PSEUDO AUTOREGRESSIVE INFERENCE USING DIFFUSION MODEL FOR PIANO AUDIO SYNTHESIS

Yuze Wang
University of Rochester
ywang349@u.rochester.edu

Chengkai Kang
University of Rochester
ckang12@u.rochester.edu

ABSTRACT

Diffusion Denoising Probabilistic Models (DDPM) [1] aim to learn the underlying data distribution of some observations. Although similar in objective to that of generative adversarial networks (GAN) [2] and variational auto-encoders (VAE) [3], a DDPM differs in its robustness towards model architecture and training procedure. Thanks to this robustness, the domain of image generation has shown remarkable results in both quality and variety. While the recent diffusion models focus on the problem domain of generating images, we would like to utilize diffusion on sound waves. In this paper, we propose a method to continuously expand the Waveform domain as a way to mimic autoregressive behavior, and a novel sampling procedure that aims to create a harmonizing result.

1. INTRODUCTION

In recent years, generative modeling has taken center stage across a multitude of scientific domains, eliciting noteworthy contributions in the fields of natural language processing [4] and image synthesis [1]. The success in these fields has culminated in generative outputs that approach human-like quality, indiscernible to the untrained eyes or ears. Despite these achievements, music synthesis remains a relatively under-explored area within the generative modeling landscape. This deficit is primarily attributable to the inherent complexities associated with music data. Specifically, musical compositions not only comprise long sequential structures but are also replete with intricate frequency spectra [5]. Consequently, raw music data in the waveform domain manifests as exceedingly dense informational entities.

Moreover, the challenges associated with music synthesis are further exacerbated by its compositional versatility. Unlike images, which are generally synthesized from a restricted palette of colors, music is born out of a rich tapestry of instrumental timbres and voices, each contributing its own unique qualities. This multiplicity of input variables presents a complex landscape for the task of effectively modeling the underlying data distribution, rendering it a compelling yet formidable research challenge.

Given the foundational commonalities between image and music synthesis—where the primary elements of images are colors, and in music, it is the fundamental frequency, denoted as f_0—we posit that advances in image synthesis techniques may be ported to the domain of music synthesis. To investigate this hypothesis, we turn our focus to diffusion techniques, which have demonstrated both simplicity and robustness in their capacity to model complex data distributions in various domains. However, the direct transposition of Denoising Diffusion Probabilistic Models (DDPM) to the domain of music synthesis is not without its challenges. Notably, the inherently non-autoregressive nature of conventional DDPM algorithms imposes a constraint of fixed sequence length on the generated output. While such limitations may be inconsequential within the context of image synthesis, they constitute a significant bottleneck for musical compositions, which frequently necessitate variable-length sequences.

In light of these challenges, the primary objective of this study is to develop a methodology that allows for autoregressive sequence inference within the diffusion framework. To this end, We proposed the use of image inpainting techniques as the inference method to mimic the behavior similar to that of an autoregressive model. We also modified the original Repaint [6] technique in favor of an algorithm that significantly reduces the reverse steps needed. In doing so, we anticipate broadening the potential applicability of DDPM techniques in the sphere of music synthesis, thereby filling an existing gap in the literature.

2. RELATED WORK

2.1 Neural Audio Synthesis

Over the course of recent years, the field of neural audio synthesis has undergone significant advancements. One of the earliest breakthroughs was Wavenet [7], as it showed an impressive result in generating audio sequences. It employed an autoregressive architecture to facilitate the direct sampling of audio sequences within the waveform domain, albeit at a computational cost. Subsequent work in the form of Vector Quantized - Variational Auto Encoder (VQ-VAE) [8] took a similar approach. Instead of directly generating new samples, VQ-VAE compresses raw waveform data into a quantized codebook, which is subsequently decoded using WaveNet. On the other hand, HiFi-GAN [9] and Rifffusion [10], both of which rely on Mel-spectrogram conditioning as opposed to raw waveform data, have also demonstrated impressive results.

