learning rate of 1×10^{-4} and a linear warmup of 200 steps. We utilize a weight decay of 1×10^{-2} and an effective batch size of 32 (achieved via gradient accumulation considering available hardware). To ensure training stability and generation quality, we maintain an exponential moving average (EMA) of the model weights with a decay rate of 0.9999. All reported results and audio samples are generated using the EMA weights. We employ mixed-precision training (FP16) to optimize memory usage without compromising generative performance.

We implement all models in Pytorch and train them using a single RTX 4080 Laptop 6BG VRAM GPU. It trains at roughly 1.7 iterations/second with a global batch size of 32.

Original Songs	Arranger	Average MCA
POP909$_F$	Human	0.395
POP909$_F$	Pop2Piano	0.402 ± 0.021
POP909$_F$	**LEAP**	$\mathbf{0.4} - - \pm \mathbf{0.0} - -$

Table 1: Average Melody Chroma Accuracy (MCA) of human-made and automatically generated piano arrangements.

3.3 DIFFUSION

We operate in the compressed latent space of a pre-trained ResNet-based Variational Autoencoder (VAE). The VAE encoder imposes a spatial downsampling factor of 8; given a velocity piano roll x of shape $1 \times 128 \times 1024$, the encoded latent $z = E(x)$ has shape $4 \times 16 \times 128$. To align the latent variance with the diffusion noise schedule, we normalize the latents using a pre-computed scaling factor (calculated as the inverse standard deviation of the training set latents) to align with unit variance (following Evans et al. (2024)).

Our diffusion framework utilizes a linear noise schedule with $T = 1000$ steps, with β ranging from 1×10^{-4} to 2×10^{-2}. Following recent findings in audio generation, we employ velocity prediction (v-prediction) Salimans & Ho (2022) as the training objective, optimizing the Mean Squared Error (MSE) between the predicted and target velocity fields.

The model is conditioned on two distinct inputs: Audio Context and Complexity Class. We utilize a frozen, pre-trained MERT model to extract semantic acoustic features from the input pop audio. These features are injected into the DiT via single self-attention layers, by concatenating the features with the latent tokens such as $[\text{Tokens}_{\text{latents}}, \text{Tokens}_{\text{audio}}]$, which is called *Joint attention*, following Esser et al. (2024). To control the arrangement density, we discretize the note density into 5 complexity classes. These are embedded via a learnable lookup table and added to the time-step embedding. During inference, we employ Classifier-Free Guidance (CFG) to modulate the strength of these conditions, using the DDIM sampler Song et al. (2022) for efficient generation in few steps.

3.4 GENERATION RESULTS

In this section, we evaluate the performance of LEAP using quantitative metrics. First, we compute the mean squared error (MSE) between the generated piano-roll images and the ground-truth piano rolls over all test pairs. Our model achieves an MSE of approximately 8×10^{-3}, which we find to be qualitatively reasonable given the inherent variability of musical arrangements. Since each piano-roll chunk in the dataset is preprocessed with an associated difficulty-level label, we evaluate the model under controlled conditions by conditioning generation on the *ground-truth* difficulty label.

For an explicit comparison with the original Pop2Piano model, we compute the average Melody Chroma Accuracy (MCA; see Section 3.1 for details) for both human-made and automatically generated piano arrangements. For Pop2Piano, we directly report the values provided in the original paper, assuming reproducible experimental conditions. The results are summarized in Table 1. Overall, LEAP achieves higher MCA scores than the previous state-of-the-art model, indicating improved melodic consistency with the source material.

3.5 SUBJECTIVE EVALUATION

We first conduct a user study to evaluate the LEAP ability to reconstruct piano roll representations. Despite the limited training time, the VAE is able to reconstruct a wide variety of piano rolls with good fidelity. Overall, only minimal differences are observed between the reconstructed piano rolls and their ground truth counterparts. This subjective evaluation indicates that the VAE learns meaningful latent representations suitable for subsequent DiT training.

4 CONCLUSION

In this paper, we introduced LEAP, a novel automatic piano arrangement model for generating piano covers directly from pop music with controllable difficulty levels. Using the same preprocessing pipeline, we explore an extension of Pop2Piano by introducing difficulty-level controllability as an additional conditioning signal. As shown in our experiments, the model achieves promising results at the VAE stage. However, due to computational constraints, it currently struggles to generate piano arrangements as convincing as those produced by state-of-the-art methods such as Pop2Piano. We strongly believe that, given additional time and computational resources, this model could serve as a solid foundation for further research in music transcription and arrangement, particularly through a public release.

