Music Composition using Unconditional Generative Models

Émile Esmaili
Columbia University
New York City, NY
ede2110@columbia.edu

***Abstract*—In this paper we propose a novel approach to music composition by formulating it as an unconditional image generation problem. We evaluate the performance of training a DCGAN and a DDPM model on a curated dataset of sheet music to assess the capabilities of machines to learn music composition. The results are promising in our view and show that future work in this domain could lead to very interesting advances**

***Index Terms*—GAN, DDPM, Unconditional Image Generation, Music Generation, Computer Vision**

I. Introduction

Music Composition is an artistic task that has been seen as inherently human for centuries. With recent advances in the domain of Generative models, more and more artistic fields have been challenged by state-of-the-art AI models. In music, this mainly through audio generation. We take a rather novel approach to this problem by formulating music composition as an image generation problem, rather than focusing on audio. The rationale here is that unlike handwriting, music compositions, and most notably sheet music, have a very streamlined, rigorous and invariant structure, that we believe can be learned from a machine if trained properly.

II. Related Work

There has been prolific research in the domain of music generation through audio, mostly through the usage of Generative Adversarial Networks (GANs), using the seminal work of Goodfellow et al. [1], including the MuseGAN architecture of Dong et al. [2]. On the other hand, Generative vision models have boomed ever since, and GANs are now being challenged by the denoising diffusion probabilistic models (DDPM) architecture for unconditional image generation proposed by Ho et al. [3]. However, uses of pure Vision approaches to the music composition problem have been very sparse in the literature. That is why we believe applying state of the art method like GANs or DDPMs could lead to interesting results for our problem.

III. Methods

A. Data

The data consists of a sample of 9427 sheet music scores for video games found online, downloaded as PDFs and converted manually to images by us. We then removed images that we deemed problematic (blank pages, or sheets with only one line). We made this propietary dataset available online for anyone to use here.
Please note that we do not use data augmentation techniques like flipping the images for instance as we believe they will compromise the algorithm's ability to learn the inherent structure of the sheet music.

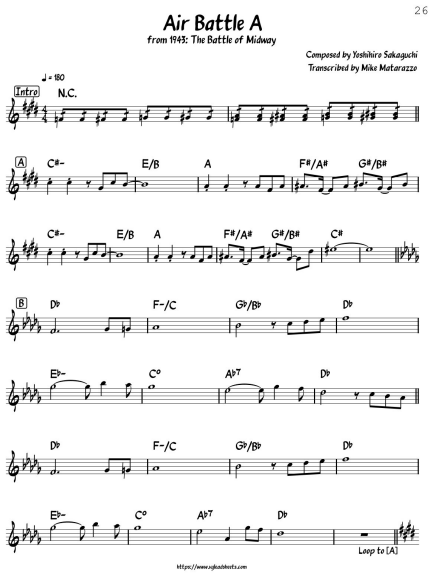

Figure 1. a sheet music from our dataset

B. Models

1) DCGAN (baseline): Our baseline model is the DCGAN, using the architecture proposed by Radford et al. [4]. We define the Generator G and the Discriminator D to compete in a minimax two-player game. D and G play a minimax game in which D tries to maximize the probability it correctly classifies reals and fakes ($logD(x)$), and G tries to minimize the probability that D will predict its outputs are fake ($log(1 - D(G(z)))$).
the GAN loss function is:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}\big[logD(x)\big] \quad (1)$$

$$+\mathbb{E}_{z\sim p_z(z)}\big[log(1 - D(G(z)))\big] \quad (2)$$

The model architecture we used was similar to the original one from the paper.

2) DDPM: We trained a DDPM using the formulation of Ho et al. [3] but with the improvement suggested by Nichol and Dhariwal [5] to include a cosine noise scheduler instead of a linear one, with the following formula in terms of the variance schedule β_t at time t :

$$\tilde{\alpha_t} := \prod_{s}^{T} 1 - \beta_s = \frac{f(t)}{f(0)} \tag{3}$$

$$f(t) = \cos(\frac{t/T + s}{1 + s} \cdot \frac{\pi}{2}) \tag{4}$$

The number of steps is $T = 1000$ to be consistent with the literature. Due to technical constraints we resize images to be 128x128 before normalizing them.
We use the DDPM cosine scheduler in combination with the U-net to denoise the encoded image. Our U-net uses self-attention, with 2 ResNet layers per block, and channels that double as the dimensions shrink. We use the AdamW optimizer with weight decay. the learning rate starts at 10^{-4}, with 1 gradient accumulation step. The batch size is 16 and we use 500 warm-up steps for the cosine scheduler

