

PAT 464/564 (Winter 2026)

# Generative AI for Music & Audio Creation

## **Lecture 12: Variational Autoencoders**

Instructor: Hao-Wen Dong

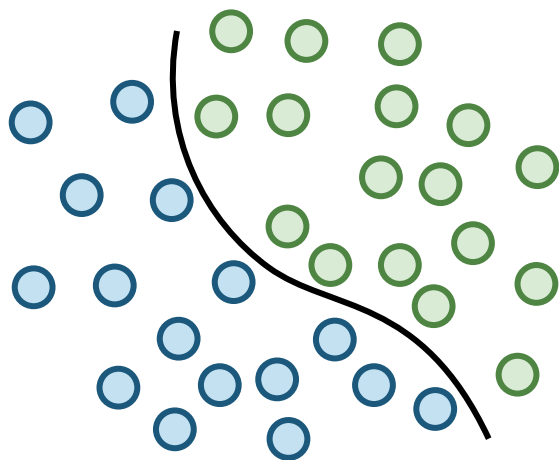
# Representative Types of Deep Generative Models

- **Deep autoregressive models**
  - Recurrent neural network (RNN)
  - Long short-term memory (LSTM)
  - Transformer model
- **Deep latent variable models**
  - Variational autoencoder (VAE) **Today's topic!**
  - Generative adversarial network (GAN)
  - Diffusion model
  - Flow-based model
- *And many others...*

# Deep Latent Variable Models

# Discriminative vs Generative Models

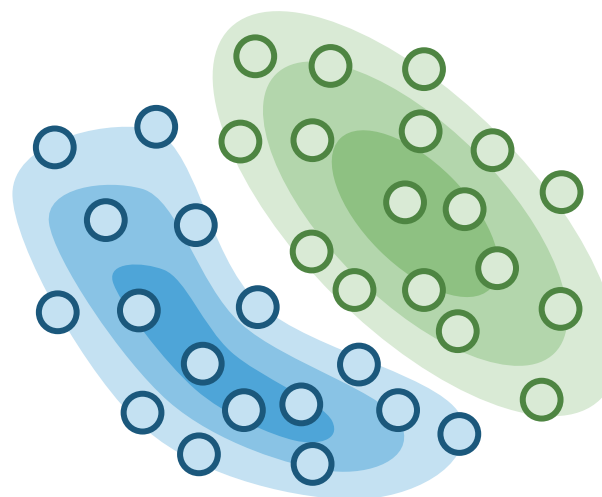
**Discriminative**



**Discriminative models learn the decision boundary**

$$P(y|x)$$

**Generative**



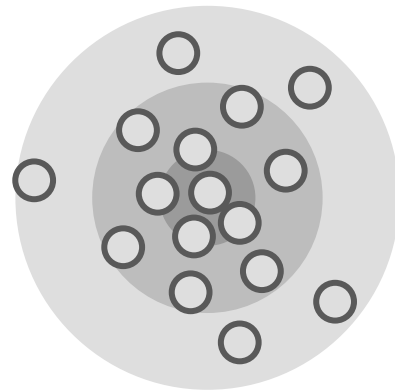
**Generative models learn the underlying distribution**

$$P(x) \text{ or } P(x|y)$$

# Deep Latent Variable Models

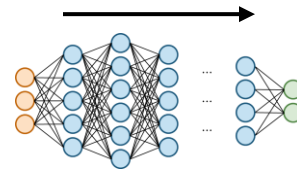
- **Intuition:** Learn to map a known distribution to the data distribution

Known distribution

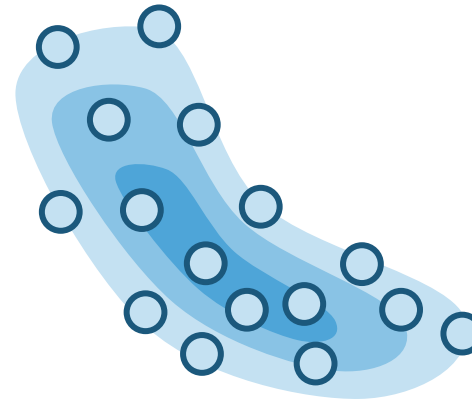


$P(z)$

$P(x | z)$



Data distribution

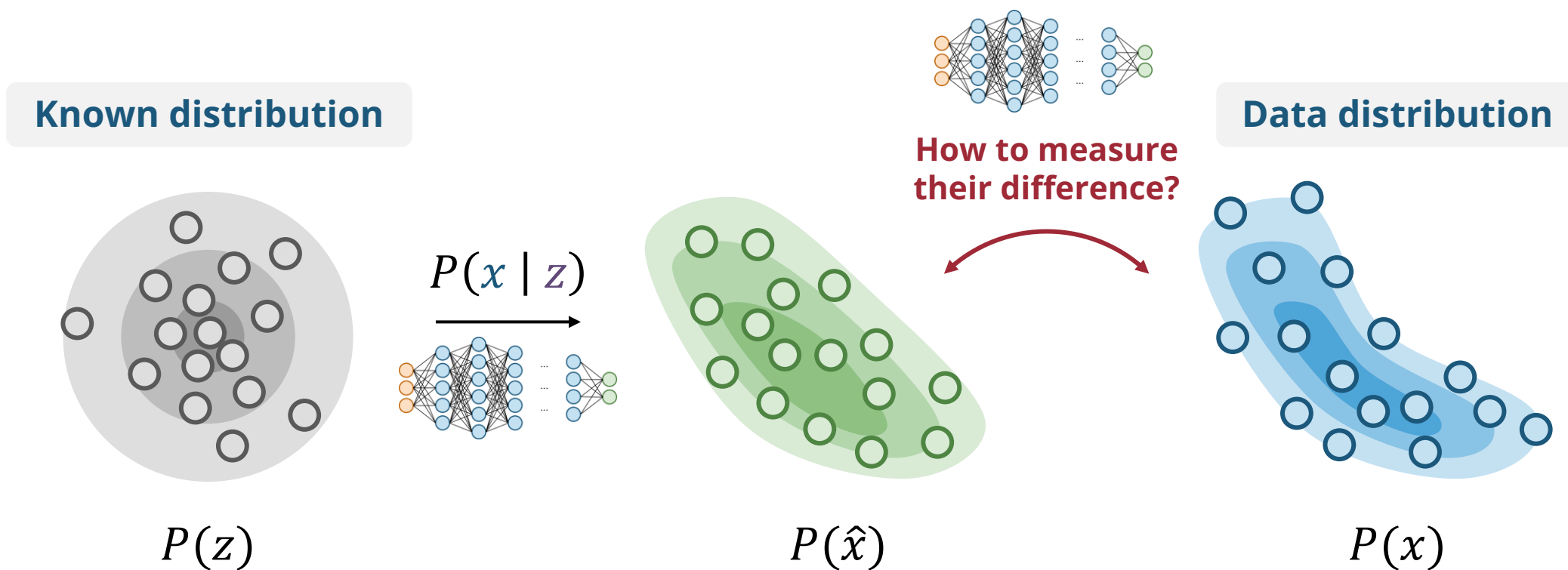


$P(x)$

$$P(x) = P(z) P(x | z)$$

# Deep Latent Variable Models

- **Intuition:** Learn to map a known distribution to the data distribution



# Deep Latent Variable Models

- **Intuition:** Learn to map a known distribution to the data distribution

What we want the model to learn!

$$P(x) = P(z) P(x | z)$$

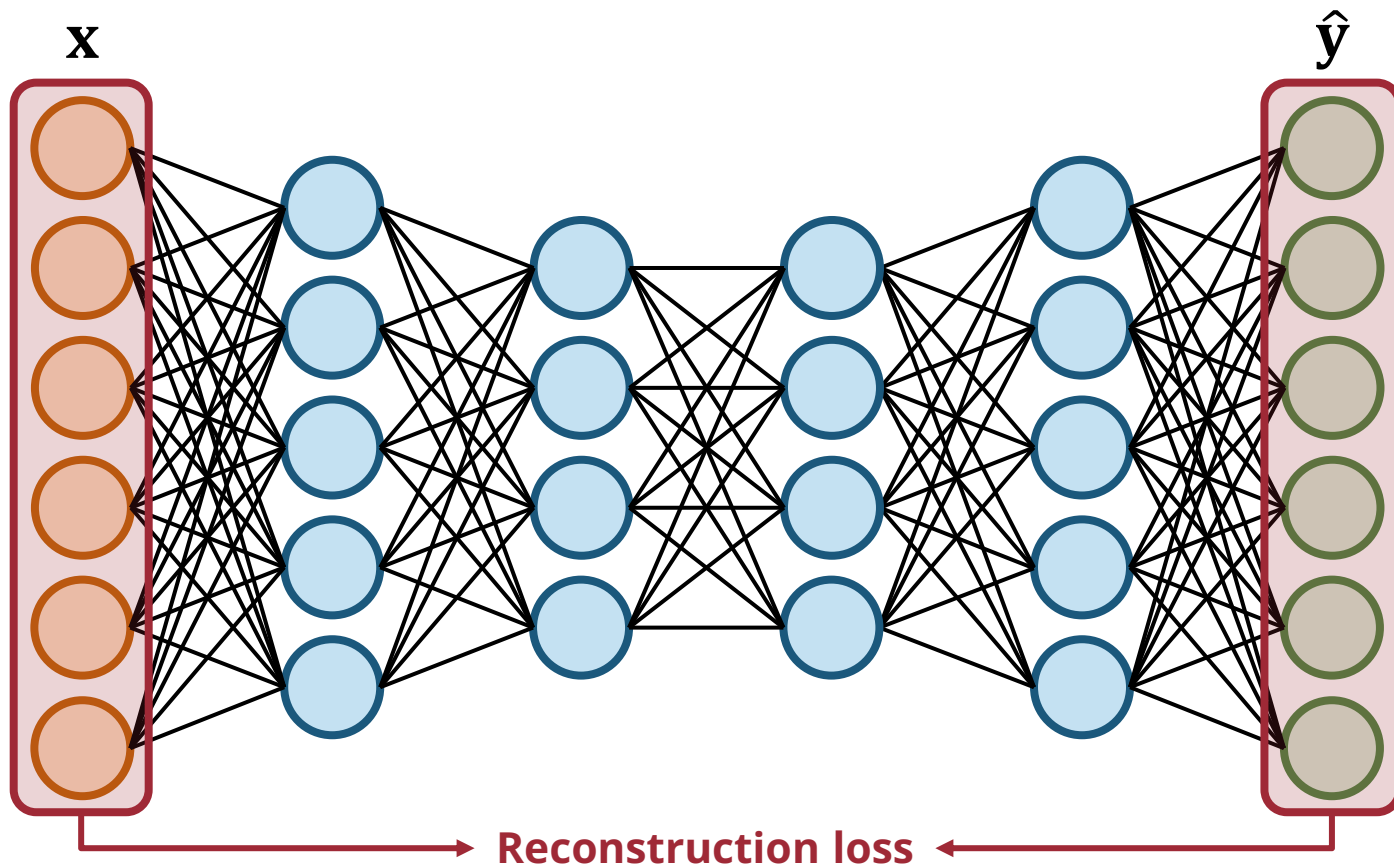
Diagram illustrating the decomposition of the data distribution  $P(x)$  into the latent distribution  $P(z)$  and the conditional distribution  $P(x | z)$ .

The equation  $P(x) = P(z) P(x | z)$  is shown. The term  $P(x | z)$  is highlighted with a red box. An upward arrow points from the boxed term to the text "What we want the model to learn!". A blue arrow points from  $P(x)$  to the label "Data distribution". A purple arrow points from  $P(z)$  to the label "Latent distribution".

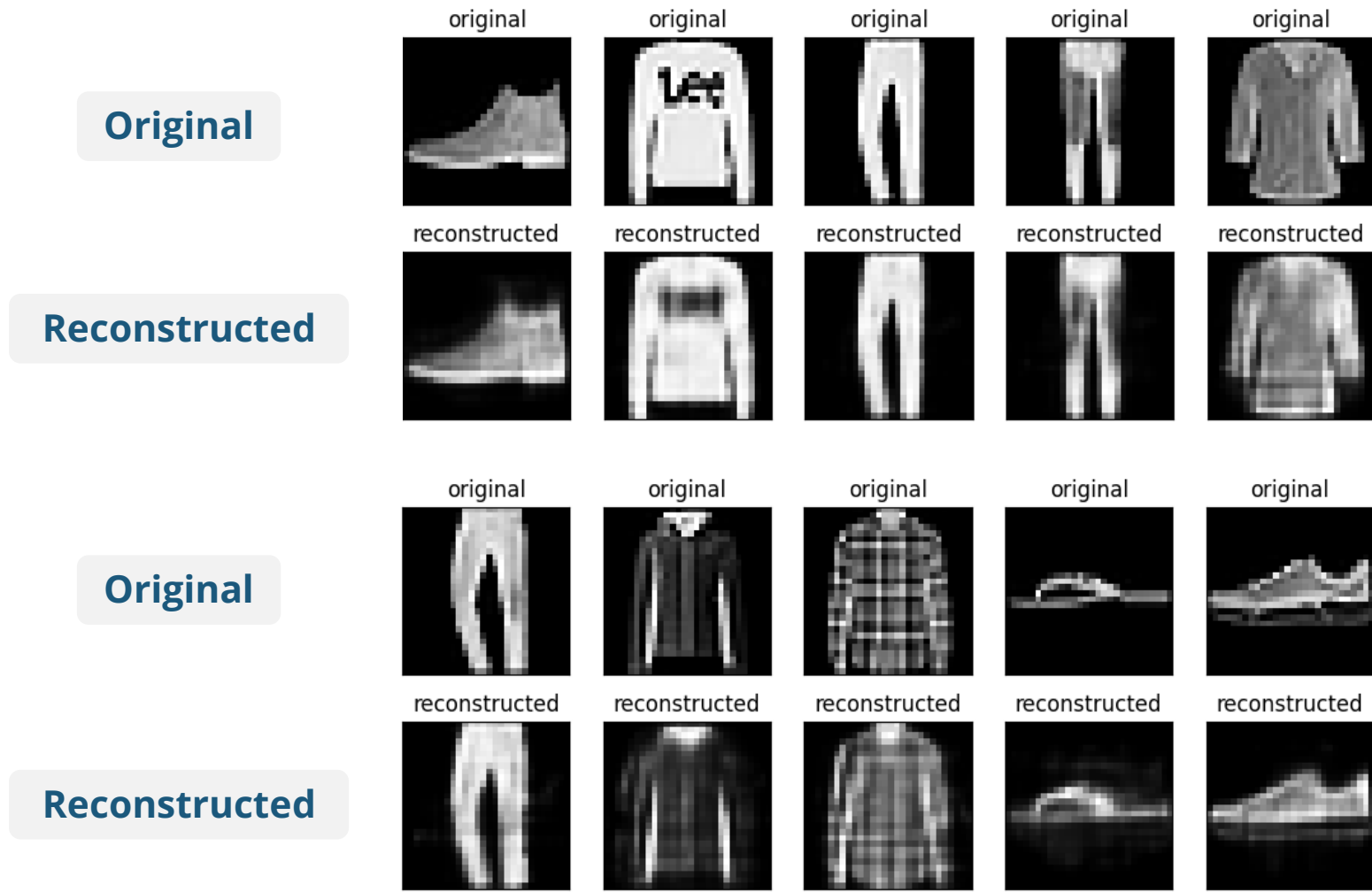
# Autoencoders

# Autoencoders

- A neural network where the **input and output are the same**



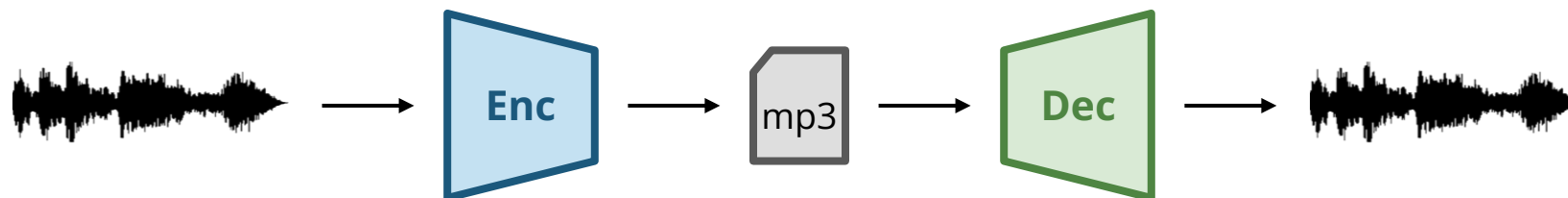
# Autoencoders: Reconstruction Examples



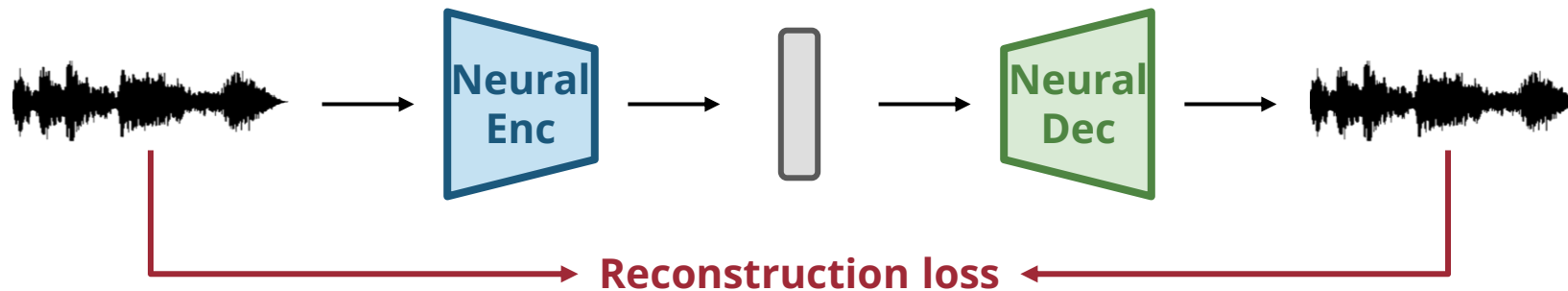
(Source: tensorflow.org)

