Technische Universität Berlin

Quality & Usability Lab, Faculty IV

Quality and Usability Lab: Generative AI and Art

Timbre Transfer and Video Generation

Final Report Advanced Study Projects 2024

Name	Matr.-Nr.	E-Mail
Rui Zhao	490839	rui.zhao@campus.tu-berlin.de
Deniz Mert Tecimer	487675	tecimer@campus.tu-berlin.de

Supervisor(s): Vera Schmitt, Premtim Sahitaj

March 17, 2024

1 Introduction

In this article, you will explore the current state of development in generative models, primarily divided into two parts: audio synthesis and music generation, as well as video generation (with a focus on the current state-of-the-art model, OpenAI Sora Brooks et al. (2024)). The music generation task focuses on the timbre transfer use case followed by a video generation task using the generated output.

2 Audio Synthesis and Music Generation

2.1 Related Work

In recent years, the adoption of generative AI, exemplified by products like ChatGPT (OpenAI, 2023), has been notable for its success. While electronic music synthesizers have a history dating back to the 1960s and 1970s (Pekonen and Välimäki, 2011), endeavors in human speech synthesis trace back approximately two centuries (Gold, 1978).

Goodfellow et al. (2014) introduced GANs(Generative Adversarial Networks) that generate new samples by learning the probability distributions from training data samples. It consists of two main components, a generator and a discriminator competing with each other. The generator tries to trick the discriminator by generating realistic samples and the discriminator tries to differentiate between real and fake(generated) samples. Donahue et al. (2018) compare the performance of different methods of generating audio samples using GANs. Their model WaveGAN(Donahue et al., 2018), based on PixelCNN (van den Oord et al., 2016), uses the input as raw audio, whereas their other model SpecGAN(Donahue et al., 2018) utilizes spectrograms, image-like representations of audio in the time-frequency domain, and transforms the audio-to-audio generation task into image-to-image generation. Even though they are motivated by image synthesis methods such as DCGAN(Radford et al., 2016) for both models, they denote that the images are less likely to contain periodicity than audio. This may result in the discriminator to reject images that contain periodicity. Therefore, they propose phase shuffling by applying phase perturbation at each layer of the discriminator. Due to the phase information loss in spectrogram generation, they estimate it through the Griffin-Lim algorithm(Griffin and Lim, 1984) to reconstruct audio. Evaluation of the results shows that the WaveNet model(Donahue et al., 2018) performs better than the SpecGAN(Donahue et al., 2018) due to different attributes persisting in audio and image representations. Its success and potential led many other researchers to adopt WaveNet(Donahue et al., 2018) for various specific tasks or as a baseline to compare their proposed methods in audio generation. Zhang et al. (2022) published their work on successfully generating dolphin vocal sounds through WaveGAN(Donahue et al., 2018), which can be used as a bionic sonar without creating noise pollution in the marine environment. They denote that the recording environment is quiet, and therefore, the data is relatively clean. However, it contains low-frequency water noise due to the swimming movements of other dolphins, which is cleared through finite impulse response(FIR) filters. ORCA-WHISPER (Bergler et al., 2022) is another animal vocal sound generation model developed using the TiFGAN-architecture(Marafioti et al., 2019), an improved version of WaveGAN/SpecGAN(Donahue et al., 2018). Due to the data shortage in the field, they aim to provide a data augmentation model by generating new spectrogram samples based on orca call type. Bergler et al. (2022) suggest a revised iteration of TiFGAN, where two additional layers are integrated into both the generator and discriminator. This extension aims to enhance the time and frequency resolution within the discriminator, potentially improving its performance. To tackle the increased model complexity, they reduce the original feature map dimensions. Even though they work with a few data samples for six categories of orca call types, they do not suggest merging all the data to obtain a single model. On the contrary, Bergler et al. (2022) reports that this results in intermediate samples which does not belong to any category. Kim et al. (2023) proposes a more generic model for data augmentation in animal sound classification. They underline that each animal generation task often requires its own model, which causes

limitations on storage and computation resources. The waveform presents the signal intensity over time, whereas the spectrogram captures periodicity through frequency-time features. The existing GAN-based methods use either waveform or spectrogram as their input, which only by itself may not capture all necessary features persisting in complex animal sounds. GANs are also affected by the noise in the input data and may result in the loss of semantics in the output. Therefore, they offer a model that utilizes both input forms by using the WaveGAN(Donahue et al., 2018) and the SpecGAN(Donahue et al., 2018) models together. DualDiscWaveGAN(Kim et al., 2023) uses a single generator and two discriminators guided by class labels. The generator produces waveform samples containing features of the respective class and the discriminator of the WaveGAN(Donahue et al., 2018) tries to differentiate the real samples from the fake ones. Then, the generated and real samples are converted into spectrograms, which are then evaluated by the discriminator of the SpecGAN(Donahue et al., 2018).

Rumelhart et al. (1986) introduced autoencoders as a feature reduction/compression method which reconstructs its input. They consist of an encoder and a decoder component. The encoder maps the input into a fixed latent space, which is then used by the decoder to reconstruct the input. Mapping samples without any regularization may cause discontinuity in the latent space and randomly sampled points may not be decoded to a meaningful output as the similar inputs are not regularized to be close to each other (Ghosh et al., 2020). Variational Autoencoders(VAEs) regularize the probability distribution to match the normal distribution through sampling on the latent space, resulting in a continuous and smooth latent space. In such space, a random sample can be mapped to an interpretable output Bank et al. (2021). Another improvement is the vector quantized VAE(VQ-VAE)(van den Oord et al., 2018), which creates a discrete latent space instead of a continuous one. Even though random sampling for generation is not possible and the mapped samples(the latent) can only be chosen from the embedded training data, training it with supplementary new data points is applied for generation tasks. VQ-VAE encodes samples based on the nearest neighbor look-up to group similar sample embeddings (van den Oord et al., 2018). VQ-VAE 2(Razavi et al., 2019) extend this method by employing multiple hierarchical latent maps that encode high and low-level features and hierarchical decoders. Furthermore, the decoders can generate the output in different scales, containing a detailed or general structure. Guei et al. (2024) aims to enrich the data available for ecological projects by generating new samples through VQ-VAE 2 architecture(Razavi et al., 2019). As an improvement, they introduce some augmentation methods such as noise perturbation and interpolation.

