Pictures Of MIDI: CONTROLLED MUSIC GENERATION VIA GRAPHICAL PROMPTS FOR IMAGE-BASED DIFFUSION INPAINTING

Anonymous Authors
Affiliation1

ABSTRACT

Recent years have witnessed significant progress in generative models for music, featuring diverse architectures that balance output quality, diversity, speed, and user control. This study explores a user-friendly graphical interface enabling the drawing of masked regions for inpainting by an Hourglass Diffusion Transformer (HDiT) model trained on MIDI piano roll images. To enhance note generation in specified areas, masked regions can be "repainted" with extra noise. The non-latent HDiT's linear scaling with pixel count allows efficient generation in pixel space, providing intuitive and interpretable controls such as masking throughout the network and removing the need to operate in compressed latent spaces such as those provided by pretrained autoencoders. We demonstrate that, in addition to inpainting of melodies, accompaniment, and continuations, the use of repainting can help increase note density yielding musical structures closely matching user specifications such as rising, falling, or diverging melody and/or accompaniment, even when these lie outside the typical training data distribution. We achieve performance on par with prior results while operating at longer context windows, with no autoencoder, and can enable complex geometries for inpainting masks, increasing the options for machine-assisted composers to control the generated music.

0. INTRODUCTION

Music generation with deep learning models has seen tremendous progress in recent years, enabling algorithmic composition of creative and expressive musical pieces. However, a key challenge that remains is providing users with intuitive control over the generative process. While text prompts and conditioning on attributes like melodies, chords, or timbres have shown promising results [1–3], there is a need to explore richer interactive interfaces that extend user control of constraints on the generative process beyond mere text inputs.

The investigation in this paper is inspired by a typical songwriting composition task in which we seek to create a melody given some accompaniment and a rough idea of when or where we would like the melody to rise and fall [4]. We imagine an application in which the user could draw on the piano roll a rough profile for the "shapes" of the regions in which to melody is to be added, and have the generative model fill in the notes in way that sounds appropriate, given the accompaniment. This process is somewhat akin to the "graphic notation" movement in 20th-century music composition championed by composers such as John Cage [5], Cornelius Cardew [6], and Karlheinz Stockhausen [7], by which performers were encouraged to improvise within certain confines indicated by the shapes drawn by the composer. In this study we imagine not so much the performers improvising, but rather the model generating a composition. Such visual interfaces for generative models have been present in the image-generation domain for quite some time, such as with NVIDIA's Gau-GAN [8], and continue today as with the recent "Semantic Image Synthesis" work of Park et al [9].

An example of such a "prompt" is shown graphically in Figure 1. One may view the prompt as a constraint on the space of generative possibilities. Such constraints have long been linked with creativity and composition [10], as they both limit the field and provide structure around which to generate. For a task such as this, one may then ask the question which generative model architectures are best suited for this kind of control. Transformer-based models for musical audio generation such as MusicGen [11] so far have incorporated text prompts and conditioning on melody, however the introduction of arbitrary inpainting masks poses significant challenges for such an architecture.

Diffusion models have emerged as a powerful framework for generative modeling, achieving state-of-the-art results in domains like image synthesis [12, 13] and audio generation [3, 14, 15], and even symbolic music genera-

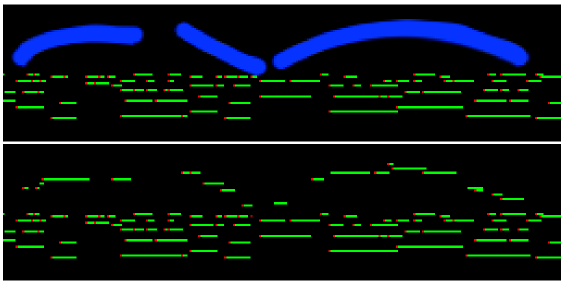

Figure 1: The Motivating Idea. Top: MIDI piano roll image of a sample "graphical prompt" of rough shapes (in blue) of pitches for melody generation given accompaniment (green lines). Bottom: Sample generated output.

—
Preprint. Corresponding author: [anonymized]

tion [16]. A particularly exciting capability is their support for inpainting or infilling, where partial data is provided as a conditioning context, and the model fills in the missing regions [15, 17–19]. This opens up opportunities for user control via sketching or sculpting of the desired output of the kind shown in Figure 1.