# Codec is an Autoencoder

## Traditional Codec

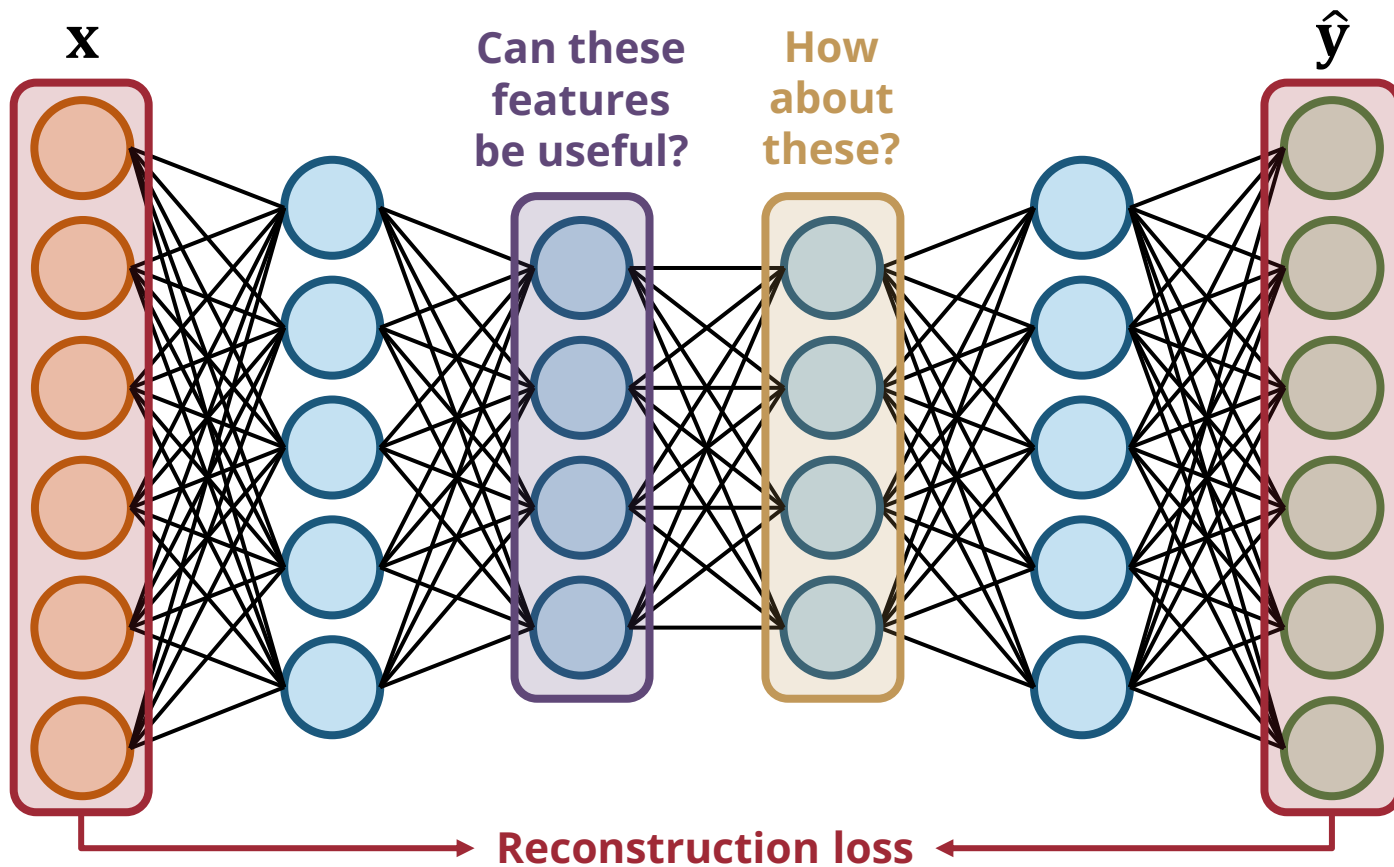


## Neural Codec



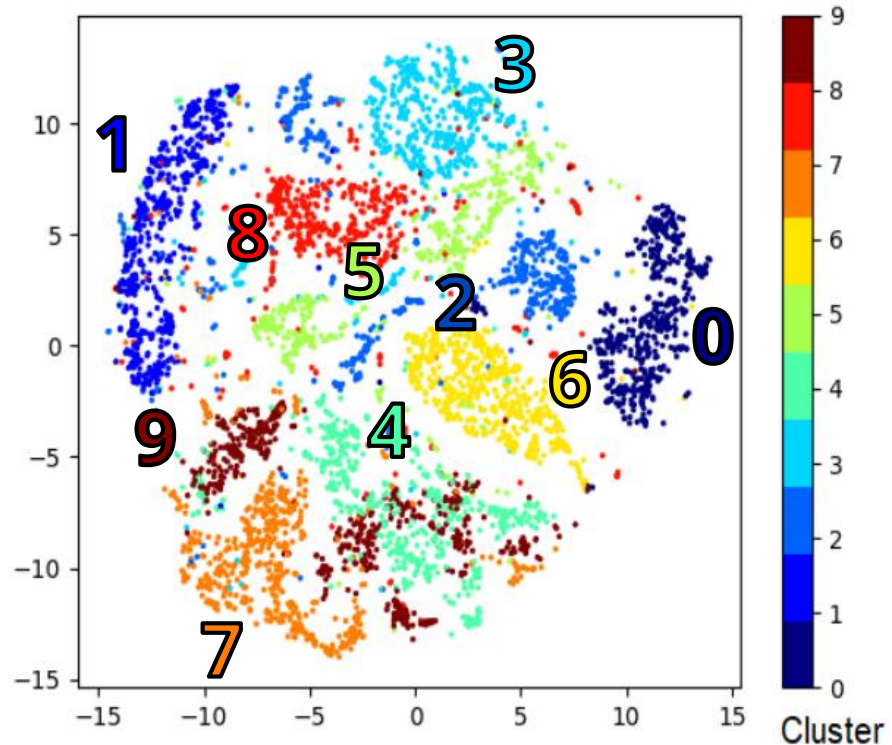
# Autoencoders

- A neural network where the **input and output are the same**



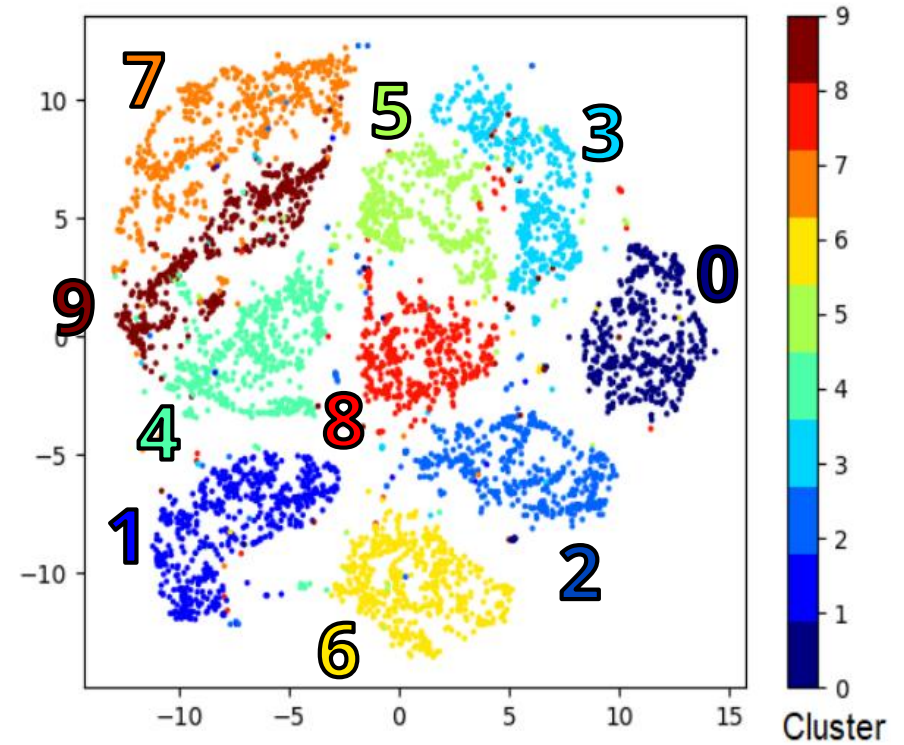
# Unsupervised Clustering with an Autoencoder

Raw pixels



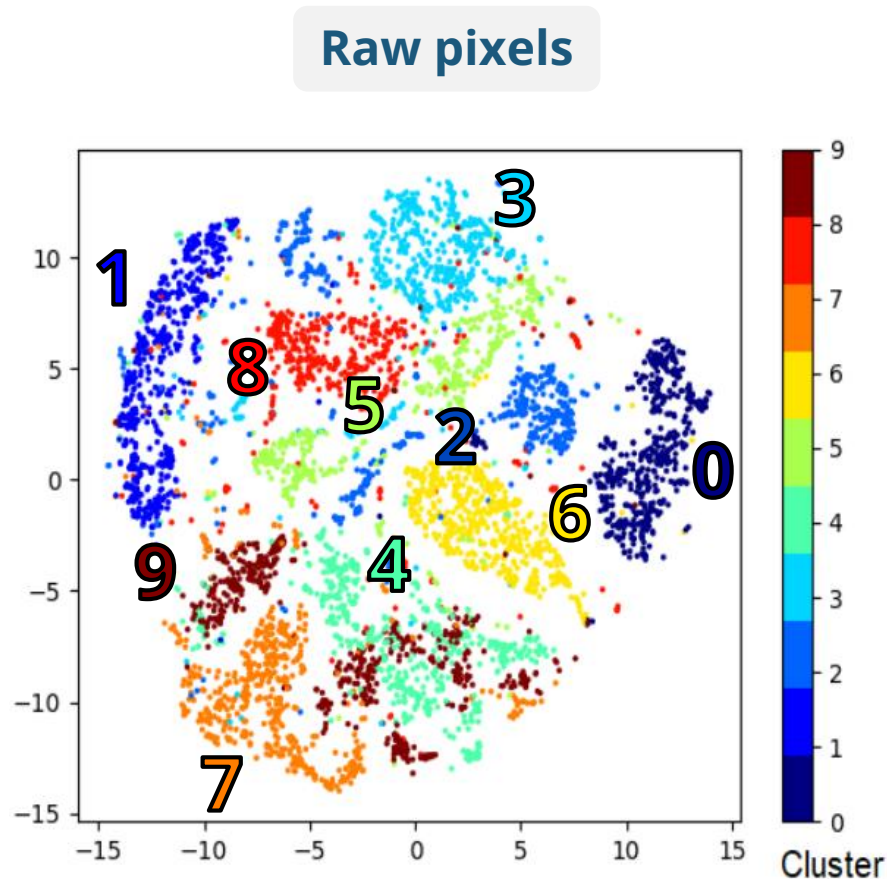
(Source: Aljalbout et al., 2020)

AE-encoded

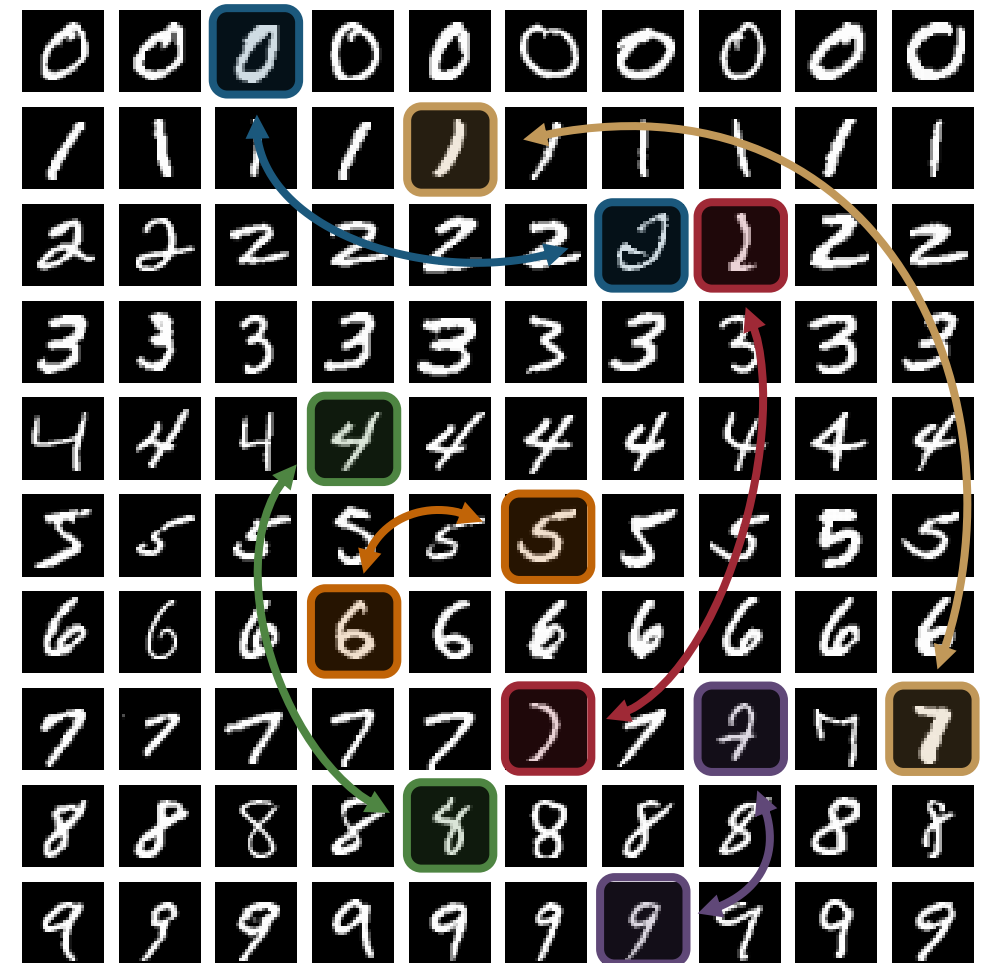


(Source: Aljalbout et al., 2020)

# Unsupervised Clustering with an Autoencoder



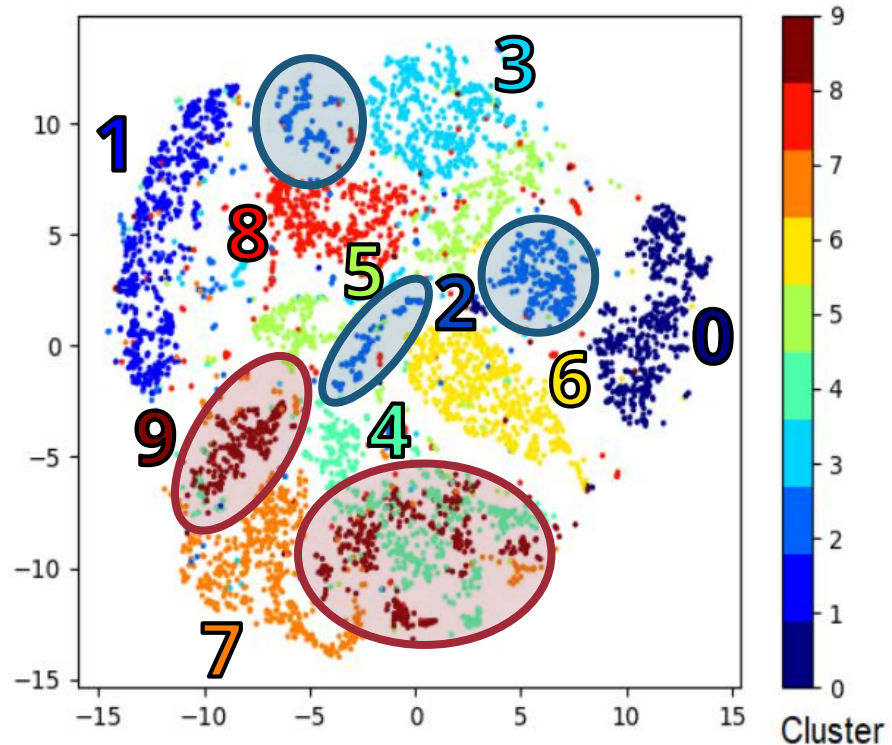
(Source: Aljalbout et al., 2020)



Elie Aljalbout, Vladimir Golkov, Yawar Siddiqui, Maximilian Strobel, and Daniel Cremers, "Clustering with Deep Learning: Taxonomy and New Methods," *arXiv preprint arXiv:1801.07648*, 2018.