Wang et al. (2024) compare different rule- and AI-based methods in their work and explore the current state in the intelligent music generation domain. They also examine the literature proposing different representations of audio, i.e. waveform, spectrograms, and music fragments, i.e. MIDI events. They classify these works based on their purpose in four categories: melody generation, arrangement generation, audio generation, and style transfer. The motivation for music generation systems relies mainly on cost reduction in commercial music production, personalization possibilities, assistance for producers, and broad application areas such as music education, etc.

Qian et al. (2019) introduced the AutoVC model using autoencoder architecture consisting of two encoders, the content encoder, the speaker encoder, and a decoder. This architecture decouples the content from the style, allows many-to-many style transfer, and aims to produce the given content with the given speaker style embedding. Inspired by Qian et al. (2019), Wu et al. (2023) decouple the style from the content. The style encoder conditions both the content encoder and the decoder. During training, they are both conditioned on the same style, which allows timbre transfer when the style code of another learned instrument is fed to the decoder. Swarnamalya et al. (2023) follow a similar decoupling approach to the artist's style transfer using VQ-VAE with a diffusion model(Ho et al., 2020), which adds noise to the input gradually and learns to reconstruct the input from the noise. They denote that the model is inefficient and can not capture the tune of the content audio perfectly, which can be fixed via a better decoder or by enriching the variability in the dataset by adding more artists. Besides audio-to-audio generation, Schneider et al. (2023) introduces the diffusion magnitude autoencoder for text-to-audio generation utilizing a specialized version of U-Net(Ronneberger et al., 2015) for noise removal. In addition,

they provide their Text2Music dataset, which contains 50K text-music pairs corresponding to 2500 hours in total. Engel et al. (2020) employ encoders extracting the fundamental frequency, amplitude, and style of the input audio. The decoder generates the harmonics conditioned on the extracted fundamental frequency and the transfer function of the FIR filter, which are then used for generating the output through an additive synthesizer. Tatar et al. (2021) generate music by interpolating between two latent representations on the latent space learned through VAE trained on spectrogram data. Caillon and Esling (2021) propose a two-stage training approach utilizing both a VAE and a GAN for style transfer. In the first phase, the model learns the representations via VAE training and the second phase fine-tunes the decoder through a GAN-based training with a discriminator. In addition to RAVECaillon and Esling (2021), Valenzuela (2023) provides a method for generating latent codes for RAVE via denoising diffusion probabilistic modelsSchneider (2023), and allow generating new samples through latent space exploration methods such as interpolation. Wang et al. (2024) underlines that current systems can generate high-quality short-length polyphonic music, yet they struggle to effectively handle longer sequences and are prone to introduce repetition or chaotic features.

2.2 Methodology used in Timbre Transfer

Regarding the evaluation of Wang et al. (2024), the timbre(style) transfer approach using the RAVE modelCaillon and Esling (2021) was employed to hinder the constraints of generating long sequences.

Caillon and Esling (2021) follow a two-stage training approach with waveform representation of audio using a VAE and a GAN for representation learning and fine-tuning the decoder. In the first phase, representation learning, they train the encoder-decoder architecture. Once the loss converges, the second phase, the adversarial training, starts. The encoder weights are frozen and the decoder is fine-tuned through adversarial training as in GAN architecture to yield realistic generations. This mitigates the model to learn perceptually irrelevant features due to different phase variations which may yield different waveforms without perceptual variance. They utilize multi-band decomposition as proposed by Engel et al. (2020) and multiband spectral distance to calculate the similarity between the input and generated signals. They also provide post-training analysis of the latent space which allows a trade-off between the reconstruction fidelity and the latent space compactness. The fidelity value, together with the dataset determines the latent space dimensionality and allows the previously mentioned trade-off. As dimensionality increases, the model can also map lower-level features and generate higher-quality output, which also causes increased complexity. Besides their extensive research, Caillon and Esling (2021) also provide their implementation and support through a Discord channel, which is maintained and supervised by the team. The project contains a Colab notebook with default settings for training the network (Horta, 2023). The model was trained with a sampling rate of 48kHz. The first phase was set to 1000000 iterations as recommended in the Discord channel followed by 362099 phase two iterations heuristically. The model was trained around 25-30 hours. Even though it is not mentioned in the paper (Caillon and Esling, 2021), the implementation supports Wasserstein regularization (Petzka et al., 2018) which can be used in GANs and adds the "work" to transform prior distribution to posterior distribution (the Wasserstein distance) to the loss. Besides the mentioned variations, the default configurations were used during training.

For the experiment, the dataset was created by scraping videos containing animal sounds from YouTube[1]. The parts containing background noise, i.e. sounds of other animals, audible human speech, wind, etc. were manually removed from the data using Wavacity(Hilss, 2023), an open-source audio editor. The overall dataset corresponds to around 2.5 hours of different animal sounds, which still may contain some noise due to human error. The dataset is imbalanced and the duration of the training data for each animal can be seen in Table 1. The dataset is accessible through the shared link. [2]

[1] `https://www.youtube.com/`

[2] `https://drive.google.com/drive/folders/1fos1LitKtLA3RBuX9riyPorjnv-2sCKa?usp=sharing`

