

# Hi! I'm **Marco Pasini**



ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA



JOHANNES KEPLER  
UNIVERSITÄT LINZ



**Sony CSL**

**Bachelor in**  
Industrial Engineering  
(2020)

**Master in**  
Artificial Intelligence  
(2022)

**Research Intern at**  
Sony CSL Paris  
(2023)

Hi! I'm **Marco Pasini**

**Voice** Conversion

Hi! I'm **Marco Pasini**

**Voice** Conversion → **Music Style**  
Transfer

Hi! I'm **Marco Pasini**





## Fast Infinite Waveform Music Generation

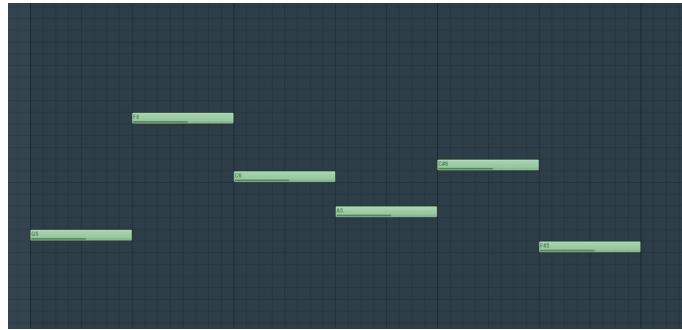
Marco Pasini, Jan Schlüter



Institute of  
Computational  
Perception

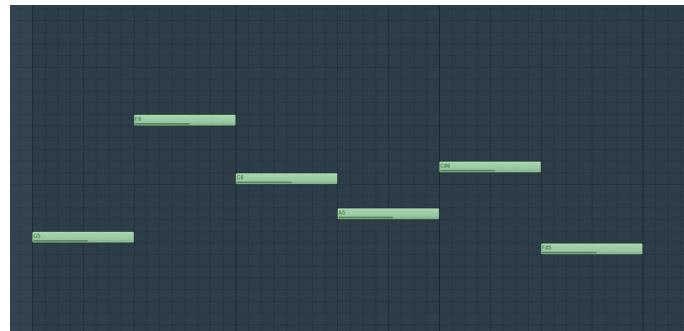
# Music Generation

# Music Generation

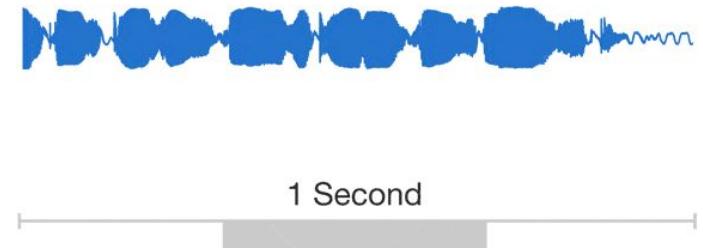


**Symbolic  
Constrained!**

# Music Generation



Symbolic  
Constrained!



Waveform  
**No Constraints!**

# State-of-the-art

# State-of-the-art: OpenAI Jukebox



## State-of-the-art: OpenAI Jukebox

**Encodes music** into discrete codes



## State-of-the-art: OpenAI Jukebox

Encodes music into discrete codes

Generates sequences with transformers



# State-of-the-art: OpenAI Jukebox

**Encodes music into discrete codes**

**Generates sequences with transformers**

**Genre/Lyrics Conditioning**



## State-of-the-art: OpenAI Jukebox

**Encodes music into discrete codes**

**Generates sequences with transformers**

**Genre/Lyrics Conditioning**



**Slow!** 8 hours for 1 minute of music

# State-of-the-art: OpenAI Jukebox



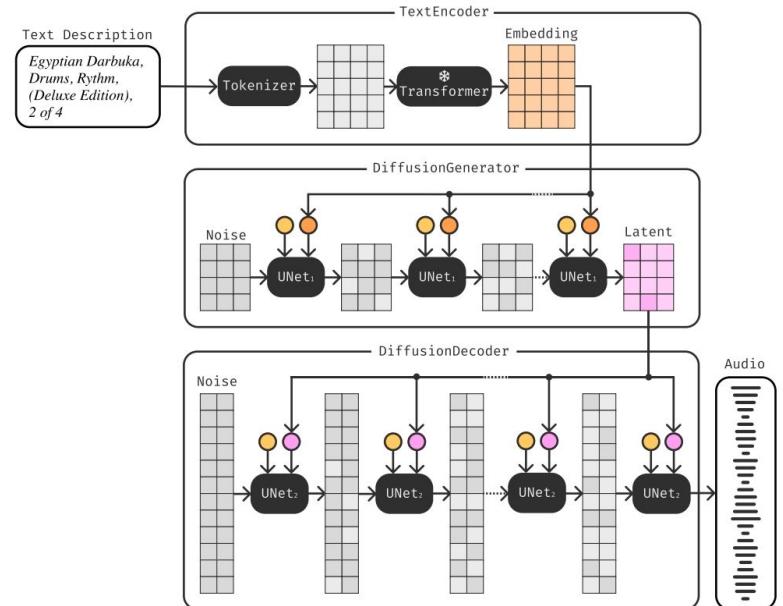
***“Pop, in the style of Beatles”***

# State-of-the-art: OpenAI Jukebox



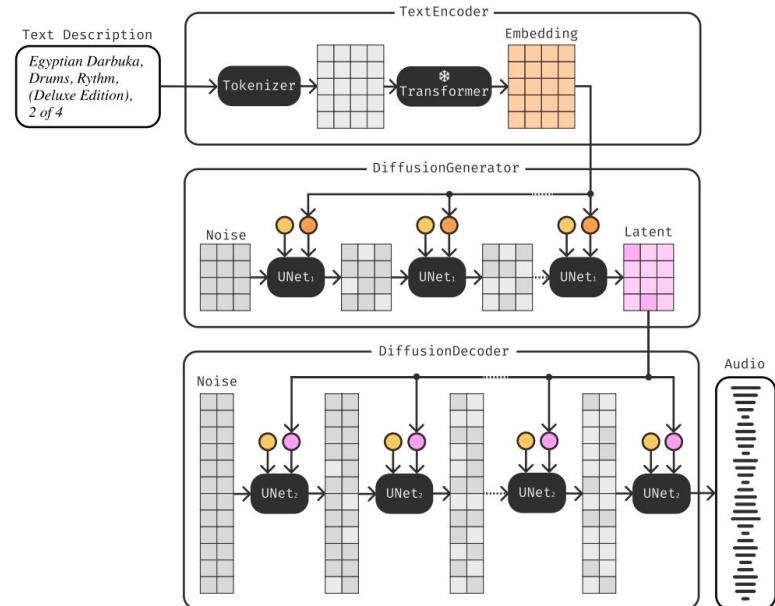
***“Pop, in the style of Beatles”***

# State-of-the-art: Moûsai



# State-of-the-art: Moûsai

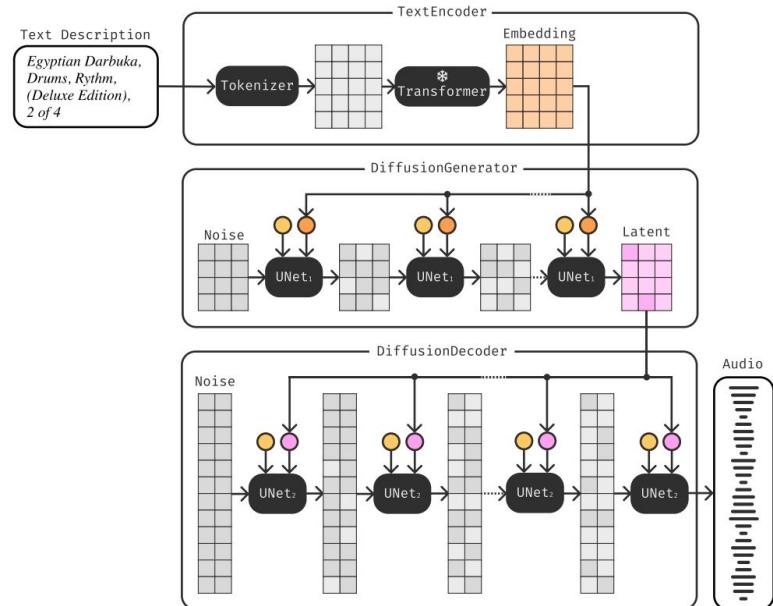
## Latent Diffusion approach



# State-of-the-art: Moûsai

Latent Diffusion approach

Text prompts Conditioning

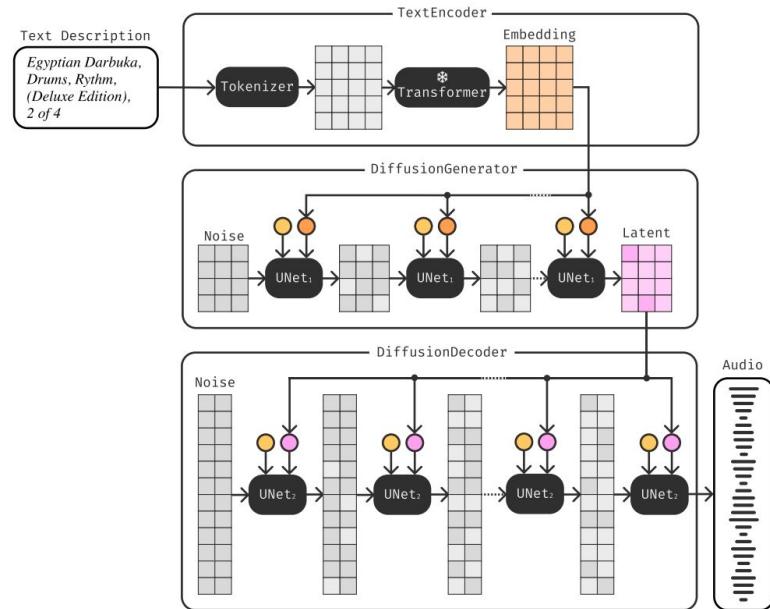


# State-of-the-art: Moûsai

Latent Diffusion approach

Text prompts Conditioning

48 kHz stereo music

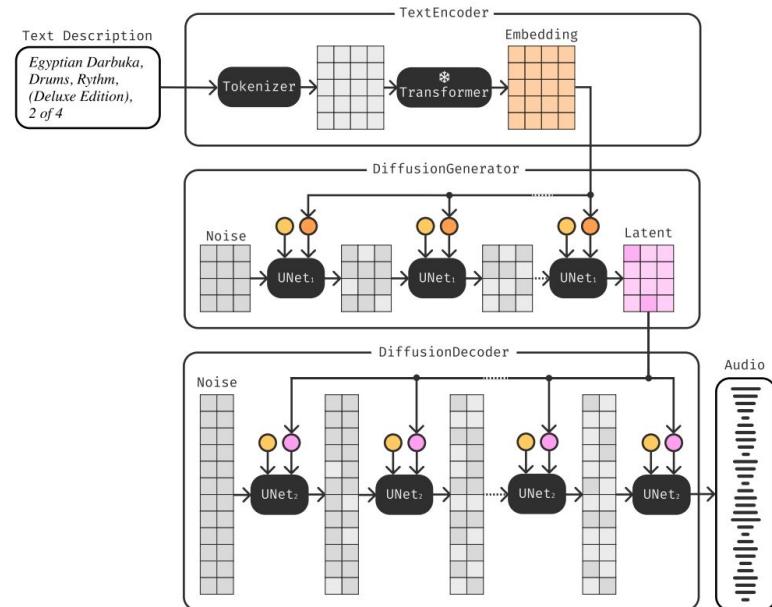


# State-of-the-art: Moûsai

Latent Diffusion approach

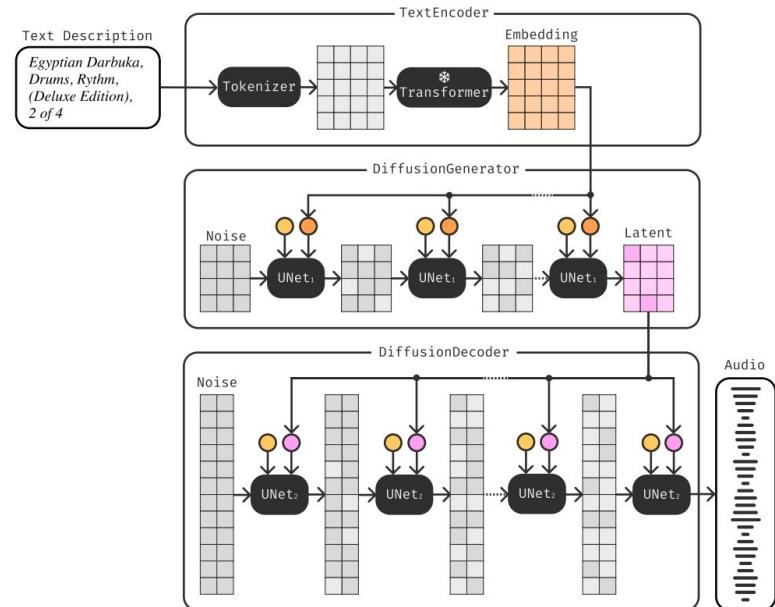
Text prompts Conditioning

48 kHz stereo music



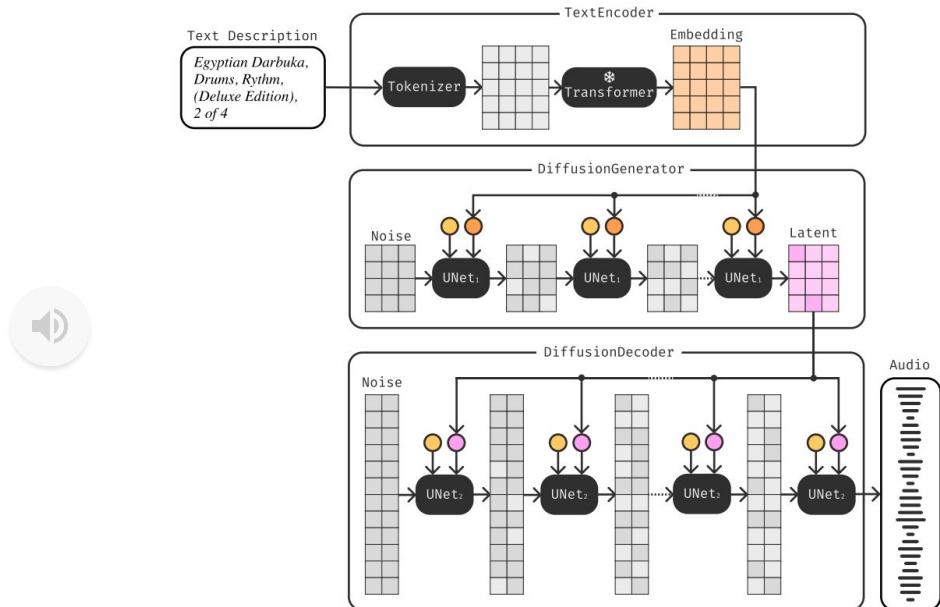
Slow/**Expensive** Training

# State-of-the-art: Moûsai



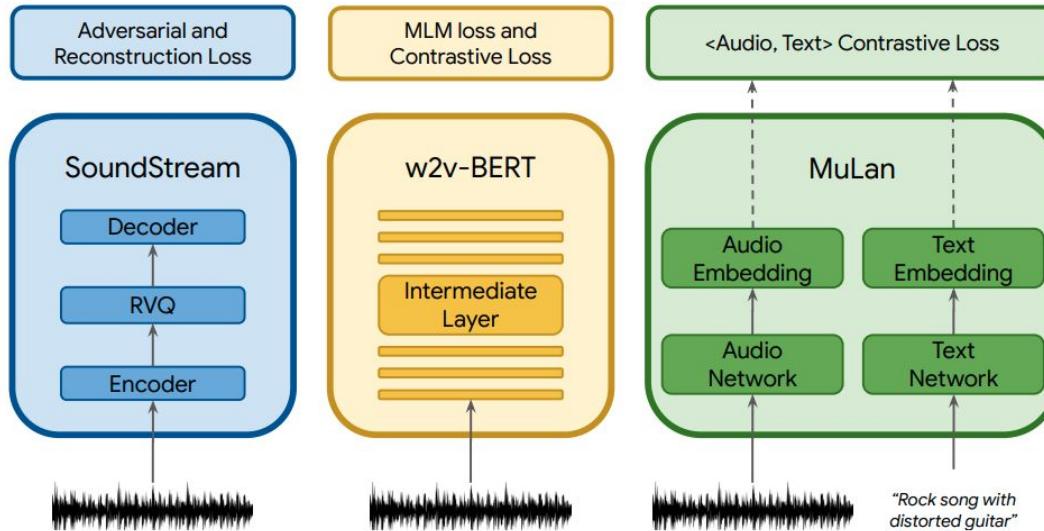
**"Electro House (Remix), 2023, 3 of 4:"**

# State-of-the-art: Moûsai



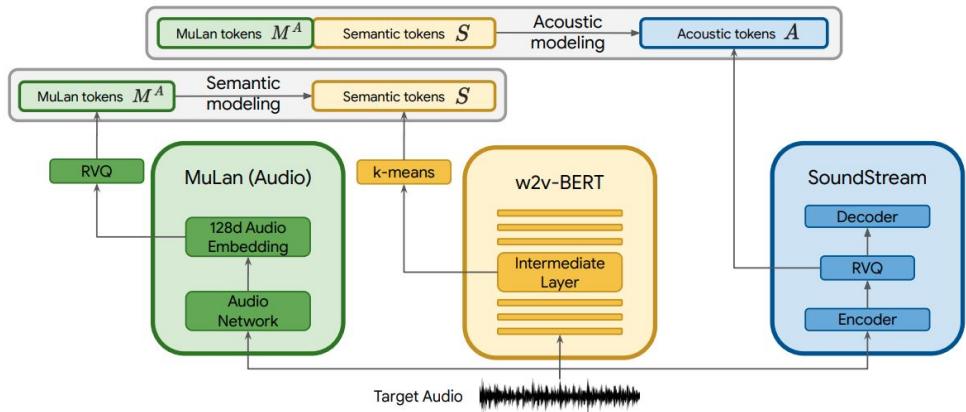
**"Electro House (Remix), 2023, 3 of 4:"**

