Audio Reactive AI Generated DJ Visuals

Matt Reed
Department of Computer Science
Stanford University
Stanford, CA 94305
mattreed@stanford.edu

Adam Zhao
Department of Computer Science
Stanford University
Stanford, CA 94305
adamzhao@stanford.edu

Hannah Kim
Department of Computer Science
Stanford University
Stanford, CA 94305
hkim24@stanford.edu

Abstract

In the world of DJ performances, the integration of graphics and audio is critical to creating an immersive audience experience. Traditional visual methods rely on static, pre-made imagery, which can stifle the fluidity and adaptability of performances. Using the Real-valued Non-Volume Preserving (REAL-NVP) model, we developed a novel visual generator that responds to auditory inputs in real-time. This discovery enables the smooth flow of visual displays that adapt to the music's beats and rhythms, finding an ideal balance between visual quality and computational efficiency. Our paper outlines the mythologies and algorithms that comprise our system, as well as in-depth analyses of performance metrics and user-centered enhancements. By outperforming baseline VAE models in generation versatility, our work provides DJs with an innovative tool to augment their performances and opens up new avenues for artistic expression in live music events.

1 Introduction

For DJ performances, the synergy between music and visuals plays a significant role in enhancing the overall experience. While the combination of visuals with music can indeed elevate a DJ's performance, the process of creating these visuals has traditionally been resource-intensive and inflexible. Typically, DJs have had to synchronize their sets with premade visuals, which limits their creative freedom and adaptability during performances.

The motivation behind this project stems from a recognized limitation in the current state of visuals used in DJ performances. In the existing landscape, these visuals are typically pre-made and do not possess the capability to react in real-time to the music being played by the DJ. This lack of responsiveness presents a challenge, as it necessitates DJs to plan and synchronize their sets with visuals that remain static throughout the performance.

Another motivation for this project is rooted in the observation of the numerous applications and improvements seen in Generative Adversarial Networks (GANs) across various domains. GANs have demonstrated their potential to generate realistic and dynamic content; however, despite the strides made in GAN technology, the field of DJ performances remains largely untapped when it comes to real-time, adaptive visual generation. This untapped potential represents an exciting opportunity. By integrating GANs and other network architectures into the generation of visuals that dynamically respond to the DJ's music, we hope to revolutionize the way DJs and visual artists collaborate.

2 Related Work

Our project draws insights from various related works in the fields of text-to-video generation, audio-to-text algorithms, and latent space interpolation. While some of our work aligns with existing research, a significant portion of our work will be innovative and does not have much prior groundwork laid out.

The paper "CogVideo: Large-scale Pretraining for Text-to-Video Generation via Transformer"..[1] by Wenyi Hong et al. explores text-to-video generation using transformer-based models. It specifically investigates the process of pretraining on a large-scale dataset to facilitate the translation of textual descriptions into coherent video sequences. The paper provides insights into the application of transformers in bridging the gap between text and visual content generation.

In the paper "Attention-Based Models for Speech Recognition" [2] by Jan Chorowski et al., attention-based models are studied in the context of speech recognition. The research aims to increase the accuracy and efficiency of speech recognition systems through the use of attention mechanisms. The findings offer valuable techniques for transcribing spoken language into text, which can be relevant when considering audio inputs, such as music, in our project.

The paper "Density Estimation using Real NVP" [3] introduces the concept of density estimation using Real NVP models. The paper provides insights into modeling complex data distributions and reversible data transformations, which is related to our project's aim to generate visuals in response to DJ performances, particularly in the context of latent space interpolation.

3 Approach

3.1 Proposed methods

We propose an innovative solution leveraging the capabilities of REAL-NVP, which stands for Real-valued Non-Volume Preserving.

The key idea behind REAL-NVP is to model the transformation between two probability distributions. Given an input data point, REAL-NVP defines a series of invertible functions that transform it into an output data point. Importantly, these transformations are designed to be reversible, meaning you can go back from the output data point to the input data point.

REAL-NVP models are part of a broader family of normalizing flow models, which aim to transform a simple probability distribution (e.g., a Gaussian distribution) into a more complex one to capture the underlying data distribution.