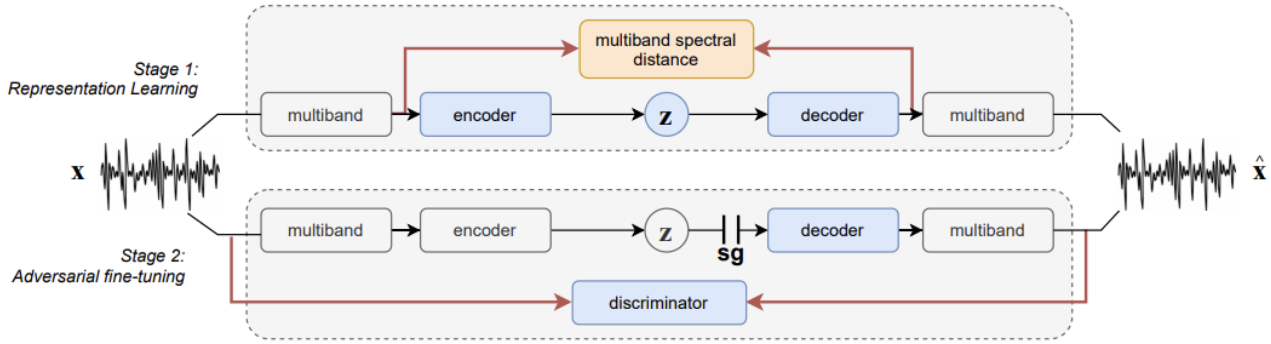

Figure 1: The overall RAVE architecture with two-stage training. Blue blocks are optimized and grey blocks are frozen or fixed operations. Caillon and Esling (2021)

Animal	Cat	Lion	Whale	Dog	Elephant	Bird	Monkey	Horse	Chicken	Leopard	Tiger	Donkey	Wolf
Duration (mm:ss)	17:01	09:57	16:34	14:33	12:26	22:39	14:43	17:16	20:39	00:07	07:55	05:00	05:23

Table 1: Animal sound durations used for training

The trained model is exported and converted into a format supported by Neutone(Neutone Inc., 2023) following Masuda (2023). Neutone provides a virtual studio technology(VST) plugin for DDSPEngel et al. (2020) and RAVE(Caillon and Esling, 2021) models to be used in a digital audio workstation(DAW) and allows access to their configurations. The input song[3] was processed as 148 beats per minute(BPM) with the model through the trained model in Neutone on Waveform 12Tracktion Inc. (2023). The input song is a techno song containing animal sounds in some parts, which allows evaluating the model performance both on the reconstruction of animal sounds and timbre transfer. The interface of Neutone can be seen in Figure 2, in which wet corresponds to the output amplitude and dry to input amplitude. In custom settings, A determines the chaos(the amplitude of noise added to the latent representation), B which dimension of the latent code to be modified, C the scale factor applied to the selected dimension, and D the offset value scaled with C and applied on the selected dimension through B. In our experiment, we do not introduce any noise or modification to the latent code during inference.

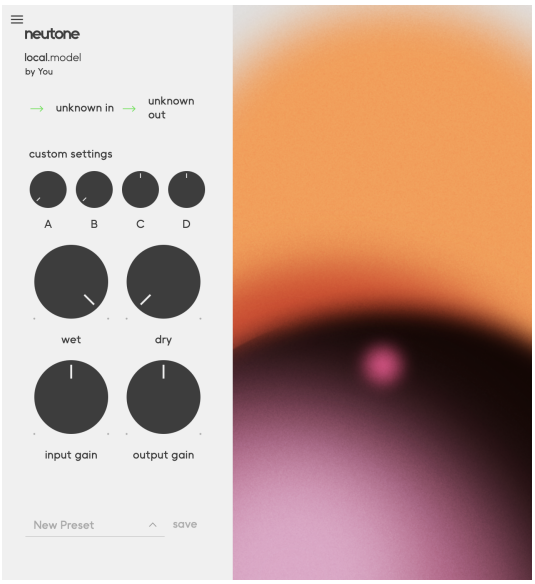

Figure 2: The user interface of Neutone(Neutone Inc., 2023)

[3] https://soundcloud.com/akr4tek/granjatek

2.3 Timbre Transfer Results and Analysis

Some parts of the training loss shown in Figure 3 are missing due to connection interruption between Colab and Google Drive. However, the overall trend is visible. After achieving successful convergence, the peak observed after the one-millionth iteration was attributable to the effects of adversarial training. Even though the loss increased, the output became more realistic as the discriminator's error increased in classifying real samples correctly throughout the adversarial training (see Figure 4).