Nevertheless, conditioning upon the Mel-Spectrogram entails a loss of information relative to the original data distribution, thereby introducing a degree of imprecision during the generative process. Recent methodologies [11,12], have attempted to address this issue by compressing the raw audio data into a latent space, conditioned by an autoregressive decoder to yield high-fidelity audio outputs. Distinctively, these models utilize a cascaded form of residual quantized codebook, thereby facilitating a more accurate discrete representation compared to predecessor models like VQ-VAE [8].

2.2 Diffusion

Diffusion models [1] present several advantages over adversarial methodologies, particularly in terms of their straightforward L2 loss objective function and the stability of their training regime, making them well-suited for applications in image synthesis [13,14]. For instance, DiffWave [15] leverages diffusion-based techniques and adapts them to a custom vocoder architecture. Extensions of this model, such as PriorGrad [16] further refine the DiffWave [15] by introducing a better noise distribution. Instead of the standard Gaussian noise, the author extracts the energy of the conditioned Mel-Spectrogram and adopts the prior noise distribution to the target audio. WaveGrad [17] is similar but instead of the discrete noise level, it is conditioned on the continuous noise level. Hierarchical diffusion model for singing voice generation [18] on the other hand, extends PriorGrad [16] in a cascade diffusion style, where the several diffusion models are combined together. The base model learns the low sample representation and the latter models learn to upscale the input. Such a process is inspired by the super-resolution cascade technique [19], which generates sample at high fidelity. A more recent approach in producing high-quality sample is first to compress the data into a latent representation, after applying the diffusion process, it is decoded back into the original data domain. This was first used by latent diffusion [14] for image synthesis, and forms the very idea of multi-band diffusion [20] for music synthesis.

Even though autoencoder-based architectures are inherently non-autoregressive, there have been concerted efforts to apply diffusion techniques in an autoregressive framework. TimeGrad [21] seeks to tackle time-series forecasting challenges using DDPM and incorporates a RNN [22, 23] to encode prior window information for conditional diffusion. This autoregressive adaptation of diffusion is particularly pertinent in the realm of video generation, which inherently consists of temporally-linked image sequences. Residual Video Diffusion [24] improved the TimeGrad approach [21] by generating a residual to a deterministic next-frame prediction.

3. METHOD

3.1 DDPM

At a very high level, the diffusion model samples noises from a Gaussian distribution and add these noises to the original data. After sufficient number of steps, the data becomes pure noise. Then, the model tries to learn how to remove the noises to reconstruct the original data. More formally, it is a two-step process where the data distribution is first gradually destroyed by adding Gaussian noise, and later gradually denoised by removing the predicted noise.

The noising process, or forward diffusion process is just a simple Markov process:

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{1}$$

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t|\sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{2}$$

Here, β_t is a fixed variance schedule with $\beta_t \in (0, 1)$. However, since the normal distribution can be parameterized as $z = \mu + \sigma\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$, the result of the Markov process at any timestep t can be calculated in a single step. let $\alpha = 1 - \beta_t$ and $\hat{\alpha} = \prod_{t=1}^{T} \alpha_t$, we derive:

$$\sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon \tag{3}$$

$$\sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon \tag{4}$$

$$\sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\epsilon \tag{5}$$

$$\sqrt{\alpha_t\alpha_{t-1}\alpha_{t-2}}x_{t-3} + \sqrt{1-\alpha_t\alpha_{t-1}\alpha_{t-2}}\epsilon \tag{6}$$

$$\vdots$$

$$\sqrt{\alpha_t\alpha_{t-1}\ldots\alpha_1\alpha_0}x_0 + \sqrt{1-\alpha_t\alpha_{t-1}\ldots\alpha_1\alpha_0}\epsilon \tag{7}$$

$$x_t = \sqrt{\hat{\alpha_t}}x_0 + \sqrt{1-\hat{\alpha_t}}\epsilon, \epsilon \sim \mathcal{N}(0, I) \tag{8}$$

The denoising process, or reverse diffusion process can be thought of as approximating the posterior of the diffusion process, which can be expressed as follows:

$$p_\theta(x_{0:T}) = p_\theta(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \tag{9}$$