# State-of-the-art: Google MusicLM



# State-of-the-art: Google **MusicLM**

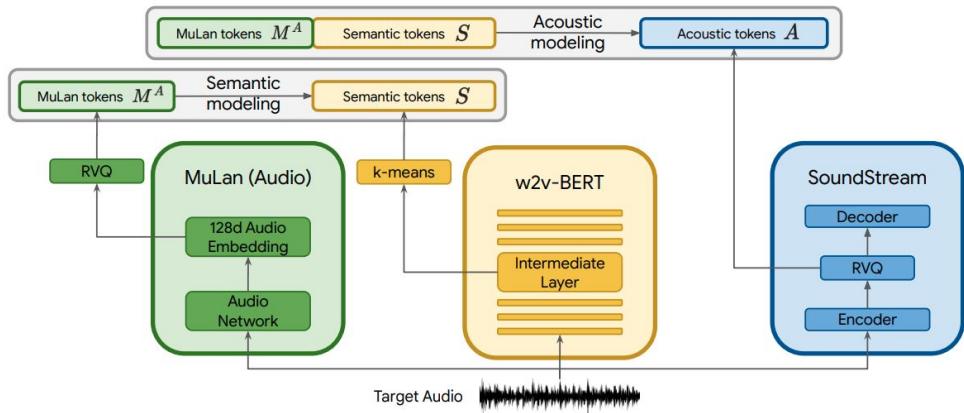
## Seq2Seq framework



# State-of-the-art: Google **MusicLM**

Seq2Seq framework

Similar to **Jukebox**

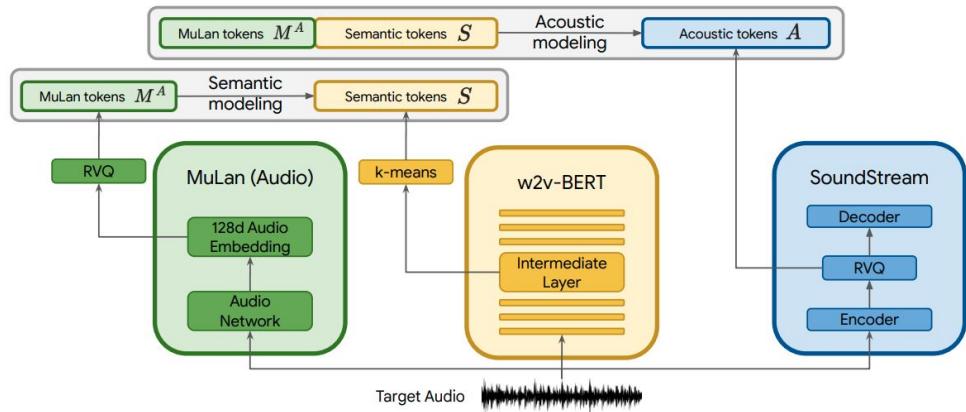


# State-of-the-art: Google **MusicLM**

Seq2Seq framework

Similar to Jukebox

Text/Melody Conditioning



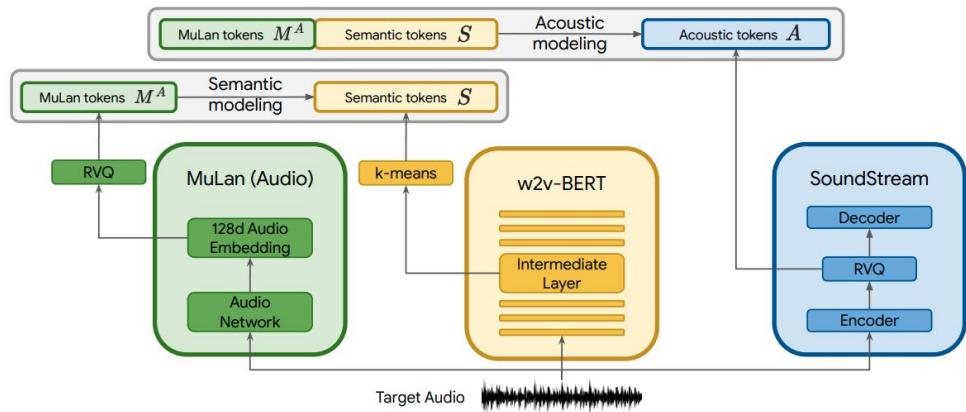
# State-of-the-art: Google **MusicLM**

**Seq2Seq** framework

Similar to **Jukebox**

**Text/Melody Conditioning**

No need for **music-text** pairs!



# State-of-the-art: Google **MusicLM**

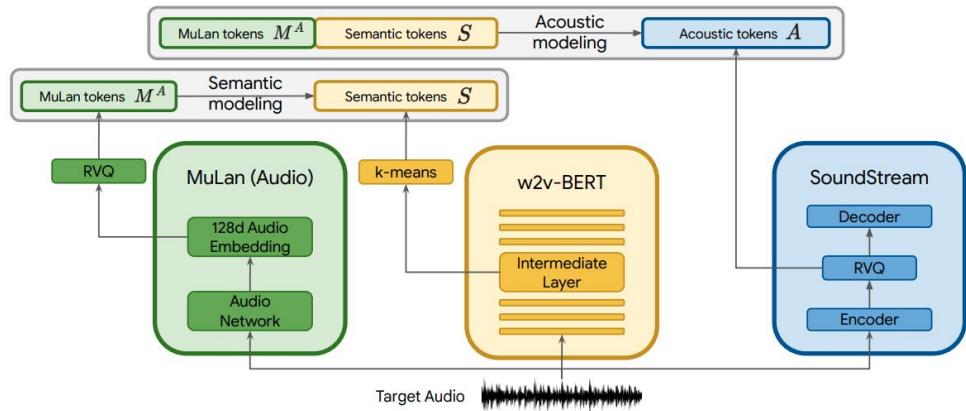
**Seq2Seq** framework

Similar to **Jukebox**

**Text/Melody Conditioning**

No need for **music-text** pairs!

24 kHz **mono** music



# State-of-the-art: Google **MusicLM**

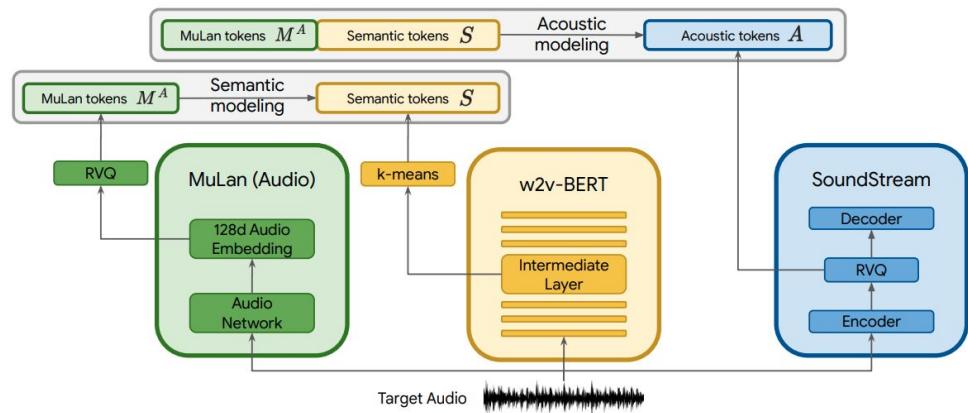
**Seq2Seq** framework

Similar to **Jukebox**

**Text/Melody Conditioning**

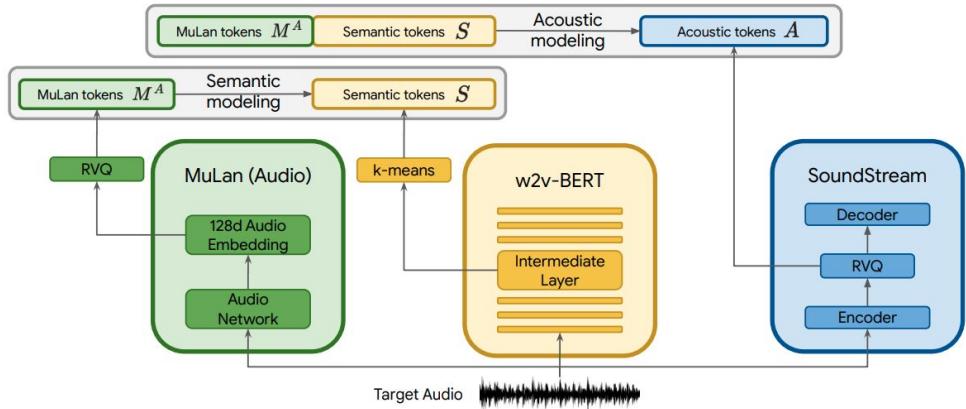
No need for **music-text** pairs!

24 kHz mono music



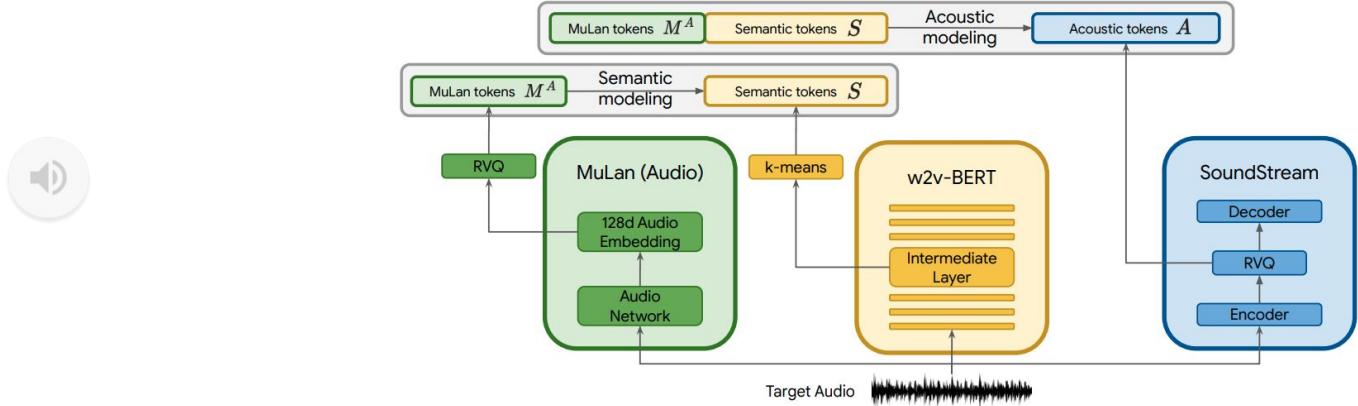
**Slow, Model/Data **not** publicly released**

# State-of-the-art: Google MusicLM



***This is an r&b/hip-hop music piece. There is a male vocal rapping and a female vocal singing in a rap-like manner. The beat is comprised of a piano playing the chords of the tune with an electronic drum backing. The atmosphere of the piece is playful and energetic. This piece could be used in the soundtrack of a high school drama movie/TV show. It could also be played at birthday parties or beach parties.***

# State-of-the-art: Google MusicLM



***This is an r&b/hip-hop music piece. There is a male vocal rapping and a female vocal singing in a rap-like manner. The beat is comprised of a piano playing the chords of the tune with an electronic drum backing. The atmosphere of the piece is playful and energetic. This piece could be used in the soundtrack of a high school drama movie/TV show. It could also be played at birthday parties or beach parties.***

## Our contribution

## Our contribution

**SOTA** models:  
**100s GPUs,  $\infty$ \$**



## Our contribution

SOTA models:

100s GPUs, ∞\$

Us:

1 Gaming GPU, 0\$



## Our contribution

SOTA models:

100s GPUs, ∞\$

Us:

1 Gaming GPU, 0\$



**How/What** can we contribute?

Our contribution:

**MUSIKA!**

Our contribution:



Generate stereo waveform **music** of **arbitrary length**

Our contribution:



Generate stereo waveform **music** of arbitrary length

**Stylistically coherent** through time

Our contribution:



Generate stereo waveform **music** of arbitrary length

**Stylistically coherent** through time

**Can be conditioned** on signals (tempo, note density)

Our contribution:



Generate stereo waveform **music** of arbitrary length

**Stylistically coherent** through time

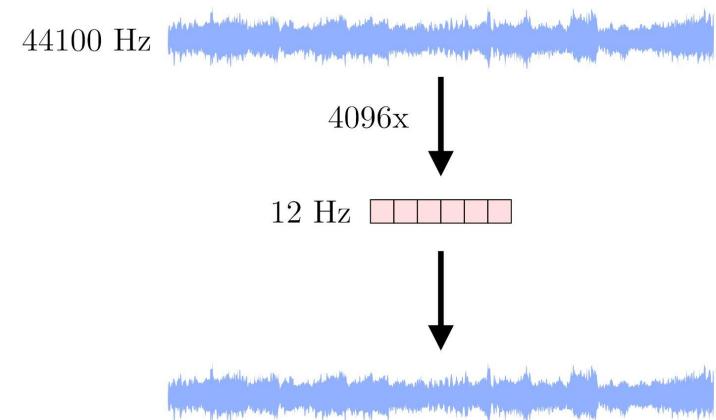
Can be **conditioned** on signals (tempo, note density)

**Fast!** Much **faster than real-time** on CPU

## How **Musika** works

# How **Musika** works

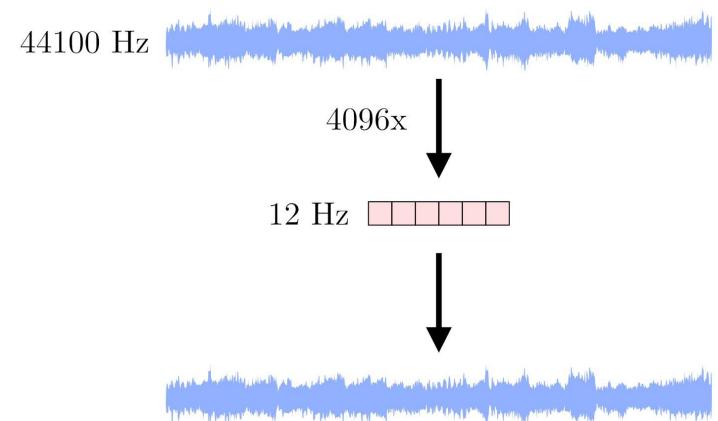
**Encode audio to compressed **invertible** representations**



# How **Musika** works

Encode audio to compressed **invertible** representations

Train **generative model** on latent representations

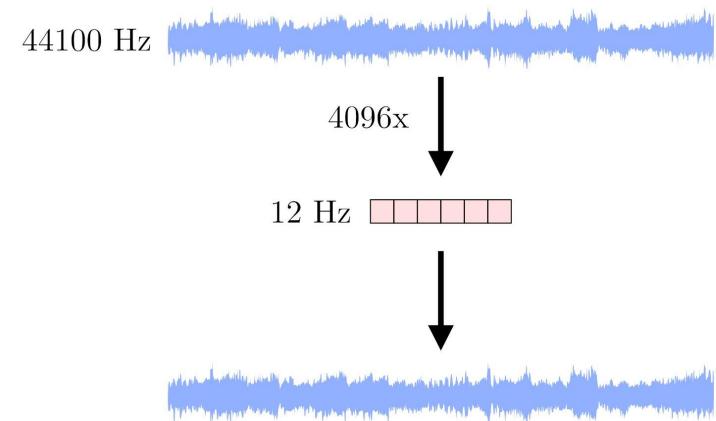


# How **Musika** works

Encode audio to compressed **invertible** representations

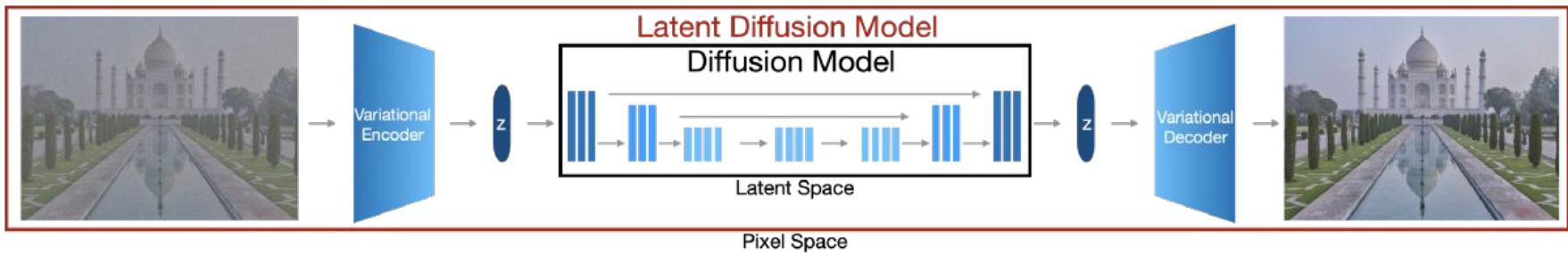
Train **generative model** on latent representations

Use **coordinate system** for infinite generation

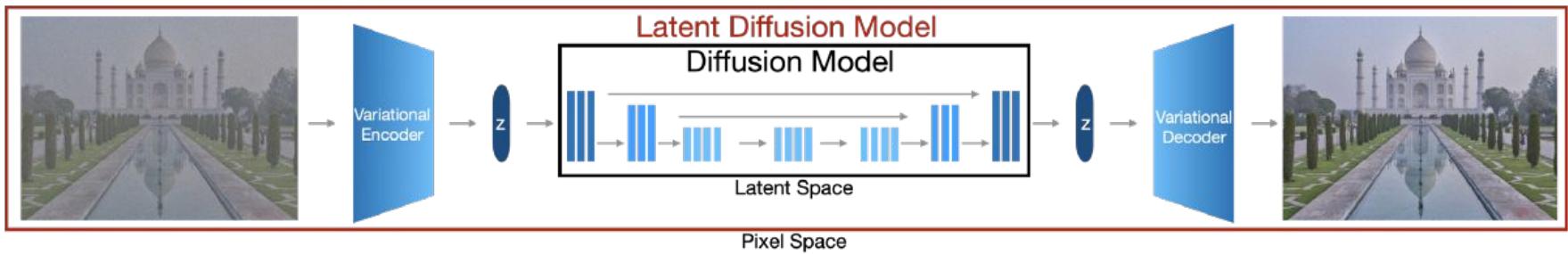


# Latent Generation

# Latent Generation



# Latent Generation



Recent common **approach** to **Fast/Efficient Generative Modeling**

# Audio **Autoencoder**

# Audio **Autoencoder**

Priorities:

# Audio **Autoencoder**

Priorities:

**Inference Speed**

# Audio **Autoencoder**

Priorities:

Inference **Speed**

**Parallel** Encoding/Decoding

# Audio **Autoencoder**

Priorities:

Inference **Speed**

**Parallel** Encoding/Decoding

**Fast** Training

# Audio **Autoencoder**

Priorities:

Inference **Speed**

**Parallel** Encoding/Decoding

**Fast** Training

Generating directly **Waveforms** (1D) is **expensive**

# Audio **Autoencoder**

Priorities:

Inference **Speed**

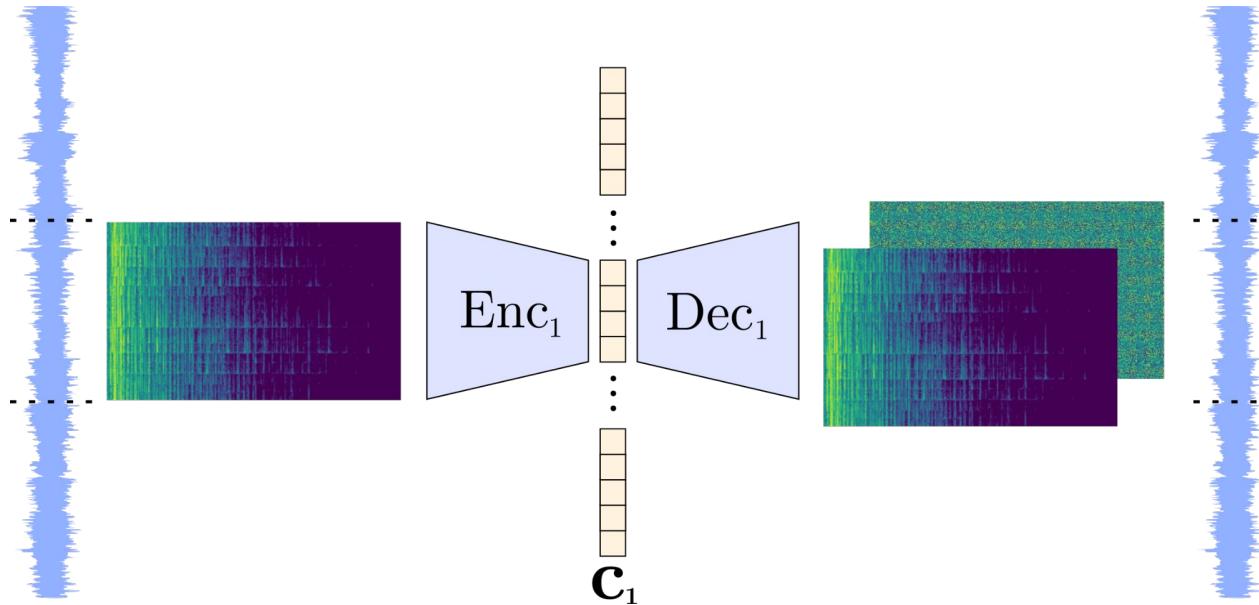
**Parallel** Encoding/Decoding

**Fast** Training

Generating directly **Waveforms** (1D) is **expensive**

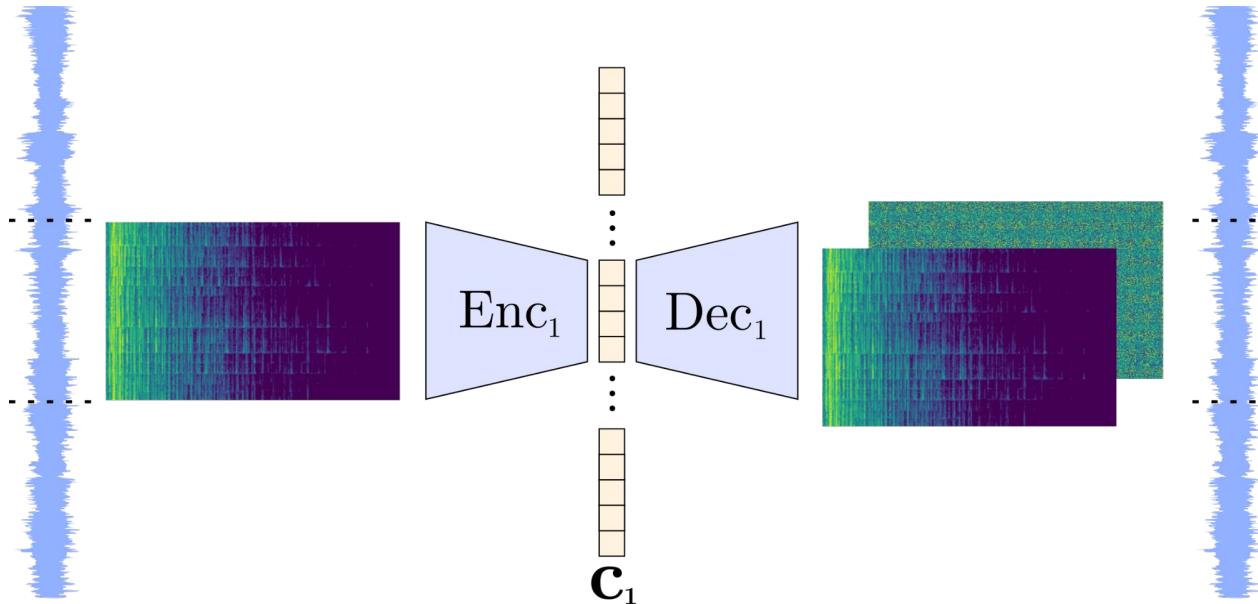
→ **Invertible 2D** Representation

## Audio **Autoencoder**: 1st Training Phase



Decoder outputs **magnitude+phase** spectrograms: **invertible!**

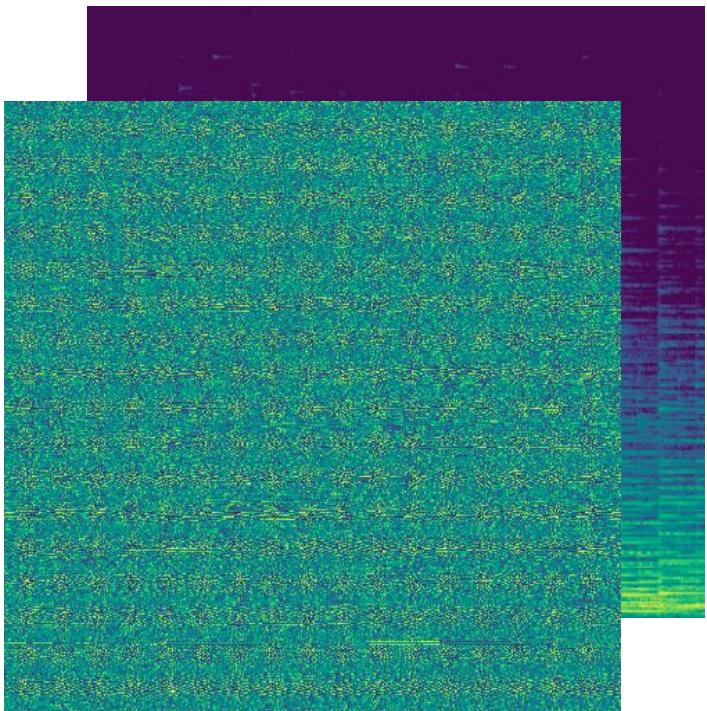
## Audio **Autoencoder**: 1st Training Phase



Only reconstruct **magnitude**, leave **phase** for later

# Magnitude and Phase

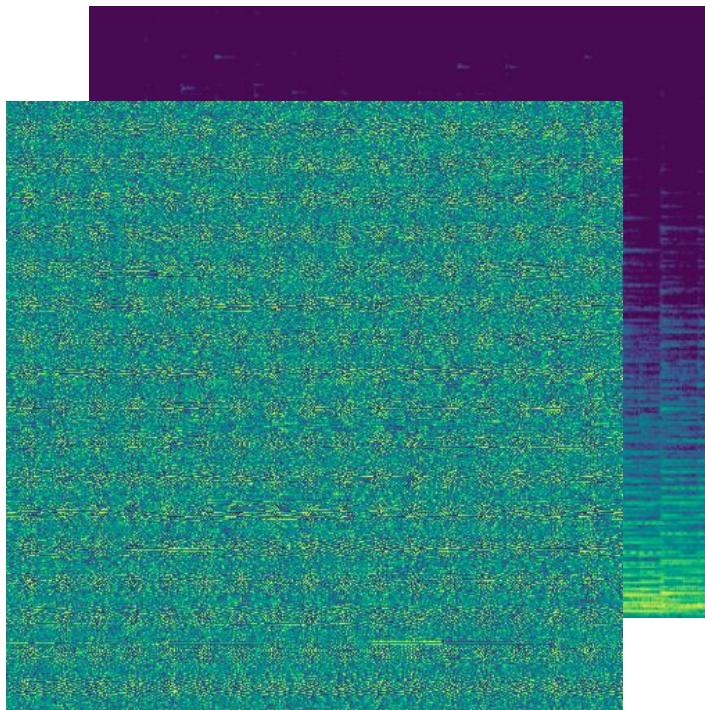
Modeling **Phase** is difficult!



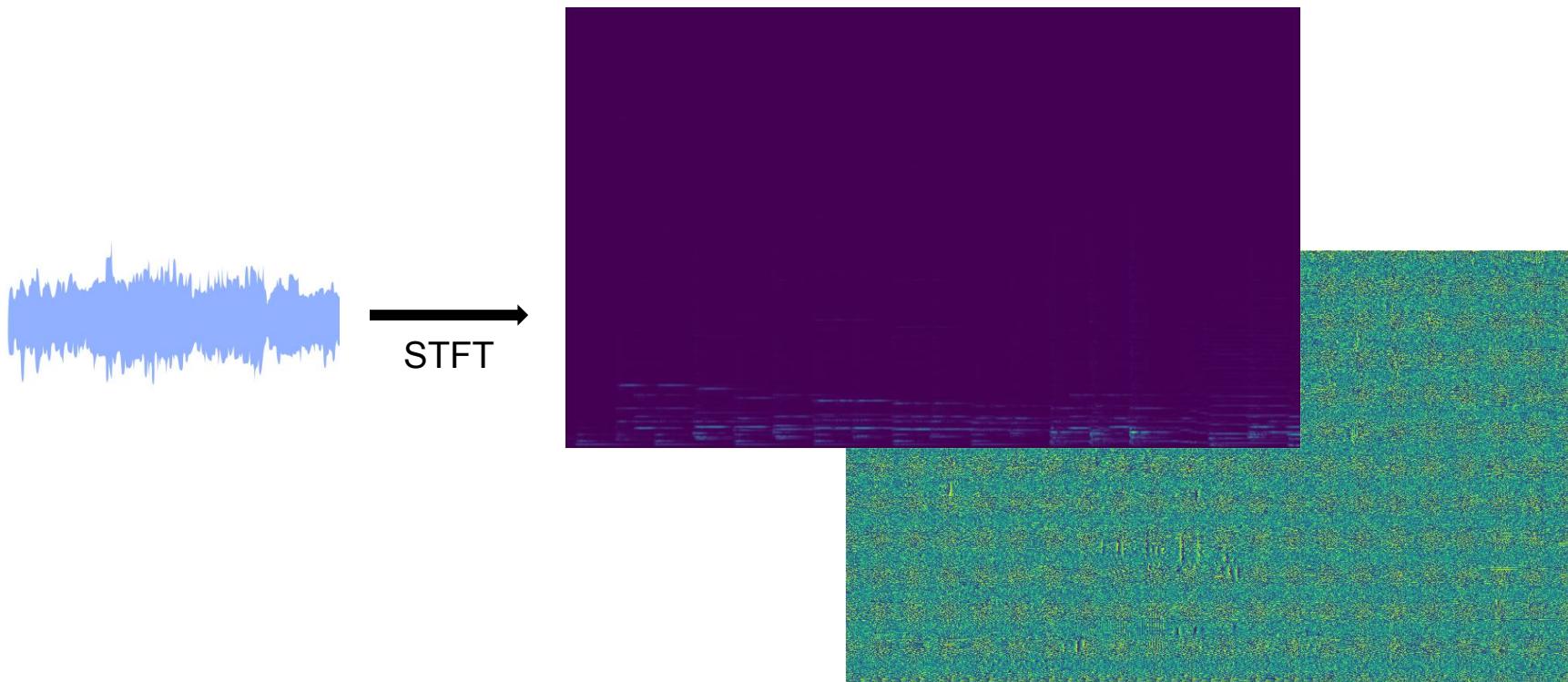
# **Magnitude and Phase**

Modeling **Phase** is difficult!