We are utilizing the CIFAR-10 dataset for training and evaluating our REAL-NVP model. CIFAR-10, or the "Canadian Institute for Advanced Research - 10", comprises a collection of 60,000 color images, each measuring 32x32 pixels. These images are divided into ten different classes, with each class representing a distinct object category such as airplanes, automobiles, cats, dogs, and more. Each class contains 6,000 labeled images.

The goal of this project is to take slices of audio as input and output a real-time smooth video that live reacts to the audio. To accomplish this, we divide the task in two. Our first model maps audio to spots in the latent space, and our second model maps those latent space spots to real color images.

The primary expected result is the generation of a sequence of images using the REAL-NVP model. These images will be both visually appealing and reflect the dynamics of the music being played by DJs. The generated images will transition smoothly from one to another as the music evolves, creating a continuous and engaging experience.

By training latent space mapping images, we hope that we can translate audio to that latent space, make a transformation on the latent representation, and then go from that revised latent representation to a new image. At no point does our method sample from the REAL-NVP model, rather we use it for its lightweight, yet sophisticated understanding of how images look in general. This ability of flow models allows for smooth image interpolation through a path that is more visually appealing than a simple pixel-wise interpolation.

Evaluating the flow model itself can be done using the loss on the test set, but the evaluation of our project as a whole will be done by a vibe check.

3.2 Technical Approach

Given an observed data variable $x \in X$, a simple prior probability distribution p_Z on a latent variable $z \in Z$, and a bijection $f : X \to Z$ (with $g = f^{-1}$), the change of variable formula defines a model distribution on X by

$$p_X(x) = p_Z(f(x)) \left| \det \left(\frac{\partial f(x)}{\partial x^T} \right) \right|$$

$$\log(p_X(x)) = \log(p_Z(f(x))) + \log \left(\left| \det \left(\frac{\partial f(x)}{\partial x^T} \right) \right| \right)$$

Exact samples from the resulting distribution can be generated by using the inverse transform sampling rule. A sample $z \sim p_Z$ is drawn in the latent space, and its inverse image $x = f^{-1}(z) = g(z)$ generates a sample in the original space. Computing the density on a point x is accomplished by computing the density of its image $f(x)$ and multiplying by the associated Jacobian determinant $\left| \det \left(\frac{\partial f(x)}{\partial x^T} \right) \right|$. Exact and efficient inference enables the accurate and fast evaluation of the model.

Algorithm 1 AIDMA - edm visuals

Input: Set S of images, Model M, Song File F

Input S into M
$r \leftarrow Latent\ Representation\ of\ Random\ s \in S$
while F is playing **do**
    Sample audio signal a
    Get latent space change d using a
    $r \leftarrow Interpolate(r, d)$
    $I \leftarrow M(r)$
    Display I
end while

Figure 1: Pseudocode

Our neural network will be using the largely same structure as in Real-NVP. Given a D dimensional vector x and some $d < D$, the transformation at each layer is defined as

$$y_{1:d} = x_{1:d} \tag{1}$$

$$y_{d+1:D} = x_{d+1:D} \odot \exp(s(x_{1:d})) + t(x_{1:d}), \tag{2}$$

where s and t stand for scale and translation, and are functions from $\mathbb{R}^d$ to $\mathbb{R}^{D-d}$, and $\odot$ is the Hadamard product or element-wise product (see Figure 2(a)). The Jacobian of this model is

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}^T} = \begin{bmatrix} \mathbf{I}_d & \mathbf{0} \\ \frac{\partial y_{d+1:D}}{\partial x_{1:d}^T} & \operatorname{diag}\left(\exp\left(s(\mathbf{x}_{1:d})\right)\right) \end{bmatrix}, \tag{3}$$

which is triangular and thus offers efficient inversion and determinant calculation. We can calculate it's determinant as $\exp\left(\sum_j s(x_{1:d})_j\right)$, and since calculating this determinant does not involve calculating the determinant of the scale or translation functions, we can make them as complicated as we want and elect to use the same deep convolutional neural networks for the task as in the original REAL-NVP codebase.