Image-based control schemes exemplified by ControlNet [20] have inspired musical audio counterparts [21,22]. The "direct" adaption of image-based music generation was demonstrated notably by Riffusion [23] for audio and Polyffusion [24] for MIDI. The latter set a new standard for quality, diversity, and controlability of outputs. MIDI data offers a convenient set of human-readable priors and very compact data representation, making it an excellent testbed for further research in contollable music generation. Polyffusion and others [25] explored diffusion models operating on piano roll representations, treating musical scores as image-like data. While highly effective, these models are typically limited to short sequences due to memory constraints, and may involve the use of compressed latent representations which typically involve basis functions that fail to correspond with humans' intuitive representations. Our work employs recent advances in high-definition image synthesis with hierarchical diffusion models [26, 27] to overcome this limitation and enable longer musical sequences.

Our key interest lies in exploring intuitive interfaces for constrained music infilling, beyond just text prompts. For example, users could sketch a rough target melody profile which the model then turns into a fully-realized composition [28]. Inpainting over the compact and human-interpretable piano roll domain allows natural specifications of structural constraints.

The key prior works similar to this study are the aforementioned Polyffusion [24] for the image-based generative diffusion of piano roll images, and Benetatos et al [29] offering a drawing-based interface for controllable melody generation. Features which this paper lacks are conditioning on chords, rhythm, and/or texture, which have been explored to great effect the aforementioned prior works. Our current codebase does not yet have these features fully implemented. We also focus only on single-instrument generation rather than multi-instrument compositions.

We believe the following are unique aspects of this paper that contribute to and advance the dialogue on controllable music generation:

1. We replicate the results of Polyffusion [24], achieving comparable outputs in terms of objective and subjective metrics, yet extending beyond earlier work to include
   (a) (4x) larger images and thus (4x) longer sequences
   (b) complex shapes for inpainting masks so users can specify regions and directions of melodic and harmonic development.
   (c) explicit modeling of note velocity
2. We apply the recent Hourglass Diffusion Transformer (HDiT) [26] that efficiently operates in pixel space on large images, thus
   (a) removing the need for a separate autoencoder as in [24]'s use of latent diffusion.
   (b) allowing for the straightforward application of complex inpainting mask shapes (that might otherwise need to be remapped into the autoencoder's latent space).
3. We explore and develop techniques (RePaint [17], nucleation) to encourage note inpainting when outside of training data distribution.
4. We investigate the limitations of inpainting-only approaches. For example, we find that while inpainting is useful for a variety of tasks, we do not find it to be an effective substitute for chord (progression) conditioning.
5. We wish to emphasize that these high quality results have been obtained using the unified, simple, even "naive" approach of simply re-appropriating an existing image diffusion codebase, without introducing specialized architectures or mathematical formalism (as in [29]), and applying inpainting methods commonly seen in image spaces.

In summary, this paper presents a simple, unified, powerful and flexible framework for controllable and interactive music generation by combining the capabilities of diffusion models, hierarchical representations, and intuitive inpainting constraints specified through a visual interface.

We provide an anonymous demo website with listening examples [1] and will release our code upon acceptance.

1. METHODS

1.1 Data Preparation

Following Polyffusion [24], we use the POP909 MIDI dataset [30] of piano arrangements of Chinese pop songs. We normalize all tempos to 120 beats per minute to standardize the piano roll representation, and perform data augmentation by transposing up to +/- 12 semitones, expanding the dataset from the initial 909 MIDI files to 22725 files. Piano roll images are rendered such that each pixel corresponds to a 16th note, and as in [24] we color note durations in green with brightness corresponding to note velocity, and add red onset markers to add an additional layer of verification to rule out spurious diffusion artifacts.

We apply the same chord detector from the Polyffusion code to all songs, through which is its found that 529 different chords exist in the augmented (i.e. variably transposed) dataset. We encode the chord information as borders along the top and bottom 8 pixels of the images, using a base-9 arithmetic scheme to separate each color by 30 color-index-values (ranging from 0 to 255) in each R,G,B channel of the image. The border size of 8 was chosen so as not to overwrite any notes, which given transpositions occur at MIDI pitches as low as 9 and as high as 115. This leaves the bottom 8 pixels and top 12 pixels available to use for other notations such as embedding chord information.

[1] Demo website: https://anonymous.4open.science/w/PoM/
(This site is fully anonymized for double-blind peer review.)