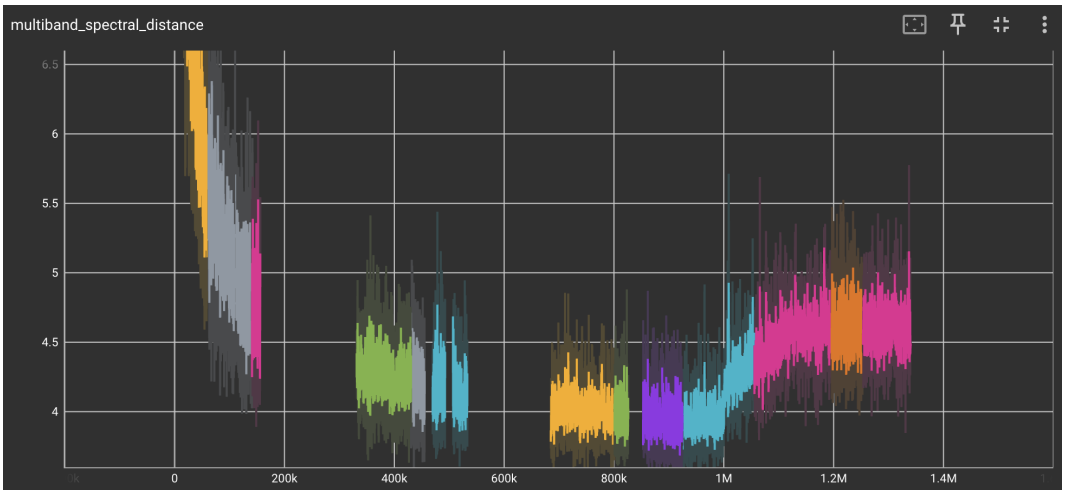

Figure 3: The training loss

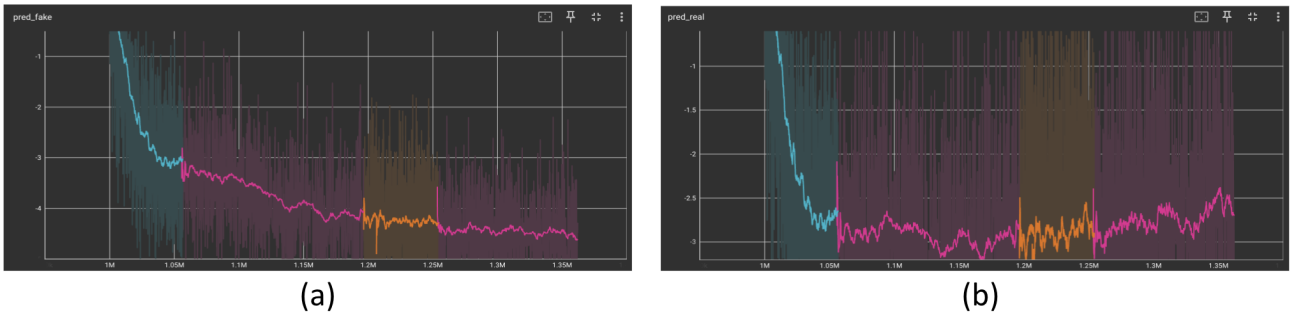

Figure 4: The performance of the discriminator on (a) fake samples (b) real samples

The generated song carries the tempo of the input song, yet loses some content-related features. The animal sounds in the song are relatively reconstructed better than recreating the input with the style of animal sounds. In some sections of the output, some animal sounds are audible. The labeled versions of the input and output songs are shared in Figure 5. There, the transfer of tempo and some harmonic features (visible as parallel line stacks) are visible. As shown in Figure 6a, the turkey sound was mapped as the chicken sound. This shows the model's generalization ability, noting that the dataset does not contain any turkey sound. The turkey sounds similar to the chicken and this model successfully mapped the similar instances together as in the example shown in Figure 6b. The cock sound in Figure 6b was reconstructed successfully with corresponding harmonics, similar to the cow sound shown at the right end of Figure 5. The dataset does not contain cow sounds, yet the model generated it using an intermediate representation, showing the generation variability of VAEs.

The produced song is longer than the input song due to unknown reasons which can be investigated further. Therefore, the parts shown in some parts of the figures (i.e. see Figure 6a) are not synchronized or hand-tuned for presentation purposes.

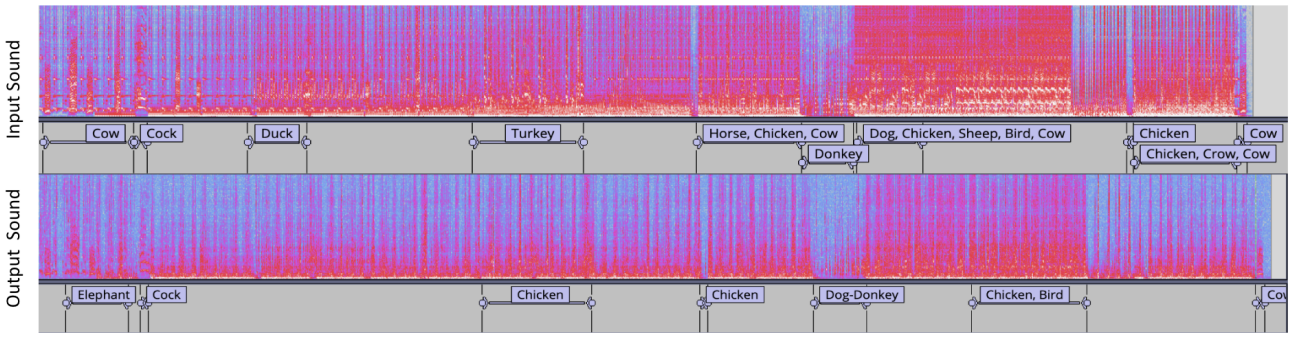

Figure 5: The overall result with audible animal labels

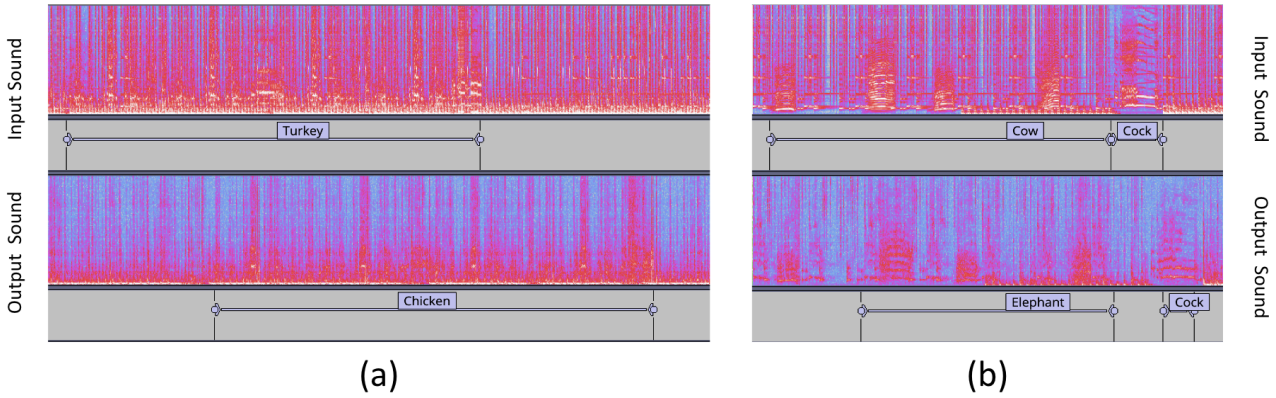

Figure 6: Comparison of animal mappings

3 Video Generation

Developed by OpenAI, Sora can be considered the current state-of-the-art (SOTA) model in the field of video generation. Brooks et al. (2024)

Due to the lack of technical details in Sora's technical report, we can better understand how Sora is designed by looking into other preceding research papers.

To comprehend Sora, some papers are insightful. On one hand, it is crucial to understand how Sora manages to accept videos of arbitrary resolutions for training and output videos of arbitrary resolutions and aspect ratios. On the other hand, it is essential to investigate how Sora processes high-resolution images, what the latent space mentioned in the technical report refers to, and why the latent space is introduced.