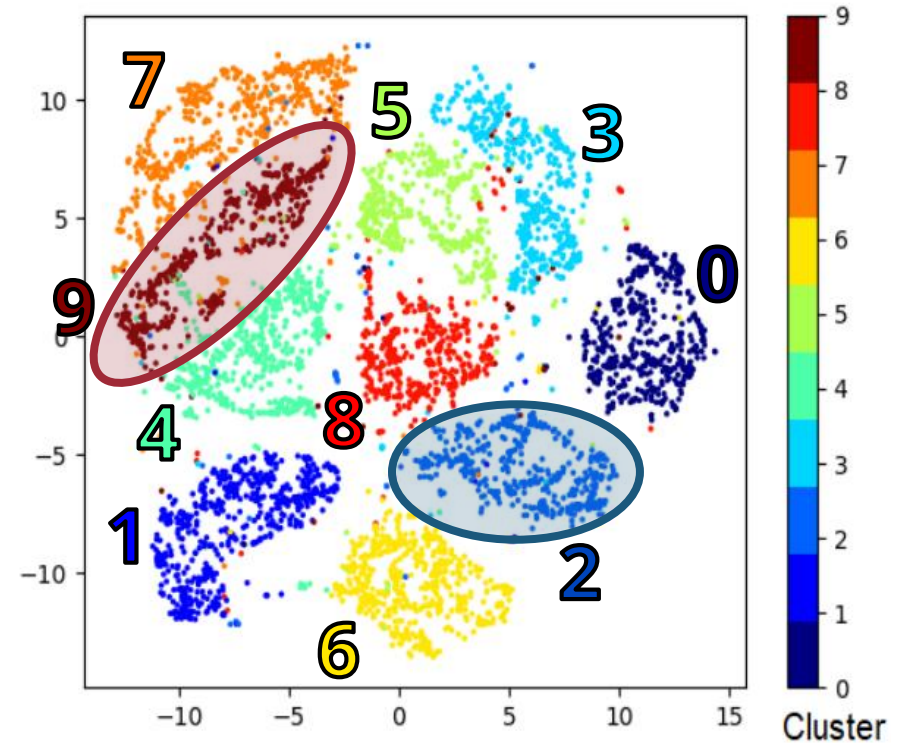
# Unsupervised Clustering with an Autoencoder

Raw pixels



(Source: Aljalbout et al., 2020)

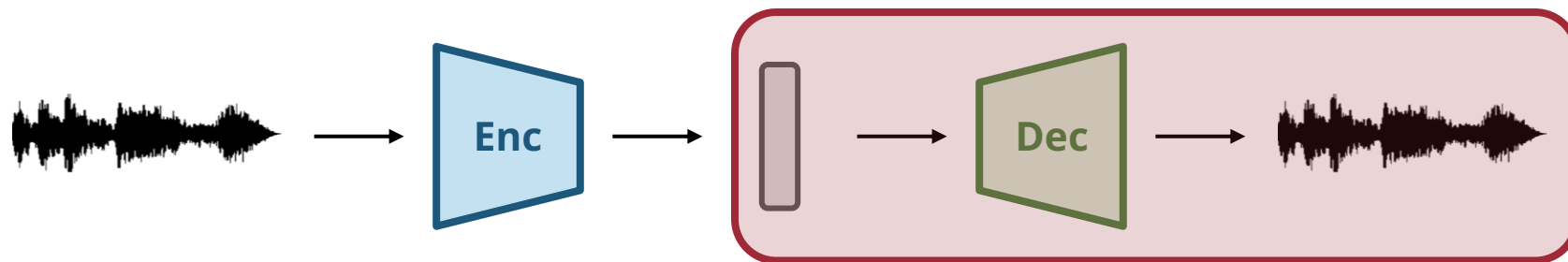
AE-encoded



(Source: Aljalbout et al., 2020)

# Variational Autoencoder (VAE)

# Autoencoder

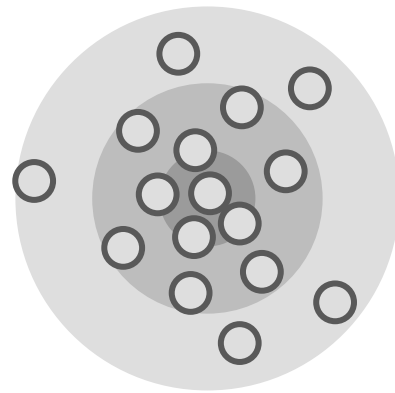


**Isn't this like a generative model?**

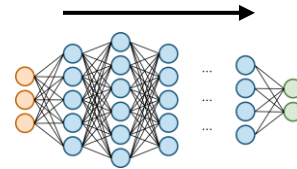
**What exactly is a generative model?**

# Generating Data from a Random Distribution

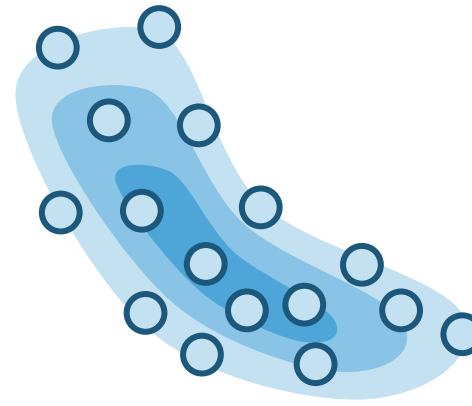
Random distribution



$P(z)$



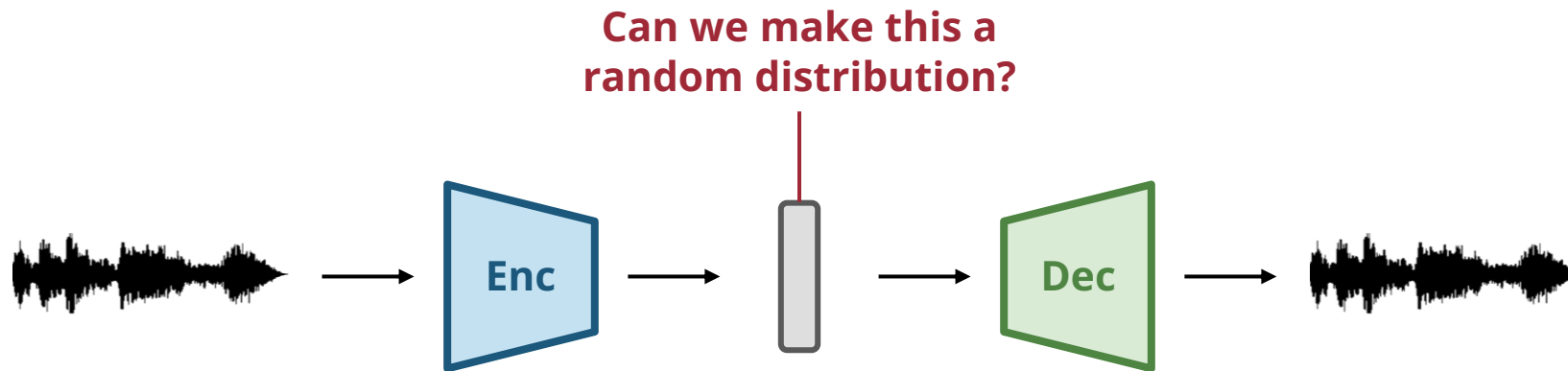
Data distribution



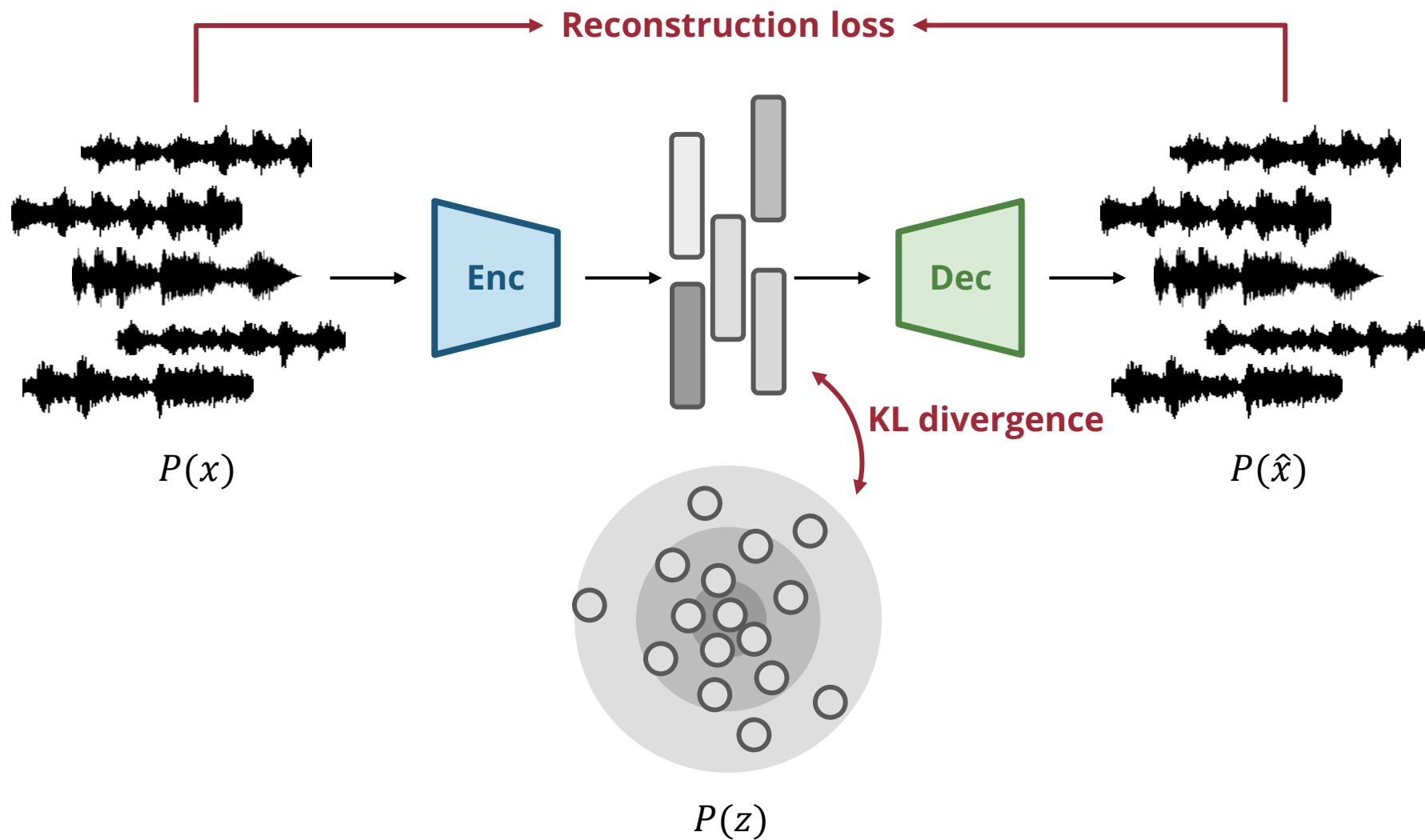
$P(x)$

If we can learn this mapping, we can then generate new samples from the data distribution

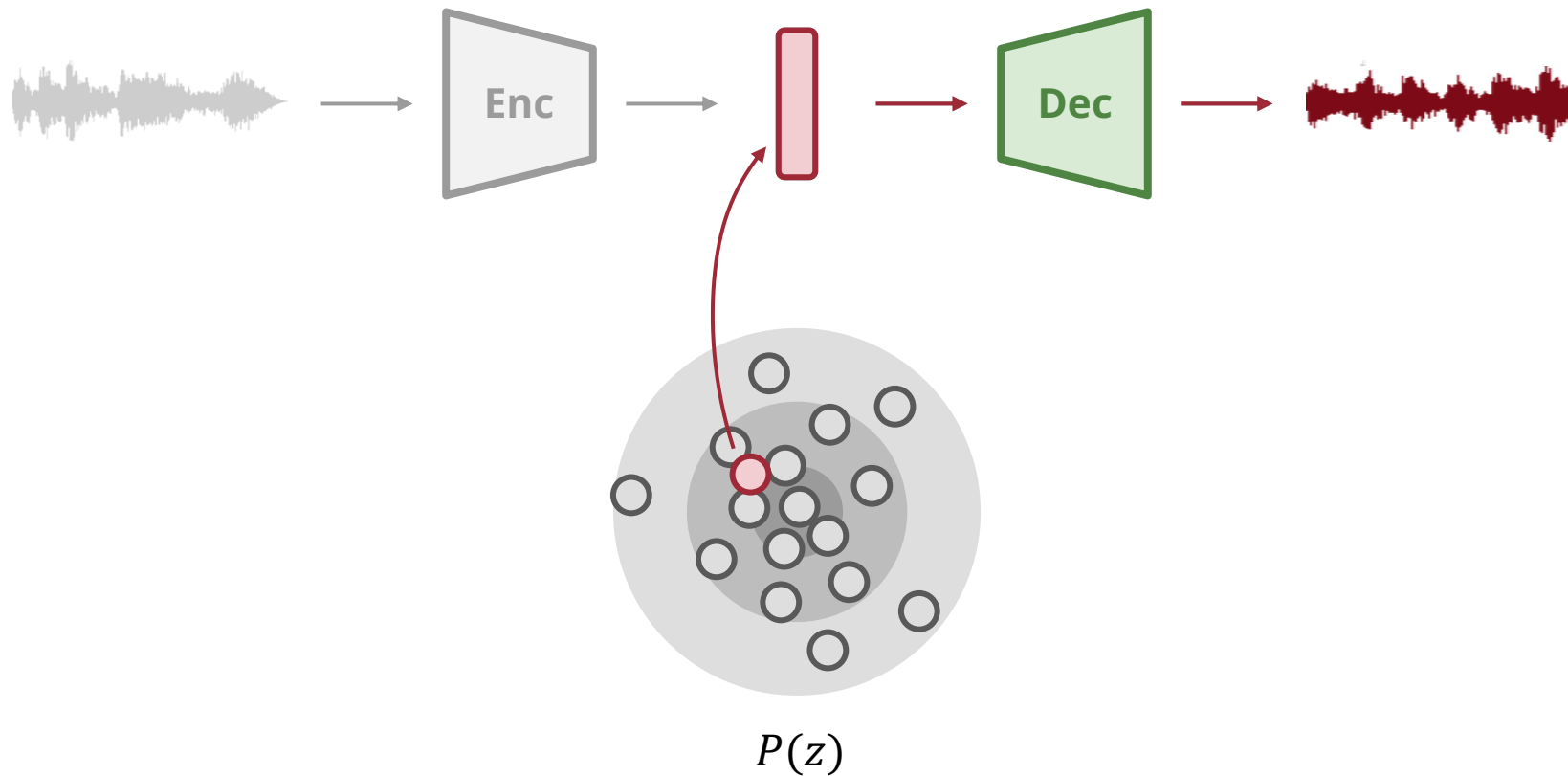
# Variational Autoencoder (VAE)



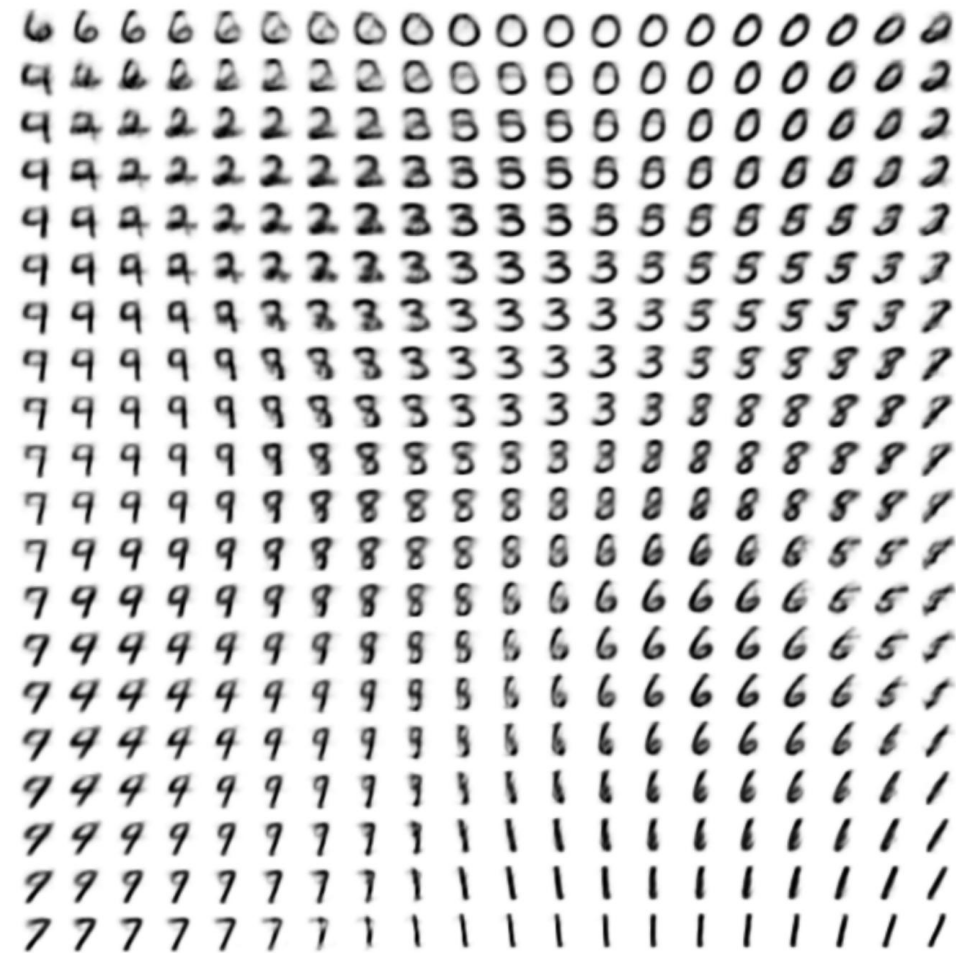
# Variational Autoencoder (VAE): Training