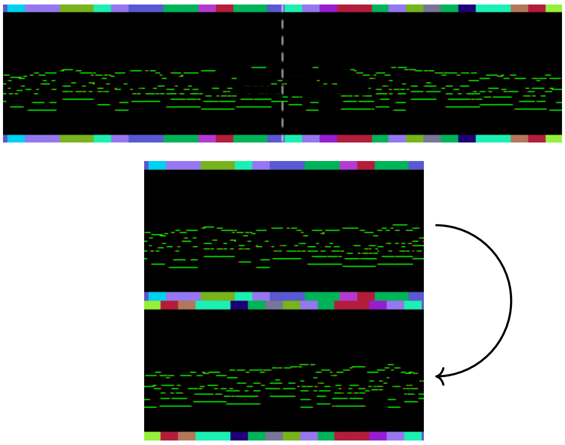

Figure 2: Sample 512x128 MIDI piano roll image. Following Polyffusion [24], we denote notes in green with onsets in red. We also add color-coded chord embeddings as borders along the top and bottom of the image. The right half of the image (after the dashed line) is "folded" underneath the left half to produce a square image suitable for example Hourglass Diffusion Transformer (HDiT) [26] pipelines. After generation, the images are restored to their rectangular format. (Although this "folding" causes a reversal of direction vs. of a simplw copy-paste, we do this so that information need not propagate all the way across the image to maintain musical continuity. In practice we observe no issues with continuity at fold boundary – the model quickly learns to adapt.

Each song in each key is rendered as one long $N_t \times 128$ pixel image where N_t is the total number of 16th notes in the song. From these full song images, will train a model by grabbing random 512-pixel windows (padding the ending with zeros as needed) for each batch of data. Since the HDiT expects square images, we convert each 512x128 image to 256x256 by putting the second half of the original image below the first. To help the network communicate effectively across the cut in the middle, we reverse the lower half of the image horizontally. Figure 2 shows this process. A window width of 512 pixels at 1 pixel per 16th note results in 32 measures, providing approximately 1 minute per image at the standardized tempo of 120 BPM.

1.2 Neural Network Architecture

We use an image-based deep learning system to learn and generate piano rolls. Employing image modalities for music representations has a long history in deep learning research, such as the use of spectrogram images for classification [31]. The closest musical audio analog of this present study might be Riffusion [23], which used Stable Diffusion [12] essentially unaltered to produce images of spectrograms that were then converted to audio via other means.

Instead of Stable Diffusion which operates in a latent space defined by a (VQ-)VAE, we use the recent Hourglass Diffusion Transformer (HDiT) of Crowson et al [26] and the code in the official `k-diffusion` repository [2] with minimal modifications that allow inpainting support. HDiT operates in pixel space and requires no VAE. From Crowson's code repository, we minimally modify the example configuration file for the 256x256 pixel Oxford 102 flowers dataset [32] to our current needs. During training, we dynamically grab batches of random 512-pixel-wide windows from each $N_t \times 128$ MIDI piano roll image. Then, as shown in Figure 2, we "fold" the right half of each windowed image under the left half to supply a 256x256 image to supply for the HDiT code.

1.3 Training

Training was performed for 48 hours on a pair of NVIDIA A6000 GPUs with a batch size of 192. Even in as little as the first 1000 training steps, the model is already putting out discernible notes in green with matching onsets in red, and corresponding chord annotation borders along the top and bottom which mirror each other. The music is sheer cacophony at this point, and the remaining two days improve the musicality. We typically see polynomial convergence of the loss function (relative to the number of steps), or slightly worse. However, eventually the model starts memorizing the data, at which point the convergence becomes exponential. We halt the training shortly after exponential convergence is first detected.

1.4 Sampling and Inpainting

Inference for diffusion models is typically referred to as sampling. It is during sampling that we do the inpainting described below. The `k-diffusion` code supports a variety of higher-order sampling algorithms. We found the Linear Multistep (LMS) sampler to yield poor results when inpainting, and settled on the stochastic variant of the DPM-Solver++(2M) solver [33] instead. (The 3M SDE version of the DPM-Solver++ also produced comparable results.)

Important musical inpainting work includes that of CRASH [15], the question of achieving usable long-term inpainted music was has received special attention in the literature [34, 35]. Our inpainting will rely on a basic binary mask as was outlined in the Polyffusion paper [24], although we must adapt the RePaint algorithm [17] slightly for the `k-diffusion` package's implementation of the mathematical formalism by Karras et al [27]. To wit: we compute the β_t parameters for Repaint using the σ_t variables of Karras via

$$\beta_t = (\sigma_t/\sigma_{\max})^2,$$

where as with comparable HDiT configurations, we use $\sigma_{\max} = 160$.