3.1 Variable Aspect Ratios and Resolutions

Dosovitskiy et al. (2021) This paper introduces the Vision Transformer (ViT), which divides an image into patches, with each patch being linearly projected into a token. Typically, the input image is resized to a fixed square aspect ratio and then partitioned into a fixed number of patches.

In language modelling, the limitation of fixed sequence lengths is typically bypassed through example packing. Tokens from multiple different examples are combined into a single sequence, which can significantly accelerate the training of language models Krell et al. (2022). Moreover, multiple examples are

packed into a single sequence to accommodate efficient training on variable-length inputs.

Inspired by example packing in natural language processing, Dehghani et al. (2023) treats images as sequences of patches (tokens). Their findings demonstrate that Vision Transformers Dosovitskiy et al. (2021) can benefit from the same paradigm, and they refer to this technique as Patch n' Pack. By employing this technique, ViTs can be trained on images at their native resolution, an approach they term NaViT (Native Resolution ViT).

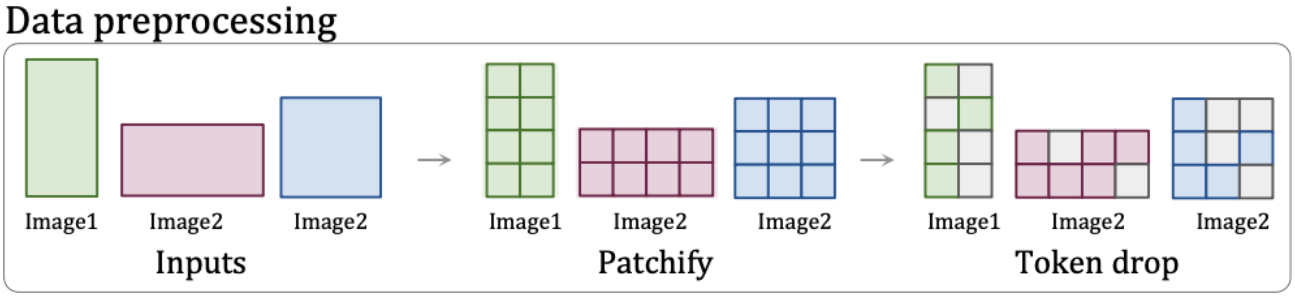

Figure 7: Patch n' Pack in Native Resolution ViT (NaViT) Dehghani et al. (2023)

Benefits of Mixed-Resolution Training: by preserving the original aspect ratios, the term "resolution" refers to the "effective resolution". In other words, it represents images with the same area as a square image of a given resolution. For instance, a square image with a resolution of 128 x 128 has the same area as images with resolutions of 170 x 96 or 64 x 256. Consequently, these non-standard aspect ratio images have the same inference cost as a standard 128 x 128 square image.

3.2 High-Resolution and Latent Space

Diffusion models (DMs) have achieved state-of-the-art synthesis results in image data and other domains, demonstrating excellent image generation capabilities. However, they face difficulties in directly processing high-resolution images. These diffusion models typically operate directly in the pixel space, requiring substantial computational resources to train powerful models and incurring high inference costs. Rombach et al. (2022) introduced latent diffusion models, an approach that significantly improves the training and sampling efficiency of denoising diffusion models without compromising their quality. They apply diffusion models in the latent space of powerful pre-trained autoencoders.

Training diffusion models in the latent space, rather than the pixel space, overcomes the limitations. Specifically, a prior network is used to encode high-resolution images into lower-resolution latent space representations. The diffusion model is then trained in this latent space to synthesize new samples, and finally, the latent space representations are transformed back to the original resolution through a decoding network.

Rombach et al. (2022), utilizing techniques e.g. latent space, breaks through the bottleneck of traditional diffusion models' difficulty in handling high-resolution images, paving a new path for high-quality image generation. The approach not only enhances quality but also improves computational efficiency.

3.3 Sora

The core work of Sora lies in finding a way to transform various types of visual data into a unified representation and perform large-scale training. Video data comes in various forms, including landscape and portrait orientations, as well as 4K high-definition videos and low-resolution 64x64 images. These

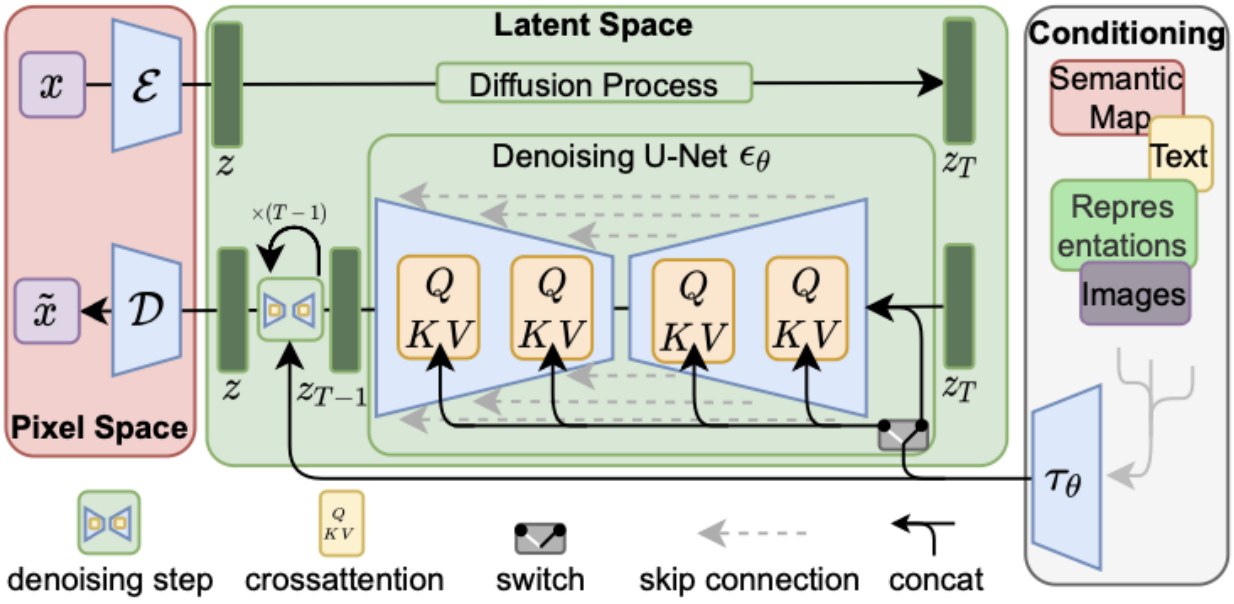

Figure 8: Latent Space in Latent Diffusion Models (LDMs) Rombach et al. (2022)

video data originate from diverse sources, have different resolutions, aspect ratios, and possess vastly different attributes.

