

Discrete Diffusion Probabilistic Models for Symbolic Music Generation

Matthias Plasser¹, Silvan Peter¹ and Gerhard Widmer^{1,2}

¹Institute of Computational Perception, Johannes Kepler University Linz, Austria

²LIT AI Lab, Linz Institute of Technology, Austria

matthias.plasser@jku.at, silvan.peter@jku.at, gerhard.widmer@jku.at

Abstract

Denoising Diffusion Probabilistic Models (DDPMs) have made great strides in generating high-quality samples in both discrete and continuous domains. However, *Discrete* DDPMs (D3PMs) have yet to be applied to the domain of Symbolic Music. This work presents the direct generation of Polyphonic Symbolic Music using D3PMs. Our model exhibits state-of-the-art sample quality, according to current quantitative evaluation metrics, and allows for flexible infilling at the note level. We further show, that our models are accessible to post-hoc classifier guidance, widening the scope of possible applications. However, we also cast a critical view on quantitative evaluation of music sample quality via statistical metrics, and present a simple algorithm that can confound our metrics with completely spurious, non-musical samples.

1 Introduction

Music generation denotes a distinct set of tasks in the burgeoning field of synthetic art generation. It subsumes a wide range of approaches and problems, both artistic and technical. Common modeling goals are creating wholly unconstrained musical material from a random seed, the sensible connection of existing music excerpts, or the continuation of a starting sequence. In terms of content, music generation can aim at the creation of individual tracks of predefined instruments, full compositions, melody accompaniments, drum loops, and many other uses.

Perhaps surprisingly to practitioners of generative models not familiar with music, the data domain of music generation is not inherent to the problem. Researchers model music generation both in the symbolic – our modeling target – and the audio domain.

Symbolic music representations build on discrete musical objects, such as notes, commonly encoded in MIDI or musicXML files. The strengths of symbolic representations lie

in their capacity to build on a wealth of thinking about musical objects and structure. The resulting representations are lightweight; they encode musical pieces in hundreds to thousands of units. However, symbolic representations do not directly encode sound.

On the other hand, the audio domain is the digital representation of sound. Audio data consists of sequences of air pressure measurements sampled at isochronous intervals. Its strengths lie in the near-complete representation of audible sound and the simple, unstructured format. However, it's precisely this lack of structure and the length of typical sequences that pose difficult modeling problems. Research in these two domains shares many similarities; however, they also have non-overlapping problems and diverging histories. A technical difference between the two music domains lies in their continuous versus discrete data distributions. This difference is mirrored in the probabilistic formulation of many generative latent space models. Among them are Diffusion Probabilistic Models (DPM) as introduced by Sohl-Dickstein et al. [2015]. Their most famous formulation — denoising diffusion probabilistic models (DDPM) — has been shown to outperform previous state-of-the-art generative models in various continuous generation tasks [Dhariwal and Nichol, 2021; Bond-Taylor et al., 2022; Sohl-Dickstein et al., 2015; Ho et al., 2020]. Consequentially, audio generation research applied DDPMs to the audio domains of speech and music [Kong et al., 2021; Hawthorne et al., 2022]. Even in symbolic music generation, Mittal et al. [2021] use a DDPM to model the continuous latent space of a symbolic music variational autoencoder.

Diffusion models have initially been proposed in both continuous and discrete formulations; however, discrete DPMs remain somewhat in the shadow of the great success of their continuous siblings.

Hoogeboom et al. [2021] extended discrete DPMs from the binomial to the categorical domain and Austin et al. [2021] proposed a more general framework for discrete denoising diffusion probabilistic models (D3PM), which includes all previous definitions of D3PMs. D3PMs are no mere theoretical innovation either; Taylor et al. [2022] demonstrate their capabilities on discrete sequences. These models, and especially the absorbing state-based formulation by Austin et al. [2021] provide the foundation for our work.

Building on these lines of research, we propose SCHmU-

Our implementation, trained model weights and samples can be found at <https://github.com/plassma/symbolic-music-discrete-diffusion>

BERT (= Symbolic Creative Hierarchic music Unmasking Bidirectional Encoder Representation Transformer), a discrete denoising diffusion probabilistic model applied directly to the discrete domain of Symbolic Music. To summarize, our contributions include:

- The first application of Discrete Denoising Diffusion Probabilistic Models to the discrete domain of symbolic music generation.
- Non-autoregressive note-level modeling of symbolic music that allows for flexible applications including inpainting at note level and accompaniment generation: SCHmUBERT can infill arbitrary masked tokens from single notes or short motifs to accompaniment generation by masking entire tracks.
- State-of-the-art performance: Although our model has five times fewer trainable parameters than Mittal et al.'s [2021], both its unconditional generation results and its conditioned infilling results outperform those of their continuous reference model.

While our model performs very well in our quantitative evaluation, we want to caution against uncritical uptake. Employing a small algorithmic generation experiment, we discuss the limits of currently used evaluative metrics of symbolic music generation, especially concerning undue aggregation of values. We use an arbitrarily chosen starting distribution of notes and simulated annealing to create a piece that metrics fail to distinguish from a reference piece but is clearly visually and sonically different.