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}|\mu_\theta(x_t, t), \gamma I) \tag{10}$$

Since the forward procedure is fixed, we are only interested in learning the parameterized $p_\theta(x_{t-1}|x_t)$. Our parameter θ can be optimized by maximizing the evidence lower bound (ELBO) [25].

$$logp(x) = log\,\mathbb{E}_q[\frac{p(x_{0:T})}{q(x_{1:T}|x_0)}] \tag{11}$$

$$\begin{aligned} logp(x) = \;& \mathbb{E}_q[logp_\theta(x_0|x_1)] \\ & - D_{KL}(q(x_T|x_0)||(x_T)) \\ & - \sum_{t=2}^{T} \mathbb{E}_q[D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)] \end{aligned} \tag{12}$$

Note that deriving (12) from (11) requires us to rewrite the encoder transitions as $q(x_t|t_{t-1}) = q(x_t|x_{t-1}, x_0)$, this can intuitively be understood as knowing the original data distribution helps lowering the variance during our Monte Carlo estimation. However, rewriting does not affect the result of the Monte Carlo estimation since the extra term is superfluous under the Markov property. Given (12), Ho et al. [1] claims that optimizing the last KL-divergence only is sufficient for the model to converge. Therefore, our objective function can be written as

$$argmin \frac{1}{2\sigma_q^2(t)} \frac{\hat{\alpha}_{t-1}(1-a_t)^2}{(1-\hat{\alpha}_t)^2}[||\hat{x}_\theta(x_t,t)-x_0||_2^2] \quad (13)$$

In the original experiment, Ho et al. [1] found out that ignoring the scaling terms at the front of the L2 loss leads to better training results, therefore our final objective becomes a simple L2 loss. In the original paper [1], Ho et al. used another loss function $||\hat{\epsilon}_\theta(x_t,t)-\epsilon||_2^2$ that optimizes for the noise difference. This interpolation is equivalent to the above equation along with score matching interpolation $||s_\theta(x_t, t - \nabla logp(x_t))||_2^2$ [25,26].

3.2 Pesudo Autoregressive Inference

Recall that an autoregressive model predicts the probability of a subsequent token based on its predecessors. In other words, this model uses accumulated historical data to forecast the next token or sample. This process is mathematically expressed as:

$$logp(x) = \prod_{i=1}^{D} p(x_i|x_{<i}) \quad (14)$$

Therefore, in replicating this autoregressive approach, our inference model must incorporate spatial dependencies in its predictions. We selected Diffwave [15] for this purpose due to its structural similarities with WaveNet [7]. Diffwave, adapting WaveNet's architecture, effectively captures temporal information during the generation process. However, this sequence creation is confined to individual generations, resetting with each new generation. As a result, each sampling instance in Diffwave [15] disregards previous generations.

Our goal, then, is not simply to generate new samples using Diffwave [15], but rather to extend the generated audio, imitating the autoregressive model's functionality. The objective is to create extended audio data $data^{new}$ of length D, building upon existing audio $data^{known}$ of shorter length B. This approach can be conceptualized as predicting $data^{new}$ based on $data^{known}$

$$logp(data^{new}) = p(data^{new}|data^{known}) \quad (15)$$

It's important to note that Diffwave's maximum generation capacity is D, aligning with the maximal context and generation length of a traditional autoregressive model. To circumvent the limitation of generation length inherent in autoencoder models, we can create longer final audio data by stacking multiple generated "frames," as illustrated in Figure 1.