Table I
U-NET ARCHITECTURE FOR THE DDPM (IN-OUT-CHANNELS)

Downsampling block	**Upsampling block**
ResNet 128-64-128	ResNet 64-128-128
ResNet 64-32-128	ResNet 32-64-128
ResNet 32-16-256	ResNet 16-32-256
ResNet 16-8-256	ResNet 8-16-256
ResNet 8-4-512 self-attention	ResNet 4-8-512 self-attention
ResNet 4-2-512	ResNet 2-4-512

IV. EXPERIMENTS

A. DCGAN

We train the DCGAN model for 20 epochs. Because of the low resolution of the images, the model is not scalable, unless we apply super-resolution to it. We tried to use a 128x128 variant of the 64x64 DCGAN formulation but the discriminator completely oulearns the generator. We decided to focus on DDPMs instead

B. DDPM

We train the DDPM for 50 epochs using the aforementioned hyperparameters. Even though the resolution bottlenecks the results, we find them promising. We do tend to notice a 'renoising' effect after 30 epochs, leading us to question the nature of the loss function used by Ho et al. [3], and Nichol and Dhariwal [5].

Sheet music looks very 'noisy', and we believe that L2 loss is not harsh enough to penalize small errors, which might be crucial to the learning process in our case. We decide to modify

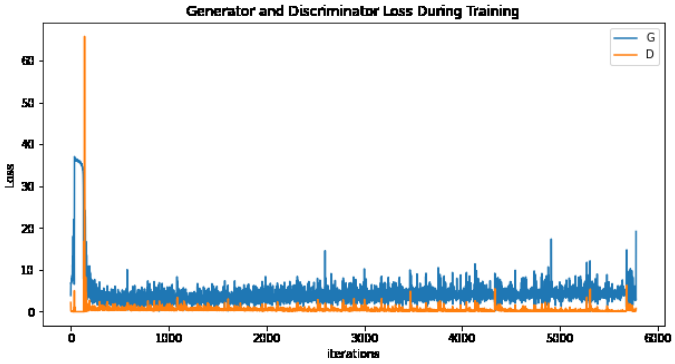

Figure 2. DCGAN Loss curves

Figure 3. DCGAN Loss generated images

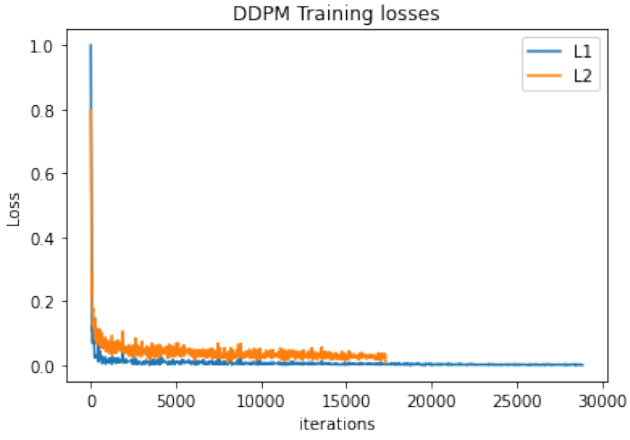

Figure 4. DDPM training

the DDPM algorithm to use L1 loss or its smooth variants like Huber loss:

$$L_\delta = \begin{cases} \frac{1}{2}(y-\hat{y})^2 & if\ |(y-\hat{y})| < \delta \\ \delta((y-\hat{y}) - \frac{1}{2}\delta) & otherwise \end{cases} \quad (5)$$

We retrain the model for 50 epochs, which takes approximately 9 hours on a NVIDIA V-100 GPU. The results are better but we still tend to see re-noising in the later epochs as seen in figure 6.

A last experiment we wished to try is using super-resolution on our output images. We used a pre-trained latent stable diffusion model as pioneered by Rombach et al. [6] that performs 4x upscaling.
The result, shown in figure 8, is disappointing as the upscaling looks cartoonish and loses some of the information given by our model output. We still believe it is an interesting avenue to explore.