Sora employs multiple techniques to gradually compress and extract the core content from videos.

The first step involves transforming the raw video data into low-dimensional latent space features. Converting the original images into latent space features before processing allows for the preservation of key feature information from the original images while significantly reducing the data and information volume.

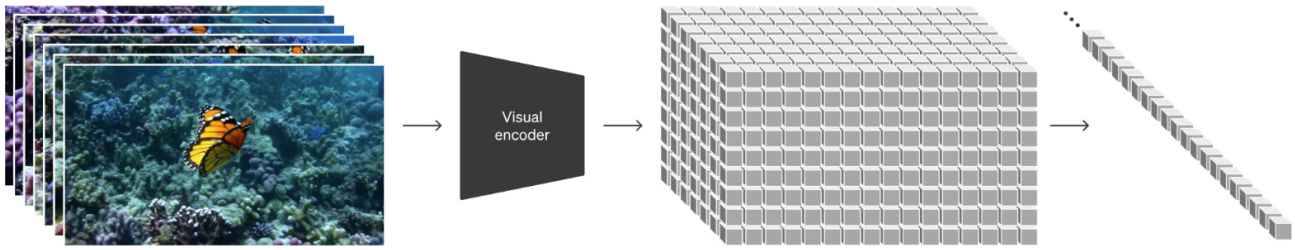

Figure 9: Compressing videos into a lower-dimensional latent space, and decomposing the representation into spacetime patches. Brooks et al. (2024)

The second step involves further decomposing the video features into spacetime patches.

tokens and patches: In large language models (LLMs), the basic unit is a "token." In the context of language models, a token typically represents the smallest text unit but is not limited to a single word. By breaking down text into tokens, the model can more flexibly handle the diversity of language. Whether it's simple words, complex phrases, or punctuation marks, the model can learn and understand them in a unified way. The main task of GPT is to predict the next token, while Sora's primary task is to predict the next patch.

In techniques like Vision Transformer (ViT) Dosovitskiy et al. (2021), static images are divided into multiple small, equal-area patches to enable more efficient processing and understanding of these images. These patches are converted into a series of vectors, each carrying not only the visual information of the patch but also its positional information within the original image. This approach allows the model to understand the content of each patch and its relative position within the entire image. In the original Vision Transformer (ViT), each patch must be of the same fixed size, and the original image must be square.

Sora likely employs an implementation similar to NaVit Dehghani et al. (2023), allowing it to handle inputs of different resolutions and aspect ratios during training by using a technique called Patch n' Pack when composing spacetime patches.

Patches alone are insufficient because videos consist of multiple images. Therefore, Sora uses spacetime patches. Spacetime patches are a technical concept specifically designed for video data. Videos are essentially a series of images (frames) that change over time. As a result, processing videos requires considering not only the spatial information of each frame but also the temporal changes of these images.

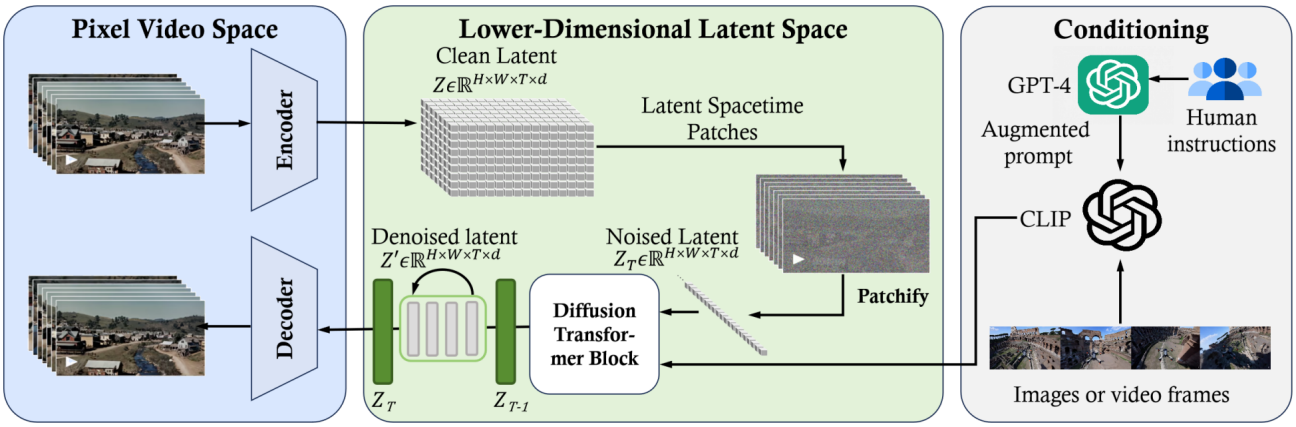

Figure 10: Sora's possible workflow based on Sora technical report HPC.AI (2024)

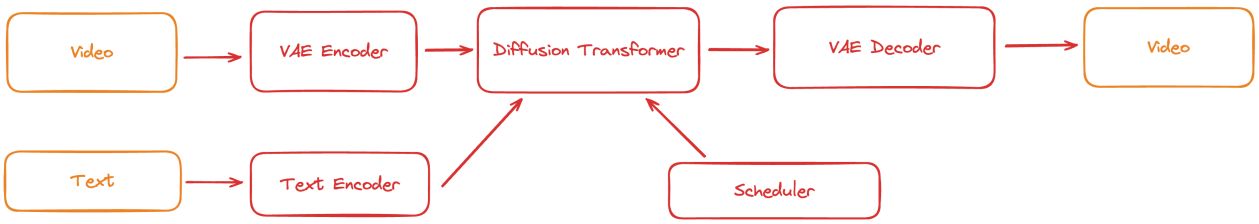

Figure 11: Sora's possible training pipeline HPC.AI (2024)

training process: Based on the Sora's technical report and related papers, Sora's training process can be roughly outlined as follows:

First, the raw video is converted into a sequence of frames. The sequence of frame images can be compressed into lower-dimensional latent space using OpenAI's own trained encoder/decoder model. Next, the latent space are further decomposed into spacetime patches. Finally, these spacetime patches are transformed into trainable vectors, enabling the training of the diffusion model.