# Variational Autoencoder (VAE): Generation

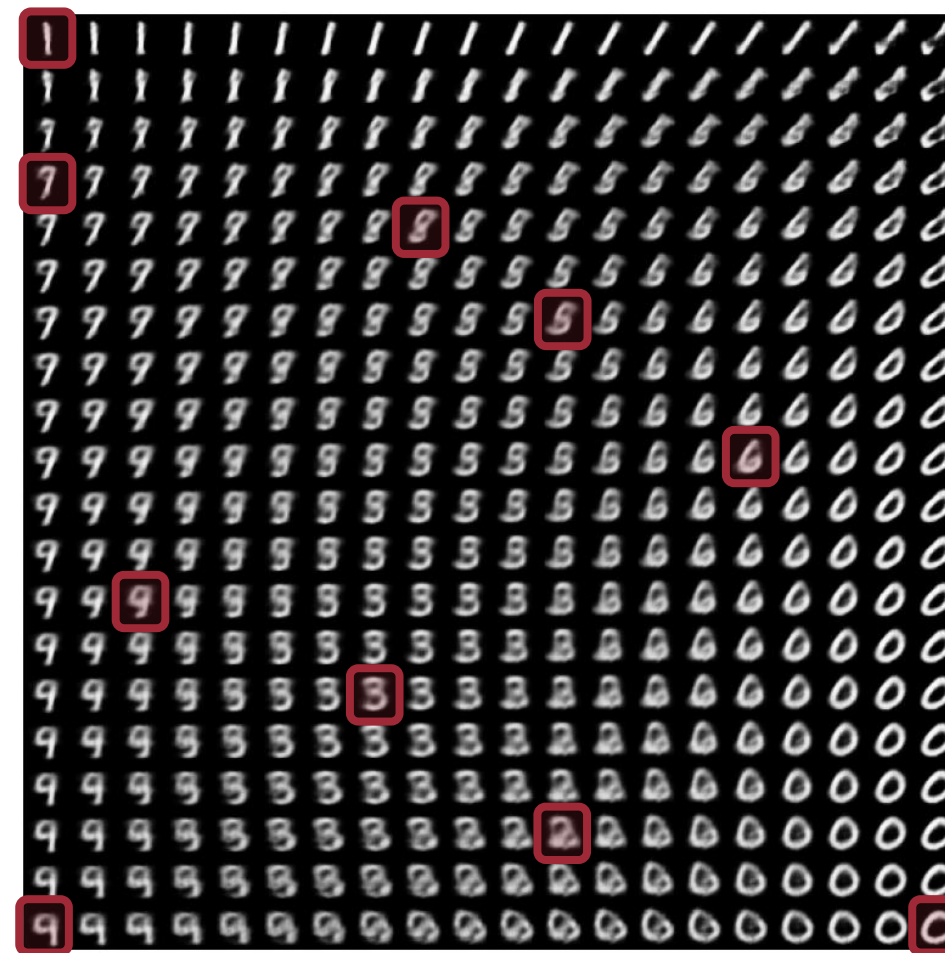
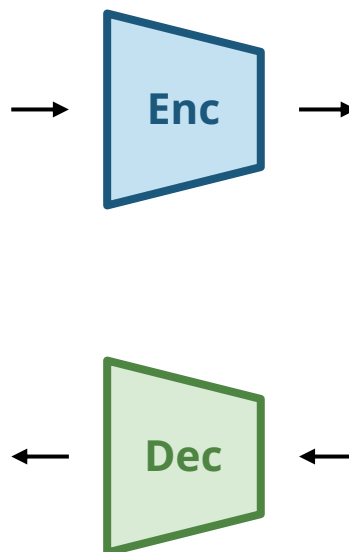


# Variational Autoencoder (VAE): Generated Examples



(Source: Kingma & Welling, 2014)

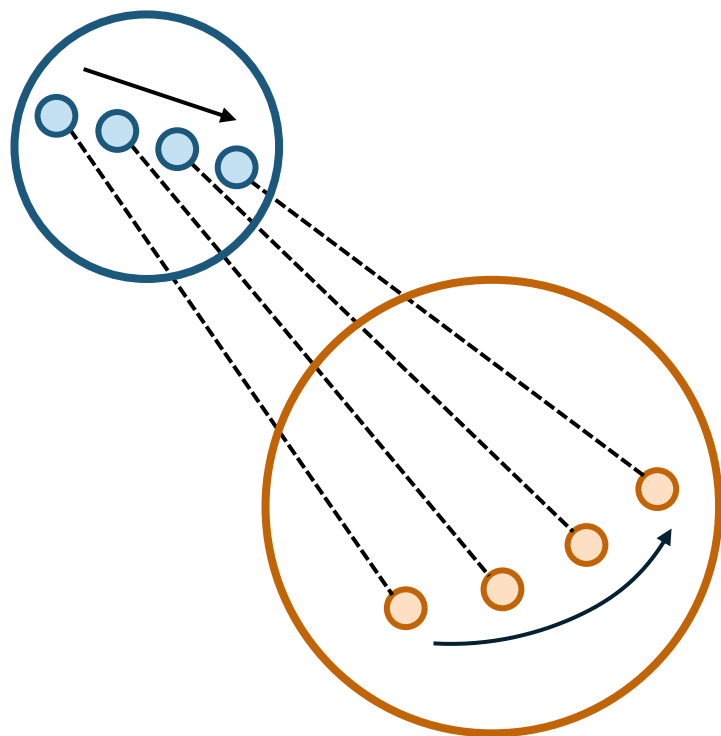
# What does a VAE learn?



(Source: tensorflow.org)

# Latent Space Interpolation of a VAE

Latent space



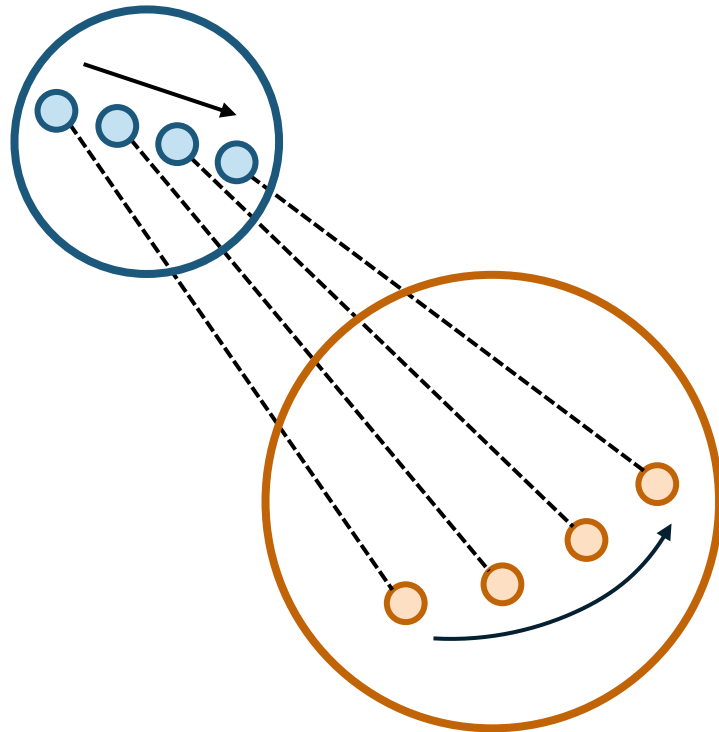
Data space



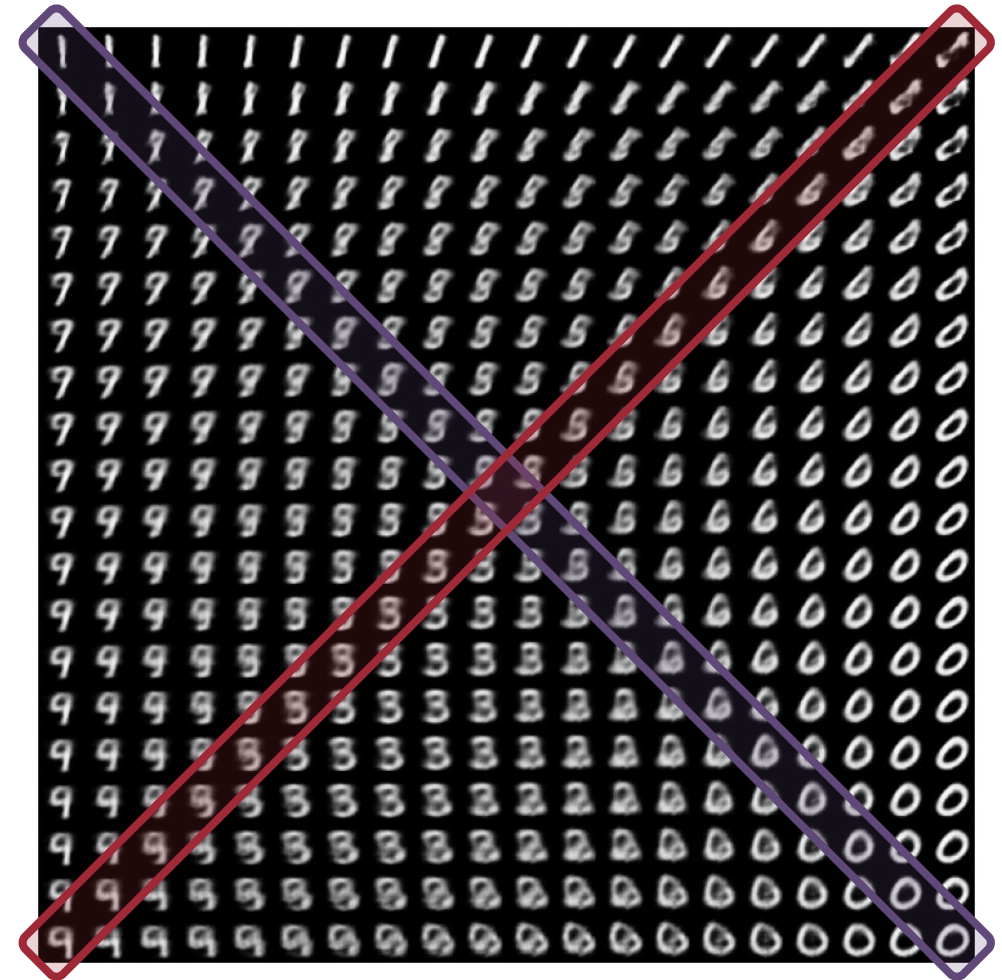
(Source: tensorflow.org)

# Latent Space Interpolation of a VAE

Latent space

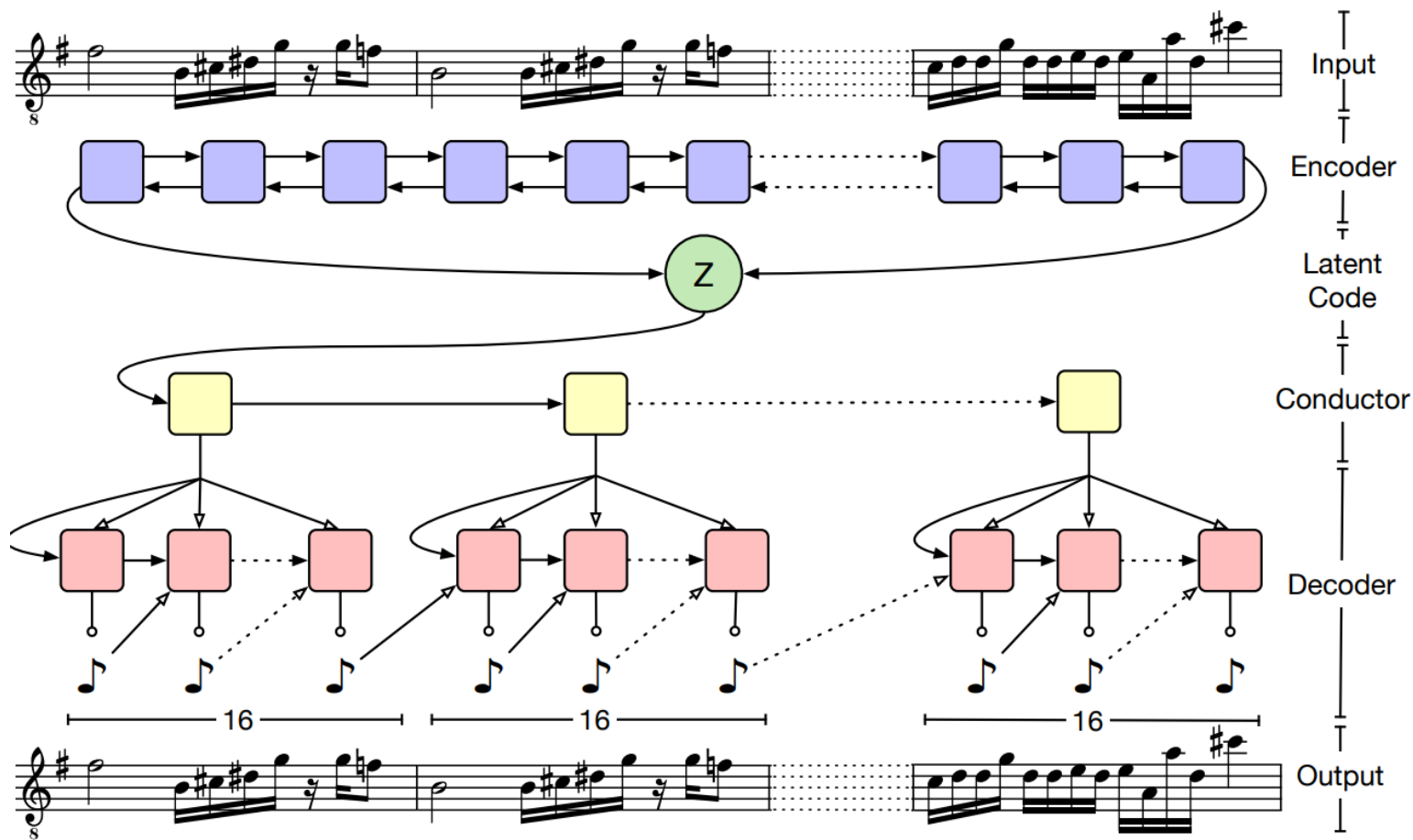


Data space



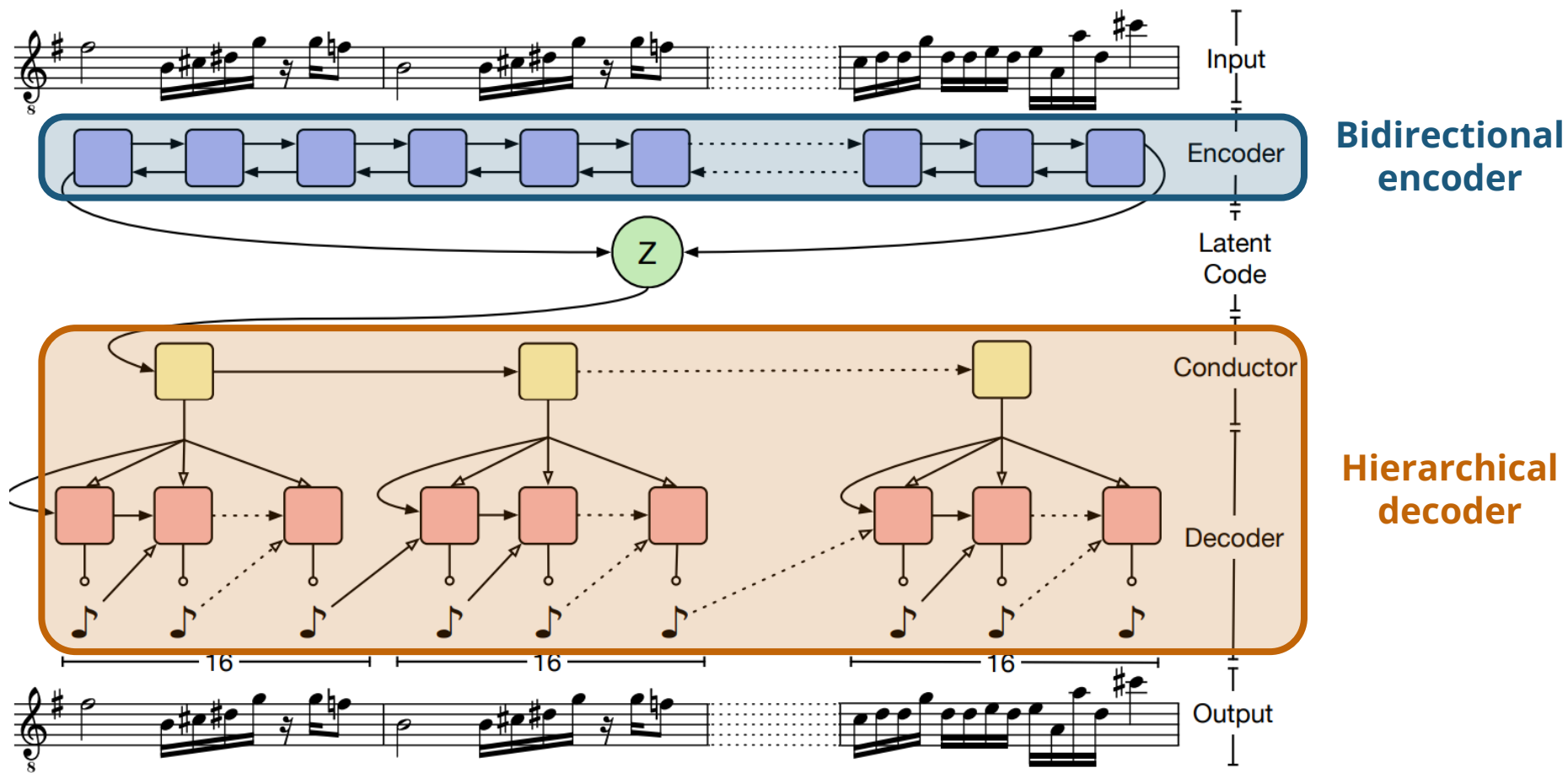
(Source: tensorflow.org)