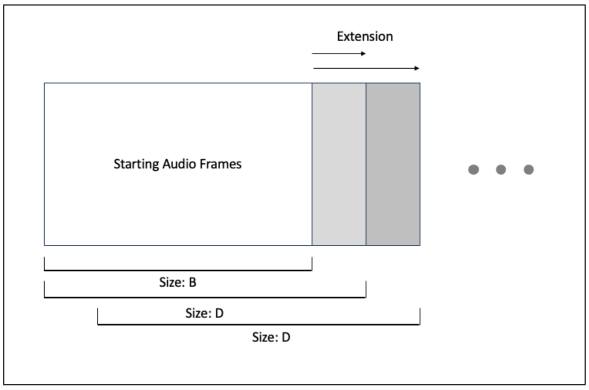

Figure 1. Stacking of audio frames

3.3 Image Inpainting and Resampling

Extending audio sequences can be understood as an audio inpainting problem. Here, our objective is to predict an unknown audio segment of size D - B. Drawing inspiration from RePaint [6], we approach this using diffusion models:

$$x_{t-1}^{known} \sim \mathcal{N}(\sqrt{\hat{\alpha}_t}x_0, (1-\hat{\alpha}_t)I) \quad (16)$$

$$x_{t-1}^{unknown} \sim \mathcal{N}(\hat{x_\theta}(x_t^{new},t), \beta I) \quad (17)$$

$$x_{t-1}^{new} = x_{t-1}^{known} + (D-B) \odot x_{t-1}^{unknown} \quad (18)$$

In these equations, The + indicates the concatenation of two 1-D matrices, while $\odot$ signifies the selection of a length segment. Since the diffusion reverse step from x_t to x_{t-1} relies solely on x_t, we modify the reverse process at each time step t by incorporating the known region, ensuring that x^{new} includes the conditional information from x^{known}.

A significant challenge with this method is achieving harmony between x^{known} and $x^{unknown}$ in the resulting x^{new}. While the diffusion model $\hat{x_\theta}$ attempts to harmonize the overall data distribution at each time step, it struggles to produce a consistent harmonized distribution across t, due to:

1. The sampling of x^{new} excludes the $B \odot x^{unkown}$ region, leading to a loss of crucial information in each reverse step.

2. The diminishing β value during the diffusion process limits the model's ability to introduce significant changes to the latent distribution at lower t values.

The original RePaint [6] paper addressed this by adding extra steps for harmonization. Specifically, it re-noises $x_t^{new} \sim \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t I)$ essentially allowing the model to backtrack in the diffusion process. This backtracking provides the opportunity to find a new path that better integrates the generated and unknown distributions. However, this solution significantly extends the diffusion process duration, as it requires multiple of t steps to complete due to the backtracking operation.

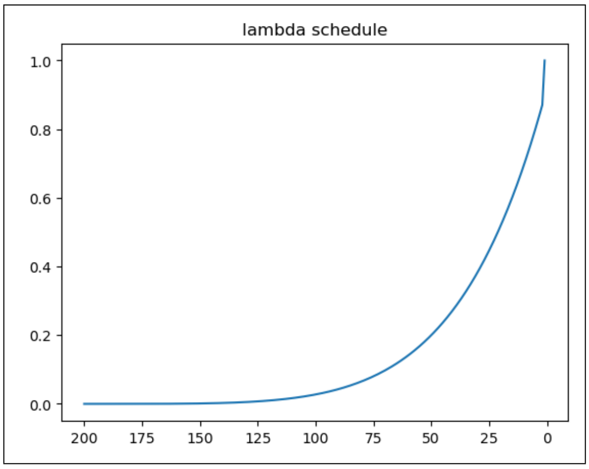

Figure 2. Reverse Lambda schedule. At the final step, lambda must equal 1 to avoid generating a x_0 that is a mix of x^{known} and $x^{unknown}$.

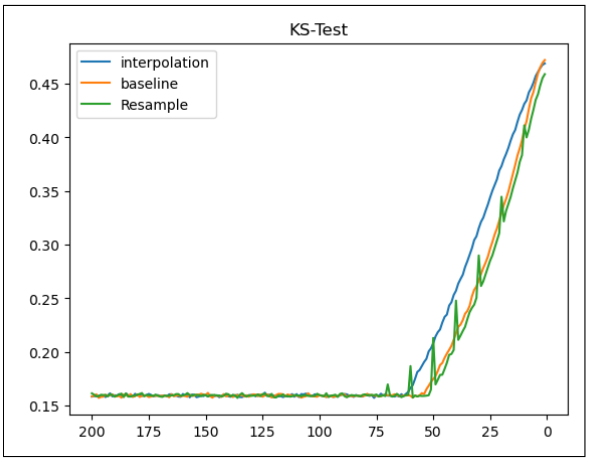

Figure 3. Kolmogorov-Smirnov Test for different methods against the original at different t, higher the better.