Highlighting a hard-to-quantify aspect of music generation—the model's capacity for interactivity and control—we present a small post-hoc conditioning experiment. SCHmUBERT is susceptible to classifier guidance by separately trained classifiers. As a proof of concept, we use a note density classifier to successfully guide SCHmUBERT to predefined note densities per measure with moderate loss of sample quality.

The remainder of this paper is structured as follows: Section 2 introduces symbolic music generation and discrete denoising diffusion probabilistic models. Section 3 details our model, representation, data, architecture, and training procedure. The trained model is then compared to a baseline model [Mittal et al., 2021] on two tasks - infilling and unconditional generation - the results of which are presented in Section 4. Finally Section 5 discusses the evaluation metrics in the broader context of symbolic music generation and concludes the paper.

2 Background & Related Work

Computation was barely a theoretical vision when researchers started to muse about the possibility of composing music automatically and algorithmically, and the interest hasn't faded since (Ada Lovelace as cited in [Aiello, 2016]). Many algorithms for symbolic music generation have been investigated and published. In recent years, most algorithmic systems have been based on AI techniques, either evolutionary algorithms or, more commonly, deep neural networks [Civit et al., 2022].

2.1 Symbolic Music Generation

This section introduces several deep learning (DL) symbolic music generation models. While we do not directly compare all these models, we want to give a minimal overview of the most influential DL models of symbolic music as they happen to roughly map the space of available representations, architectures, learning models, and generation paradigms: MusicVAE [Roberts et al., 2018] is a music generator based on variational autoencoders (VAE). VAEs [2014] are latent space models consisting of an encoder and a decoder network.

MusicVAE uses bidirectional recurrent neural networks as encoders and decoders. The model employs one of the earliest fixed-grid tokenized MIDI representations and produces musical material in two bar segments.

MidiNet takes a different approach and uses a generative adversarial network (GAN) [Yang et al., 2017; Goodfellow et al., 2014]. Architecturally, MidiNet employs convolutional neural networks (CNN) for both generator and discriminator, and consequentially, represents its symbolic music data as fixed-size piano rolls.

Attention-based Transformer encoders are the foundational architecture of several recently published non-autoregressive music generation models. These models are sometimes presented as general pre-trained music understanding models providing embeddings suitable to diverse downstream tasks beyond music generation. This is warranted by applying a key technique that enabled the success of large-scale language pre-training: non-sequential unmasking [Devlin et al., 2019].

MusicBERT is a Bidirectional Encoder Representation Transformer (BERT) that learns to predict masked tokens in a symbolic music sequence [Zeng et al., 2021]. An octuple token is introduced that contains higher-level information than just the note (e.g. time signature and tempo), the masking occurs at measure level.

MidiBERT-Piano is another BERT-style unmasking model [Chou et al., 2021]. MidiBERT-Piano is trained exclusively on piano music - single track and non-monophonic - and evaluated both on generation and downstream classification tasks.

Mittal et al. [2021] proposed a symbolic music generation system using DDPMs. This model builds on the two-bar monophonic MusicVAE proposed earlier [Roberts et al., 2018]. A continuous DDPM is then trained on sequences of latent MusicVAE embeddings. This model was designed for longer-term structure via hierarchical modeling — steering the generation process of a latent space model (MusicVAE) with another model— and uses DDPMs in the continuous (latent) domain.

To the best of our knowledge, Mittal et al.'s is the only published diffusion model for symbolic music generation, albeit with the diffusion model applied indirectly. It serves as our reference model: We keep a MusicVAE-like token representation and Mittal et al.'s evaluation metrics [2021].

2.2 Diffusion Probabilistic Models

Diffusion Probabilistic Models (DPMs) generate data by inverting a Markovian data corruption process. Although Sohl-Dickstein et al. [2015] introduced DPMs for continuous and

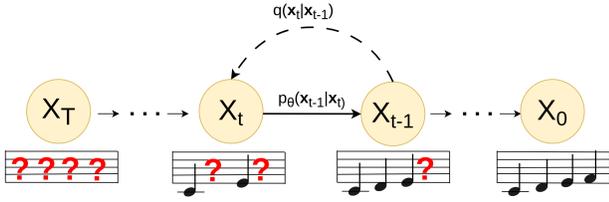


Figure 1: A score example illustrating the absorbing state diffusion process

discrete (binomial) domains, the former—under the name of Denoising Diffusion Probabilistic Models (DDPMs)—have received relatively more attention in recent work [Ho *et al.*, 2020; Mittal *et al.*, 2021; Dhariwal and Nichol, 2021]. Hoogeboom *et al.* [2021] extended DDPMs from the binomial to the categorical domain, and Austin *et al.* [2021] propose a more general framework for Discrete DDPMs (D3PMs), which includes all previous definitions of D3PMs.

The Markovian data corruption process $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ is called forward diffusion process, and in T steps transforms data from the complex target distribution $p(\mathbf{x}_0)$ into a distribution $p(\mathbf{x}_T)$, from which sampling without access to target data is possible. Once the Markovian reverse diffusion process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is learned, it can be used to transform samples from $p(\mathbf{x}_T)$ to samples from the target distribution $p(\mathbf{x}_0)$.