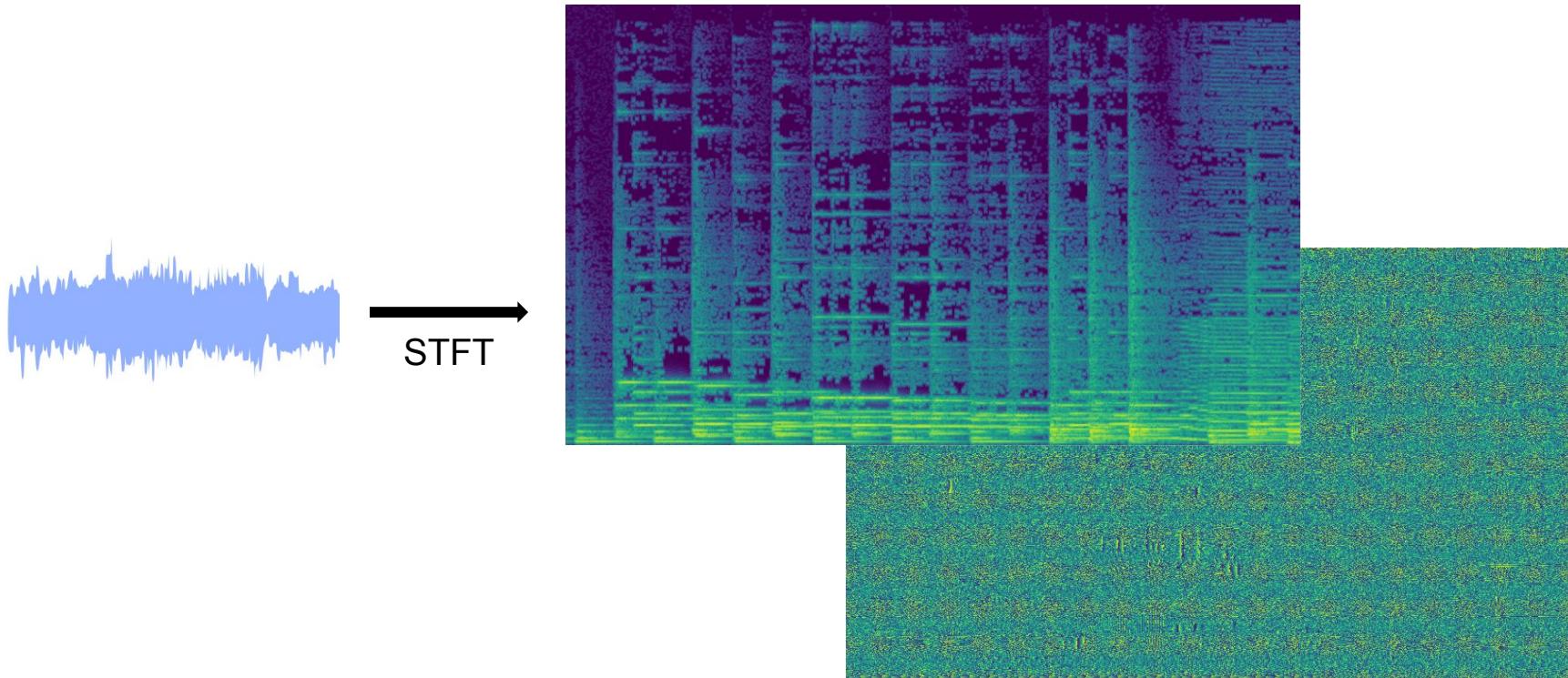
**Lack of Structure**



## Magnitude and Phase: STFT



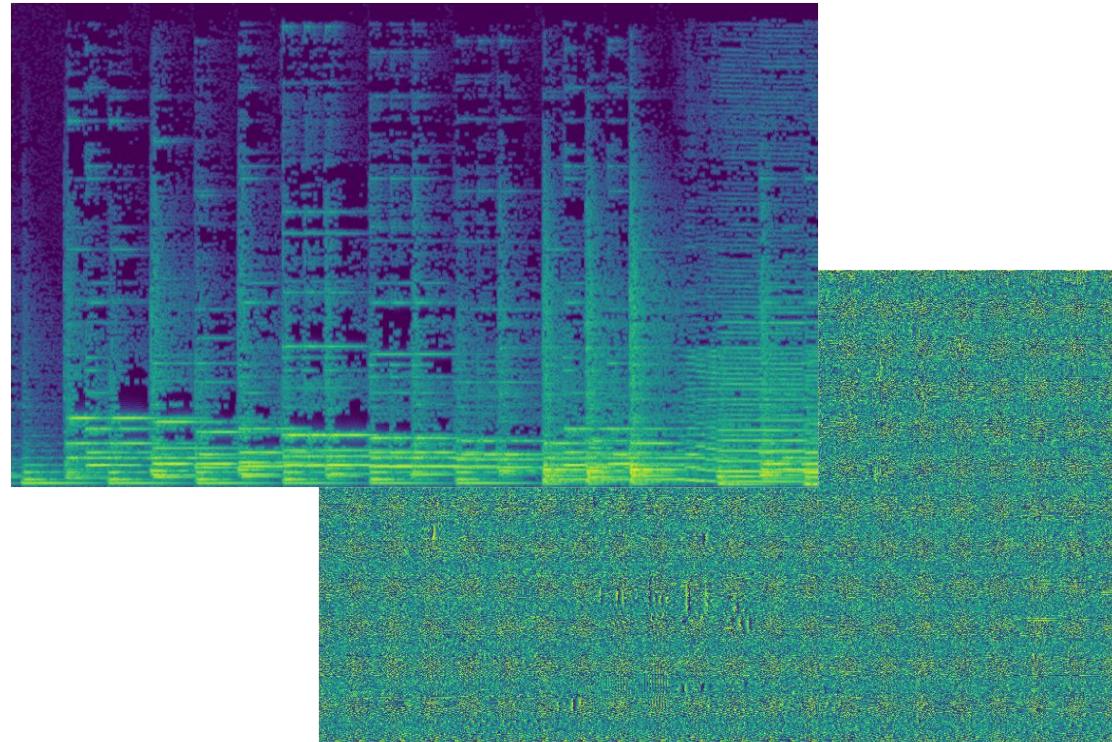
## Magnitude and Phase: STFT



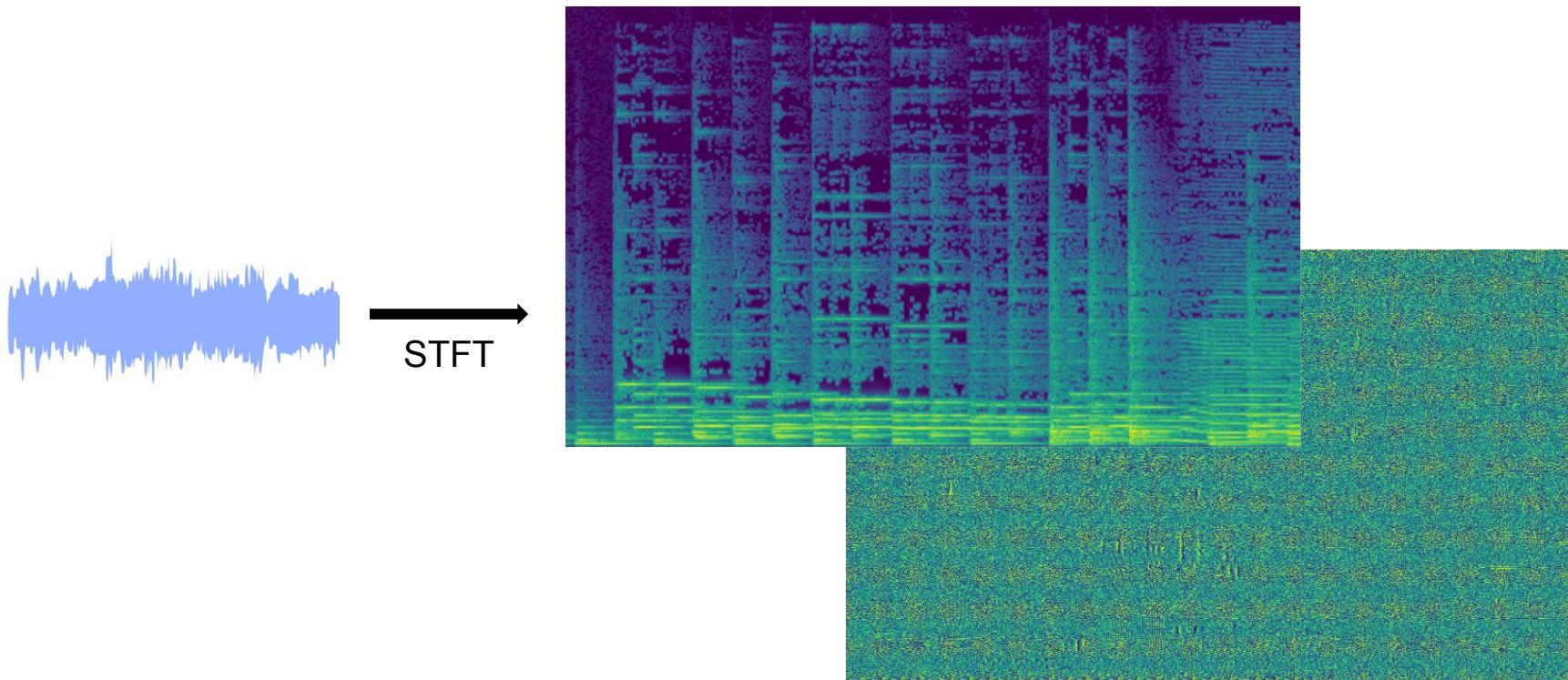
# Magnitude and Phase: STFT



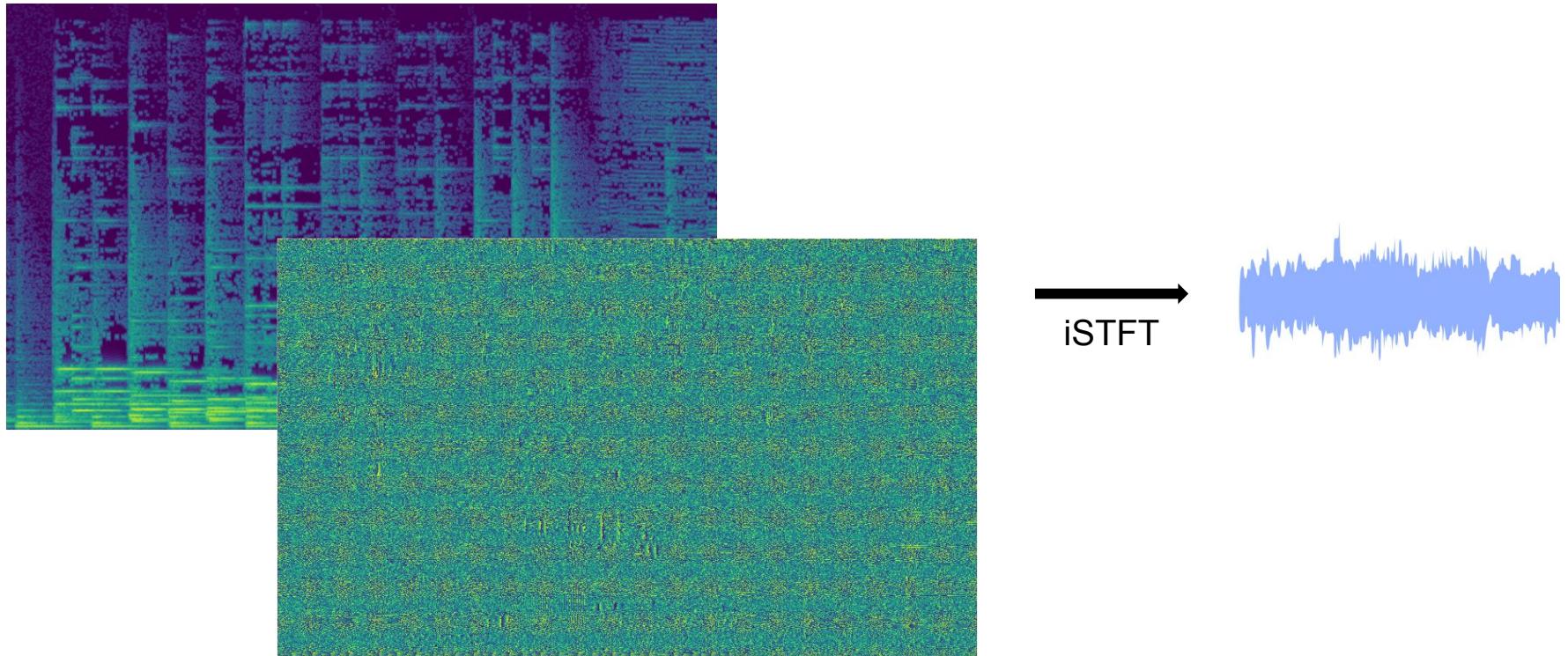
→  
STFT



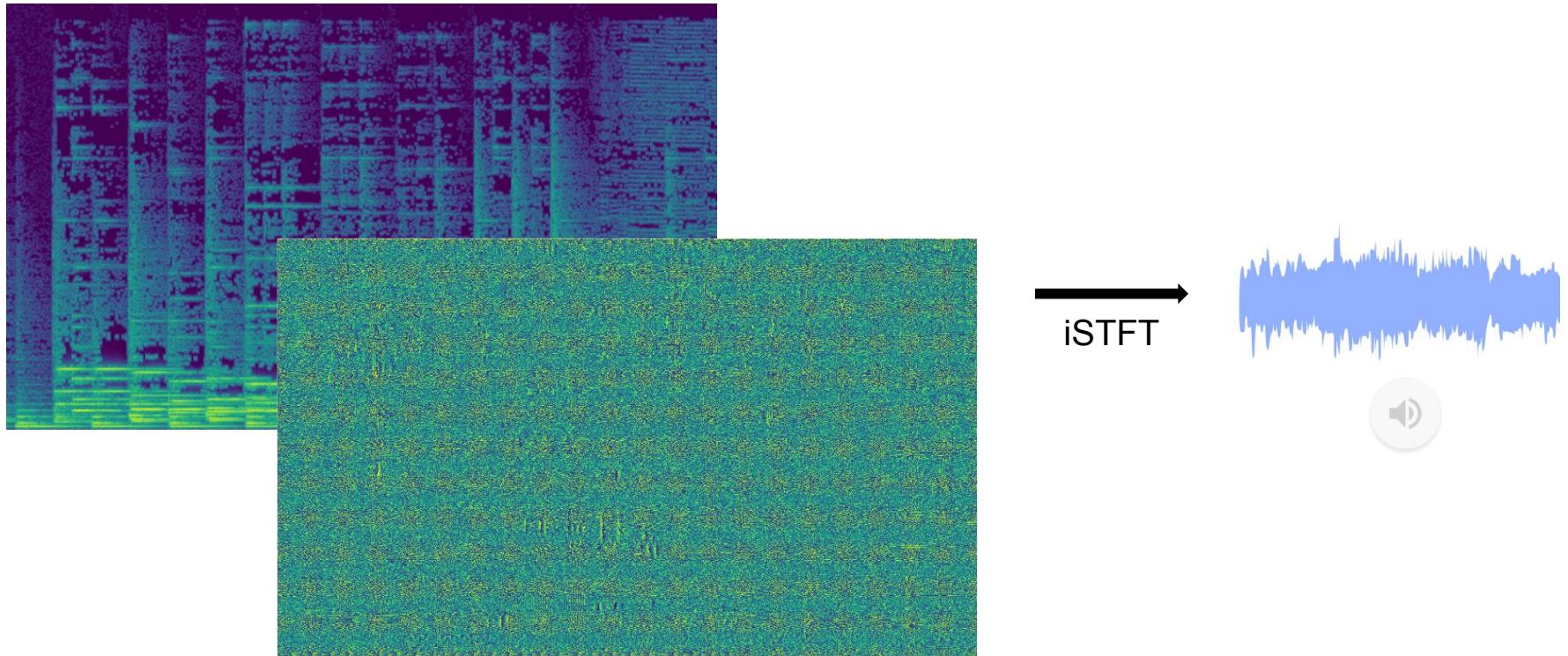
## Magnitude and Phase: STFT



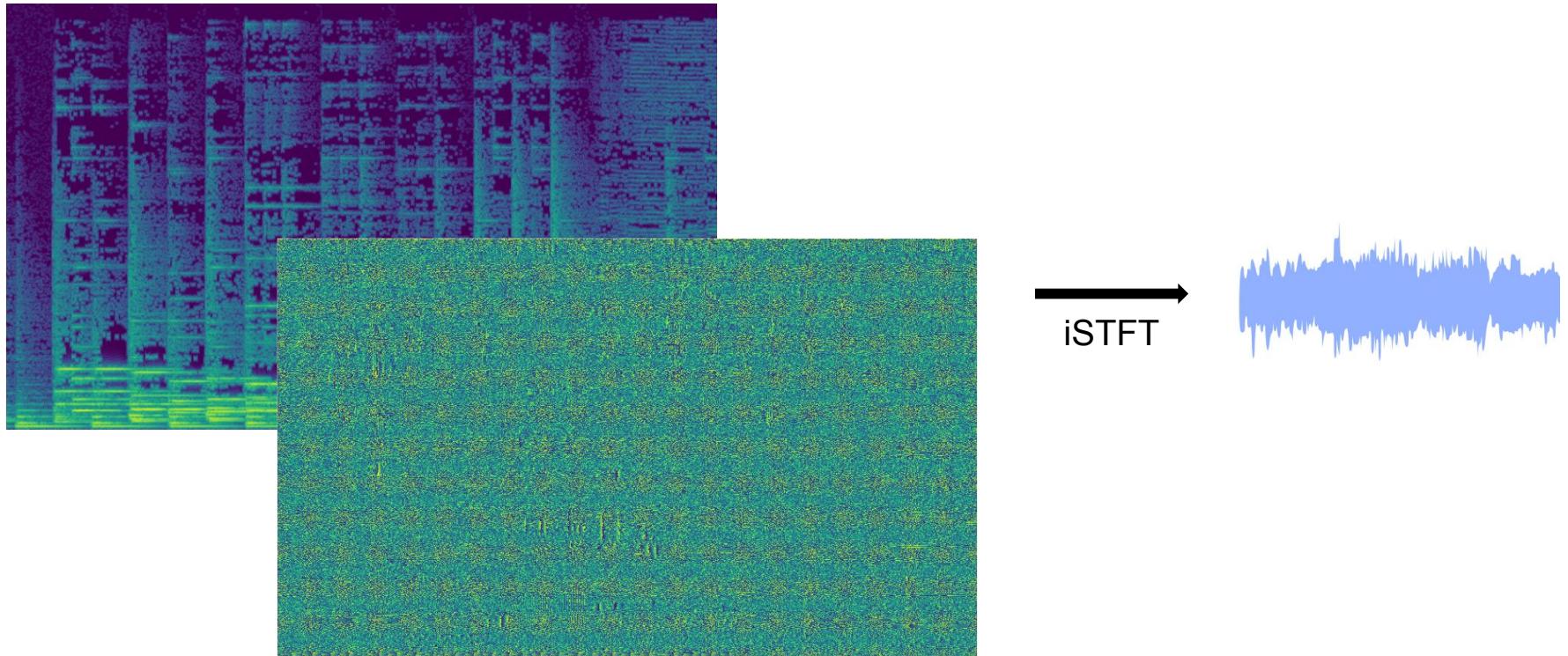
## Magnitude and Phase: inverseSTFT



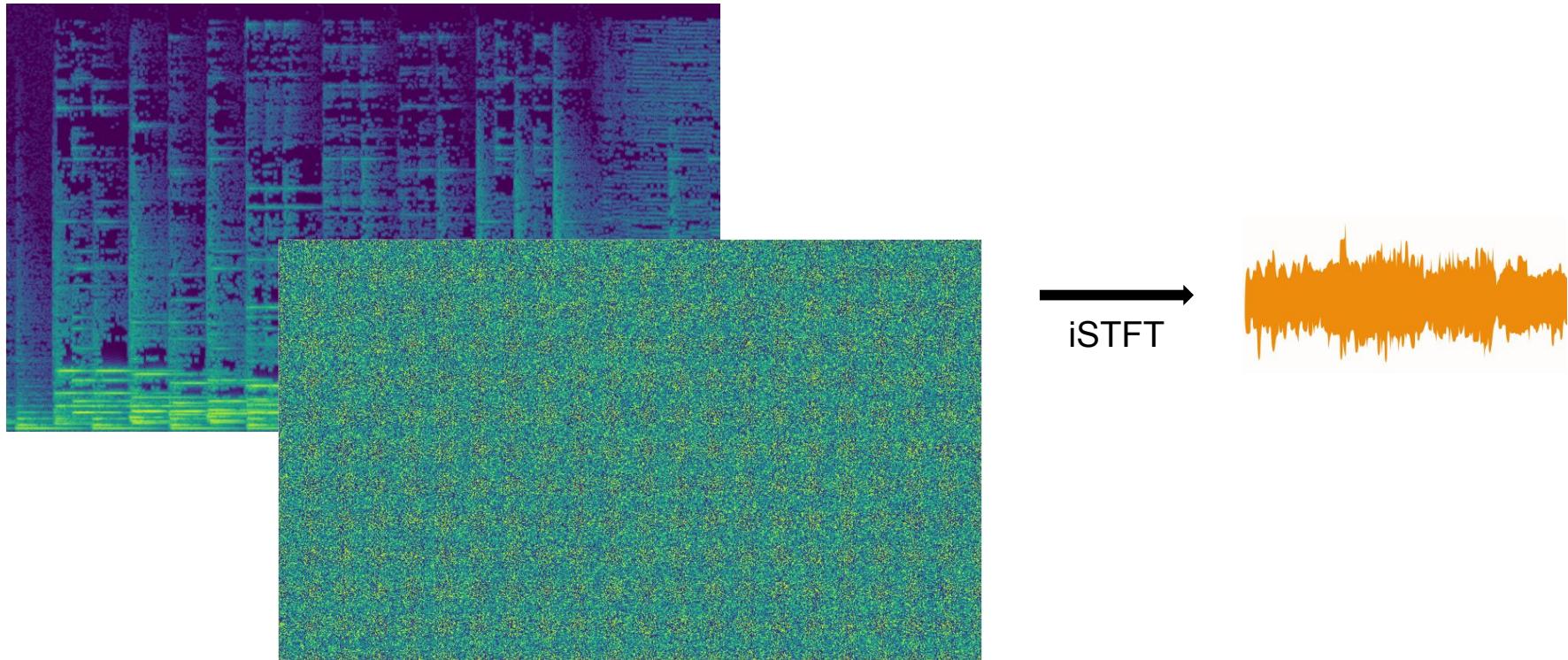
## Magnitude and Phase: inverseSTFT



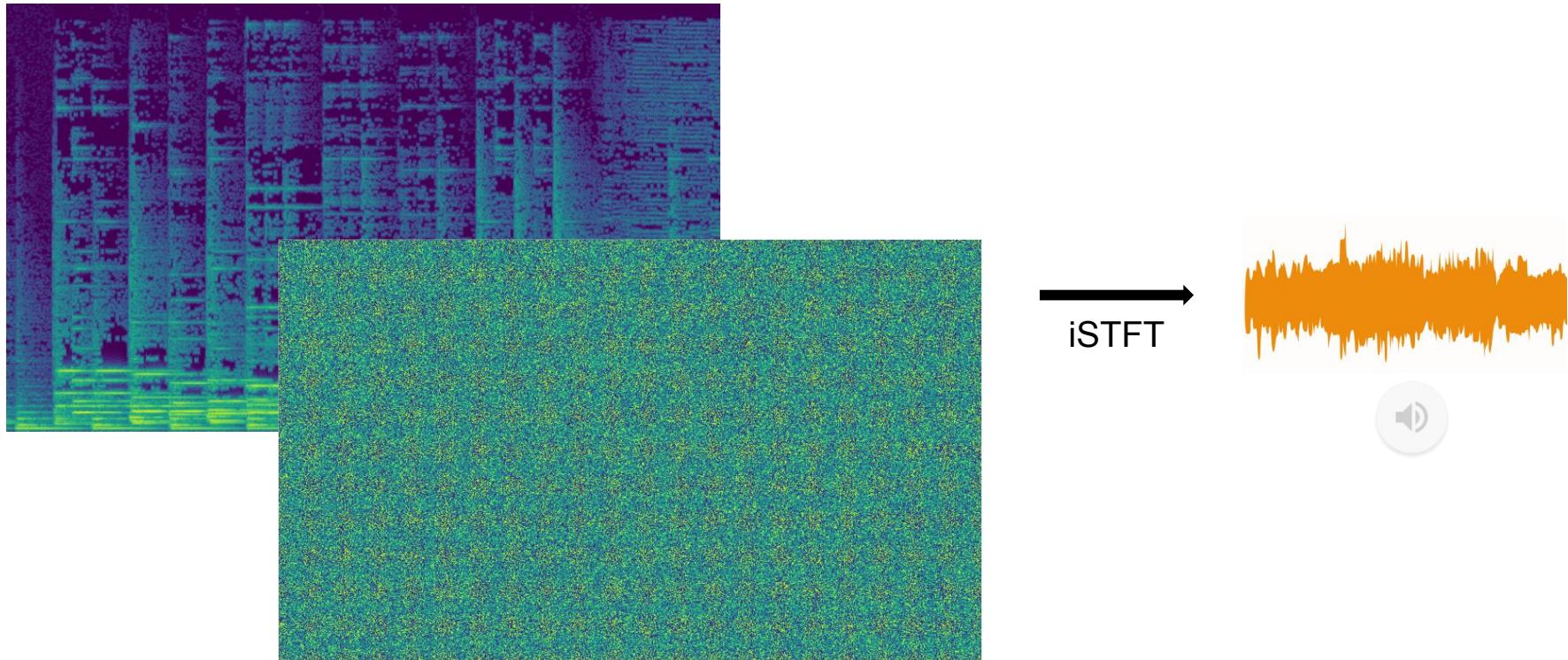
## Magnitude and Phase: inverseSTFT



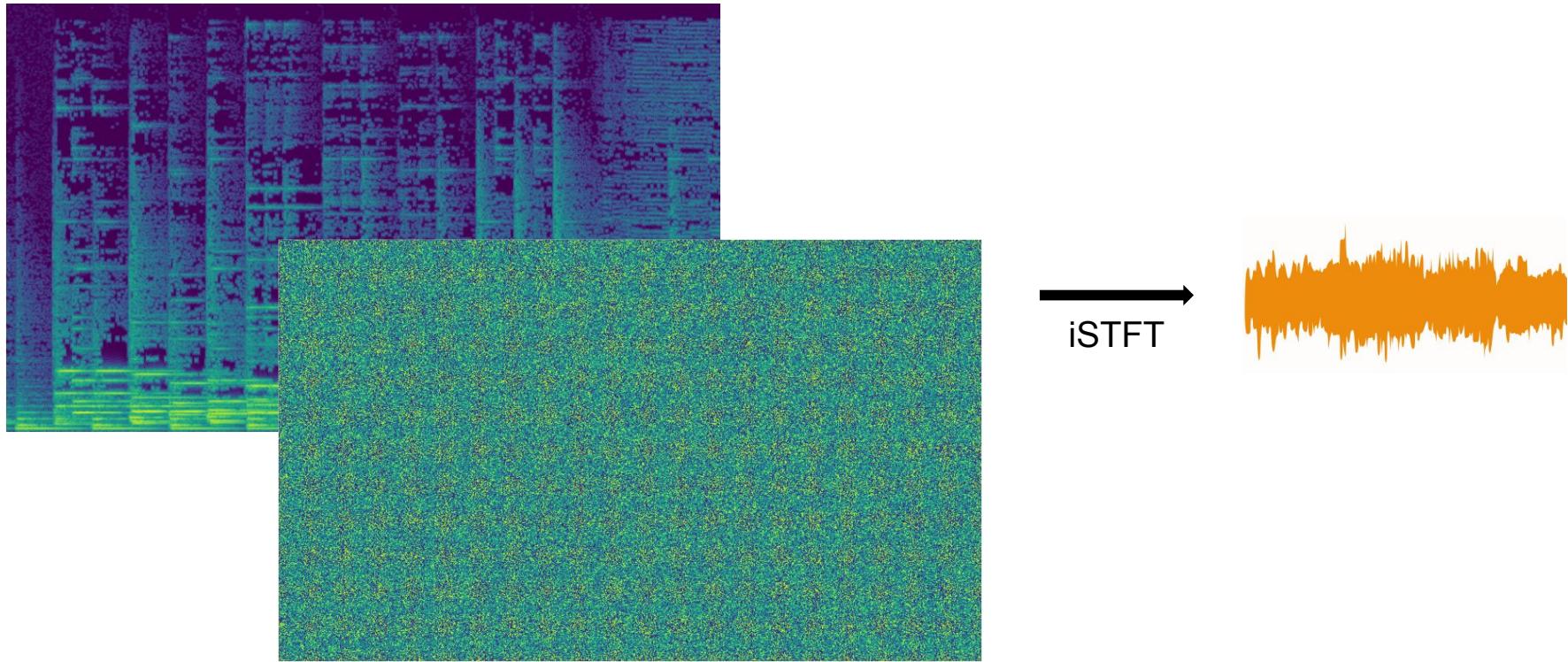
## Magnitude and random Phase: inverseSTFT