Additionally, note that the forward and inverse operations are just as efficient, meaning that sampling and inference are just as efficient as well. This is crucial in a live audio setting, where latency kills performance.

$$\begin{cases} y_{1:d} = x_{1:d} \\ y_{d+1:D} = x_{d+1:D} \odot \exp(s(x_{1:d})) + t(x_{1:d}) \end{cases} \tag{4}$$

$$\iff$$

$$\begin{cases} x_{1:d} = y_{1:d} \\ x_{d+1:D} = (y_{d+1:D} - t(y_{1:d})) \odot \exp(-s(y_{1:d})) \end{cases} \tag{5}$$

In order to handle masking, we're using a binary mask b, and using the functional form for y,

$$y = b \odot x + (1 - b) \odot (x \odot \exp(s(b \odot x)) + t(b \odot x)) . \tag{6}$$

We use two partitionings that exploit the local correlation structure of images: spatial checkerboard patterns, and channel-wise masking . The spatial checkerboard pattern mask has value 1 where the sum of spatial coordinates is odd, and 0 otherwise. The channel-wise mask b is 1 for the first half of the channel dimensions and 0 for the second half. An illustration that was presented in the original paper is presented in Figure 2.

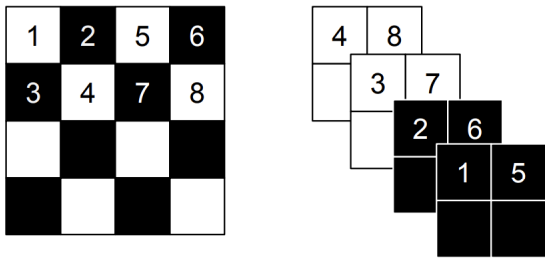

Figure 2: Masking Scheme [3]

For the models presented here, both $s(\cdot)$ and $t(\cdot)$ are rectified convolutional networks.

In order to increase the efficiency of forward propogation, Real NVP utilizes a trick which factorizes out half of the dimensions at regular intervals and models tham as Gaussians. [3] In addition, batch and weight normalization are used to improve the propogation of the training signal.

Our full model must be able to map from the audio space to the image space, and we have already seen how the second half of this algorithm operates: the REAL-NVP model allows for a useful reconstruction of realistic images from a latent space. The first half, from audio to latent space, is the remaining piece. While a deep neural network approach would have been preferred, for the scope of this project, we only had the time and resources to attempt a few simple function mappings. The idea that is central to all of these is that the user of this program is able to input a set of images that they like which can be passed through the flow network backwards to get their corresponding latent space representations. These latent space points serve as the map that the following methods use to sail the ocean of the latent space:

- Weighted Average: The FFT of the input audio is split into n sections then passed through a softmax with temperature scaling. Using these values, a weighted average of the user's n image latent space representations is computed to find the resulting latent space position. This didn't work very well in practice because, with no temperature scaling, the weighted average was too consistent, but with high-temperature scaling, it was too random. Either way, this defeats the whole purpose of the neural network which was smooth and consistent interpolation.
- Magellan Method: Named after his circumnavigation of the globe, this method utilizes a simple rotation around the input images, interpolating between adjacent input images' latent vectors in a circle. The speed of interpolation is based on the RMS of the audio chunk for that frame. This method appears very smooth and pleasant so is very good for chill music.
- Random Exploration: starting from a selected input image, this method morphs an image with a randomly selected other input image. The amount of morph is proportional to

the audio's RMS. This works well for EDM songs that should be more flashy or need a nice-looking image that gets distorted each time the kick hits

This leaves us with the final model shown in Figure 3:

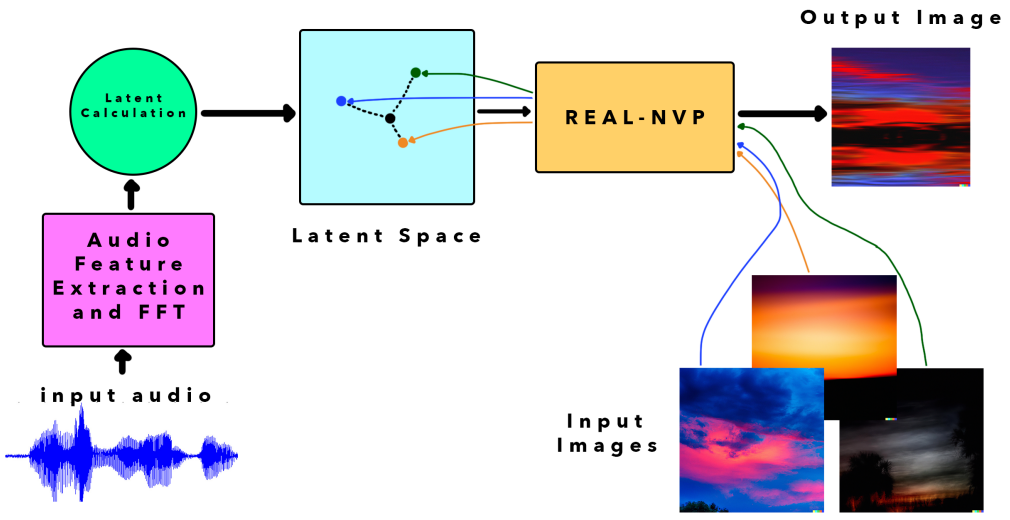

Figure 3: Model Pipeline

3.3 Evaluation/Experimental procedures

Our baseline model was implemented using a VAE, which performed decently, but was limited to interpolating between images in whatever dataset that it was trained on.

We trained two iterations of the Real-NVP model, one with the full number of parameters and one with greatly reduced parameters for faster sampling. The full model could only be feasibly trained for one epoch, whereas the reduced model trained for 10 epochs. We trained on the CIFAR-10 dataset, as it gives us a large set of images for the model to familiarize itself on.

Due to both the subjective nature of evaluating visuals as well as the lack of useful testing datasets and methods, we used human evaluation to iterate on and improve our models. Explain your baseline again, possibly in more detail Explain any experimental procedures (training details, test data, etc.)

3.4 Results

During the training of our flow model, two main metrics were used for evaluation: negative log likelihood loss, and the bits per dimension achieved, both standard in image-based tasks. Our results can be seen in Figure 5.

As stated before, the VAE baseline is limited to images in the dataset it is trained on, whereas our model will work with any set of images we give it. An example can be seen in Figure 4

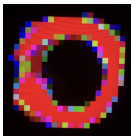

Figure 4: VAE image

Interpolation examples generated from the two models, using images we provided, can be seen in Figure 6. It should be noted that although our numerical results are not stellar, that ends being fine

	Lighter Model	Larger Model
Bits Per Dimension (BPD)	8.44	30.4
Test Loss	1.8e+4	6.46e+4 (after one epoch ~ 4 hrs)

Figure 5: Comparison of Lighter and Larger Model for BPS / Loss

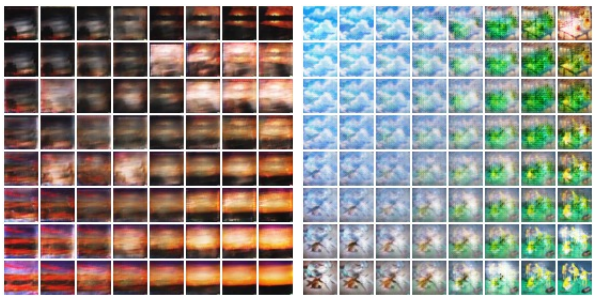

Figure 6: Comparison of Lighter (left) and Larger (right) Model for Interpolation Quality

since we only need the flow model to learn how to go from latent space to an image. Our images, especially those from the larger model, blend seamlessly and are smooth.