In the continuous case, the forward diffusion process adds small amounts of Gaussian noise each step, resulting in \mathbf{x}_T converging towards an isotropic Gaussian distribution. In [Austin *et al.*, 2021]’s framework for Discrete Denoising Diffusion Probabilistic Models, the forward transition probabilities for scalar discrete random variables $x_t \in \{1, \dots, K\}$ are expressed using matrices: $q(x_t = j|x_{t-1} = i) = [\mathbf{Q}_t]_{ij}$. Given a set of random variables \mathbf{x}_t in one-hot notation, the forward diffusion process can be written as:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \text{Cat}(\mathbf{x}_t; \mathbf{p} = \mathbf{x}_{t-1}\mathbf{Q}_t) \quad (1)$$

where $\text{Cat}(\mathbf{x}, \mathbf{p})$ is a categorical distribution, over the one-hot vector \mathbf{x} , and probabilities are given by the vector \mathbf{p} . $\mathbf{x}_{t-1}\mathbf{Q}_t$ denotes the (one-hot) vector-matrix product and amounts to selecting a specific column from the transition matrix. This transition matrix allows encoding domain-specific knowledge into the forward, and thus into the trainable reverse diffusion process: Transition probabilities between states reflect similarities between them. [Austin *et al.*, 2021] define four flavors of D3PM, each characterized by its type of transition matrix:

The **uniform D3PM** has a matrix with uniform transition probabilities between all categories, $p(\mathbf{x}_T)$ is a uniform distribution over all categories.

The **discretized Gaussian D3PM** samples transition probabilities from a Gaussian PDF, and thus imposes higher transition probabilities between closer states. This is very suitable for ordinal data, such as brightness in image pixels.

The **Absorbing state D3PM** (see figure 1) introduces an additional category that does not occur in the domain: the Absorbing state. This Absorbing state indicates corrupted data, and is similar to the [MASK] token in BERT-like models [Devlin *et al.*, 2019]. In the forward diffusion process, variables

are masked out randomly, they can only transition into the Absorbing state, but not leave it (thus absorbing). In the reverse diffusion process, on the other hand, variables can only transition if they are in the Absorbing state.

Token embedding distance D3PMs are suitable where similarities between categories are not ordinal, but still can be distinguished. Austin *et al.* mention text at character level encoding, where vowels are likely to be more similar to each other than consonants as an example. These similarities can be introduced to the forward transition matrices.

Training

The generative distribution $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ can be approximated using a neural network. Given an \mathbf{x}_t , the neural network predicts the reverse diffusion transition probabilities.

With access to \mathbf{x}_0 , these reverse transition probabilities can be explicitly expressed using the forward diffusion process:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \quad (2)$$

$$= \text{Cat}\left(\mathbf{x}_{t-1}; \mathbf{p} = \frac{\mathbf{x}_t\mathbf{Q}_t^\top \circ \mathbf{x}_0\bar{\mathbf{Q}}_{t-1}}{\mathbf{x}_0\mathbf{Q}_t\mathbf{x}_t^\top}\right) \quad (3)$$

As the forward diffusion process is defined as a Markov chain, $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = q(\mathbf{x}_{t-1}|\mathbf{x}_t)$.

Training the generative distribution amounts to minimizing the negative log-likelihood of real samples, which typically is done indirectly by minimizing the evidence lower bound (ELBO):

$$L_{\text{vb}} = \mathbb{E}_{q(\mathbf{x}_0)} \left[\underbrace{D_{\text{KL}} [q(\mathbf{x}_T | \mathbf{x}_0) || p(\mathbf{x}_T)]}_{L_T} \right] + \sum_{t=2}^T \underbrace{\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} [D_{\text{KL}} [q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) || p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)]]}_{L_{t-1}} - \underbrace{\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} [\log p_\theta(\mathbf{x}_0 | \mathbf{x}_1)]}_{L_0} \quad (4)$$

A typical DDPM training process minimizes L_{vb} stochastically by minimizing randomly sampled subterms. For efficient sampling of subterms of L_{vb} , x_t can be inferred from x_0 in closed form in a single step:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \text{Cat}(\mathbf{x}_t; \mathbf{p} = \mathbf{x}_0\bar{\mathbf{Q}}_t) \quad , \text{where } \bar{\mathbf{Q}}_t = \prod_{i=1}^t \mathbf{Q}_i \quad (5)$$

Connection to Other Generative Models

Although the sampling process of D3PMs is very similar to that of autoregressive models, there is a crucial difference: Autoregressive models are usually optimized to predict the next token in a sequence, while DDPMs refine their whole, joint output sequence simultaneously. This is a consequence of how their tasks are formulated: DDPMs are optimized for fixed sequence lengths. In contrast, the sequence length in autoregressive models is flexible both during training and inference. An Absorbing state D3PM could be used to sample

autoregressively, by unmasking tokens in a fixed (linear) order at the cost of sample quality and diversity [Austin *et al.*, 2021].

Diffusion models share some striking similarities with VAEs. Loosely speaking, diffusion models can be seen as Variational Autoencoders with their latent dimension equal to their data dimension. Their encoder is T layers deep and has no trainable parameters, while their decoder holds all trainable parameters and has a depth of T times the depth of $p(\mathbf{x}_{t-1}|\mathbf{x}_t)$ (T repeated applications). The model $p(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is trained at each of the T layers using intermediate results from the forward diffusion process, which avoids the vanishing gradient problem of neural networks with such large depths. At the other extreme of depth are one-step diffusion models. In particular, BERT [Devlin *et al.*, 2019] can be seen as a one-step Absorbing state diffusion model: its Absorbing state is the [MASK] token, which the model fills in in a single step [Austin *et al.*, 2021].