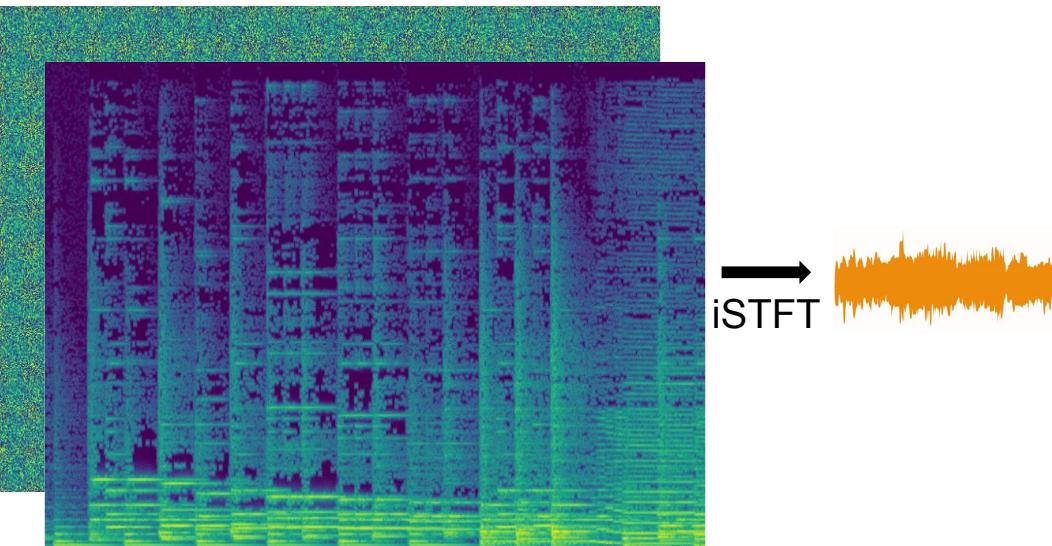
## Magnitude and random Phase: inverseSTFT



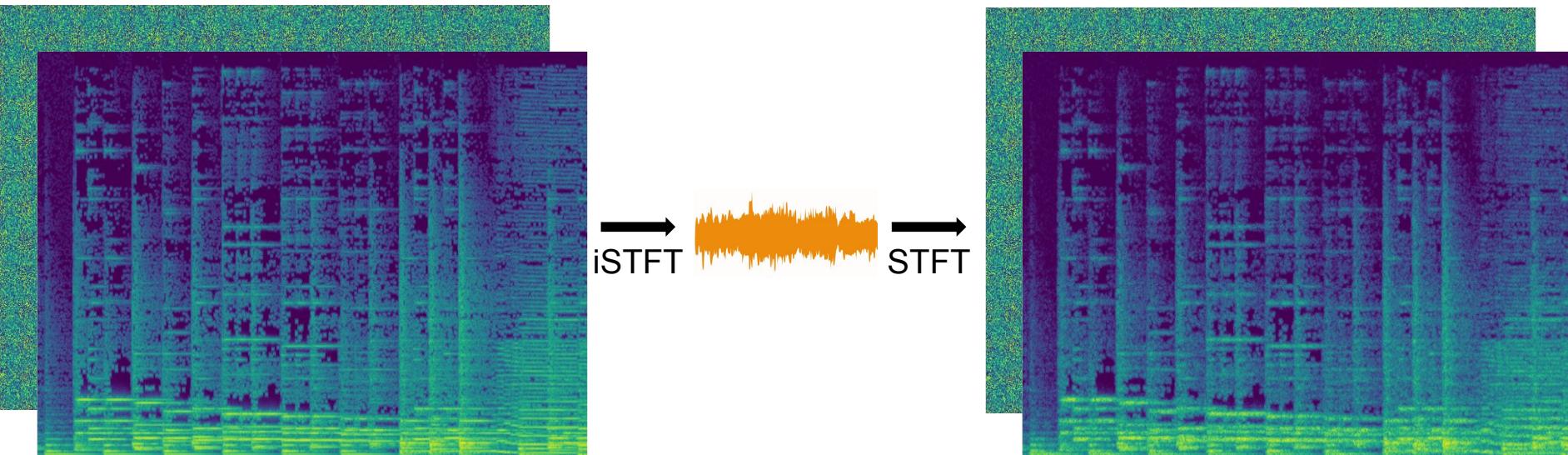
## Magnitude and random Phase: inverseSTFT



## Magnitude and random Phase: inverseSTFT



## Magnitude and random Phase: inverseSTFT



## Magnitude and random Phase: inverseSTFT

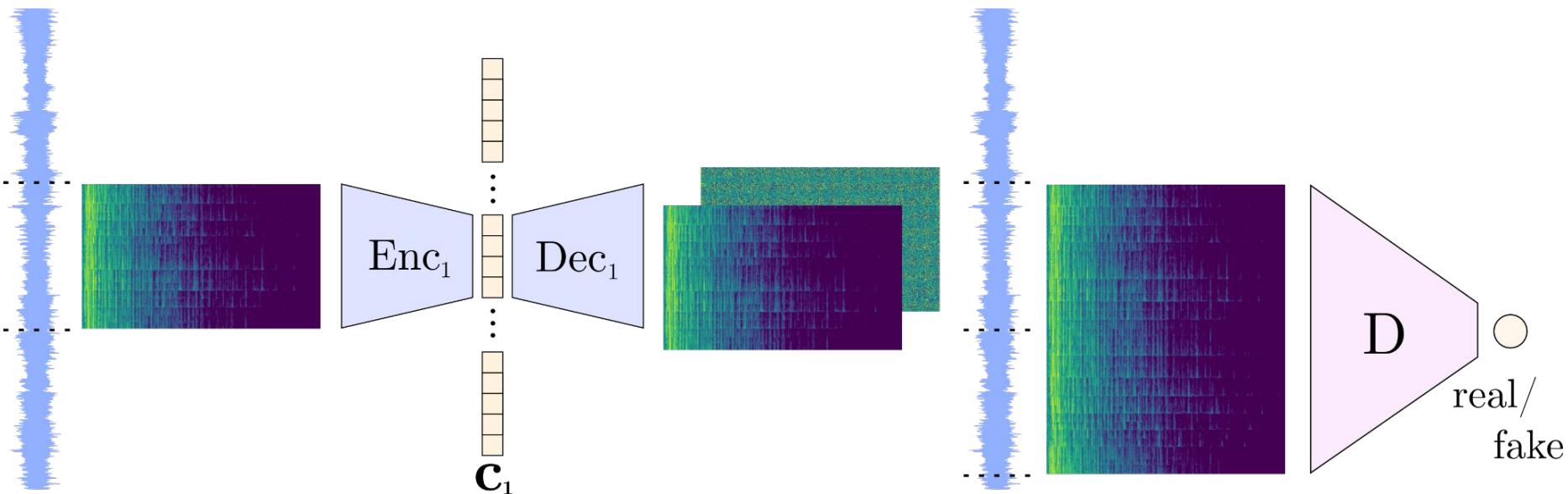


## Magnitude and random Phase: inverseSTFT



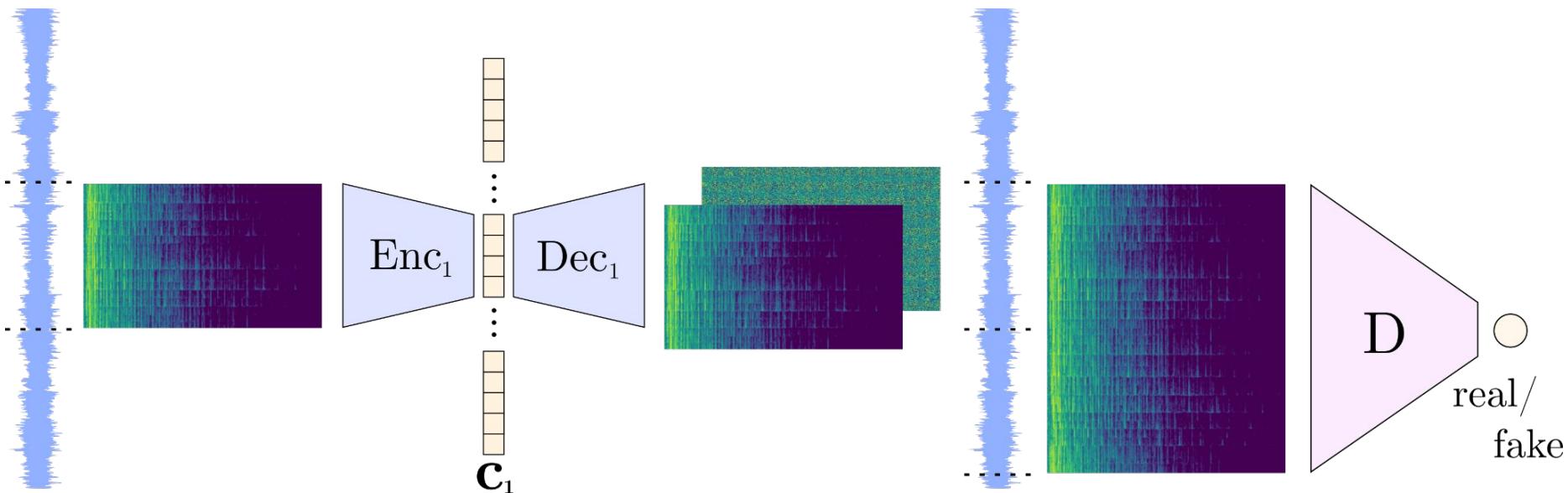
Use **Magnitude** instead of **Waveform** to model **Phase!**

## Audio Autoencoder: 2nd Training Phase



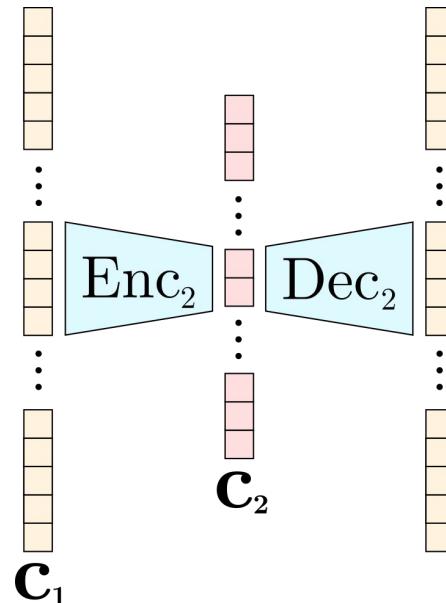
**Adversarial objective** models **phases** indirectly

## Audio Autoencoder: 2nd Training Phase



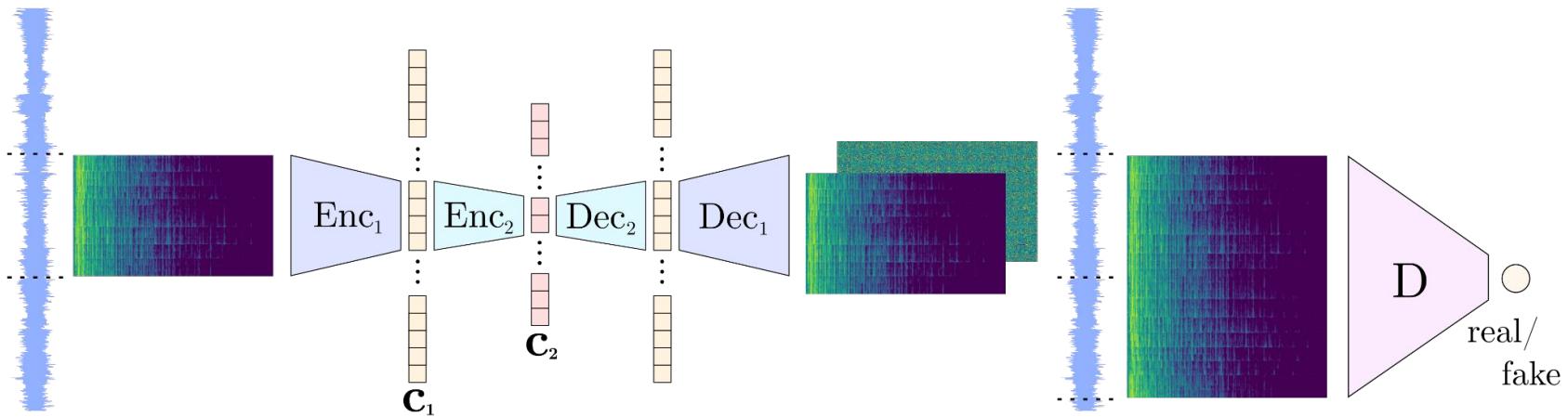
**Adversarial objective removes boundary artifacts**

## Audio **Autoencoder**: 3rd Training Phase



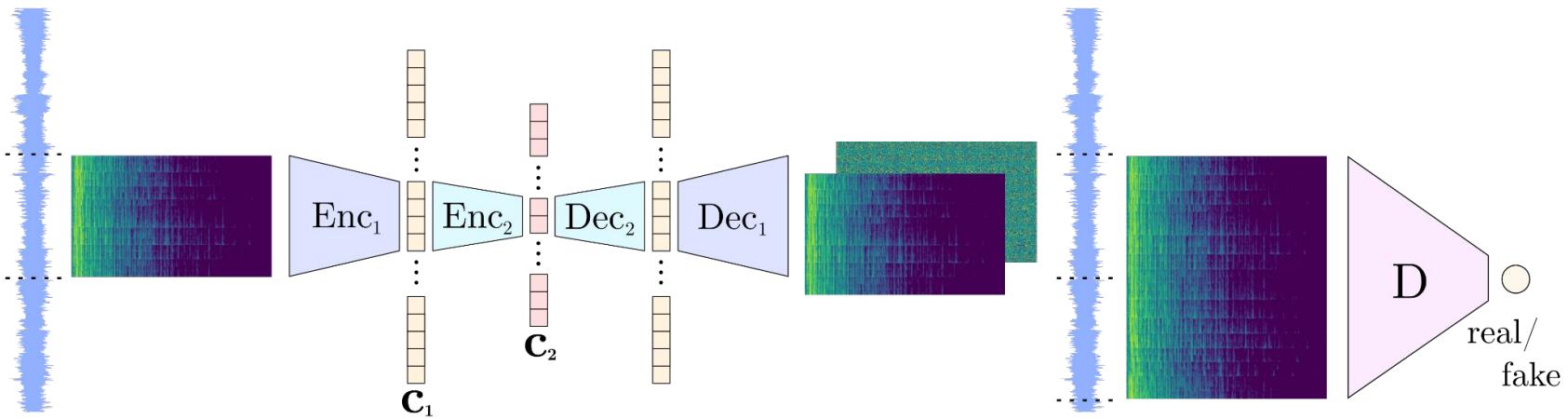
Reconstruct **representations** from **1st** level Autoencoder

## Audio **Autoencoder**: 4th Training Phase



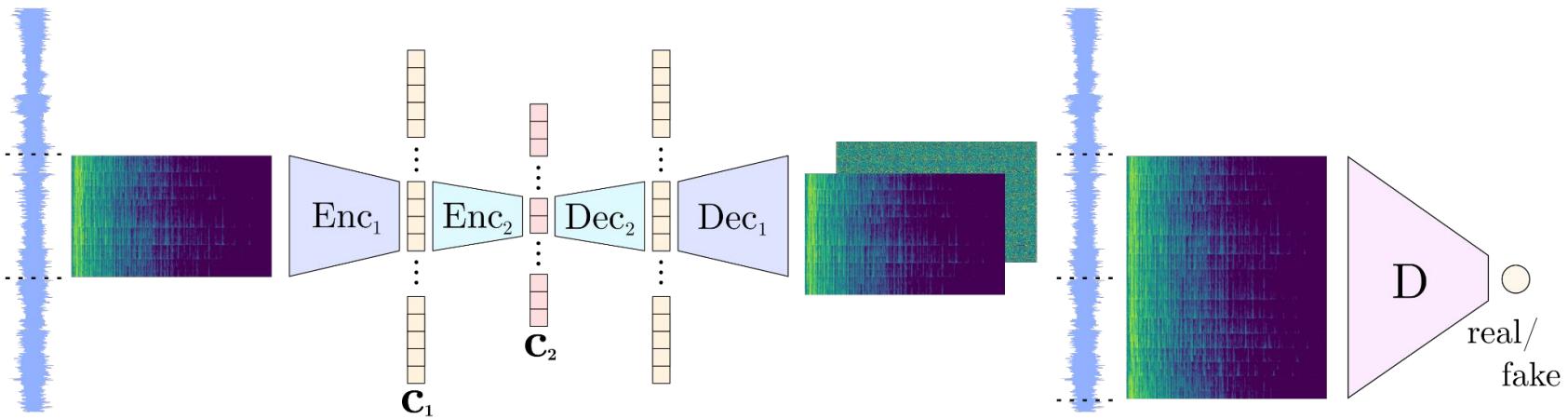
**Gradients through frozen 1st level Autoencoder**