4 Analysis

4.1 Frame Rate and Visual Quality Assessment

For evaluation, we first ran a comparative analysis between two variants of our model: a 'lighter' model, which is less computationally demanding, and a 'larger' model, which is more resource-intensive but potentially capable of generating higher-quality images. This comparison allowed us to explore the trade-offs inherent in the design and implementation of generative visual systems. Specifically, we measured the bits per dimension—a metric that gives insight into the complexity and detail of the visual data generated—and the test loss, which indicates the model's accuracy in generating images that are faithful to the desired output. The outcomes of these measurements, which are crucial in deciding the optimal model for different performance contexts, are illustrated in Figure 5.

Next, we assessed the quality of image interpolation for our two models. When evaluating the lighter and larger mode's interpolation between two static images, it became apparent that the larger model was more adept at creating intermediate frames that were visually coherent and aesthetically pleasing. The larger model managed to produce a sequence of images where each frame logically progressed from the starting image to the ending image, maintaining the integrity of visual elements throughout the transition.

On the other hand, the lighter model, while optimized for less intensive computation, did not perform as well in this aspect. Its attempt at interpolation resulted in a series of frames that seemed to lack a coherent transition, with images appearing abruptly or randomly rather than blending smoothly from one to the next. This suggests that the larger model's advanced capabilities significantly outperformed the lighter model in terms of creating a seamless visual bridge between two given images.

4.2 Model Comparison

The baseline Variational Autoencoder (VAE) model, which was capable of generating only the trained MNIST images, served as the starting point for our advancements. While the VAE model showed

proficiency in recreating these specific types of images, its scope was limited to the dataset it was trained on, providing little flexibility beyond that.

Our flow model, in contrast, demonstrated an ability to generate a diverse array of images, moving beyond the constraints of a single dataset. This capacity for generating arbitrary images marks a substantial leap in versatility, opening up a broader spectrum of creative possibilities.

4.3 Audience Engagement and Feedback Collection

Feedback was actively collected from a sample audience, focusing on the visual appeal and the synchronization of the generated images with the music. This feedback was crucial for refining our interpolation techniques and ensuring that the end product resonated well with the intended users. The insights gained from audience feedback were crucial in guiding further refinements of the model, ensuring that the visuals not only met technical standards but also aligned with audience expectations for a compelling audio-visual experience.

5 Conclusion

Our study successfully demonstrated a novel technique for producing audio-responsive visuals for DJ sets. Using REAL-NVP models, we were able to dynamically sync visual elements with the changing beats and rhythms of DJ music. This synchronization is both a technical and artistic accomplishment, considerably improving the entire experience of a DJ performance.

Our work is distinguished by its real-time responsiveness and adaptation to the music being performed. This is an enormous improvement over traditional methods, which require DJs to rely on pre-made images, restricting their creative flexibility. Not only does our technique free DJs from these limits, but it also offers new opportunities for creative expression in live performances. This work demonstrates the potential of combining powerful computational models with artistic domains.

Looking ahead, there are a few directions to explore. An important limitation in our current system is the high pixelation of images, which is primarily due to constraints in computational power. In the future, we hope to work on improving our algorithms' computational efficiency while employing more powerful hardware. It is also worthwhile to investigate image upscaling and enhancing methods. This would allow us to create higher-resolution, less pixelated, and more visually appealing graphics. Improving picture quality is critical, especially given the high-definition visual demands in modern DJ performances. By fixing this issue, we can dramatically improve the visual experience, making it more immersive and engaging to audiences. Furthermore, while our present system performs well in terms of real-time performance, there is always potential for improvement in terms of lowering latency and improving the fluidity of visual transitions. Another possible future project is to use machine learning techniques to design visuals that represent the DJ's particular style or cater to audience preferences, therefore increasing the interactive and personalized component of the DJing experience.

Our code can be found at this repo:
https://github.com/matttreed/Real-EDM

References

[1] Wenyi Hong et al. *CogVideo: Large-scale Pretraining for Text-to-Video Generation via Transformers*. 2022. arXiv: 2205.15868 [cs.CV].

[2] Jan Chorowski et al. *Attention-Based Models for Speech Recognition*. 2015. arXiv: 1506.07503 [cs.CL].

[3] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. *Density estimation using Real NVP*. 2017. arXiv: 1605.08803 [cs.LG].