Regarding the textual description of videos, OpenAI applies the recaptioning technique introduced in DALL·E 3. The technical report for DALL·E 3 explains this video description technique in detail. Essentially, they fine-tuned a model to clearly describe the content of a video according to a specific format.

This description content is more detailed than the original title information of these videos. By using such a formatted approach for systematic description, Sora can clearly understand the specific content of each video segment.

This textual description content can be matched and trained with the previous spacetime patches during the training stage. As a result, the model can understand and correspond the text descriptions with the video spacetime patches.

OpenAI also utilizes GPT to transform users' brief prompts into more detailed descriptions. In other words, the user's simple description is transformed into a detailed descriptive statement similar to those used during training. This enables Sora to generate high-quality videos that accurately follow the user's prompts.

The Sora's technical report also confirms that the Sora model performs better when trained on large-scale data. The larger the training data, the better the results.

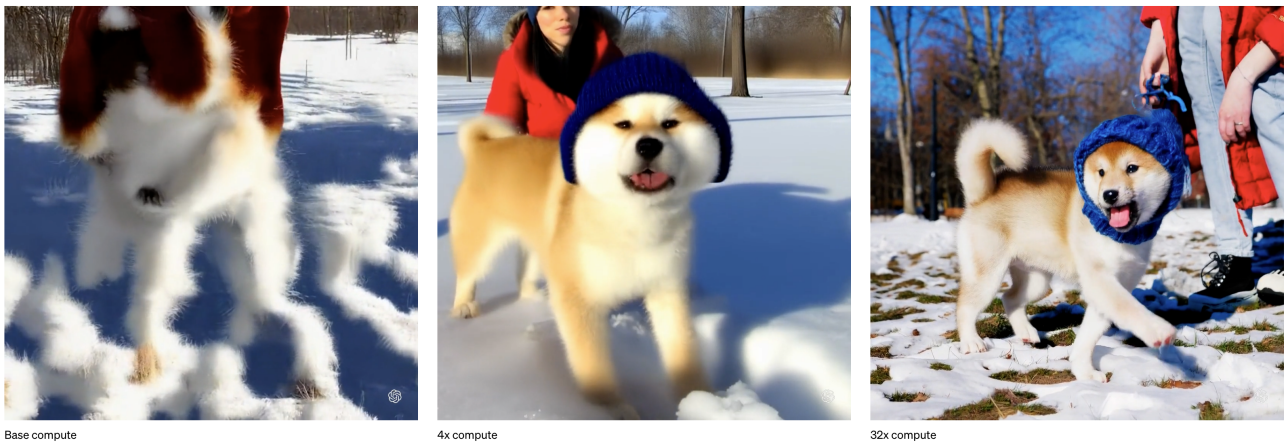

Figure 12: Sample quality improves markedly as training compute increases Brooks et al. (2024)

This characteristic perfectly aligns with OpenAI's advantage in massive computing power.

4 Discussion

4.1 Timbre Transfer

By analyzing the result, we can conclude that the model can successfully regenerate animal sounds but lacks musicality. The model capabilities can be further examined by applying different envelopes, filters, etc. to understand and improve the predictability of the output. As a future work, the model can also be run on a MIDI file to understand the effect of each note.

Further research can be conducted by focusing the dataset on a similar animal group and implementing new methods for transferring the musicality of the input. Engel et al. (2020) denotes replacing the deterministic autoencoder with a VAE as a possible improvement in his work. The RAVE model(Caillon and Esling, 2021) can be a potential candidate for improvement. However, RAVE is a data-hungry model, which requires more than 2-3 hours of data and a long and costly training duration.

4.2 Video Generation

Although primarily focused on video synthesis, the Sora model demonstrates interesting capabilities related to simulating the physical world:

- 3D Consistency: In videos generated by Sora, scene elements maintain a high degree of consistency in three-dimensional space as the viewpoint moves and rotates.
- Temporal Consistency: Objects and characters can persist over periods, maintaining their appearance even when occluded or temporarily leaving the frame.
- World Interaction: Accurate modeling of interactions, such as leaving marks on a canvas or consuming food.
- Simulate Digital Environments: The model can simulate environments like video games to a certain extent.

However, as a general world simulator, Sora still has some limitations:

- Physical Inaccuracies: The model struggles to accurately capture complex physical interactions, such as glass shattering.
- Limited World Knowledge: The model cannot precisely simulate some interactions due to insufficient understanding of the world.
- Hallucinations: Objects and scene elements sometimes appear or disappear unnaturally.

5 Conclusion

Through analysis and evaluation of the current state-of-the-art generative models, particularly in the domains of audio synthesis, music generation, and video generation, we have demonstrated the immense potential and versatility of generative AI in artistic creation.

In the realm of audio synthesis and music generation, we explored techniques such as GANs, autoencoders, and variational autoencoders, and how they can generate high-quality audio samples by learning data distributions. Specifically, we discussed how these models can enhance data diversity and richness through data augmentation and style transfer while preserving audio characteristics.

For video generation, the analysis of the Sora model emphasized the importance of combining large-scale data training with advanced video processing techniques. By training on video data with varying resolutions and aspect ratios, Sora showcased the ability to generate high-quality video content with temporal consistency and three-dimensional spatial consistency.

Generative AI holds immense promise for applications in artistic creation and content generation, particularly in simulating physical worlds and creating virtual environments. Despite challenges and limitations, such as physical inaccuracies and limited world knowledge, we believe that these issues can be addressed through continuous technological improvements and larger-scale data training.