RePaint involves repeatedly stepping backwards once re-introducing noise at each stage of the reverse diffusion process, with the number of repaint loops (or "repaints") given by some parameter U. For generations in Section 2.1 we use $U = 1$ (i.e., regular diffusion sampling with

[2] https://github.com/crowsonkb/k-diffusion

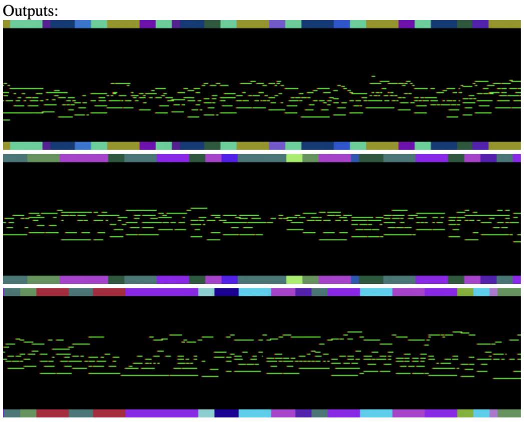

Figure 3: Undirected generation. Here we see model generates various assortments of melody, accompaniment, and "chord borders". Refer to the demo website for listening examples.

no re-paints), but in Section 2.2 we increase U as a way to generate higher note densities in drawn inpainting regions.

Somewhat inspired by the recent work of Lin et al [36] who "algorithmically render all symbolic controls into the audio format before inputting them to MusicGen," we pass chord information into the model as part of each image. Our hope was to achieve results comparable to those obtained from chord-based conditioning but using inpainting alone instead. Rather than supply chords as a conditioning signal, we render "color embeddings" of the chords and affix them to "borders" of unused very high and low pitches, which we observe from our (transposed) dataset to comprise the top and bottom 8 pixels of our 128-pixel-tall MIDI piano roll images.

2. RESULTS

2.1 "Standard" Inpainting Tasks

Figure 3 shows sample outputs from letting the model generate in an undirectected fashion, based on the learned distribution of training data, with no inpainting. We see differences in melody, accompaniment, and chords. Regarding the latter: although the model is able to reliably produce colored chord-marker borders that match on the top and bottom of each image, we find that these colors, when decoded, do not correspond to the chords as detected independently using a chord-extraction code (e.g. from [24]. This demonstrates a limitation of the "all-inpainting" approach in this paper: any chord borders shown in this paper are thus "decorative" and may be cropped out.

Figures 4 through 6 show examples of common musical inpainting tasks. In Figure 4, we mask out the notes in the upper half of the piano roll. The model fills in melody lines consistent with the underlying chord structure and accompaniment. Figure 5 shows the reverse, generating accompaniment given a melody.

Not shown are the *negative* results from our inpainting studies: anything involving the chord markers – inpainting notes given chords and vice versa – does not work, in

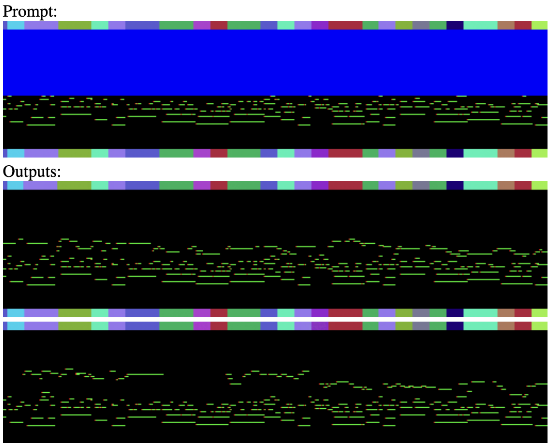

Figure 4: Melody inpainting. The top portion of the piano roll is masked out in blue (top image), and the model generates melodies (shown in the bottom 3 images) that fit the accompaniment notes and the chords (denoted by colored bars along the top and bottom).