# MusicVAE: A VAE for Symbolic Music (Roberts et al., 2018)



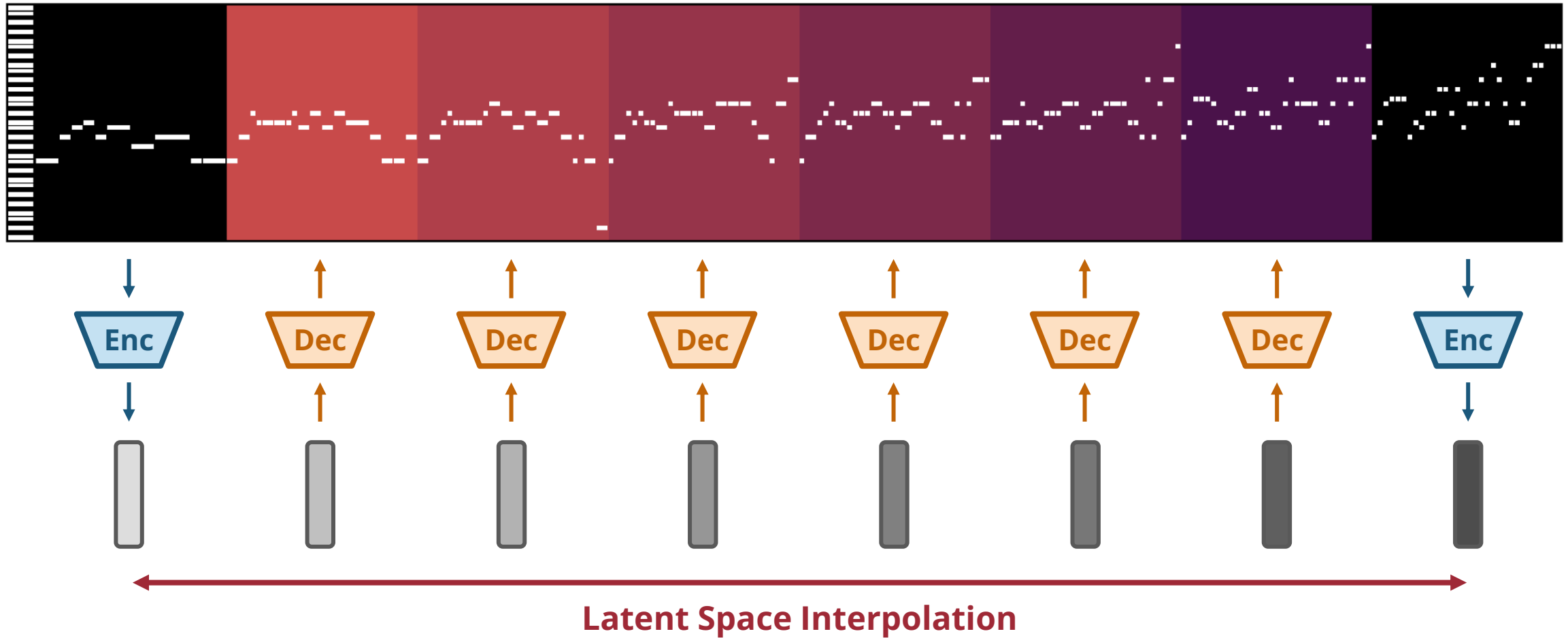
(Source: Roberts et al., 2018)

# MusicVAE: A VAE for Symbolic Music (Roberts et al., 2018)



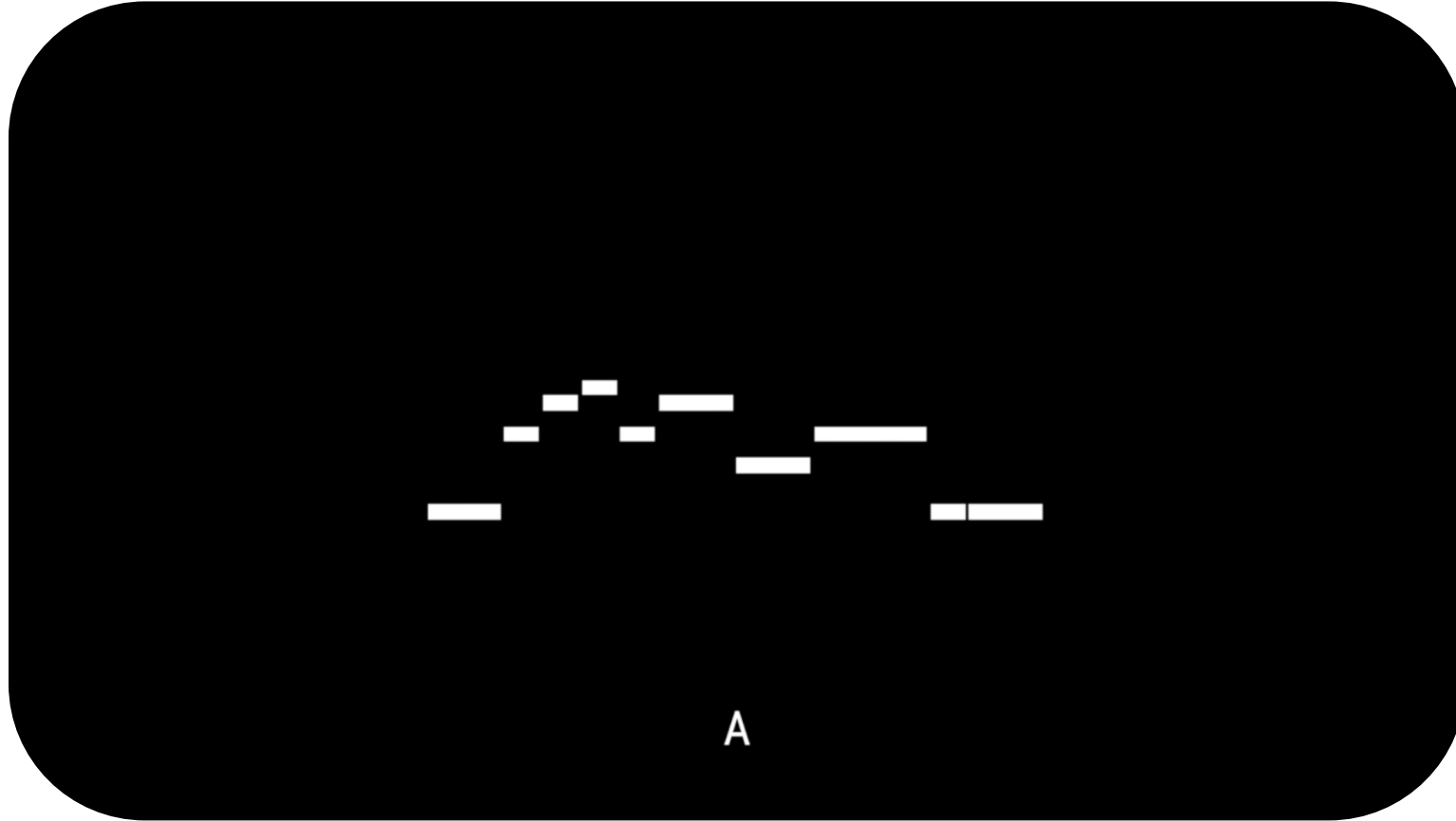
(Source: Roberts et al., 2018)

# Latent Space Interpolation for MusicVAE (Roberts et al., 2018)



(Source: Roberts et al., 2018)

# Latent Space Interpolation for MusicVAE (Roberts et al., 2018)

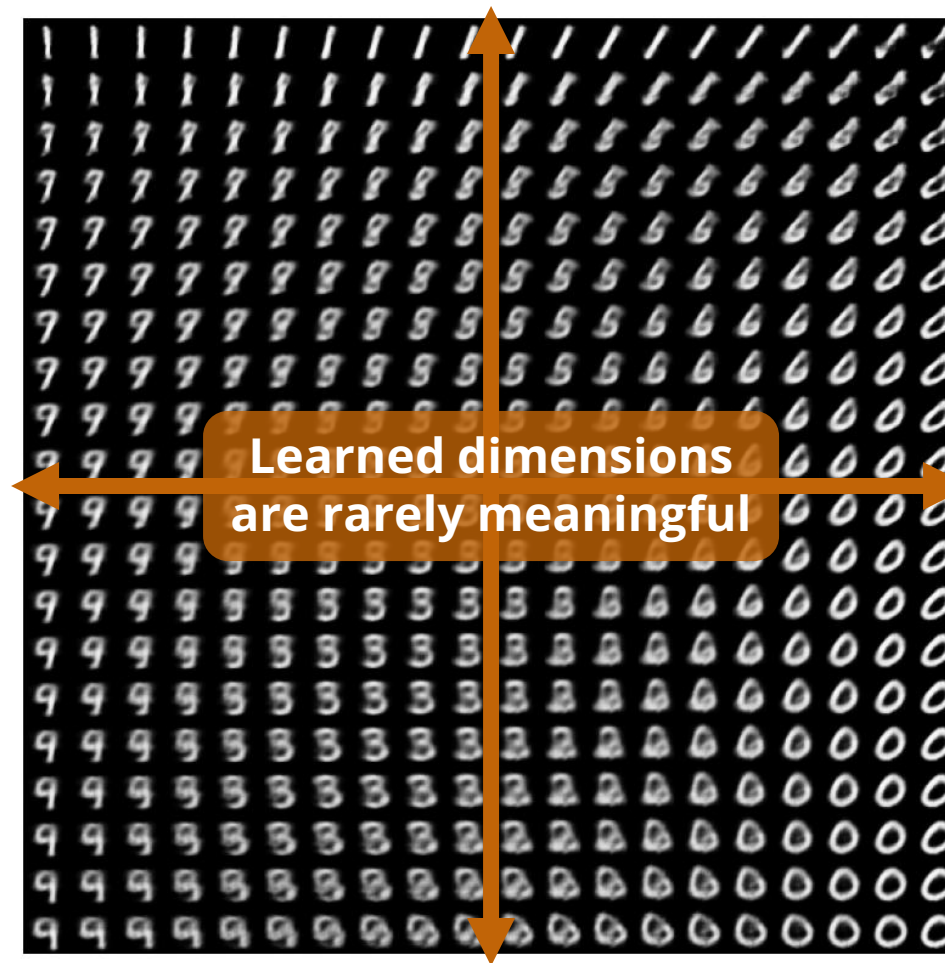
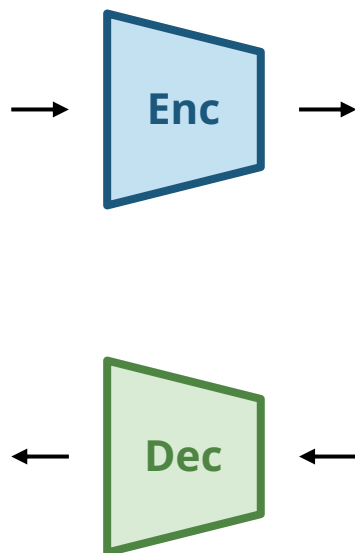


(Source: Roberts et al., 2018)

[goo.gl/magenta/musicvae-examples](https://goo.gl/magenta/musicvae-examples)

# Disentangling the Latent Variables

# What does a VAE learn?



(Source: tensorflow.org)

# Disentangling the Latent Variables

VAE



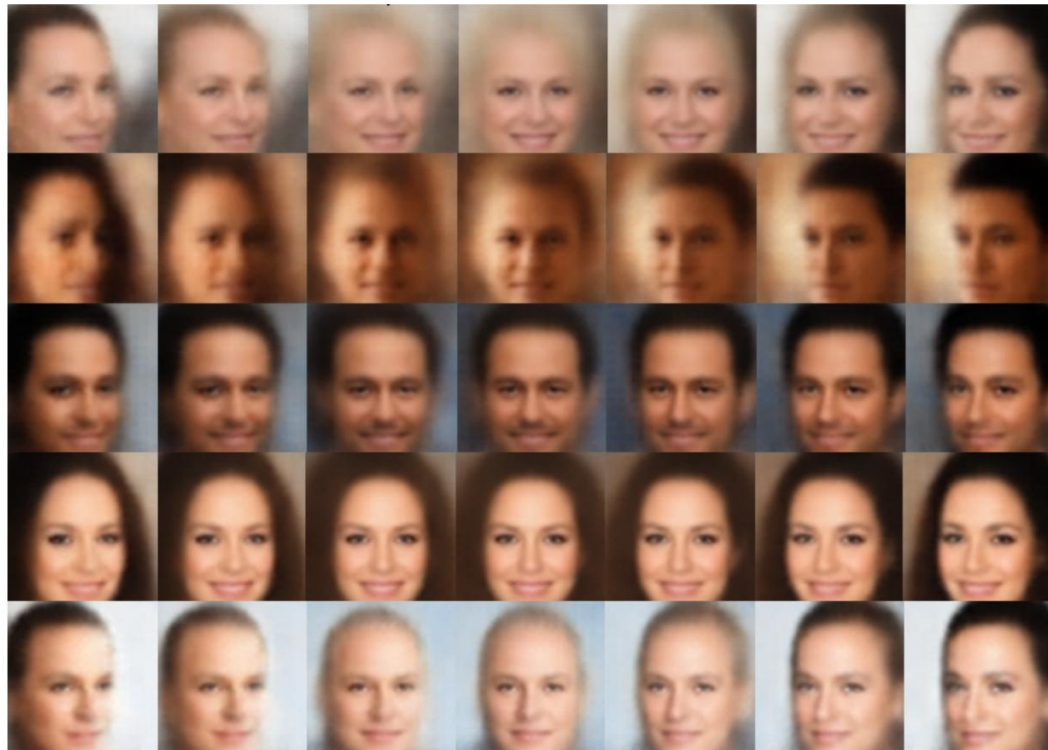
Guided VAE



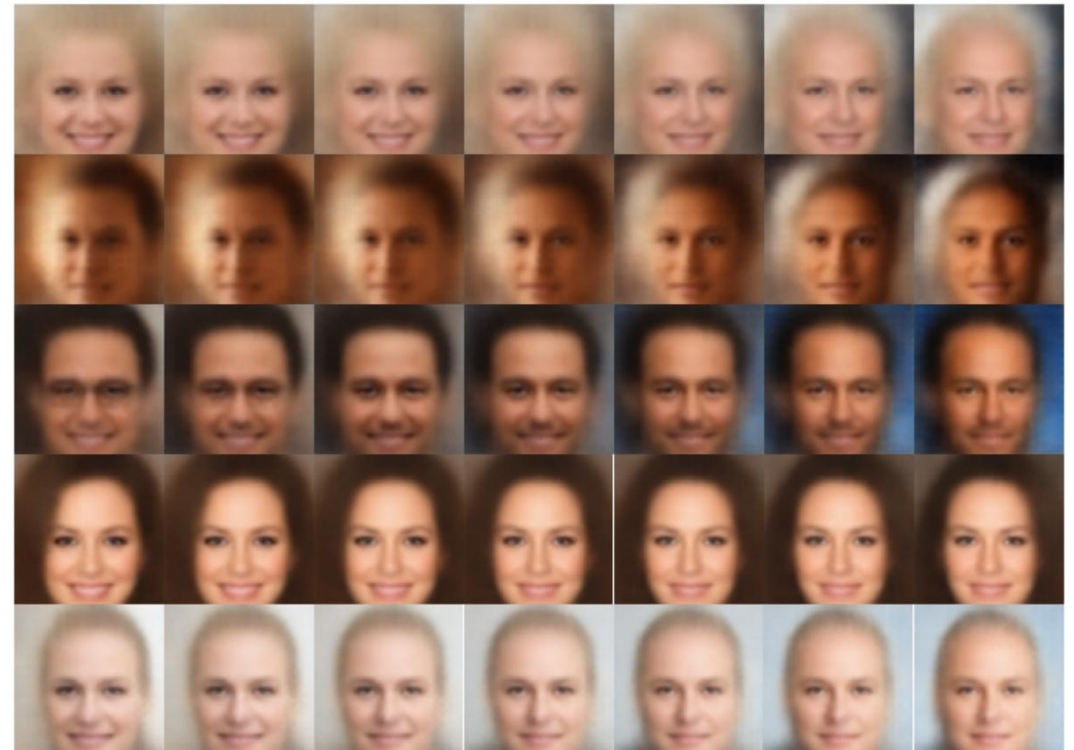
(Source: Ding et al., 2020)