3 Experiments

In this section, we introduce the components of SCHmUBERT. We discuss the D3PM employed, the data representation, model architecture, and training hyperparameters. In the section 4, we compare two models against a state-of-the-art competitor [2021] in both unconditional generation and conditional infilling tasks.

3.1 Absorbing State Diffusion Probabilistic Model

We choose our model to be an Absorbing State Denoising Diffusion Probabilistic Model (ASD3PM), as this offers three advantages over the other flavors of D3PMs: Firstly, ASD3PMs are capable of directly infilling in any sample containing masks. Secondly, ASD3PMs can be trained using a simplified loss parameterized only on \mathbf{x}_0 . Thirdly, this simplified parametrization allows adapting the number of diffusion steps at sample time [Bond-Taylor *et al.*, 2022].

In our ASD3PM, the forward diffusion process $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ masks out tokens with probability $\frac{t}{T}$, while the reverse diffusion process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ recovers the masked out tokens. Like Bond-Taylor *et al.* [2022], we directly predict $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$, which in training enables respecting the error for all masked out tokens, instead of only those unveiled in the current reverse diffusion step. Prediction of all tokens jointly models all possible \mathbf{x}_{t-1} instead of only a specific one. The single step prediction $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ can be sampled from $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$ by keeping predicted tokens in \mathbf{x}_0 only with probability $\frac{1}{T-t}$. Bond-Taylor *et al.* [2022] further find that assigning less importance to the loss at timesteps close to T (many masks) boosts training. Hence, they reweigh the ELBO by $\frac{T-t-1}{T}$:

$$\mathbb{E}_{q(\mathbf{x}_0)} \left[\sum_{t=1}^T \frac{T-t-1}{T} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \left[\sum_{[\mathbf{x}_t]_i=m} \log p_\theta([\mathbf{x}_0]_i | \mathbf{x}_t) \right] \right] \quad (6)$$

The inner sum in equation 6 forms the joint (log) probability of all random variables in \mathbf{x}_0 given \mathbf{x}_t , which results in the cross-entropy between \mathbf{x}_0 and \mathbf{x}_t under the expectation of $\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}$.

3.2 Data & Representation

We train our models on the Lakh MIDI dataset (LMD) [Rafel, 2016], which consists of more than 170,000 multi-track MIDI files, and to our best knowledge, remains the largest publicly available symbolic music dataset. Following our reference model [Mittal *et al.*, 2021; Roberts *et al.*, 2018], we use a time-quantized representation with 16 steps per bar resolution. We extract $\frac{4}{4}$ monophonic melodies and trios with lengths between 16 and 64 bars using Magenta’s MusicVAE [Roberts *et al.*, 2018] pipelines. The trios consist of a monophonic melody, a monophonic bass track, and a polyphonic drum track. Each step of the melody and bass track can have one of 90 values: 88 pitch values, 1 value for note-off, and 1 to continue the previous note. The drum track encodes note-on for 9 independent drums simultaneously and hence can have one of 512 ($= 2^9$) values per step. In sum, a melody piece with a length of 64 bars is a sequence of 1024 indices, while a trio piece of the same length is a sequence of 1024 triplets of indices. We randomly transpose the melody and bass tracks to augment our training data.

3.3 Model Architecture

Figure 2 shows the melody-only architecture of the neural network estimating transition probabilities of $p(\mathbf{x}_0|\mathbf{x}_t)$. First, indices are embedded in a 128-dimensional vector. A 1D CNN with kernel size 4 and stride 4 then computes first summarization—which we refer to as the lower hierarchy level—of 4 adjacent 128-dimensional embeddings into one 512 dimensional embedding, effectively reducing the sequence length to a quarter, while increasing the embedding dimension. The upper hierarchy level, a GPT-like [Brown *et*

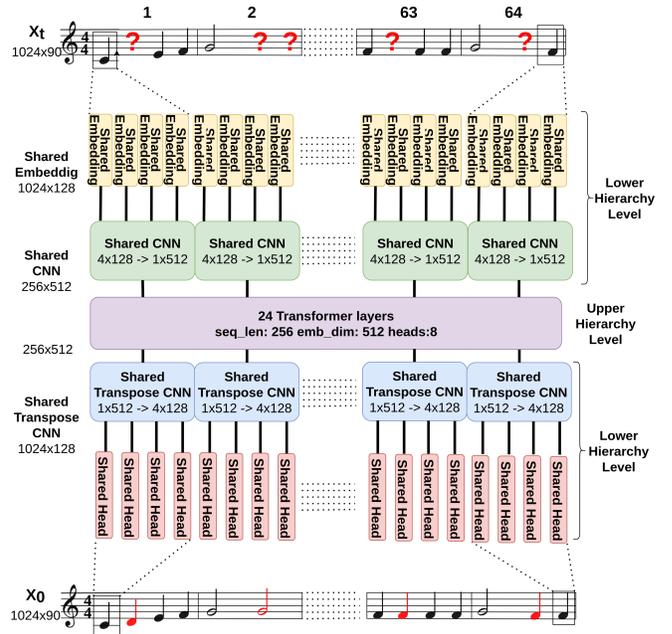


Figure 2: SCHmUBERT architecture. Four adjacent embeddings are convolved into one representation whether they encode a quarter note (such as the examples in the figure) or any other musical material.

al., 2020] stack of 24 transformers processes the sequence of compressed embeddings. The shared 1D Transpose CNN decompresses the sequence to its original length before a shared head maps the sequence of 1024x128 back to a sequence of indices.