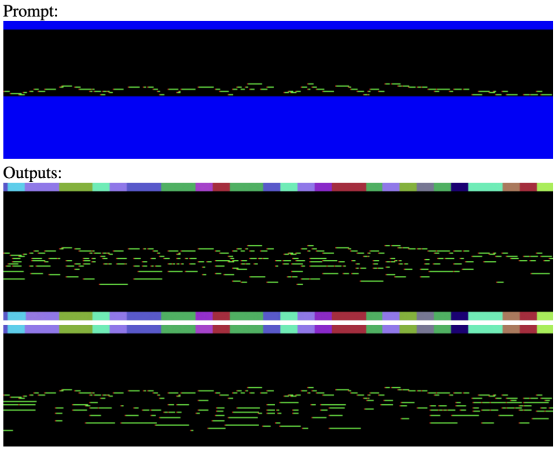

Figure 5: Accompaniment inpainting, given a melody. It is noteworthy that the chords shown in the top and bottom borders are the same for each output even though the accompaniment notes differ.

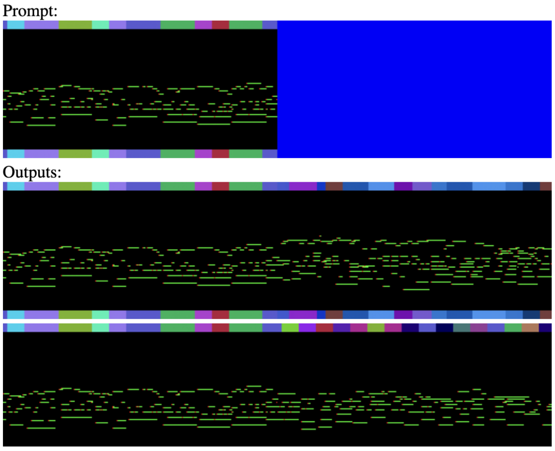

Figure 6: Continuation / "outpainting," offering novel continuations from the same starting music.

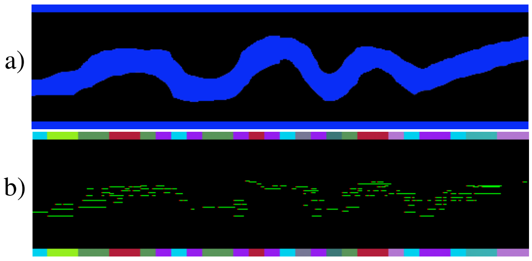

Figure 7: Generation constrained by "creative drawing" of inpainting mask, Case 1.

the sense that the generated chord markers appear to be arbitrary and do not match the notes at in any significant sense, and the notes generated do not follow the chords requested. It is possible that a revised inpainting scheme such as RePaint [17] would be sufficient to allow these tasks to be completed via inpainting alone, although finding the proper implementation of RePaint for multi-step k-diffusion [27] integration is thus far an open question. Otherwise, inpainting alone will need to be abandoned in favor of chord based conditioning, the effectiveness of which was demonstrated by Min et al. [24].

2.2 "Creative" Inpainting Tasks

We now consider cases where a user may have an idea of the overall shape that they would like the music to follow, expressed via drawing the mask via some graphical interface. (For this paper, these were drawn via GIMP and/or the Mac's Preview app. [3]) The model is tasked with adding notes that "make sense" and meet the user's criteria.

The examples shown in Figures 7 and 8 are deliberately "extreme" in the sense of featuring pitch ranges that fluctuate over larger intervals and shorter timescales than is found in the training data distribution. As a result of this domain mismatch, the model typically produces relatively few notes meeting the user's criteria, instead outputting mostly silence.

It is possible to "brute force" the creation of satisfactory(-looking) outputs by generating a large number of example outputs and ranking them according to the area of the mask filled in by notes (i.e., by taking the dot-product of the mask and the generated image). These satisfactory outputs are therefore low-probability events. The examples shown in Figures 7 and 8 are thus *automatically cherry-picked* via this ranking method, obtained by generating 100 outputs at a time and taking only the top 2 outputs in the ranking. An alternative to this brute force method, namely "seeding nucleation," is explored in the next subsection.

Given that these examples differ significantly from the training data distribution, it is perhaps unsurprising that the generated music is not especially "musical" or aesthetically pleasing in the sense of pop tunes. Thus the utility of such extreme masking cases may yet have little practical utility without additional and more varied training data, conditioning signals, and/or diffusion guidance. This remains an area for further study.

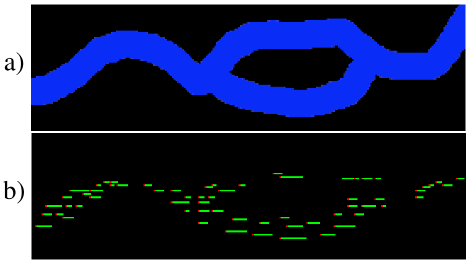

Figure 8: Generation constrained by "creative drawing" of inpainting mask, Case 3. Here the user requests that the bass line and melody diverge leaving a sizable gap in between, and then converge.