3.4 Interpolation Guidance and Reverse Lamdba Schedule

Addressing the harmonization issue in audio inpainting necessitates tackling the two identified challenges. Firstly, rather than discarding the $B \odot x_{t-1}^{unknown}$ segment, we should integrate it with the x^{known} region. This integration can be achieved through interpolation [1], as defined by $x^{combined} = \lambda x^{unknown} + (1-\lambda)x^{known}$. Although one might think that adding x^{known} and $x^{unknown}$ directly would be intuitive; such an operation would incur numeric blowup, therefore an interpolation scale is needed to ensure the latent blend is within a numeric bound. The blend of $x^{unknown}$ and x^{known} still influences the $(D-B) \odot x_{t-1}^{unknown}$ region, given Diffwave's reliance on temporal dependencies during generation, meaning later parts are generated considering this mixture.

The interpolation creates a latent distribution that is intermediate between $x^{unknown}$ and x^{known}. To ensure the final x_0 accurately reflects x^{known}, we ultimately wanted an extreme instead of a mix. We've designed our λ schedule as an exponential function (shown in fig 2, eq 19) that converges to 1 as t nears 0, tailored specifically for 200 steps in the diffusion process. This choice is driven by the fact that most denoising activity occurs at lower t values [15], necessitating more dramatic changes for effective latent shaping [1] .

Initially, it seemed logical to increase λ for x^{known} as t approached 0, ensuring the diffused x_0 matches the diffused x_0^{known}. However, this approach did not resolve the unharmonized $(D-B) \odot x^{unknown}$ issue due to the diminishing β problem (issue 2). This problem implies that when x^{known} significantly influences the latent space, the model is restricted in its ability to effect changes. Counterintuitively, by inverting the λ value, we found that the final data distribution still aligns with x^{known}. We hypothesize that the denoising process at a high t value aims to create a uniformed noised distribution, which at a lower t value it aims to denoise the said distribution. Therefore, by incorporating a high value of x^{known} in the early steps, we direct $x^{combined}$ to target an outcome incorporating x^{known}'s distribution. The later steps can then focus on harmonization and denoising.

$$\lambda = \begin{cases} 1, & \text{if } t = 0 \\ INVERSE(0.3704e^{0.5t} - 1)^4, & \text{otherwise} \end{cases} \tag{19}$$

$$x_{t-1}^{combined} = (1-\lambda)x_{t-1}^{known} + \lambda(B \odot x_{t-1}^{unknown}) \tag{20}$$

$$x_{t-1}^{new} = x_{t-1}^{combined} + (D-B) \odot x_{t-1}^{unknown} \tag{21}$$

4. EXPERIMENT

4.1 Model setup and training

For our project, we configured Diffwave [15] following the original architecture proposed by the authors. This setup included 30 residual layers and a maximum of 64 residual channels. Although Diffwave is capable of conditioning on Mel-Spectrograms during both training and inference, we decided not to use this feature. Our focus was on generating raw waveforms through diffusion, making Mel-Spectrogram conditioning unnecessary for this project.

Our dataset was sourced from Kaggle and comprised 14 monophonic piano sounds of varying lengths. The audio files were sampled at 22.05 KHz, and the total duration of the dataset amounted to 1,217 seconds or approximately 20.283 minutes.

During the training phase, we randomly selected four 5-second audio clips ('snapshots') for each training step, feeding these into Diffwave. The training process extended over approximately 730,000 steps, and we observed the final L2 loss stabilizing around 0.03. The entire training duration was roughly 14 days, conducted on a single A5000 GPU with a 24 Gb memory capacity.