LIMITATION & FUTURE WORKS

Our reliance on the data preprocessing pipeline, inherited from related work, involves time quantization to the 8th-note beats. This design decision inhenrently restricts the model's ability to generate arrangements featuring complex rythmic variations, such as triplets, 16th notes, or trills. Consequently the true micro-timing and expressive qualities of a human performance, like *rubato* or nuanced tempo deviations, are not fully captured in the symbolic domain.

Despite operating in the VAE latent space, the sampling process of diffusion models requires numerous sequential steps of denoising. This makes the inference stage computationally expensive and slow compared to a single forward pass of the original auto-regressive Transformer, (posing a challenge for real-time creative applications).

ACKNOWLEDGMENTS

This paper was written as part of a Creative Machine Learning course project, in which the primary objective was to propose a novel method within a limited time frame. We would like to thank our professor, Philippe Esling, Professor and Researcher at IRCAM Paris, for giving us the opportunity to experiment with a model that has the potential to be further developed in future work.

As part of this assignment, we would also like to acknowledge the contributions of each group member. We thank Emilio Picard for the model architecture and training, Loïs Guerci for data preprocessing, and Julien Verstraete and Louis Gosselin for their work on the theoretical background, bibliography, and their contributions to the paper writing.

REFERENCES

Shunya Ariga, Satoru Fukayama, and Masataka Goto. Song2guitar: A difficulty-aware system for generating guitar solo covers polyphonic audio of popular music. In *Proceedings of the 18th International Society for Music Information Retrieval Conference*, pp. 568–574. ISMIR, October 2017. doi: 10.5281/zenodo.1417501. URL https://doi.org/10.5281/zenodo.1417501.

Jongho Choi and Kyogu Lee. Pop2piano : Pop audio-based piano cover generation, 2023. URL https://arxiv.org/abs/2211.00895.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. URL https://arxiv.org/abs/2403.03206.

Zach Evans, Julian D. Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open, 2024. URL https://arxiv.org/abs/2407.14358.

Josh Gardner, Ian Simon, Ethan Manilow, Curtis Hawthorne, and Jesse Engel. Mt3: Multi-task multitrack music transcription, 2022. URL https://arxiv.org/abs/2111.03017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer, 2018. URL https://arxiv.org/abs/1809.04281.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

Yizhi Li, Ruibin Yuan, Ge Zhang, Yinghao Ma, Xingran Chen, Hanzhi Yin, Chenghao Xiao, Chenghua Lin, Anton Ragni, Emmanouil Benetos, Norbert Gyenge, Roger Dannenberg, Ruibo Liu, Wenhu Chen, Gus Xia, Yemin Shi, Wenhao Huang, Zili Wang, Yike Guo, and Jie Fu. Mert: Acoustic music understanding model with large-scale self-supervised training, 2024. URL https://arxiv.org/abs/2306.00107.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D. Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. *ICML*, 2023.

Gautam Mittal, Jesse Engel, Curtis Hawthorne, and Ian Simon. Symbolic music generation with diffusion models. *ISMIR*, 2021.

Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021. URL https://arxiv.org/abs/2102.09672.

William Peebles and Saining Xie. Scalable diffusion models with transformers. *ICCV*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models, 2022. URL https://arxiv.org/abs/2202.00512.

Kentaro Shibata, Eita Nakamura, and Kazuyoshi Yoshii. Non-local musical statistics as guides for audio-to-score piano transcription. *Information Sciences*, 566:262–280, August 2021. ISSN 0020-0255. doi: 10.1016/j.ins.2021.03.014. URL http://dx.doi.org/10.1016/j.ins.2021.03.014.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. URL https://arxiv.org/abs/2010.02502.

Hirofumi Takamori, Takayuki Nakatsuka, Satoru Fukayama, Masataka Goto, and Shigeo Morishima. Audio-based automatic generation of a piano reduction score by considering the musical structure. In Ioannis Kompatsiaris, Benoit Huet, Vasileios Mezaris, Cathal Gurrin, Wen-Huang Cheng, and Stefanos Vrochidis (eds.), *MultiMedia Modeling*, pp. 169–181, Cham, 2019. Springer International Publishing.