# Disentangling the Latent Variables

Rotation



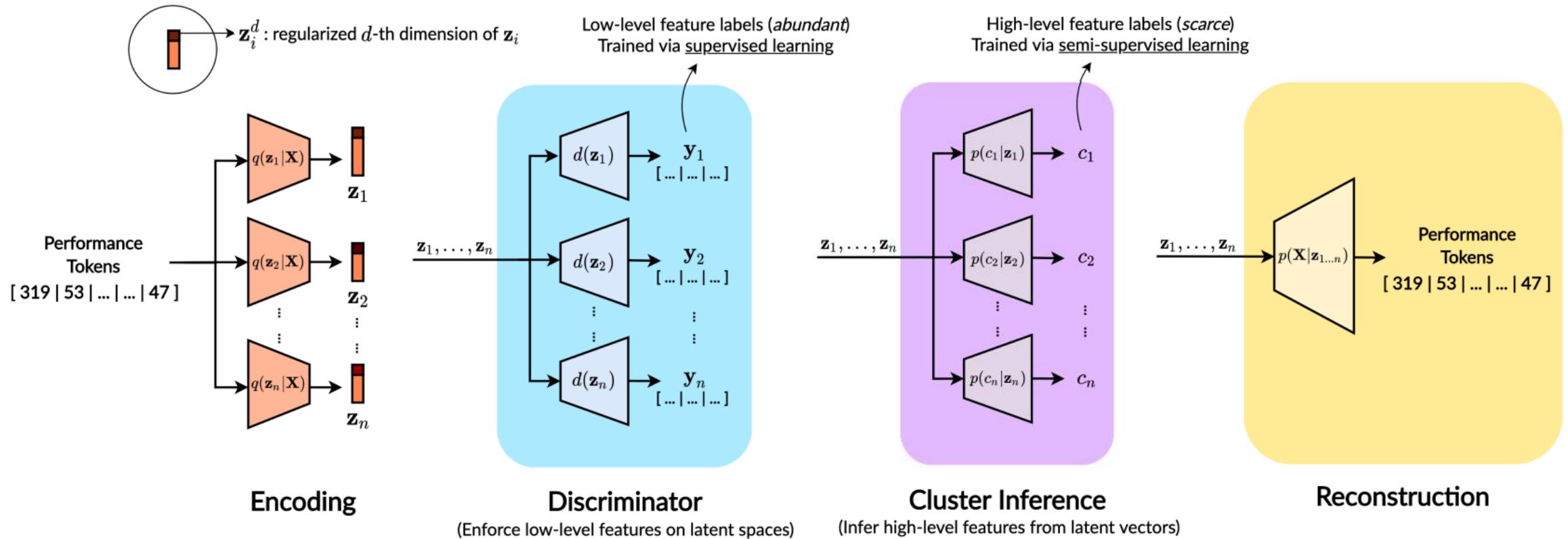
Smile



(Source: Higgins et al., 2017)

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner, " [\$\beta\$ -VAE: Learning Basic Visual Concepts with a Constrained Variational Framework](#)," *ICLR*, 2017.

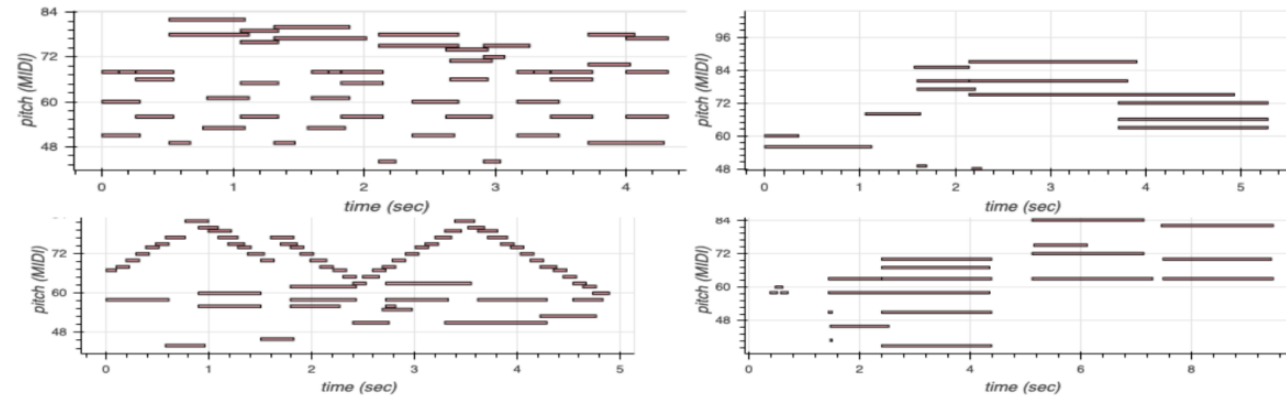
# Music FaderNet (Tan & Herremans, 2020)



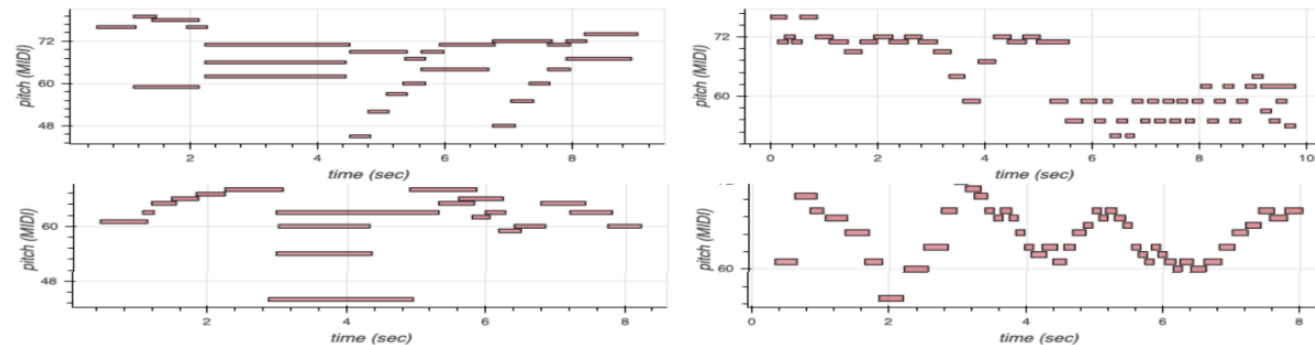
(Source: Tan & Herremans, 2020)

# Music FaderNet (Tan & Herremans, 2020)

## High Arousal → Low Arousal



## Low Arousal → High Arousal

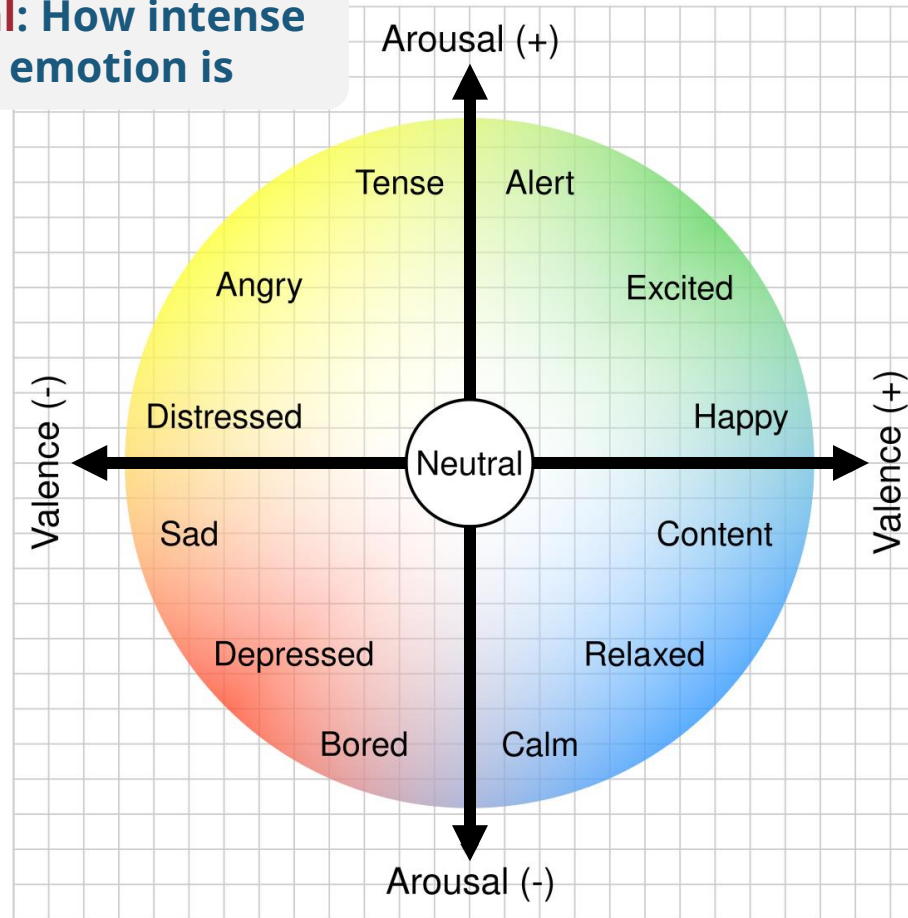


(Source: Tan & Herremans, 2020)

[music-fadernets.github.io](https://music-fadernets.github.io)

# Valence-Arousal Model for Emotion

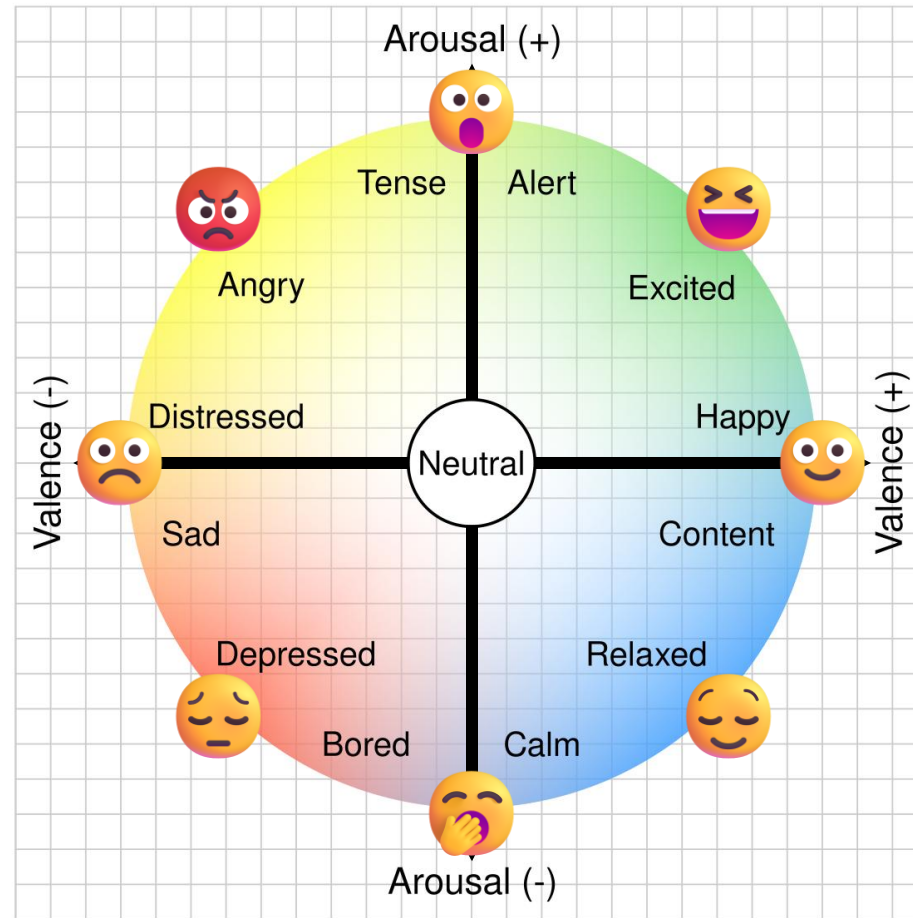
**Arousal:** How intense the emotion is



**Valence:** How pleasant the emotion is

(Source: mrAnmol)

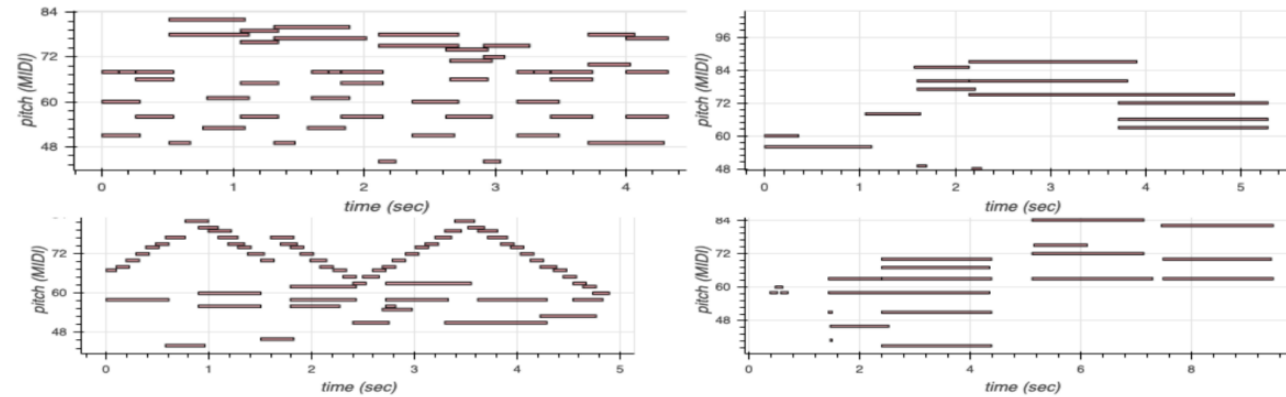
# Valence-Arousal Model for Emotion



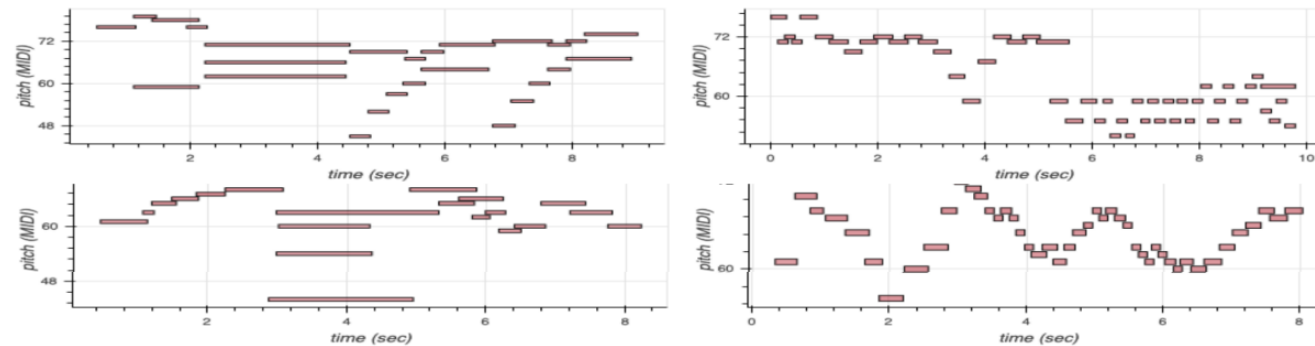
(Source: mrAnmol)