# Audio Autoencoder



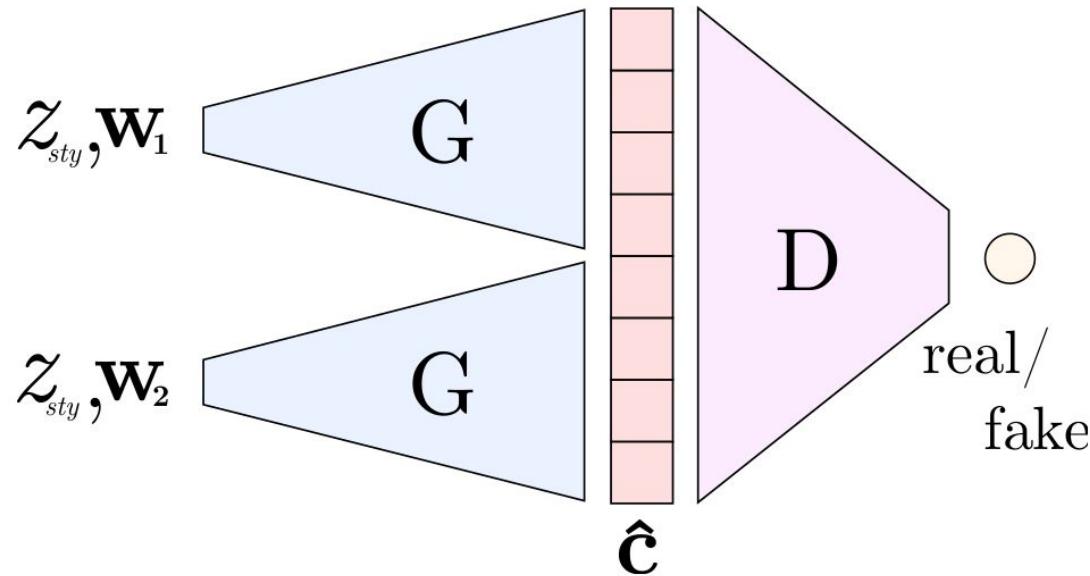
**Why 2-stage design?**

# Audio Autoencoder



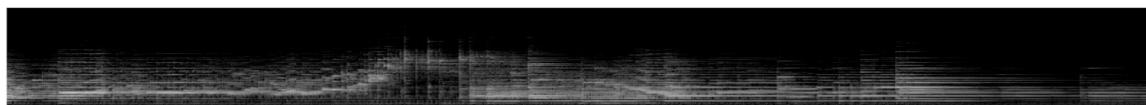
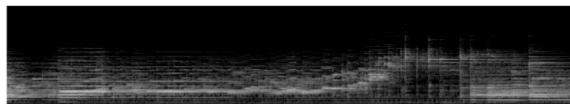
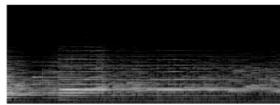
Why 2-stage design? Reconstruct **phase** + semantic **compression**

## Latent Generator



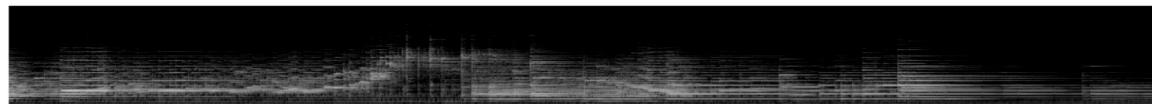
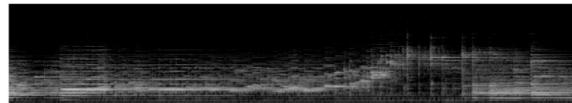
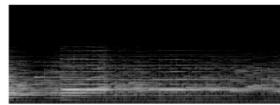
A **Latent GAN** generates **new** sequences of **latent vectors**

# Arbitrary Length Generation



# Arbitrary Length Generation

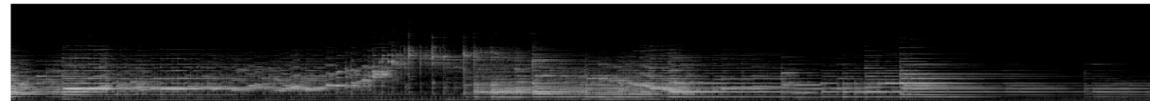
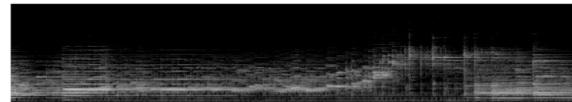
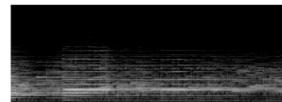
**Autoregressive** models (Jukebox): Trivial



# Arbitrary Length Generation

**Autoregressive** models (Jukebox): Trivial

**Non-Autoregressive** models: not Trivial!

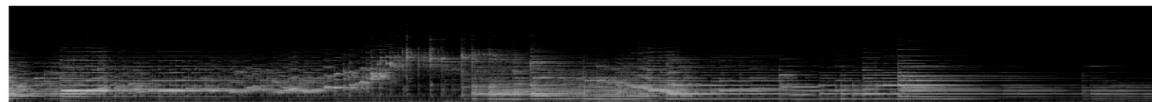
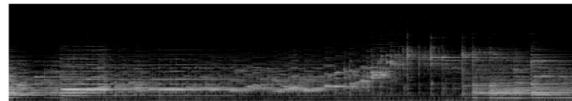
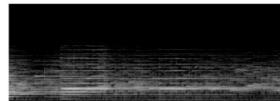


## Arbitrary Length Generation

Autoregressive models (Jukebox): Trivial

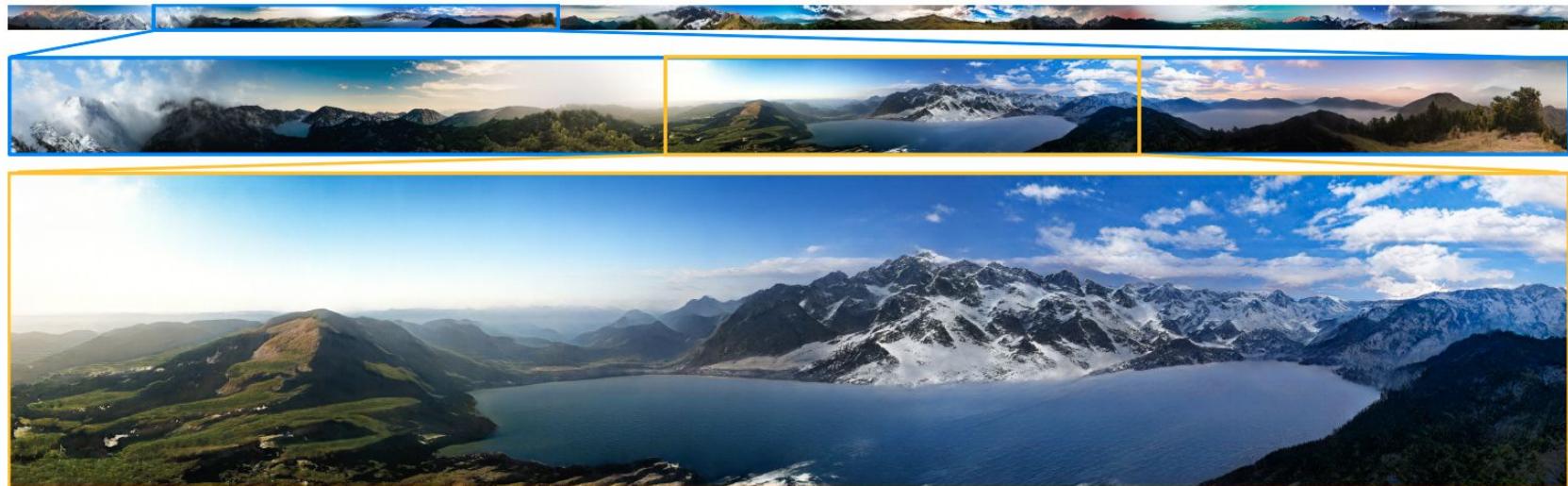
Non-Autoregressive models: not Trivial!

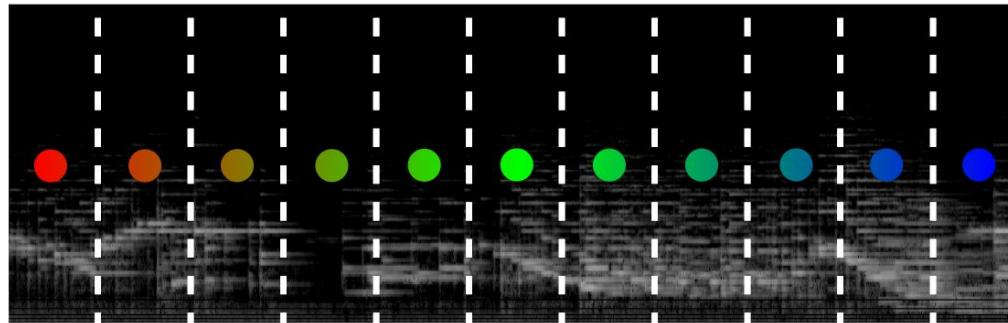
Ensure **coherent** parallelly generated **samples**



# Aligning Latent and Image Spaces to Connect the Unconnectable

ICCV 2021





Alignment

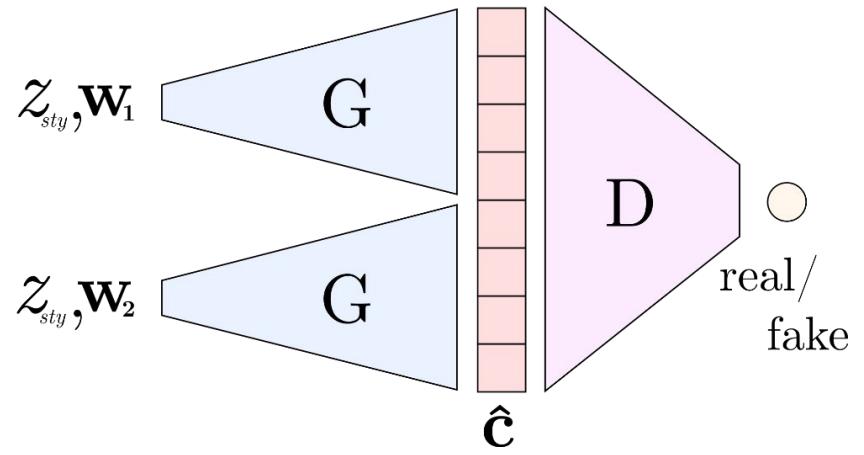
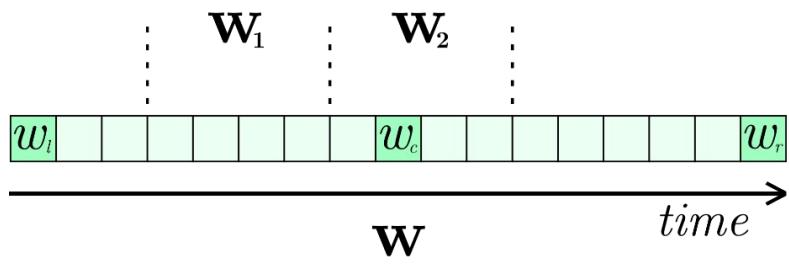


Interpolation

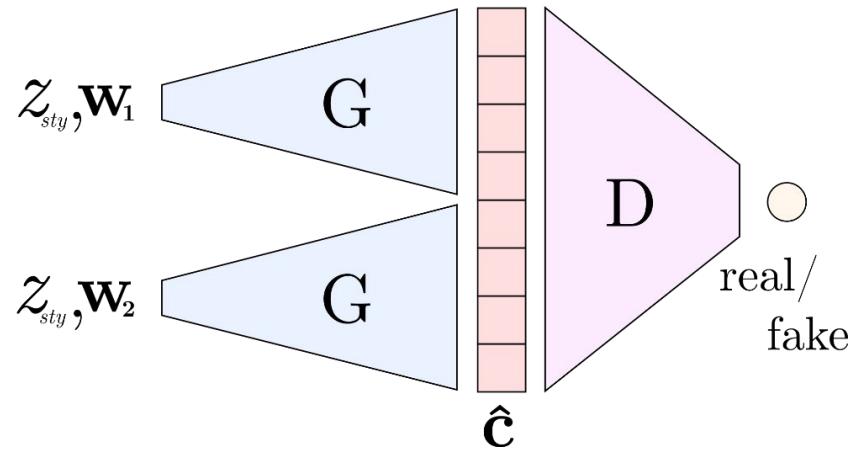
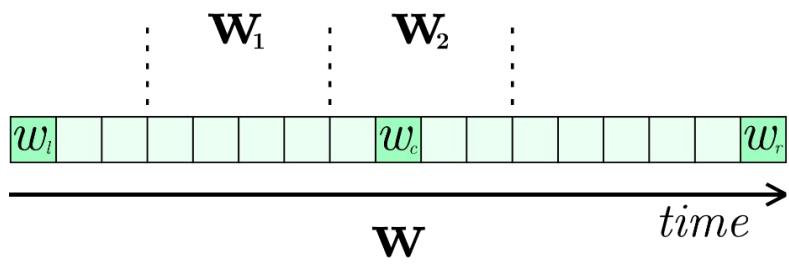


Anchors

Training **aligns latent coordinates** with generated **sequences**



Training **aligns latent coordinates** with generated **sequences**

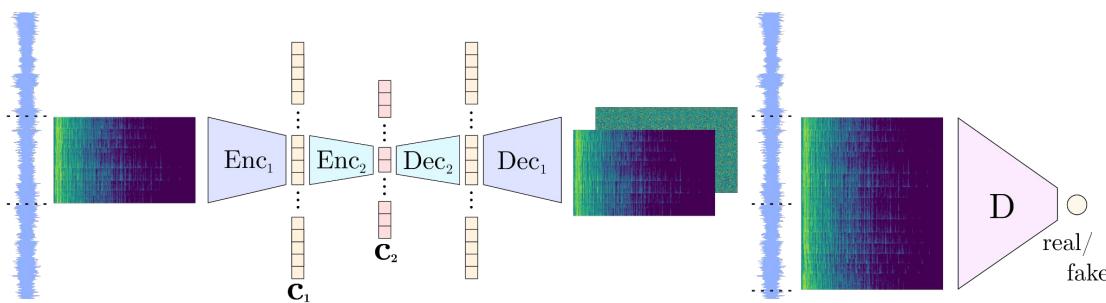


**Single random vector is used as conditioning for style**

# Implementation

**Autoencoder:**

Fully convolutional, **1D** Convs

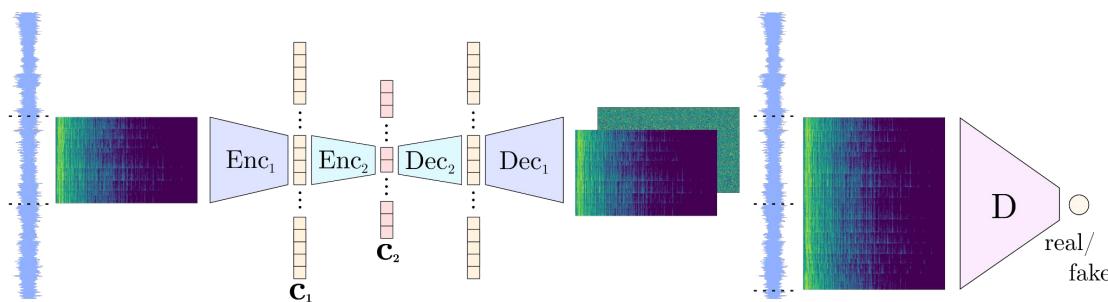


# Implementation

**Autoencoder:**

Fully convolutional, **1D Convs**

**High freq.** resolution spectrograms



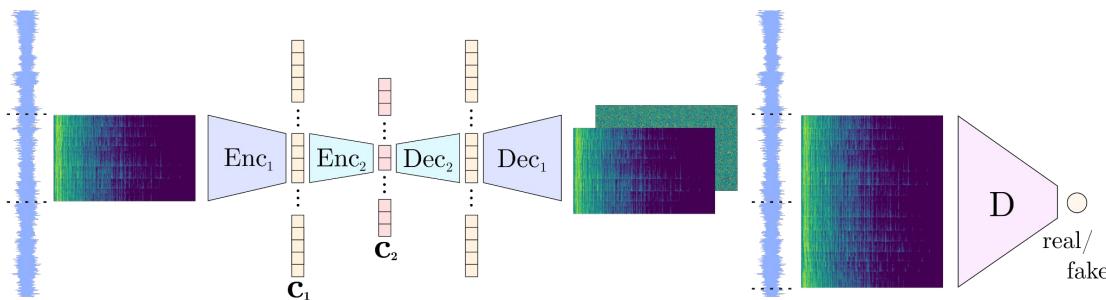
# Implementation

## Autoencoder:

Fully convolutional, **1D** Convs

High freq. resolution spectrograms

**Critic** with **2D** Convs



# Implementation

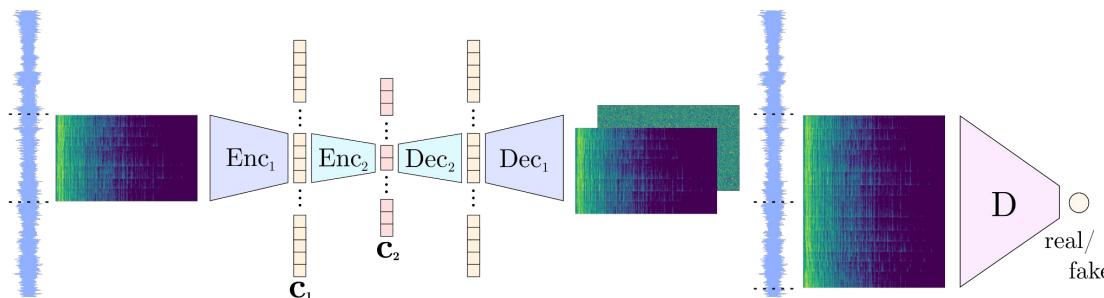
## Autoencoder:

Fully convolutional, **1D** Convs

High freq. resolution spectrograms

**Critic** with **2D** Convs

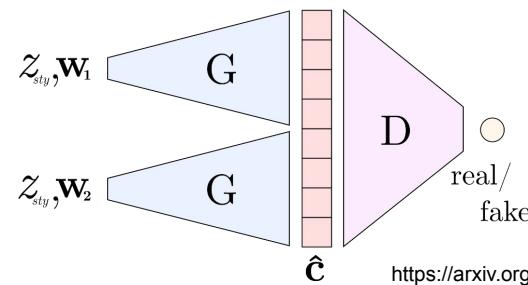
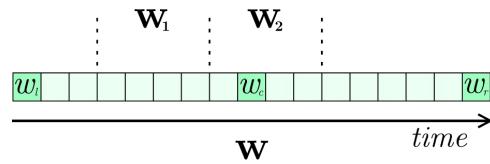
**Multi-scale** spectral distance to aid **reconstruction**



# Implementation

## Latent GAN:

FastGAN with **Residual 1D Conv. Blocks**

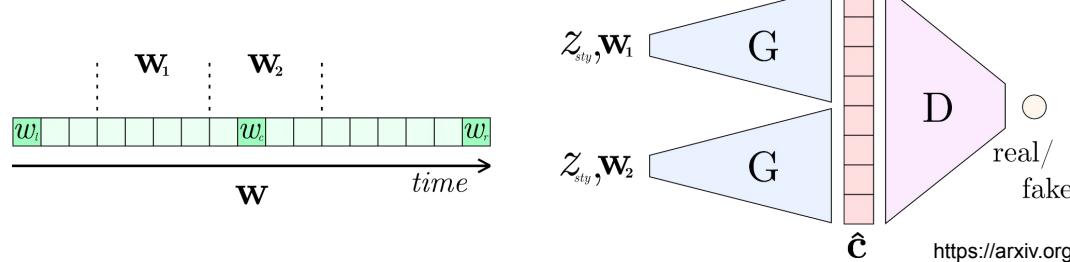


# Implementation

## Latent GAN:

FastGAN with Residual 1D Conv. Blocks

Spatial AdaIN with conditioning signal



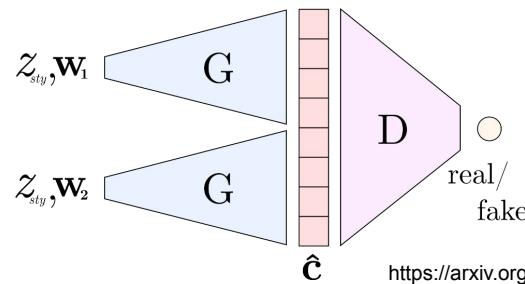
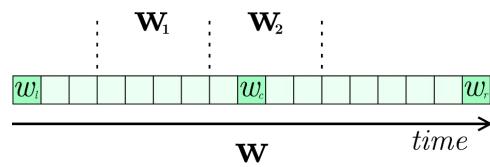
# Implementation

## Latent GAN:

FastGAN with **Residual 1D Conv. Blocks**

Spatial AdaIN with conditioning signal

**R1** Gradient Penalty



# Implementation

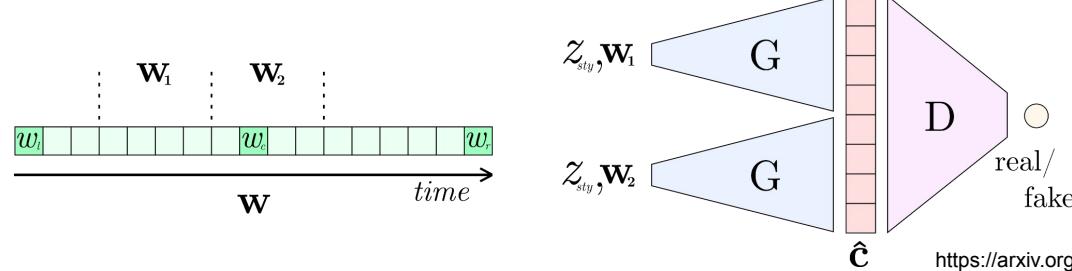
## Latent GAN:

FastGAN with **Residual 1D Conv. Blocks**

Spatial **AdaIN** with conditioning signal

**R1** Gradient Penalty

**Stereo representations stacked channel-wise**



## Experiments

**Autoencoder is trained on general music dataset (SXSW)**

# Experiments

**Autoencoder is trained on general music dataset (SXSW)**



## Experiments

Autoencoder is trained on general music dataset (SXSW)

It produces **genre-agnostic** compressed **representations!**

## Experiments

Autoencoder is trained on general music dataset (SXSW)

It produces genre-agnostic compressed representations!