[1] Indeed, we found that using a linear schedule results in a worse reconstructing quality

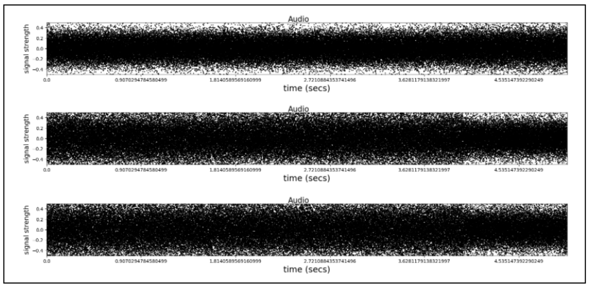

Figure 4. Wavform representation for x at $t = 25$, from top to bottom: our Method, do-nothing, and Repaint. The abscissa is in seconds

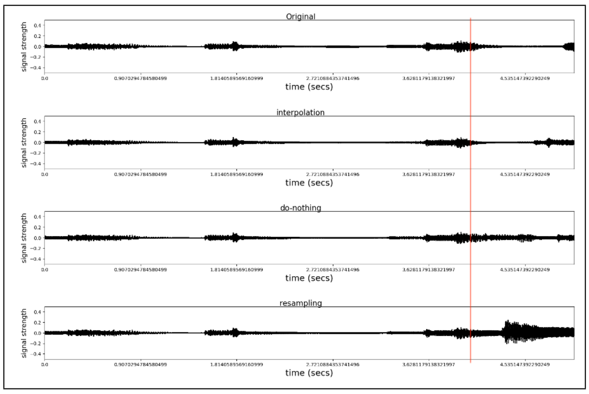

Figure 5. Wavform representation for x at $t = 0$, from top to bottom: original, our Method, do-nothing, and Repaint. The area beyond the redline is the generated content.

4.2 Comparision Study

Given the subjective nature of music and the constraints of our resources, we approached the evaluation of interpolation guidance empirically. In our experiment, we tasked various algorithms with reconstructing a 5-second audio segment, using a 4-second window as a reference. We compared our algorithm's performance against both a Resampling method (with jumping and backtrack length of 10) and a do-nothing baseline (eq 16-18), focusing on the output sequence and the denoising history.

Our method consistently produced a uniform noise distribution throughout the denoising process, particularly noticeable at $t = 25$. In these instances, the distinction between the generated and reference segments was minimal. Numeric analysis using Kolmogorov-Smirnov Test shows that our method has a higher similarity to x^{known} as shown in fig 3. This uniformity contributed to a more harmonized final output, as evident in x_0. As shown in fig 5, the impulse shape closely mirrors the original audio relative to other methods.

We extended our analysis by generating three different 8-second audio segments from 5-second starting points using various methods. We chose an 8-second duration as it is sufficiently long to demonstrate melodic structure while short enough for Diffwave to maintain reference to the original audio. To validate the coherence and superiority of our method's tempo, we conducted a pitch analysis using the Yin algorithm [27], with results presented in fig 6.

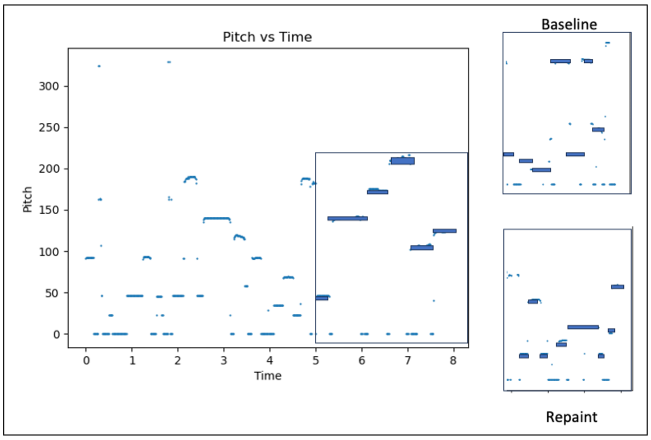

Figure 6. Pitch analysis of 8-second audio generated using different algorithms. Abscissa is in seconds. Anything Beyond 5 seconds are generated parts which are marked square marker. The right-hand sides are the generated parts using baseline and Repaint (top to bottom)

Visually inspecting, the pitch structure of the audio generated by our method yields a rather organized shape when compared to others.