# Music FaderNet (Tan & Herremans, 2020)

## High Arousal → Low Arousal



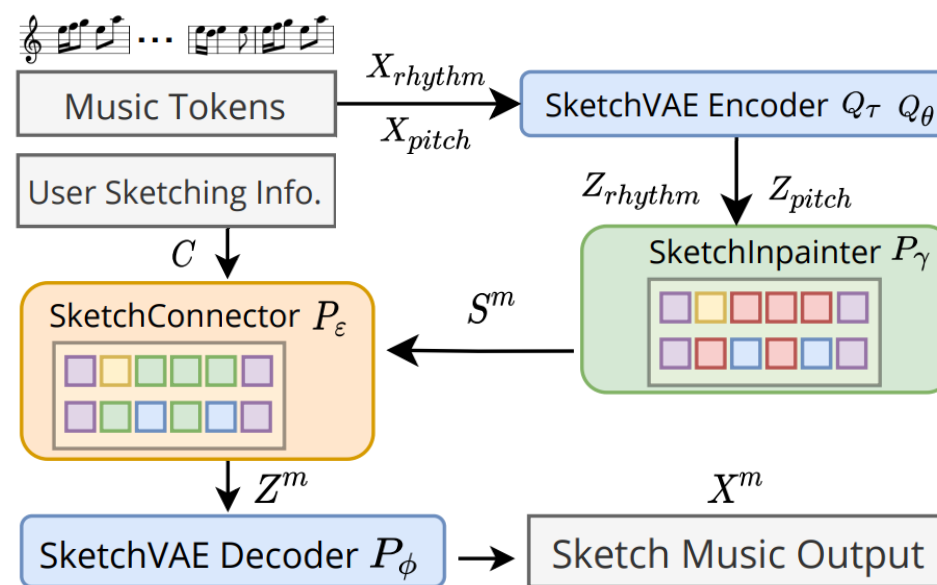
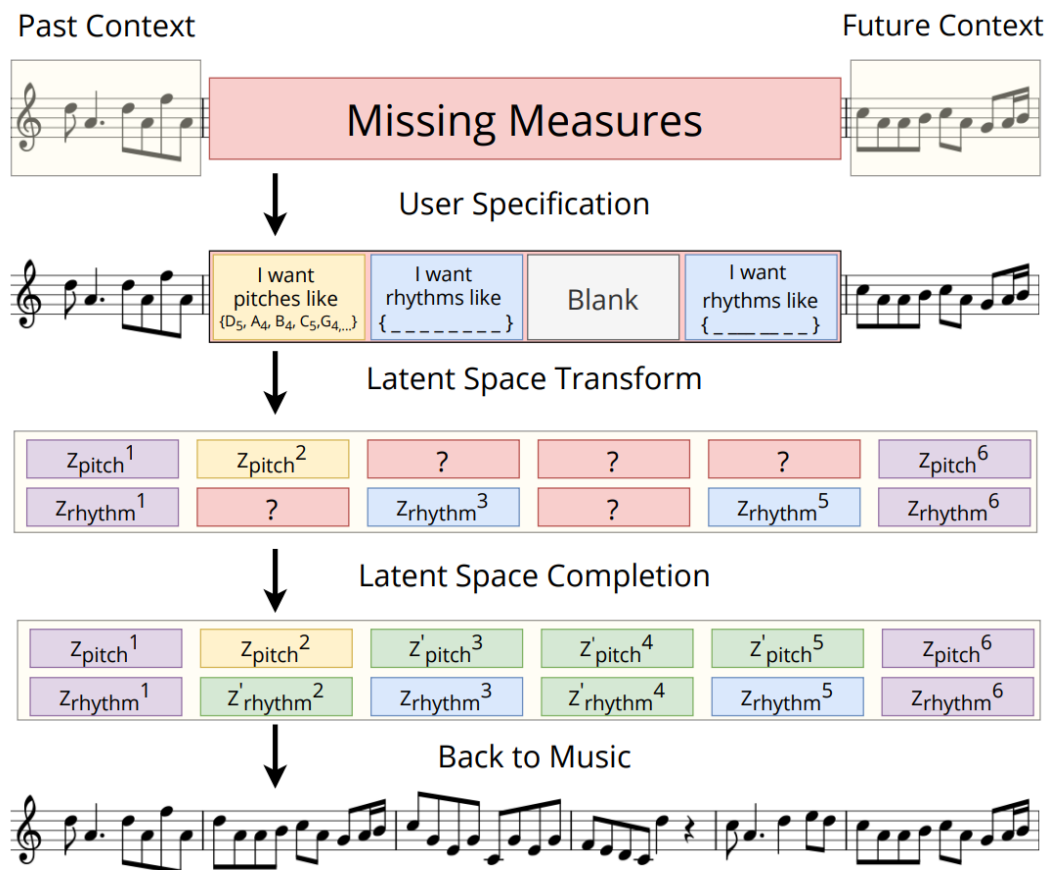
## Low Arousal → High Arousal



(Source: Tan & Herremans, 2020)

[music-fadernets.github.io](https://music-fadernets.github.io)

# Music SketchNet (Chen et al., 2020)



(Source: Chen et al., 2020)

# Music SketchNet (Chen et al., 2020)

The diagram illustrates the Music SketchNet architecture. It consists of four staves: Original, Control Pitch, Control Rhythm, and Control Both. The score is divided into three sections: Past Context, Generation, and Future Context. The Control Pitch staff shows blue notes with chord sets: {Ab5, Db6, Eb6, Gb6}, {C6, Eb6, Db6, F6, Db6}, {F6, Gb6, Ab6, Ab6, F6}, and {Db6, F6, Ab6, Bb6, Db6}. The Control Rhythm staff shows pink bars representing rhythmic sketches. The Control Both staff shows a grey bar labeled "No Sketch" during the Generation phase. The Original staff shows the resulting musical output.

(Source: Chen et al., 2020)

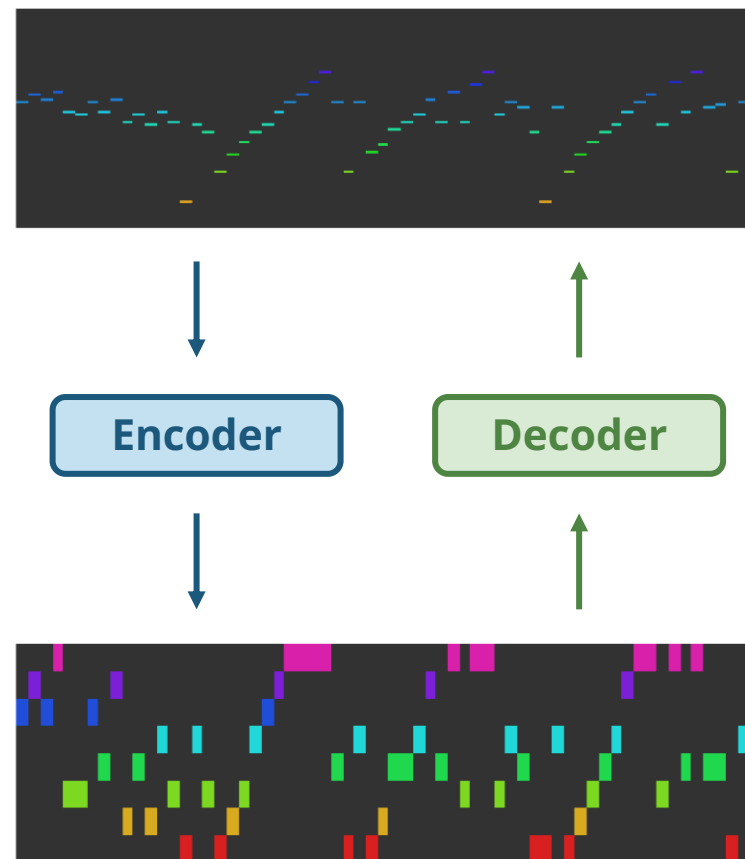
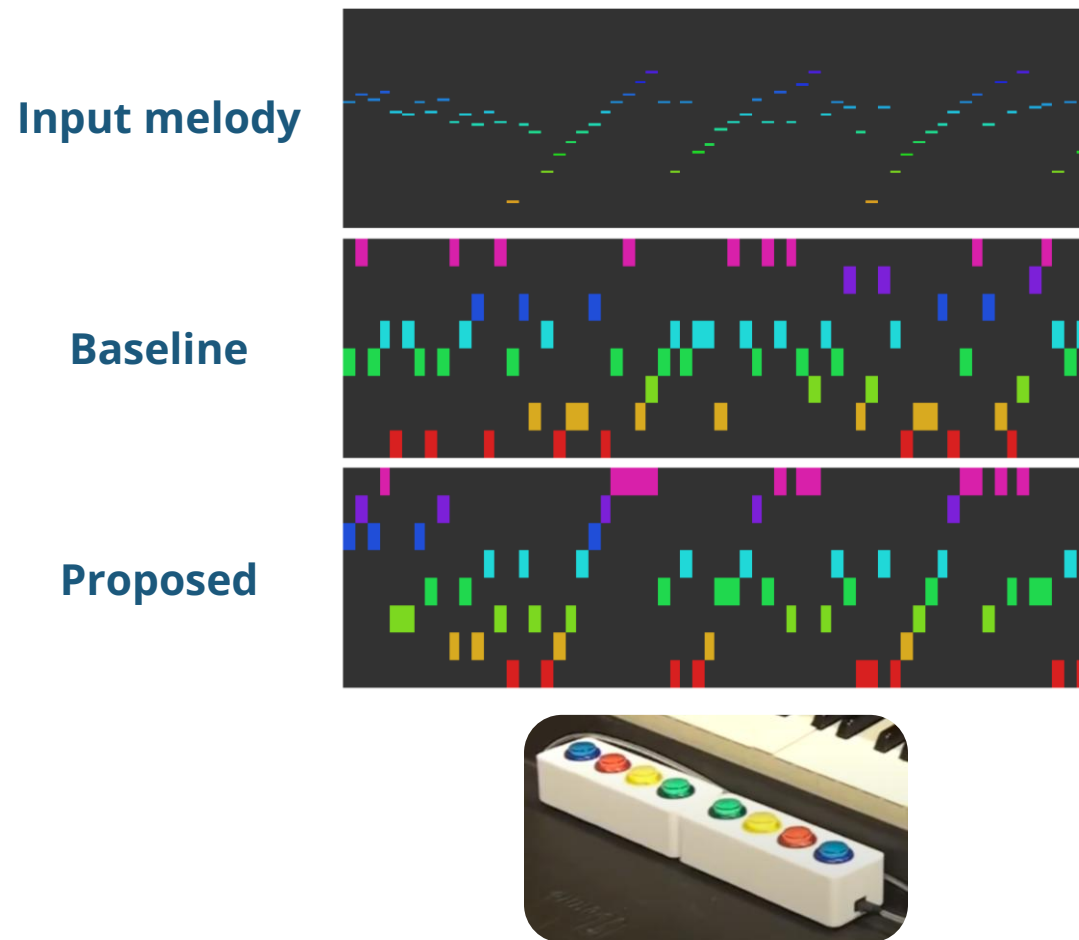
# Piano Genie

# Piano Genie (Donahue et al., 2019)



[youtu.be/YRb0XAnUpIk](https://youtu.be/YRb0XAnUpIk) & [magenta.tensorflow.org/pianogenie](https://magenta.tensorflow.org/pianogenie)

# Piano Genie (Donahue et al., 2019)



(Source: Donahue et al., 2019)

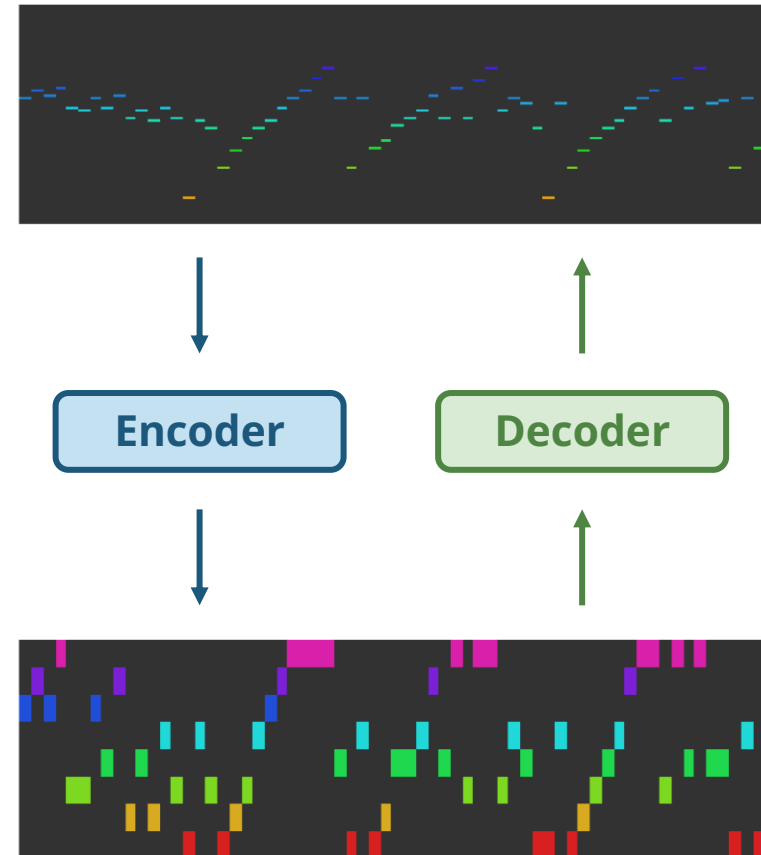
# Piano Genie (Donahue et al., 2019)

$$L = L_{\text{recons}} + L_{\text{margin}} + L_{\text{contour}}$$

$$L_{\text{recons}} = -\sum \log P_{\text{dec}}(\mathbf{x}|\text{enc}(\mathbf{x}))$$

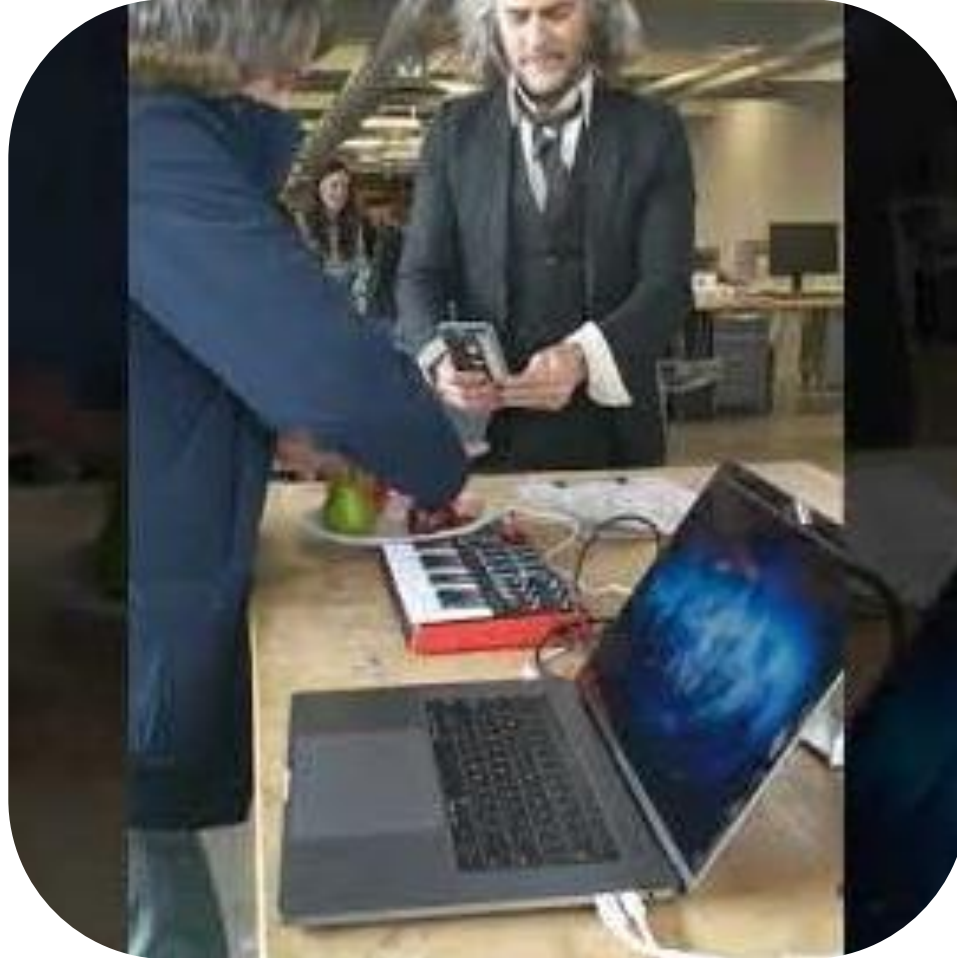
$$L_{\text{margin}} = \sum \max(|\text{enc}_s(\mathbf{x})| - 1, 0)^2$$

$$L_{\text{contour}} = \sum \max(1 - \Delta\mathbf{x}\Delta\text{enc}_s(\mathbf{x}), 0)^2$$



(Source: Donahue et al., 2019)

# Fruit Genie (2019)



[youtu.be/HoVs4kC68no](https://youtu.be/HoVs4kC68no)

# Fruit Genie Live (2019)



[youtu.be/L4wvXrPmIkU](https://youtu.be/L4wvXrPmIkU)