2.3 Increasing Inpainted Note Density

Depending on how closely or poorly the user's visual prompt conforms to typical data in the training data distribution, the model may have a respectively higher or lower probability of providing matching outputs. In Figure 9a), we see an attempt to generate notes in a mask shape that extends higher than typical melodic ranges for POP909. The results of these generations can be "hit or miss" in that there can be a low probability of notes appearing at these high pitch values as demonstrated in Figure 9b). Applying the automated brute-forced cherry-picking described in Section 2.2, we can get more notes as shown in Figure 9c), however this is computationally needlessly expensive. More effective is to combine the ranked cherry-picking method for a smaller number of generated outputs in which we increase the RePaint parameter U from 1 to 2. These results are shown in 9d).

2.4 Evaluation

In theory, a desirable model would generate music that is both of high quality and high diversity. While the diversity of outputs can be measured statistically, it is difficult to objectively define "quality." Thus as a proxy for quality, we tend to objectively measure the statistical similarity of the generated data to the training data and supplement this with subjective evaluations using human preferences measured via listening tests.

For objective diversity measurements, it's appropriate to use the standard deviation of MIDI pitches σ_{P} since these are approximately normally distributed. The note durations, however, are heavily skewed toward small values, and thus a metric such as the inter-quartile range $\mathrm{IQR}_{\mathrm{D}}$ [4] is better suited to convey the statistical spread of data rather than the standard deviation (which can be heavily influenced by the the long tail). Table 1 summarizes these measurements for various generative outputs. For our objective similarity measurements, we compare the training data with generated data by computing the average overlapped area of pitch distribution ($\mathcal{D}_P$) and duration distribution

[3] We also have a Gradio GUI demo in the works, pending the resolution of a reported issue with Gradio's image editor.

[4] This IQR covers 25% to 75% of the sorted data points.

Name of Dataset / Model	$\sigma_P \uparrow$	$\mathrm{IQR}_D \uparrow$	$\mathcal{D}_P \uparrow$	$\mathcal{D}_D \uparrow$	$\mathcal{D}_{\mathrm{KL},P} \downarrow$	$\mathcal{D}_{\mathrm{KL},D} \downarrow$
Original MIDI	0.46	10.6	n/a	n/a	n/a	n/a
Piano Roll Images (±12 semitones)	0.88	13.6	n/a	n/a	n/a	n/a
PoM (ours)	1.0	11.7	0.87	0.95	0.036	0.14
Polyffusion (our execution)	0.38	11.0	0.88	0.95	0.0050	0.13

Table 1: Objective Metrics. For MIDI pitches (P) and note durations (D), we measure the diversity of values via the standard deviation of pitches σ_P and the inter-quartile range of durations (IQR_D). Similarity to the training data is computed via the average overlapped area of pitch distribution ($\mathcal{D}_P$) and duration distribution ($\mathcal{D}_D$), as well as Kullback–Leibler divergences $\mathcal{D}_\mathrm{KL}$. All values should be taken as +/- 1 on the last digit.

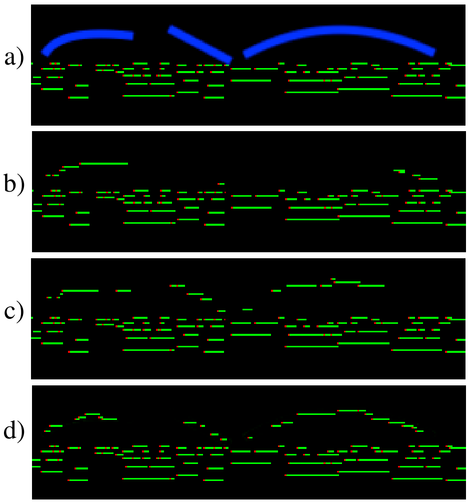

Figure 9: Driving Inpainting. a) The user's promp veers a bit high compared to the training data. b) Typical output shows few notes generated. c) By brute force (cf. Section 2.2) we can get more notes. d) More efficiently, we can get more notes by increasing the RePaint parameter U from 1 to 2. As we increase U to 3, 4, or more, we get more notes but they increasingly show little musical coherence and tend toward apparent randomness. Our attempts to spell out words in the MIDI in the style of Jacob Collier [37] have so far produced nothing close to his level of musicality.