We train a **Latent GAN** on:

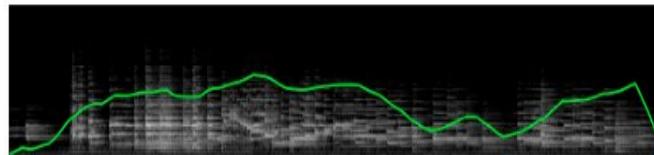
**Piano** Music dataset (*MAESTRO*), **note density** as conditioning

**Techno** Music dataset (*Jamendo*), **tempo** as conditioning

# Conditioning Signals

**Piano**

MadMom CNN **Onset** detector

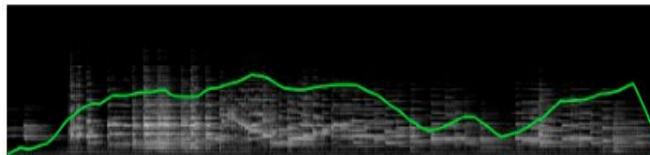


# Conditioning Signals

**Piano**

MadMom CNN Onset detector

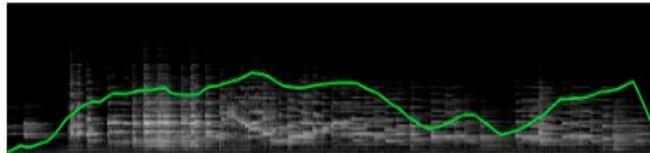
**Note density from Onsets (KDE)**



# Conditioning Signals

Piano

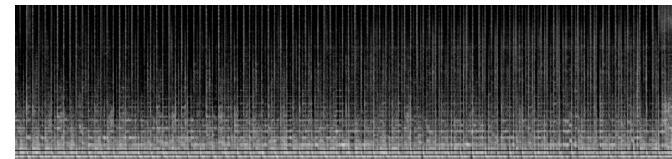
MadMom CNN Onset detector



Note density from Onsets (KDE)

Techno

Tempo-CNN to estimate **Tempo**



135 bpm

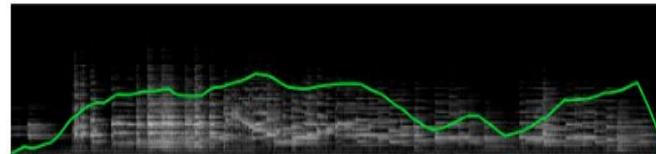
108

<https://arxiv.org/abs/1903.10839>

# Conditioning Signals

Piano

MadMom CNN Onset detector

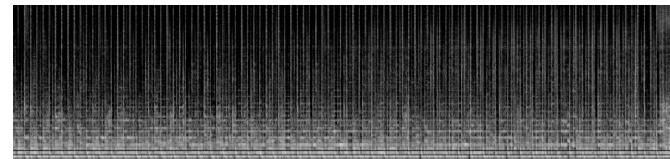


Note density from Onsets (KDE)

Techno

Tempo-CNN to estimate Tempo

Constant in training samples



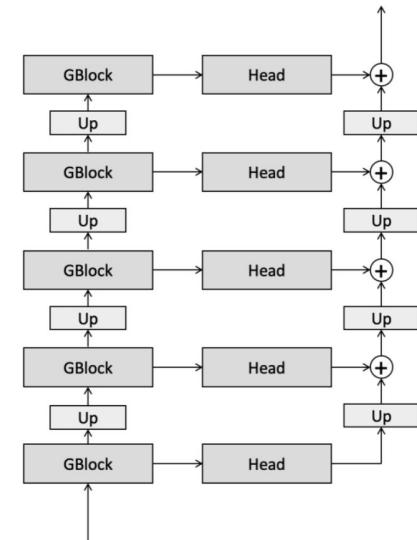
135 bpm

109

<https://arxiv.org/abs/1903.10839>

## Baseline: UNAGAN

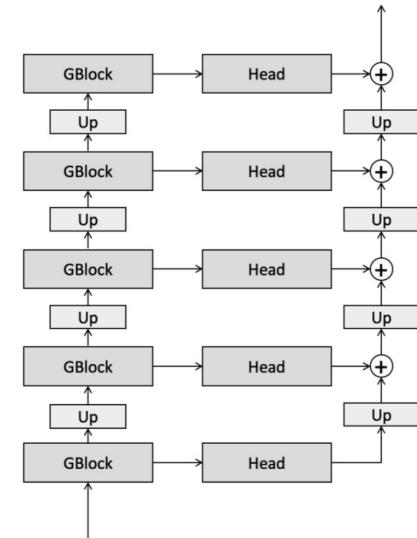
Sequence of **rand. vectors**  $\longrightarrow$  **Mel-spectrograms**



## Baseline: UNAGAN

Sequence of rand. vectors  $\longrightarrow$  Mel-spectrograms

Vectors drawn independently: **short context!**

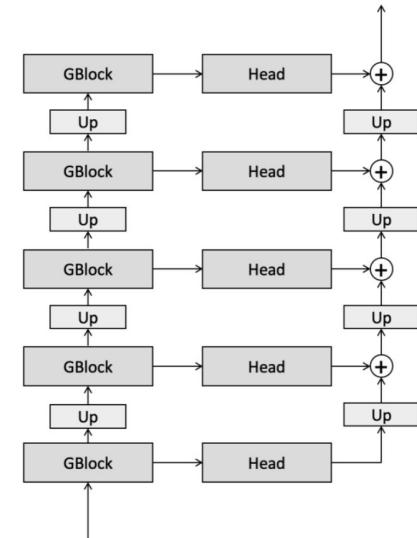


## Baseline: UNAGAN

Sequence of rand. vectors  $\longrightarrow$  Mel-spectrograms

Vectors drawn independently: **short context!**

**No parallel generation** of same sample!



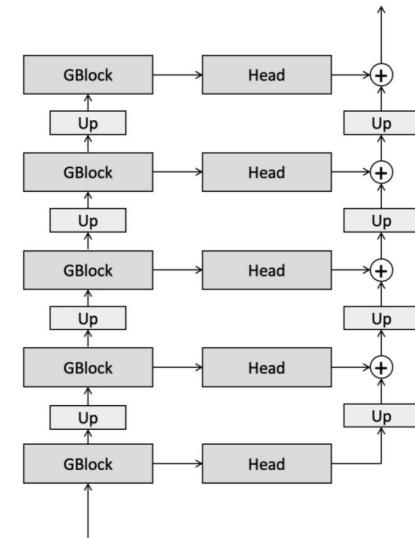
## Baseline: UNAGAN

Sequence of **rand. vectors** → **Mel-spectrograms**

Vectors drawn independently: **short context!**

No parallel generation of same sample!

Relies on **separate vocoder**



# Results



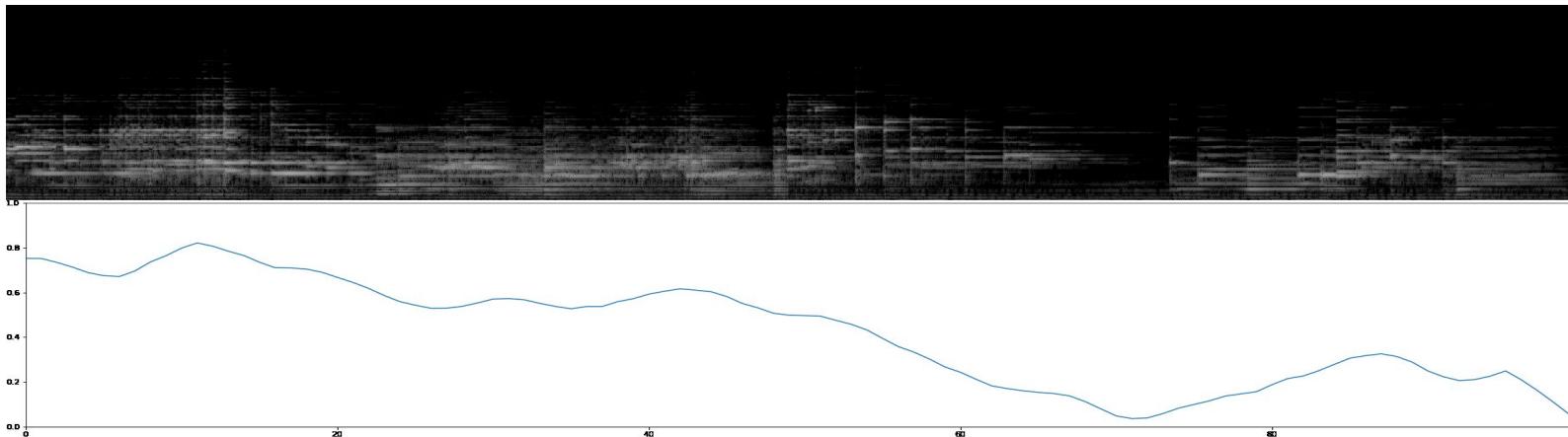
UNAGAN Piano sample (**Baseline**)

# Results



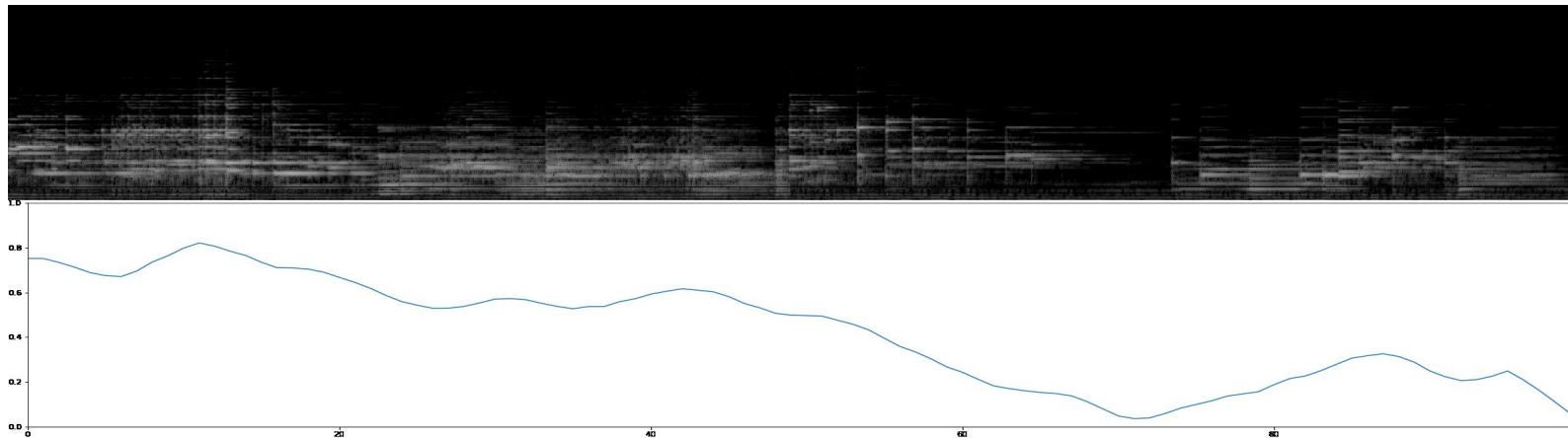
UNAGAN Piano sample (**Baseline**)

# Results



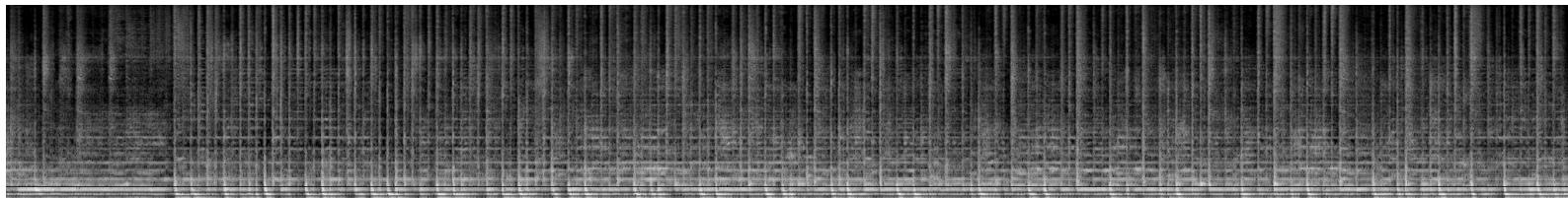
**Musika Piano** (Rand. conditioning)

# Results



**Musika Piano** (Rand. conditioning)

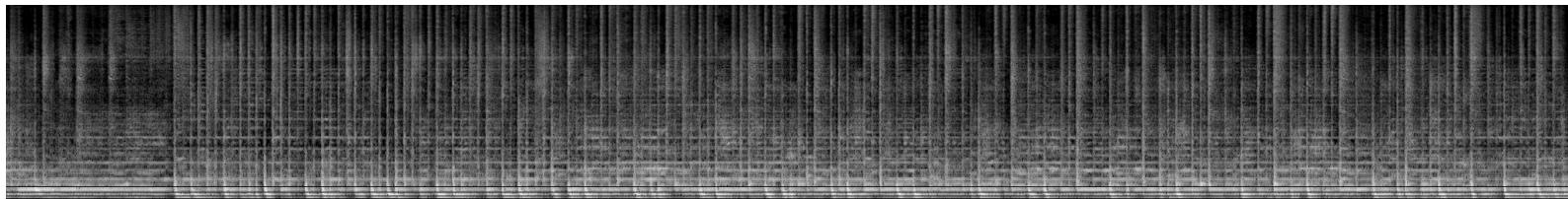
# Results



120 bpm

**Musika Techno**

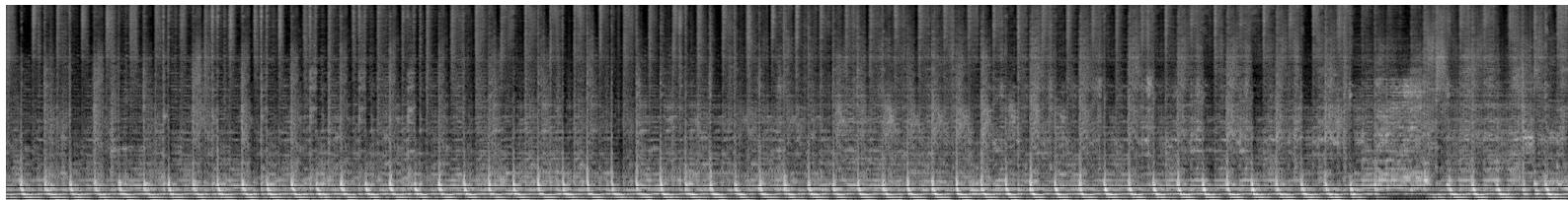
# Results



120 bpm

Musika **Techno**

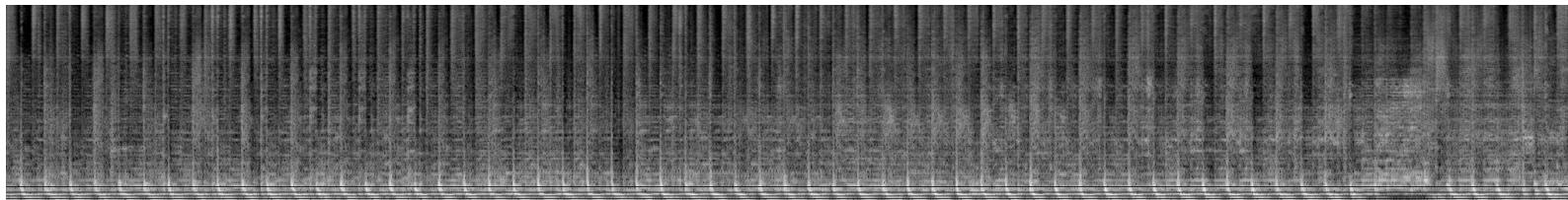
# Results



160 bpm

**Musika Techno**

# Results



160 bpm

**Musika Techno**

## Generation Speed

Model (Faster than real-time)	GPU	CPU
Musika Uncond. Piano	972x	<b>40x</b> 
Musika Cond. Piano	921x	<b>40x</b>
UNAGAN [20] Piano	28x	11x 
Musika Uncond. Techno	<b>994x</b>	39x
Musika Cond. Techno	917x	39x