Generative AI technologies are providing us with new tools and paving the way for exploring creativity and innovation. As we continue to push the boundaries of what is possible with generative models, we can anticipate a future where AI becomes an integral part of the artistic process, empowering creators to bring their visions to life in unprecedented ways.

References

Bank, D., Koenigstein, N., and Giryes, R. (2021). Autoencoders.

Bergler, C., Barnhill, A., Perrin, D., Schmitt, M., Maier, A. K., and Nöth, E. (2022). Orca-whisper: An automatic killer whale sound type generation toolkit using deep learning. In *INTERSPEECH*, pages 2413–2417.

Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., and Ramesh, A. (2024). Video generation models as world simulators.

Caillon, A. and Esling, P. (2021). Rave: A variational autoencoder for fast and high-quality neural audio synthesis. *arXiv preprint arXiv:2111.05011*.

Dehghani, M., Mustafa, B., Djolonga, J., Heek, J., Minderer, M., Caron, M., Steiner, A., Puigcerver, J., Geirhos, R., Alabdulmohsin, I., Oliver, A., Padlewski, P., Gritsenko, A., Lučić, M., and Houlsby, N. (2023). Patch n' pack: Navit, a vision transformer for any aspect ratio and resolution.

Donahue, C., McAuley, J., and Puckette, M. (2018). Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale.

Engel, J., Hantrakul, L., Gu, C., and Roberts, A. (2020). Ddsp: Differentiable digital signal processing.

Ghosh, P., Sajjadi, M. S. M., Vergari, A., Black, M., and Schölkopf, B. (2020). From variational to deterministic autoencoders.

Gold, B. (1978). Historical background. *The Journal of Criminal Law*, 42:109 – 121.

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

Griffin, D. and Lim, J. (1984). Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243.

Guei, A.-C., Christin, S., Lecomte, N., and Hervet, É. (2024). Ecogen: Bird sounds generation using deep learning. *Methods in Ecology and Evolution*, 15(1):69–79.

Hilss, A. (2023). Wavacity audio editor. `https://wavacity.com/`. Accessed 16.03.2024.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models.

Horta, M. (2023). Ravev2_training.ipynb - colaboratory. `https://colab.research.google.com/drive/1ih-gv1iHEZNuGhHPvCHrleLNXvooQMvI?usp=sharing`. Accessed 16.03.2024.

HPC.AI (2024). Open-sora: Sora replication solution with 46% cost reduction, sequence expansion to nearly a million. `https://hpc-ai.com/blog/open-sora`. Accessed: 2024-03-16.

Kim, E., Moon, J., Shim, J., and Hwang, E. (2023). Dualdiscwavegan-based data augmentation scheme for animal sound classification. *Sensors*, 23(4):2024.

Krell, M. M., Kosec, M., Perez, S. P., and Fitzgibbon, A. (2022). Efficient sequence packing without cross-contamination: Accelerating large language models without impacting performance.

Marafioti, A., Holighaus, N., Perraudin, N., and Majdak, P. (2019). Adversarial generation of time-frequency features with application in audio synthesis.

Masuda, N. (2023). Ravev2-neutone.ipynb - colaboratory. https://colab.research.google.com/drive/1qlN6xLvDYrLcAwS8yh2ecmNG_bEKlVI9?usp=sharing#scrollTo=YO1GOepQL9o5. Accessed 16.03.2024.

Neutone Inc. (2023). Neutone inc. https://neutone.ai/. Accessed 16.03.2024.

OpenAI (2023). Gpt-4 technical report.

Pekonen, J. and Välimäki, V. (2011). The brief history of virtual analog synthesis.

Petzka, H., Fischer, A., and Lukovnicov, D. (2018). On the regularization of wasserstein gans.

Qian, K., Zhang, Y., Chang, S., Yang, X., and Hasegawa-Johnson, M. (2019). Autovc: Zero-shot voice style transfer with only autoencoder loss.

Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks.

Razavi, A., van den Oord, A., and Vinyals, O. (2019). Generating diverse high-fidelity images with vq-vae-2.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986. *Biometrika*, 71:599–607.

Schneider, F. (2023). Archisound: Audio generation with diffusion.

Schneider, F., Kamal, O., Jin, Z., and Schölkopf, B. (2023). Moûsai: Text-to-music generation with long-context latent diffusion.

Swarnamalya, A., Pravena, B., Satyanarayana, V., Patel, R., and Kokila, P. (2023). Synthesizing music by artist's style transfer using vq-vae and diffusion model. In *International Conference on Computer Vision and Robotics*, pages 487–497. Springer.

Tatar, K., Bisig, D., and Pasquier, P. (2021). Latent timbre synthesis: Audio-based variational auto-encoders for music composition and sound design applications. *Neural Computing and Applications*, 33:67–84.

Tracktion Inc. (2023). Waveform free | digital audio workstation band editing software - tracktion. https://www.tracktion.com/products/waveform-free. Accessed 16.03.2024.

Valenzuela, M. H. (2023). Rave-latent diffusion. https://github.com/moiseshorta/RAVE-Latent-Diffusion. Accessed 16.03.2024.

van den Oord, A., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., and Kavukcuoglu, K. (2016). Conditional image generation with pixelcnn decoders.

van den Oord, A., Vinyals, O., and Kavukcuoglu, K. (2018). Neural discrete representation learning.

Wang, L., Zhao, Z., Liu, H., Pang, J., Qin, Y., and Wu, Q. (2024). A review of intelligent music generation systems. *Neural Computing and Applications*, pages 1–21.

Wu, Y., He, Y., Liu, X., Wang, Y., and Dannenberg, R. B. (2023). Transplayer: Timbre style transfer with flexible timbre control. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Zhang, L., Huang, H.-N., Yin, L., Li, B.-Q., Wu, D., Liu, H.-R., Li, X.-F., and Xie, Y.-L. (2022). Dolphin vocal sound generation via deep wavegan. *Journal of Electronic Science and Technology*, 20(3):100171.