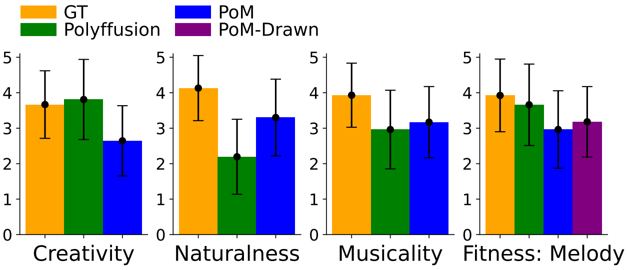

Figure 10: Subjective evaluation for unconditional generation. The error bars may suggest that, for the examples provided, listeners found all examples to be comparable. "GT" denotes ground truth music from the POP909 MIDI Dataset. The "PoM-Drawn" results are for melodies generated to follow pre-made user-drawn inpainting masks as in Figure 9, whereas "PoM" alone denotes "typical" unconditioned melody generation as in Figure 4.

($\mathcal{D}_D$) as in [24]. In addition to these we compute Kullback–Leibler divergences $\mathcal{D}_\mathrm{KL}$ of probability density functions (PDF) for each data distribution, obtained via kernel density estimation on histograms of the MIDI pitches ($\mathcal{D}_\mathrm{KL,P}$) and note durations ($\mathcal{D}_\mathrm{KL,D}$).

Subjective listening evaluations were performed by 54 volunteers, half of whom were senior Audio Engineering undergraduate majors, and the remaining half were drawn from the general population. We repeated the Polyffusion paper's criteria of "Creativity," "Naturalness" (i.e., "how likely a human musician composed the music"), overall "Musicality," and "Fitness" of a generated melody over a fixed accompaniment. Evaluators listened to two 16-second long examples drawn from the "ground truth" POP909 MIDI dataset, generations from our execution of the Polyffusion code (using their pretrained checkpoints), and our model. Results are shown in Figure 10, which indicate that listeners found all examples to be comparable in quality. Notably, the "Fitness" evaluation included examples of the "creative inpainting" variety with a Repaint value of 2 and with drawn melody shapes that, unlike those shown earlier, were of a modest range of notes, thus staying within the typical range of the training data distribution. One further finding is that requiring red onset markers for our model makes no difference to our metrics, which agrees with the Polyffusion paper [24]: "In practice, the generation process of 160 8-bar samples report zero invalid notes." Taken together these evaluations suggest that the PoM model generates output comparable to Polyffusion [24] while supporting longer contexts and arbitrary inpainting mask shapes.

3. CONCLUSIONS

We have presented a unified method for image-based diffusion inpainting that performs comparably to prior inpainting work, yet requires no additional autoencoder due to its use of a Hourglass Diffusion Transformer (HDiT). The HDiT allows for efficient processing of large images allowing longer sequence contexts than previous approaches, and also allows for the straightforward and interpretable use of complex mask geometries for inpainting. These simplifications and extensions, along with the relative ease of appropriating the `k-diffusion` codebase, suggest that image-based piano roll diffusion shows increasing promise for enabling composers to exert greater control during the generative music creation process. Future studies may include scaling images up, semi-transparent masks, and multi-instrument piano rolls.

4. REFERENCES

[1] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank, "Musiclm: Generating music from text," 2023.

[2] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, O. Teboul, D. Grangier, M. Tagliasacchi, and N. Zeghidour, "Audiolm: a language modeling approach to audio generation," arXiv preprint arXiv:2209.03143, 2022.

[3] Z. Evans, C. Carr, J. Taylor, S. H. Hawley, and J. Pons, "Fast timing-conditioned latent audio diffusion," 2024.

[4] L. Bell, "Hitmaking For Producers: The Professional Method For Producing Radio-Ready Songs. 2. Writing the Melody," May 2021. [Online]. Available: https://studio.com/louis-bell/hitmaking-for-producers

[5] J. Cage, Ed., Notations, first edition ed. Something Else Press, Jan. 1969.

[6] B. Dennis, "Cardew's 'treatise' (mainly the visual aspects)," Tempo, no. 177, p. 10–16, 1991.

[7] T. Sauer and M. Perry, Notations 21. New York, NY: Mark Batty Publisher, Apr. 2009.

[8] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," 2019.