5. DISCUSSION

While our studies have shown promising results in comparison to other audio extension methods, they do not conclusively prove the effectiveness of our approach. In an ideal scenario, we would assess our method using subjective metrics like the Mean Opinion Score (MOS). However, due to resource limitations, conducting such a study is currently beyond our scope. Additionally, our hardware and time constraints mean that our custom-trained Diffwave model cannot perfectly reconstruct tempo and melody. Presently, it produces sounds resembling piano music, but lacks organized musical notes. This limitation complicates our comparative studies, challenging our ability to produce meaningful, unbiased results. We cannot simply regenerate outputs until one method yields a satisfactory x_0.

Therefore, all algorithms start the generation using the same random distribution. The result of the algorithm yield is solely based on the diffusion process rather than good base distribution (some quality is better because it started with a better random distribution). This approach helps to mitigate biases and provides a more equitable comparison framework.

Furthermore, our method's reliance on the temporal dependency inherent in Diffwave raises questions about its applicability beyond music synthesis or even outside of this specific architectural model. Although insusceptible to the human ears, our reconstruct x^{known} is not perfect. It is unsure that such differences would be more noticeable in other problem domains, therefore, The potential for Interpolation Guidance is uncertain. We leave this area open for future exploration and encourage readers to investigate these possibilities further.

6. CONCLUSION

We presented a pseudo auto-regressive inferencing technique for the diffusion model. In particular, we proposed a novel algorithm that performs audio extension at a fraction of the time compared to that of Repaint [6] while still yielding comparable if not superior results. Because we utilize the special architecture of Diffwave, our methods produced a favorable result in our empirical studies.

7. REFERENCES

[1] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *CoRR*, vol. abs/2006.11239, 2020. [Online]. Available: https://arxiv.org/abs/2006.11239

[2] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[3] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022.

[4] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[5] M. Müller, *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. Springer International Publishing, 2015. [Online]. Available: https://books.google.com/books?id=HCI_CgAAQBAJ

[6] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. V. Gool, "Repaint: Inpainting using denoising diffusion probabilistic models," 2022.

[7] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," 2016.

[8] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," 2018.

[9] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," 2020.

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

[11] 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.

[12] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," 2023.

[13] A. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," 2021.

[14] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," 2022.

[15] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," 2021.

[16] S. gil Lee, H. Kim, C. Shin, X. Tan, C. Liu, Q. Meng, T. Qin, W. Chen, S. Yoon, and T.-Y. Liu, "Priorgrad: Improving conditional denoising diffusion models with data-dependent adaptive prior," 2022.

[17] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "Wavegrad: Estimating gradients for waveform generation," 2020.

[18] N. Takahashi, M. Kumar, Singh, and Y. Mitsufuji, "Hierarchical diffusion models for singing voice neural vocoder," 2022.

[19] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, "Cascaded diffusion models for high fidelity image generation," 2021.

[20] R. S. Roman, Y. Adi, A. Deleforge, R. Serizel, G. Synnaeve, and A. Défossez, "From discrete tokens to high-fidelity audio using multi-band diffusion," 2023.

[21] K. Rasul, C. Seward, I. Schuster, and R. Vollgraf, "Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting," 2021.

[22] A. Graves, "Generating sequences with recurrent neural networks," 2014.

[23] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf

[24] R. Yang, P. Srivastava, and S. Mandt, "Diffusion probabilistic modeling for video generation," 2022.

[25] C. Luo, "Understanding diffusion models: A unified perspective," 2022.

[26] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," 2021.

[27] F. J. Aragón-Artacho and D. Torregrosa-Belén, "A direct proof of convergence of davis-yin splitting algorithm allowing larger stepsizes," 2022.