In the trio case, each track passes through its own lower hierarchy level; 3 separate embeddings and 1D CNNs compress the track sequences independently. The compressed outputs are then summed up before the upper hierarchy level processes them. Three separate 1D Transpose CNNs compute the decompression, and separate heads predict the output probabilities.

3.4 Training

We train two versions of our model: a melody-only, and a trio version. To allow for direct comparison with Mittal et al.’s model [2021], the melody-only version is trained and evaluated on an extract of 1,000,000 melodies without augmentation.

In table 1, we provide further parameters of both training runs.

Name	Melody	Trio
Melody/Bass pitches	90	
Augmentation	no	yes
Piece shape	(1024, 1)	
Diffusion Model timesteps	1024	
Optimizer	Adam @ lr= $5 * 10^{-4}$	
Transformer layers	24	
Transformer embedding size	512	
Transformer attention heads	8	
Total parameters	76,601,088	78,303,488
Train steps	190,000	600,000
Validation loss	0.087	0.046
Dataset size	19000 Batches	37649 Batches
Batch size	50	
GPU	4x NVIDIA 2080 Ti	
Duration	24h	50h

Table 1: Parameters of training runs

4 Evaluation

In this section, we compare our two models—melody-only and trio—against a state-of-the-art competitor [2021] in both unconditional generation and conditional infilling tasks.

4.1 Metrics

We evaluate our models using Mittal et al.’s [2021] frame-wise self-similarity metrics. This metric captures local self-similarity patterns and is based on statistics of Overlapping Areas of Gaussian density functions. We first define a sequence of 4-bar windows with a hop size of 2 bars. For each window w , we model pitch and duration with a Gaussian distribution, respectively: $\mathcal{N}(\mu_{P_w}, \sigma_{P_w}^2), \mathcal{N}(\mu_{D_w}, \sigma_{D_w}^2)$. The Overlapping Area of Gaussian density functions of adjacent windows is then calculated. The mean μ_{OA} and variance σ_{OA}^2 of overlap areas are then aggregated over all window tuples

of a set of musical pieces. From these values, the relative normalized distances *Consistency* and *Variance* are calculated as follows:

$$Consistency = \max(0, 1 - \frac{|\mu_{OA} - \mu_{GT}|}{\mu_{GT}}) \quad (7)$$

$$Variance = \max(0, 1 - \frac{|\sigma_{OA}^2 - \sigma_{GT}^2|}{\sigma_{GT}^2}) \quad (8)$$

where the values μ_{GT} and σ_{GT}^2 refer to the mean and variance of overlap areas in the ground truth data, respectively. Overall, we compute four values per experiment: Consistency and Variance for pitch distributions and Consistency and Variance for duration distributions.

4.2 Results

Like in [Mittal et al., 2021], we evaluate the self-similarity metric for sets of 1000 generated pieces against 1000 pieces drawn randomly from the joined train and evaluation set as ground truth. In the infilling setting (Figure 3), the central 512 notes are masked and filled in by the model. Consistency and Variance are inferred using the original, unmasked samples as ground truth.

Table 2 shows that SCHmUBERT consistently outperforms Mittal et al.’s Diffusion on MusicVAE-latents model (marked Melody MDMVAE) in the framewise self-similarity metric. This is particularly remarkable when comparing their number of trainable parameters: SCHmUBERT has roughly 80 MIO trainable parameters, while [Mittal et al., 2021]’s hybrid model has roughly 425 MIO trainable parameters (25MIO in the continuous DDPM, 400MIO in MusicVAE). Inpainting and accompaniment experiments show that SCHmUBERT consistently deals with various scenarios.

We find that lower hierarchy level summarization significantly boosts training efficiency, while not reducing the model’s capacity compared to a model without the lower hierarchy level and an embedding dimension of 512. In the Trio setting, we alter the forward diffusion process: instead of masking notes uniformly in all three tracks, notes are either masked in one, two, or three tracks. This masking schedule improves performance in the accompaniment setting.

Setting	Unconditional				Infilling			
	Pitch		Duration		Pitch		Duration	
Metric	C	Var	C	Var	C	Var	C	Var
Train Data	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Test Data	1.00	0.96	1.00	0.96	1.00	0.96	1.00	0.91
Melody 64 bar								
Melody MDMVAE 64 bar	0.99	0.90	0.96	0.92	0.97	0.87	0.97	0.80
SCHmUBERT	0.992	0.920	0.993	0.937	0.997	0.970	0.996	0.970
Trio 64 bar								
SCHmUBERT	0.996	0.893	0.990	0.896	0.997	0.964	0.995	0.924
Melody infilling	-	-	-	-	0.999	0.999	0.978	0.973
Bass infilling	-	-	-	-	0.999	0.999	0.991	0.996
Drum infilling	-	-	-	-	0.999	0.999	0.991	0.996

Table 2: Self similarity scores, values in **bold** represent models proposed in this work

5 Discussion

In this paper, we introduce a novel diffusion model that directly models the discrete variables of symbolic music. We show this model to outperform a reference model on two tasks. However, in place of concluding this paper with a simple recapitulation of the qualities of our models, we want to shed light on some critical aspects of music generation evaluation. In this section, we will focus on two specific questions, discussing each from a general perspective first, then demonstrating them with particular small experiments in the following subsections. These experiments are not evaluated thoroughly, but only serve as proofs of concept.