[9] J. Lee, D. S. Jung, K. Lee, and K. M. Lee, "Streammultidiffusion: Real-time interactive generation with region-based semantic control," 2024.

[10] I. Stravinsky, Poetics of music: in the form of six lessons, 16th ed. Cambridge, Mass: Harvard University Press, 1970.

[11] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," 2024.

[12] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," CoRR, vol. abs/2112.10752, 2021.

[13] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, T. Salimans, J. Ho, D. J. Fleet, and M. Norouzi, "Photorealistic text-to-image diffusion models with deep language understanding," 2022.

[14] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," in International Conference on Learning Representations, 2021. [Online]. Available: https://openreview.net/forum?id=a-xFK8Ymz5J

[15] S. Rouard and G. Hadjeres, "CRASH: Raw audio score-based generative modeling for controllable high-resolution drum sound synthesis," in Proc. ISMIR, Nov. 2021.

[16] G. Mittal, J. Engel, C. Hawthorne, and I. Simon, "Symbolic music generation with diffusion models," in Proc. ISMIR, 2021.

[17] A. Lugmayr, M. Danelljan, A. Romero et al., "RePaint: Inpainting using denoising diffusion probabilistic models," in Proc. IEEE/CVF Conf. CVPR, Jun. 2022, pp. 11 461–11 471.

[18] E. Moliner and V. Välimäki, "Diffusion-based audio inpainting," 2023.

[19] C.-J. Chang, C.-Y. Lee, and Y.-H. Yang, "Variable-length music score infilling via xlnet and musically specialized positional encoding," 2021.

[20] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in IEEE International Conference on Computer Vision (ICCV), 2023.

[21] S.-L. Wu, C. Donahue, S. Watanabe, and N. J. Bryan, "Music controlnet: Multiple time-varying controls for music generation," 2023.

[22] Z. Novack, J. McAuley, T. Berg-Kirkpatrick, and N. J. Bryan, "DITTO: Diffusion inference-time t-optimization for music generation," 2024.

[23] S. Forsgren and H. Martiros, "Riffusion - Stable diffusion for real-time music generation," 2022. [Online]. Available: https://riffusion.com/about

[24] L. Min, J. Jiang, G. Xia, and J. Zhao, "Polyffusion: A Diffusion Model for Polyphonic Score Generation with Internal and External Controls," Jul. 2023, arXiv:2307.10304 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2307.10304

[25] H. Wang, "DiffuseRoll: Multi-track multi-category music generation based on diffusion model," Mar. 2023, arXiv:2303.07794 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2303.07794

[26] K. Crowson, S. A. Baumann, A. Birch, T. M. Abraham, D. Z. Kaplan, and E. Shippole, "Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers," 2024. [Online]. Available: https://arxiv.org/abs/2401.11605

[27] T. Karras, M. Aittala, T. Aila, and S. Laine, "Elucidating the design space of diffusion-based generative models," in Proc. NeurIPS, Dec. 2022.

[28] C.-P. Tan, A. W. Y. Su, and Y.-H. Yang, "Melody infilling with user-provided structural context," in Proc. ISMIR, 2022.

[29] C. Benetatos and Z. Duan, "Draw and listen! A sketch-based system for music inpainting," in Proc. ISMIR, 2022.

[30] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, G. Bin, and G. Xia, "Pop909: A pop-song dataset for music arrangement generation," in Proc. ISMIR, 2020.

[31] K. Choi, G. Fazekas, and M. B. Sandler, "Automatic tagging using deep convolutional neural networks," in Proc. ISMIR, vol. abs/1606.00298, 2016. [Online]. Available: http://arxiv.org/abs/1606.00298

[32] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in Indian Conference on Computer Vision, Graphics and Image Processing, Dec 2008.

[33] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models," 2023.

[34] E. Moliner, J. Lehtinen, and V. Välimäki, "Solving audio inverse problems with a diffusion model," in Proc. IEEE ICASSP, 2023, pp. 1–5.

[35] K. Liu, W. Gan, and C. Yuan, "Maid: A conditional diffusion model for long music audio inpainting," in Proc. IEEE ICASSP, 2023, pp. 1–5.

[36] L. Lin, G. Xia, Y. Zhang, and J. Jiang, "Arrange, inpaint, and refine: Steerable long-term music audio generation and editing via content-based controls," 2024.

[37] "Jacob collier can spell out words with a piano," Jan 2021. [Online]. Available: https://www.youtube.com/watch?v=LszYGO22azA