**Much faster** than UNAGAN!

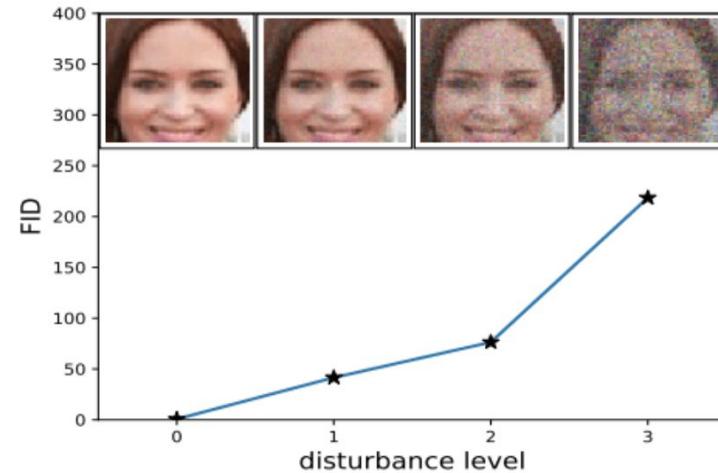
# Evaluating GANs

Years of **GANs** research **without metric!**

# Evaluating GANs

Years of **GANs** research **without metric!**

For images: IS, **FID**, KID

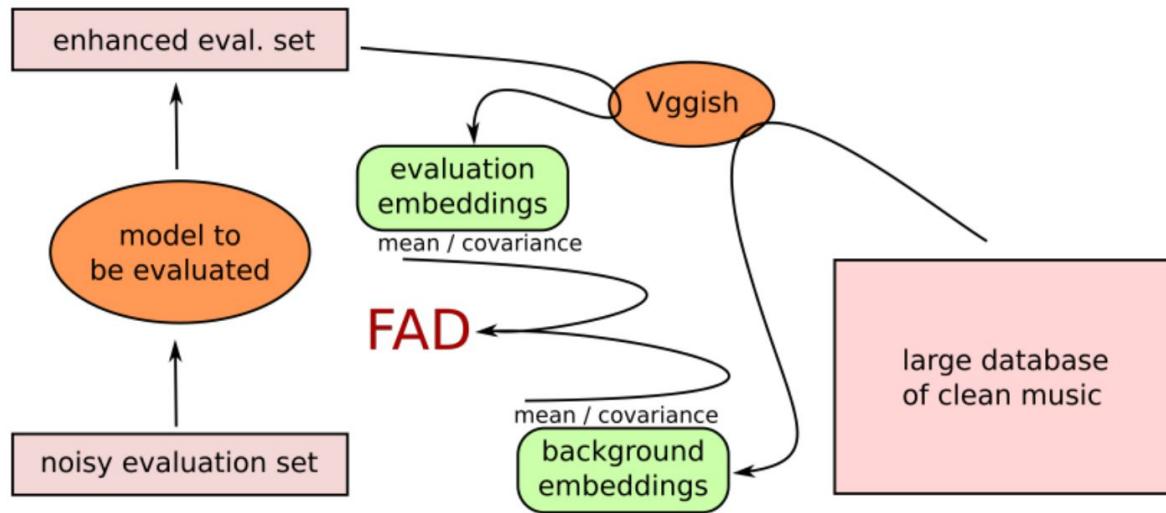


# Evaluating GANs

Years of **GANs** research **without metric!**

For images: IS, **FID**, KID

For audio: **FAD** (Frechét Audio Distance)



$$\mathbf{F}(\mathcal{N}_b, \mathcal{N}_e) = \|\mu_b - \mu_e\|^2 + \text{tr}(\Sigma_b + \Sigma_e - 2\sqrt{\Sigma_b \Sigma_e})$$

## FAD Problems

**Calculated with 1 second snippets**

## FAD Problems

Calculated with **1 second snippets**

Evaluates sound **quality, not long-range coherence**

## FAD Problems

Calculated with **1 second snippets**

Evaluates sound **quality**, not long-range coherence

**No quantitative metrics** currently exist!

---

<b>Model</b>	<b>FAD</b>
Musika Uncond. Piano	<b>1.641</b> 
Musika Cond. Piano Rand.	2.150
Musika Cond. Piano Const. 0.15	2.584
Musika Cond. Piano Const. 0.30	3.400
Musika Cond. Piano Const. 0.45	4.389
Musika Cond. Piano Const. 0.60	4.839
Musika Cond. Piano Const. 0.75	5.434
UNAGAN [20] Piano	11.183 

---

**Lower Frechét Audio Distance than UNAGAN!**

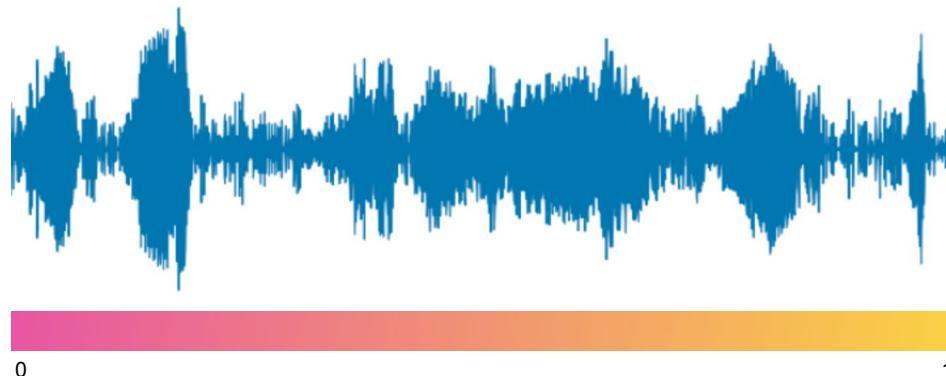
## Future/Current Work

22.05 kHz            44.1 kHz    **Not Trivial!**

## Future/Current Work

22.05 kHz → 44.1 kHz **Not Trivial!**

**Positional** conditioning for coherent **Song Structure**

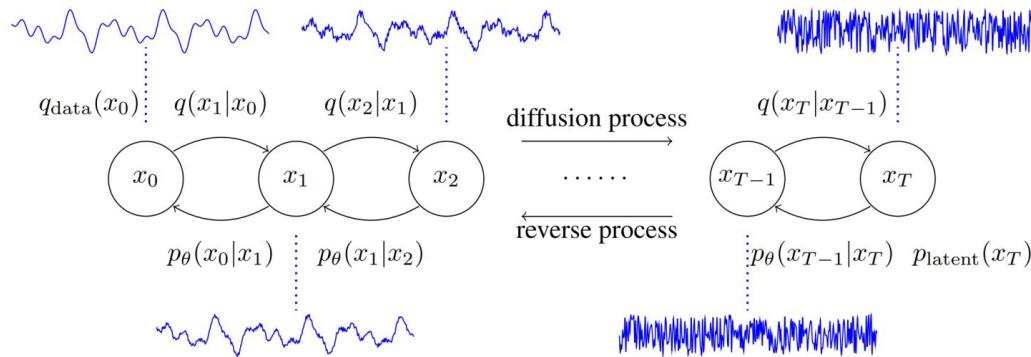


# Future/Current Work

GANs



Diffusion Models (Better with more data)



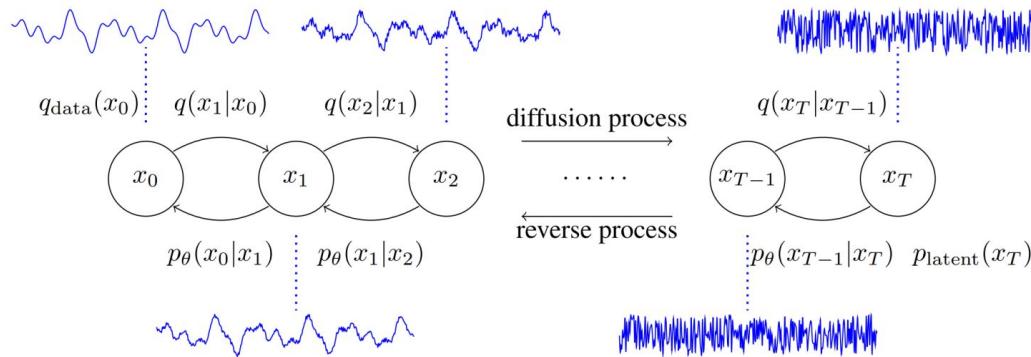
## Future/Current Work

GANs



Diffusion Models (Better with more data)

**Arbitrary length parallel generation not possible!**



# Reception

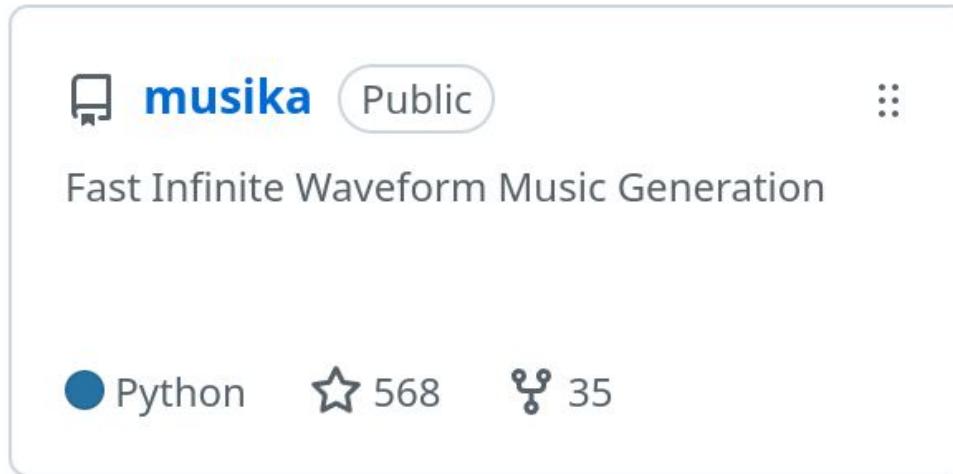
# Reception

The screenshot shows the musika! demo interface on a web browser. At the top, there's a header with navigation links: "App", "Files and versions", and "Community". Below the header, the title "musika!" is displayed in a large, bold font. A subtitle reads: "Blazingly Fast 44.1 kHz Stereo Waveform Music Generation of Arbitrary Length. Be patient and enjoy the weirdness!". On the left, there's a form for generating music. It includes a section for "Music Genre to Generate" with three radio buttons: "Techno/Experimental" (selected), "Death Metal (finetuned)", and "Misc". Another section for "Generated Music Length" has three radio buttons: "23s", "1m 58s" (selected), and "3m 57s". A slider allows users to "How much do you want the music style to be varied? (Stddev truncation for random vectors)" with a value of 1.8. At the bottom of the form are "Clear" and "Submit" buttons. To the right, there's a preview area. It shows a spectrogram titled "Log-MelSpectrogram of Generated Audio (first 23 s)". Below it is a player titled "Generated Audio" showing a progress bar at 0:00 / 1:58. The player includes standard controls like play/pause, volume, and settings.

Original work by Marco Pasini ([Twitter](#)) at the Institute of Computational Perception, JKU Linz. Supervised by Jan Schlüter.

## Popular Online Demo!

# Reception



Popular **Github** repository!

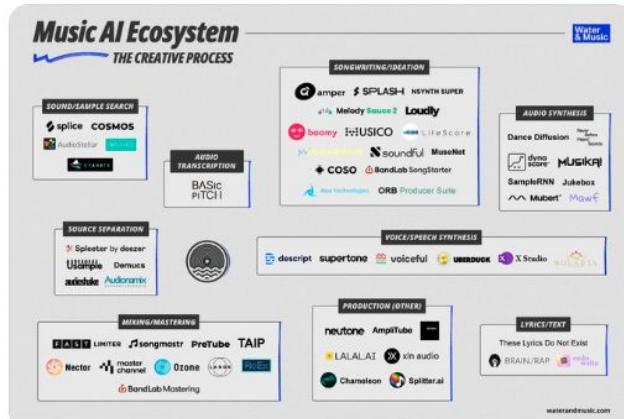
# Reception



Water & Music @water\_and\_music · 1 nov

we've seen lots of generative AI market maps making the rounds on Twitter. a huge gap: none of them cover music with the same amount of breadth as visual art and text.

so, we made our own 🤖 here's a market map of 60+ AI tools for musicians, across the entire creative process:



38

203

717



## AI Music tools Market Map

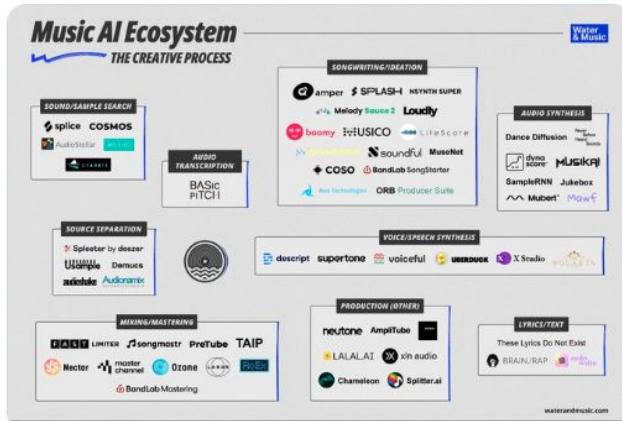
# Reception



Water & Music @water\_and\_music · 1 nov

we've seen lots of generative AI market maps making the rounds on Twitter. a huge gap: none of them cover music with the same amount of breadth as visual art and text.

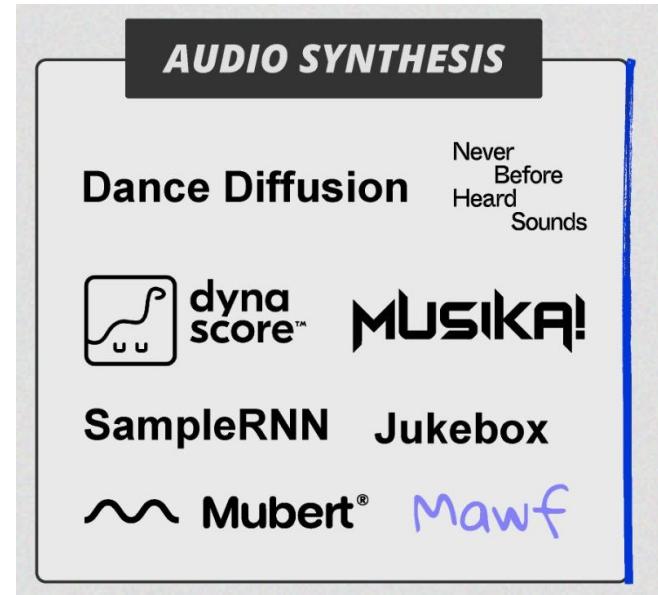
so, we made our own 🤖 here's a market map of 60+ AI tools for musicians, across the entire creative process:



38

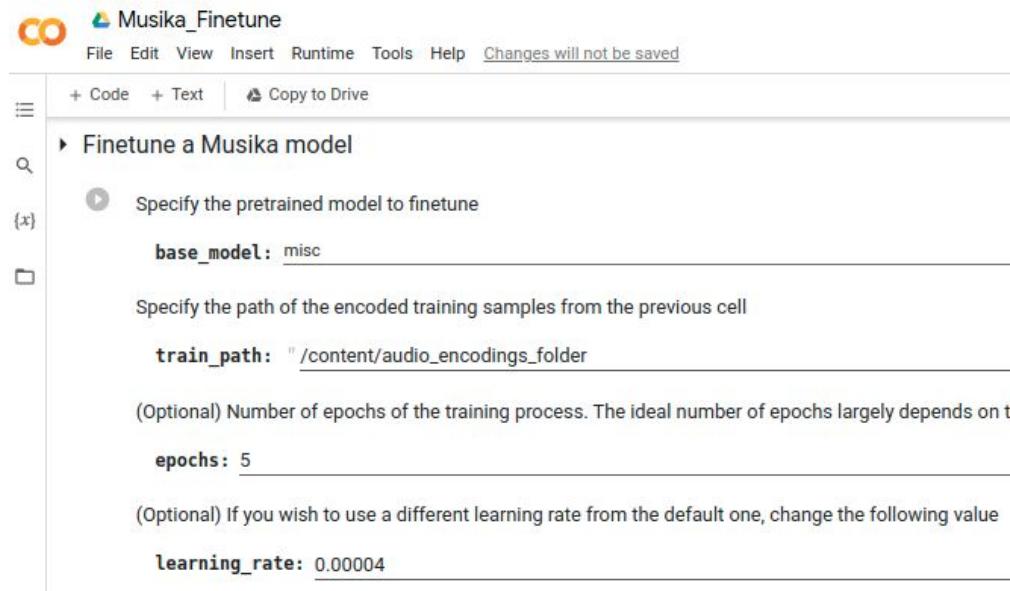
203

717



## AI Music tools Market Map

# Ecosystem

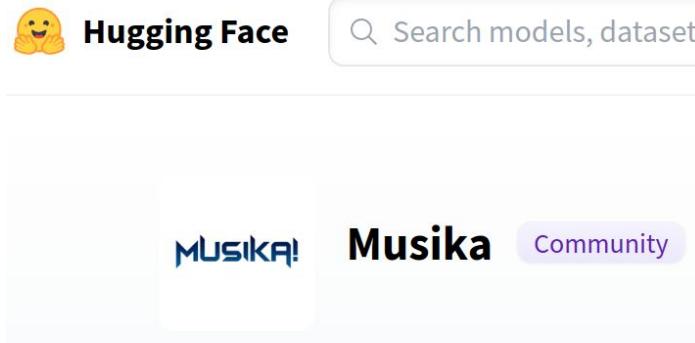


The screenshot shows a Jupyter Notebook interface with the title "Musika\_Finetune". The notebook contains the following configuration parameters:

- base\_model:** misc
- train\_path:** "/content/audio\_encodings\_folder"
- epochs:** 5
- learning\_rate:** 0.00004

**Custom Training made Easy!**

# Ecosystem



**Share your custom Model!**

# Ecosystem



Hugging Face

Search models, datasets

MUSIKA! Musika

Community

## Models 15

▲ Collapse

↑↓ Sort: Recently Updated

musika/musika-acoustics [private]

Updated 8 days ago

musika/musika-early-dm [private]

Updated 8 days ago

musika/musika-dubstep [private]

Updated 16 days ago

musika/musika-jazz-40-epochs [private]

Updated 18 days ago

musika/musika-jazz-15-epochs [private]

Updated 20 days ago

musika/musika-anime-songs

Updated 20 days ago

musika/musika-techno-ssib [private]

Updated 21 days ago

musika/musika-metal-7aw [private]

Updated 21 days ago

musika/musika-irish-jigs

Updated 21 days ago • 1

musika/bd [private]

Updated 22 days ago

musika/my [private]

Updated 23 days ago

musika/musika-halvany\_oszi\_rozsa

Updated 23 days ago

musika/musika-s3rl-happy-hardcore

Updated 24 days ago • 3

musika/musika\_misc

Updated 27 days ago

## Share your custom Model!

# Ecosystem

 Musika\_Finetune

File Edit View Insert Runtime Tools Help Changes will not be saved

+ Code + Text ⌘ Copy to Drive

▶ Finetune a Musika model

Specify the pretrained model to finetune  
base\_model: misc

Specify the path of the encoded training samples from the previous cell  
train\_path: "/content/audio\_encodings\_folder"

(Optional) Number of epochs of the training process. The ideal number of epochs largely depends on the dataset.  
epochs: 5

(Optional) If you wish to use a different learning rate from the default one, change the following variable.  
learning\_rate: 0.00004

Train

Share

 Musika\_Inference

File Edit View Insert Runtime Tools Help Cannot save changes

+ Code + Text ⌘ Copy to Drive

▶ Test model

A Gradio interface will be created to test your chosen model. Click on the public URL to access it.

Specify a model id from the [Musika Library](#).

model\_id: "musika/musika\_techno"

Test



Search models, datasets

MUSIKA

Musika

Community

Streamlined process

## Conclusion

**Solving a technical challenge:**

**Fast music generation of sufficient quality, conditioned on user input**

## Conclusion

**Solving a technical challenge:**

**Fast music generation of sufficient quality, conditioned on user input**

We **release code** and weights:

**A system can be trained on a new music domain in hours!**

## **What can Musika generate today?**

(44.1 kHz +  Structure +  Sound Quality)

MUSIKAI

# Thank you!

Huge thanks to Jan Schlüter  
and to the Institute of Comp. Perception for the support!