Starting from a general view on limitations of *statistical quality metrics* for music generation (Section 5.1), we design an algorithm to question the merits of our own evaluation metrics specifically (Section 5.2). Thereafter, we briefly discuss the idea of *model affordances*, i.e. (speculative) possibilities for user interaction and control (Section 5.3), before presenting a proof of concept for post-hoc conditioning on information not previously seen by the model (Section 5.4).

5.1 Limitations of Statistical Metrics

The metrics used in the last five years’ most cited symbolic music generation papers are negative log-likelihood, a set of musical metrics (ratio of empty bars, pitch classes per bar, qualified notes ratio, ratio of notes in patterns, tonal distance), and reconstruction accuracy, among various small-scale ablation studies [Civit *et al.*, 2022]. Two papers note no quantitative evaluation and instead opted for a listening test.

Given the variety of tasks and uses for symbolic music generation, it is unsurprising that no consensus on evaluation exists. However, the difficulties extend beyond mere disagreement and have been addressed variously in the literature

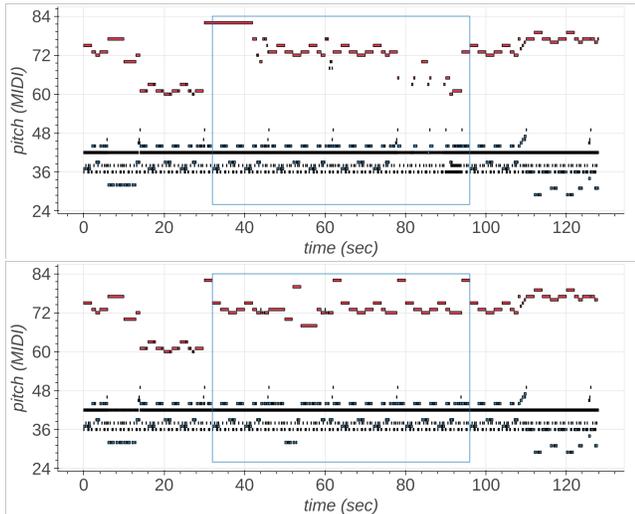


Figure 3: Trio infilling scenario: The top plot shows an original validation sample. The central 512 tokens (blue box) were masked out in all tracks, and filled in again by SCHmUBERT (bottom). The tracks are color-coded (red=melody, blue=bass, black=drum). Bass and drum tracks lie in similar pitch regions of the piano roll due to standard MIDI drum encoding.

[2016; 2001; 2017]. This section discusses the inherent theoretical and technical limitations of statistical metrics.

Following Naem *et al.* [2020], we conceive of the quality of generative models as relating to two high-level concepts: Fidelity and diversity. Fidelity denotes sample quality. For music generation, this might include attributes like structuredness, melody-harmony consistency, correctness, and richness. Listening tests commonly assess such attributes, e.g. in [Hsiao *et al.*, 2021].

Diversity, on the other hand, indicates a distributional want: the model should be able to produce a large variety of samples. In probabilistic terms, it shouldn’t suffer from mode collapse. A somewhat related aspiration is originality, an operative definition of which is that the model should not produce copies of training data samples.

The metrics commonly used in the literature for the evaluation of music generation seem to cover neither fidelity nor diversity satisfactorily. Most metrics are aggregated statistical estimates, which seem unlikely to depict musical concepts at their most granular—what musician ever thought of mismatching pitch class distributions in adjacent measures?—, and only become more impenetrable when aggregated over subsequent windows or even sets of pieces. Arguably, such broad-brush criticism is as easy to dismiss as the metrics themselves — after all, what is a *musical* metric? —, but we want to give a simple example of a failure of a specific metric: the very framewise self-similarity metric we used to evaluate our models.

5.2 Confounding Consistency and Variance Metrics

In the following, we describe a simple algorithm that creates clearly spurious, yet high-ranking “musical” material. We take inspiration from Anscombe’s famous quartet and its recent generalization based on simulated annealing [Matejka and Fitzmaurice, 2017]. Anscombe’s quartet consists of four scatter plots of 2D datasets with similar descriptive statistics but visually different distributions. Adapted to music generation, we devise a simulated annealing process to create a score with nearly indistinguishable consistency and variance values from those of a reference score in the dataset while visually similar to a chosen reference image.

The generation process starts with any binarized image in an appropriate pianoroll-like resolution (e.g. 128 X 500). We compute “music” from the image by sampling column-wise from a pitch distribution given by a normalized and filtered vertical line of the image. The resulting notes are then steered towards exhibiting similar Consistency and Variance metrics as a reference excerpt by repeatedly adding noise to pitches and durations and keeping the best proposals.