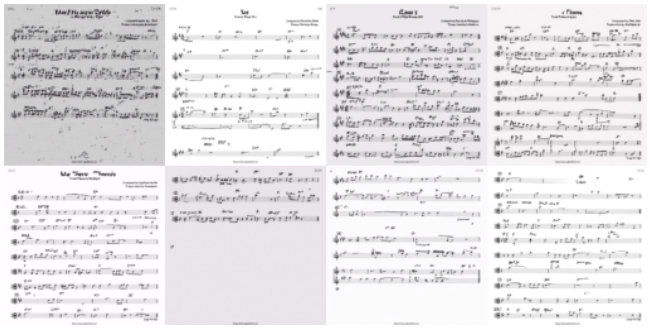

Figure 5. DDPM with L2 loss results - epoch 18

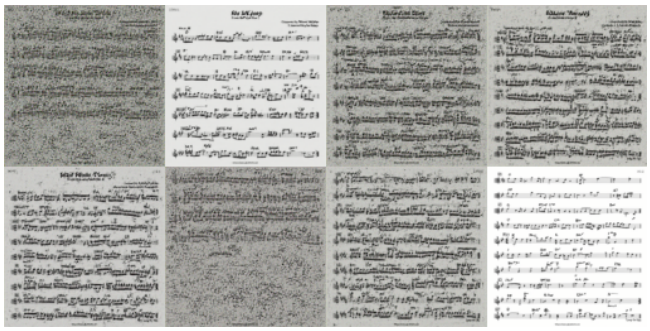

Figure 6. DDPM with L2 loss results - epoch 50

V. Conclusion

We have used state-of-the-art image generation techniques, namely DCGANs and DDPMs, to tackle the problem of sheet music composition by AI. Our findings show diffusion models are able to learn the structure inherent to sheet music but because of the low-resolution capabilities of the algorithms and GPU memory constraints, the images produced are unsatisfying, even when passed onto a super-resolution latent diffusion model. We believe however that exploring higher-resolution training or adding a super-resolution network on top of our current architecture could lead to substantial results and pave the way for AI-generated music scriptures.

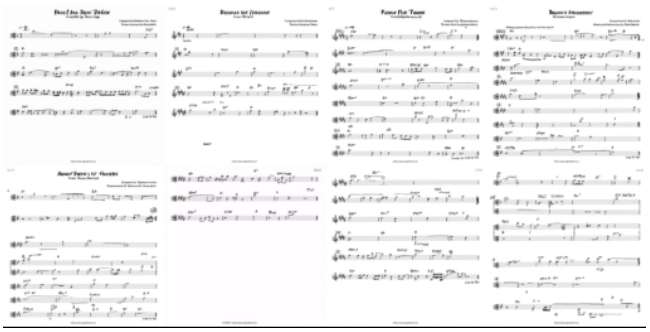

Figure 7. DDPM with L1 loss results

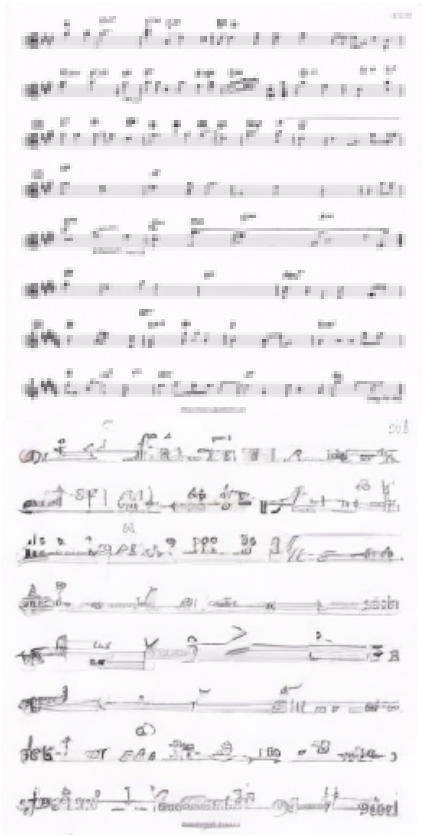

Figure 8.
Down: upscaled image using latent stable diffusiom
Up: fake image from our model

References

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL https://arxiv.org/abs/1406.2661.

[2] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and

accompaniment, 2017. URL https://arxiv.org/abs/1709.06298.

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

[4] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2015. URL https://arxiv.org/abs/1511.06434.

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

[6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. URL https://arxiv.org/abs/2112.10752.