Figure 4 illustrates the result of this process. Given an input image depicting a scene of a Super Mario Bros video game (top) and the Super Mario theme as a reference piece (bottom piano roll), this algorithm nudges a sampled set of notes (middle piano roll) toward being evaluatively indistinguishable from the reference piece while still visually close to the original image. This shows a salient oversight of our quantitative evaluation metrics: Surely we want them to be able to

distinguish between Super Mario the image and Super Mario the theme.

With this result, we want to caution against expecting this metric to directly translate to "musical quality". Note that due to the distributional nature of the metrics, this process is possible for a large variety of images and pieces. We use this example to game the evaluation metrics used for our evaluation, however we conjecture that – in keeping with Goodhart’s law [Varela *et al.*, 2014] – most distributional evaluation strategies are amenable to the same type of incursion. However, we cannot fully assess the impact of confoundable statistical metrics on evaluations of generative models, including our own.

5.3 Model Interaction and Control

In a second experiment, we discuss aspects of music generation quality that are harder to quantify but no less important; in fact, they are often pivotal for uptake by artists and instrument designers. These aspects concern a generative system’s potential for interaction and control. The ASD3PM produces human-interpretable, partially masked note sequences in its intermediate steps, and thus affords flexible interactions not only before, but also during the sampling process. This is noteworthy, as such interactions are unobtainable in comparable generative models, which rely on hard to interpret latent space representations [2018; 2021].

Our model allows for ex nihilo as well as accompaniment generation. Furthermore, the sampling process can be carried

out with unusually fine granularity, effectively enabling infilling as well as accompaniment generation with note-level resolution. ([Mittal *et al.*, 2021]’s diffusion on MVAE-latents for example only supports infilling with a resolution of 32 notes at once.) The generated sequence is of fixed length, but outpainting allows for length variability.

Diffusion models are inherently incremental, a suitable user interface can leverage this property and allow interaction with the sampling process at any timestep. Consequently, these interactions can be adaptive, e.g. by iteratively resampling notes or motifs that a user has masked.

5.4 Classifier Guidance

A particular advantage DPMs is flexible post-hoc generation conditioning. One way of conditioning is guiding the generation process using gradients of a separately trained classifier. For many DDPM formulations, this classifier needs to be trained on noisy data to provide correct gradients at different diffusion steps. However, the x_0 parametrization of our ASD3PMs allows using an off-the-shelf differentiable classifier that was only trained on real, unmasked data to guide the sampling process without the need for any retraining:

As a proof of concept, we train a small feed-forward neural network to predict note density, i.e. the number of onsets per measure. We then use this classifier to guide SCHmUBERT to predefined note densities. Classifier guidance results in high accuracy, for plausible note densities (3-12 of max. 16), around 40% of measures satisfy the target density, while almost all of the remaining measures count only one onset too many or little. The subjective sampling quality apparently only decreases a little, but this needs further evaluation.

5.5 Conclusion

In this paper, we present SCHmUBERT, a discrete denoising diffusion probabilistic model applied directly to the modeling of symbolic music tokens. SCHmUBERT outperforms reference work on two music generation tasks while being comparatively small and offering more options for conditioning. Besides presenting a state-of-the-art model, we take a critical look at music generation evaluation and caution against lopsided evaluation based only on aggregated note statistics.

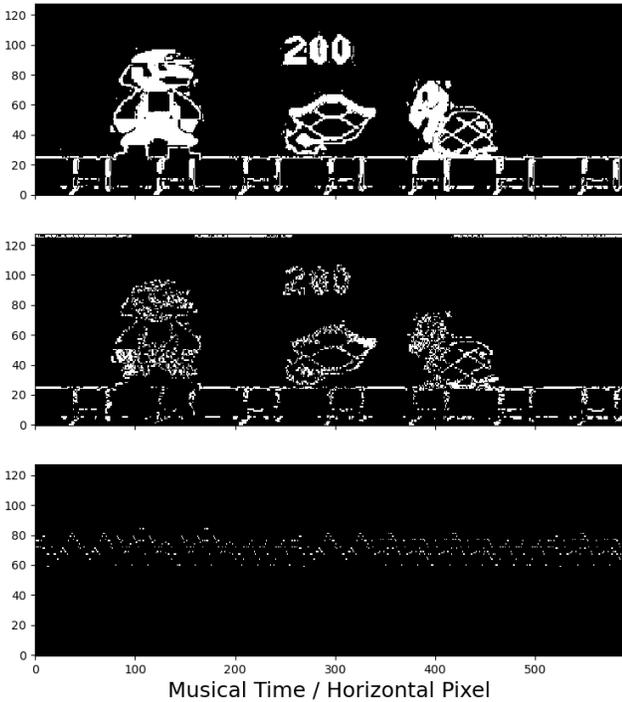


Figure 4: The Super Mario Bros video game (top), the Super Mario theme (bottom piano roll), and a sampled set of notes (middle piano roll) the self-similarity metrics cannot distinguish from the theme.

Algorithm 1 Guided Absorbing State D3PM Sampling

Input: T steps, unmasking function $f_\theta(x_t)$
 differentiable loss function $l_\phi(x_0)$, guidance scale s

```

 $x_T \leftarrow [\text{MASK}]$ 
for  $t = T, \dots, 1$  do
     $x_{0\text{logits}} = f_\theta(x_T)$ 
     $x_{0\text{probs}} = \text{softmax}(x_{0\text{logits}})$ 
     $loss = l_\phi(x_{0\text{probs}})$ 
     $x_{0\text{probs}} = -\nabla_{x_{0\text{probs}}} loss \cdot s$ 
     $x_{0\text{cond}} = \text{Cat}(x_{0\text{probs}}).sample()$ 
    with probability  $\frac{1}{t}$  replace masks in  $x_T$  with  $x_{0\text{cond}}$ 
end for
return  $x_T$ 
    
```

Acknowledgments

This work was supported by the European Research Council (ERC) under the EU's Horizon 2020 research & innovation programme, grant agreement No. 01019375 (*Whither Music?*), and the Federal State of Upper Austria (LIT AI Lab).

References

- [Agres *et al.*, 2016] Kat Agres, Jamie Forth, and Geraint Wiggins. Evaluation of musical creativity and musical metacreation systems. *Comput. Entertain.*, 14(3):3:1–3:33, December 2016.
- [Aiello, 2016] Luigia Carlucci Aiello. The multifaceted impact of ada lovelace in the digital age. *Artificial Intelligence*, 235:58–62, 2016.
- [Austin *et al.*, 2021] Jacob Austin, Daniel Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pages 17981–17993. Curran Associates, Inc., 2021.
- [Bond-Taylor *et al.*, 2022] Sam Bond-Taylor, Peter Hesse, Hiroshi Sasaki, Toby Breckon, and Chris Willcocks. Unleashing transformers: Parallel token prediction with discrete absorbing diffusion for fast high-resolution image generation from vector-quantized codes. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 170–188, Cham, 2022. Springer Nature Switzerland.
- [Brown *et al.*, 2020] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.
- [Chou *et al.*, 2021] Yi-Hui Chou, I-Chun Chen, Chin-Jui Chang, Joann Ching, and Yi-Hsuan Yang. Midibert-piano: Large-scale pre-training for symbolic music understanding. *arXiv preprint arXiv:2107.05223*, 2021.
- [Civit *et al.*, 2022] Miguel Civit, Javier Civit-Masot, Francisco Cuadrado, and Maria Escalona. A systematic review of artificial intelligence-based music generation: Scope, applications, and future trends. *Expert Systems with Applications*, page 118190, 2022.
- [Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. pages 4171–4186, June 2019.
- [Dhariwal and Nichol, 2021] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc., 2021.
- [Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Advances in Neural Information Processing Systems*, 2014.
- [Hawthorne *et al.*, 2022] Curtis Hawthorne, Ian Simon, Adam Roberts, Neil Zeghidour, Josh Gardner, Ethan Manilow, and Jesse Engel. Multi-instrument music synthesis with spectrogram diffusion. *International Society for Music Information Retrieval*, 2022.
- [Ho *et al.*, 2020] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.
- [Hoogeboom *et al.*, 2021] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 12454–12465. Curran Associates, Inc., 2021.
- [Hsiao *et al.*, 2021] Wen-Yi Hsiao, Jen-Yu Liu, Yin-Cheng Yeh, and Yi-Hsuan Yang. Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 178–186, 2021.
- [Kingma and Welling, 2014] Diederik Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.
- [Kong *et al.*, 2021] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.
- [Matejka and Fitzmaurice, 2017] Justin Matejka and George Fitzmaurice. Same stats, different graphs: generating datasets with varied appearance and identical statistics through simulated annealing. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pages 1290–1294, 2017.
- [Mittal *et al.*, 2021] Gautam Mittal, Jesse Engel, Curtis Hawthorne, and Ian Simon. Symbolic music generation with diffusion models. In *International Society for Music Information Retrieval*, 2021.
- [Naeem *et al.*, 2020] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models.

- In *International Conference on Machine Learning*, pages 7176–7185. PMLR, 2020.
- [Pearce and Wiggins, 2001] Marcus Pearce and Geraint Wiggins. Towards a framework for the evaluation of machine compositions. In *Proc. Symp. AI and Creativity in the Arts and Sciences*, 2001.
- [Raffel, 2016] Colin Raffel. *Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching*. PhD thesis, Columbia University, 2016.
- [Roberts *et al.*, 2018] Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4364–4373. PMLR, 10–15 Jul 2018.
- [Sohl-Dickstein *et al.*, 2015] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR.
- [Sturm and Ben-Tal, 2017] Bob Sturm and Oded Ben-Tal. Taking the models back to music practice: Evaluating generative transcription models built using deep learning. *J. Creative Music Systems*, 2(1), 2017.
- [Varela *et al.*, 2014] Diego Varela, Giacomo Benedetto, and Jose Manuel Sanchez-Santos. Editorial statement: Lessons from goodhart’s law for the management of the journal. *European Journal of Government and Economics*, 3(2):100–103, 2014.
- [Yang *et al.*, 2017] Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang. Midinet: A convolutional generative adversarial network for symbolic-domain music generation using 1d and 2d conditions. *ISMIR 2017*, pages 324–331, 2017.
- [Zeng *et al.*, 2021] Mingliang Zeng, Xu Tan, Rui Wang, Zeqian Ju, Tao Qin, and Tie-Yan Liu. Musicbert: Symbolic music understanding with large-scale pre-training. *arXiv preprint arXiv:2106.05630*, 2021.