# Variational Autoencoders for Audio

# Four Paradigms of Music Generation



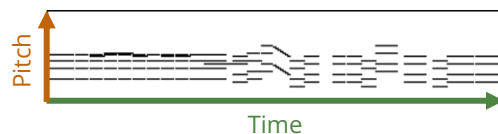
## Symbolic music generation

### Text-based

```
Program_change_0,  
Note_on_60, Time_shift_2, Note_off_60,  
Note_on_60, Time_shift_2, Note_off_60,  
Note_on_76, Time_shift_2, Note_off_67,  
Note_on_67, Time_shift_2, Note_off_67,  
...
```

### MIDI

### Image-based



### Piano roll



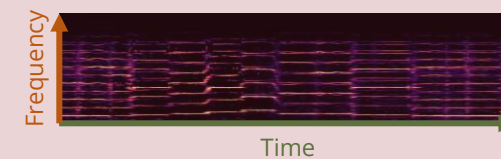
## Audio-domain music generation

### Time series-based



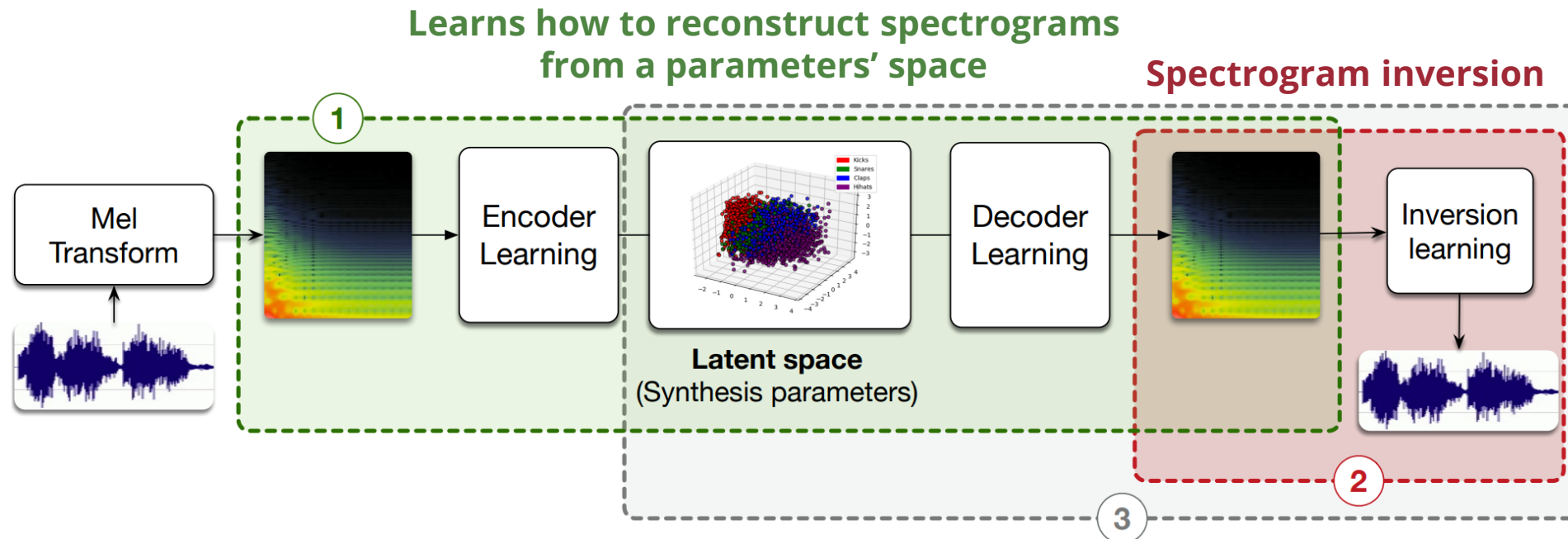
### Waveform

### Image-based



### Spectrogram

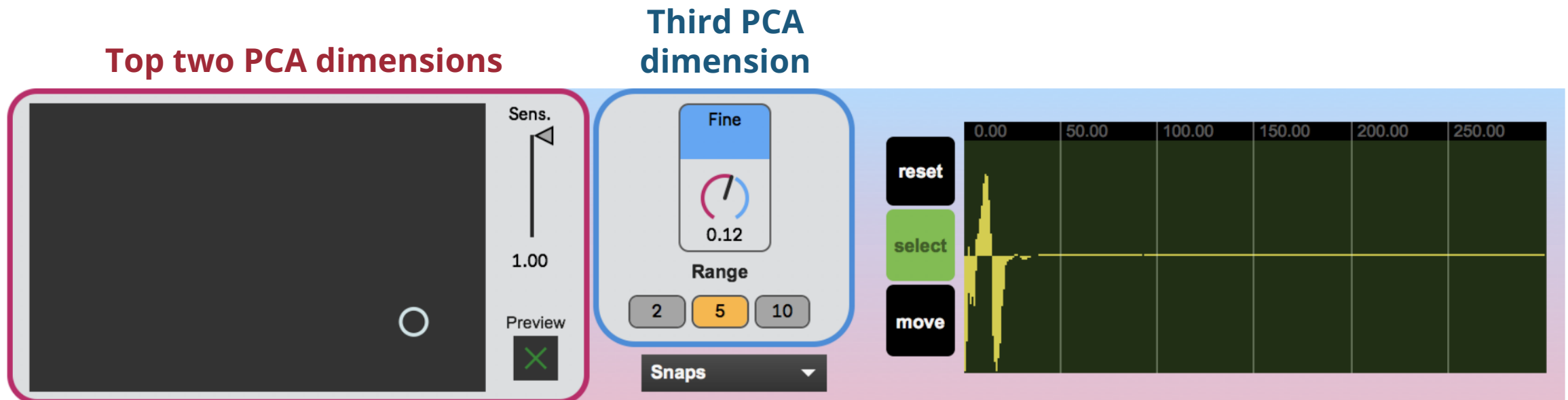
# Neural Drum Machine (Aouameur et al., 2019)



**Allows a user to interact with the model and to generate sound from the parameters' space**

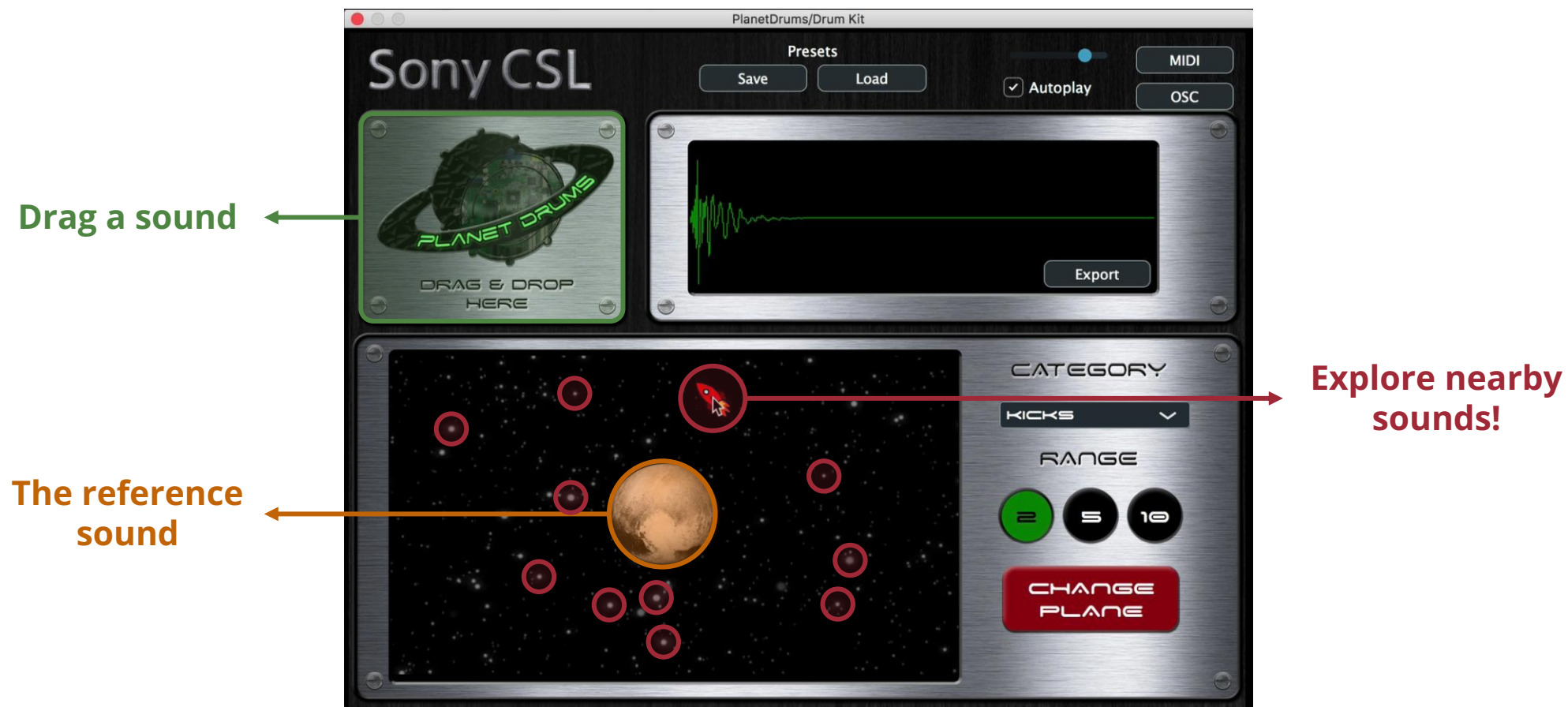
(Source: Aouameur et al., 2019)

# Neural Drum Machine (Aouameur et al., 2019)



(Source: Aouameur et al., 2019)

# Neural Drum Machine (Aouameur et al., 2019)



[drive.google.com/file/d/1DDDo0\\_KnwkWirCM4t0PT8cp6uotsfuufj/view](https://drive.google.com/file/d/1DDDo0_KnwkWirCM4t0PT8cp6uotsfuufj/view)

# Four Paradigms of Music Generation



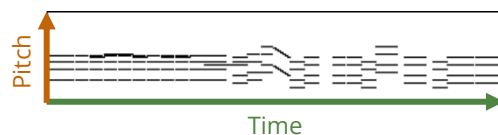
## Symbolic music generation

### Text-based

```
Program_change_0,  
Note_on_60, Time_shift_2, Note_off_60,  
Note_on_60, Time_shift_2, Note_off_60,  
Note_on_76, Time_shift_2, Note_off_67,  
Note_on_67, Time_shift_2, Note_off_67,  
...
```

### MIDI

### Image-based



### Piano roll



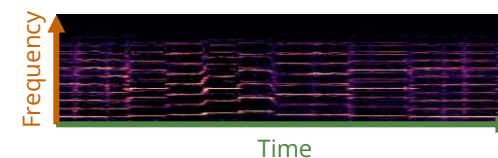
## Audio-domain music generation

### Time series-based



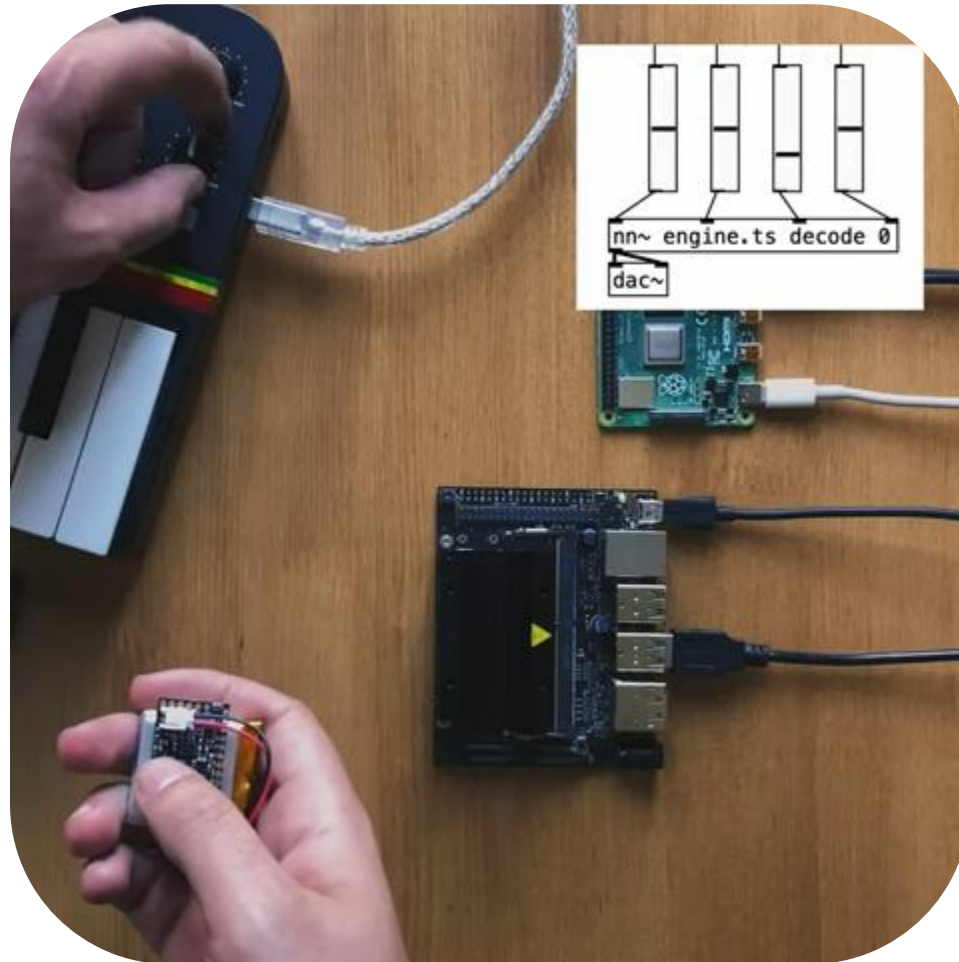
### Waveform

### Image-based



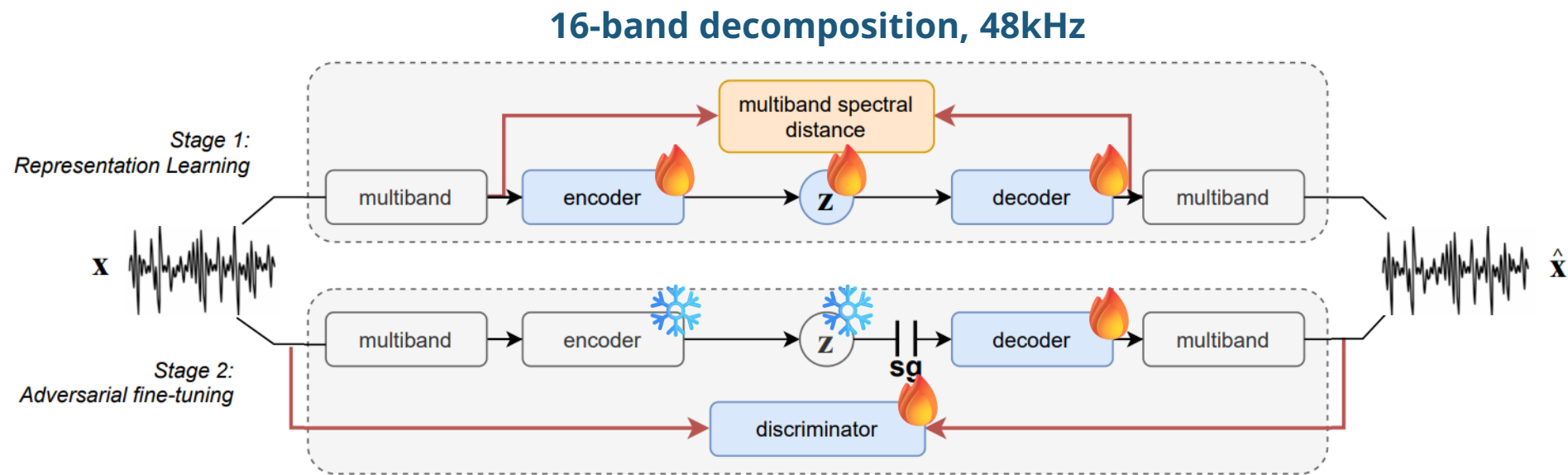
### Spectrogram

# RAVE: Real-time Audio Synthesis (Caillon & Esling, 2022)



[youtu.be/jAIRf4nGgYI](https://youtu.be/jAIRf4nGgYI)

# RAVE: Real-time Audio Synthesis (Caillon & Esling, 2022)



(Source: Caillon & Esling, 2021)

[github.com/acids-ircam/RAVE](https://github.com/acids-ircam/RAVE)

# RAVE: Real-time Audio Synthesis (Caillon & Esling, 2022)

Model	CPU synthesis	GPU synthesis
NSynth	18 Hz	57 Hz
SING	304 kHz	9.8 MHz
RAVE (Ours) w/o multiband	38 kHz	3.7 MHz
<b>RAVE (Ours)</b>	<b>985 kHz</b>	<b>11.7 MHz</b>

**Realtime capable on CPUs & GPUs**

(Source: Caillon & Esling, 2021)

[anonymous84654.github.io/RAVE\\_anonymous](https://anonymous84654.github.io/RAVE_anonymous)

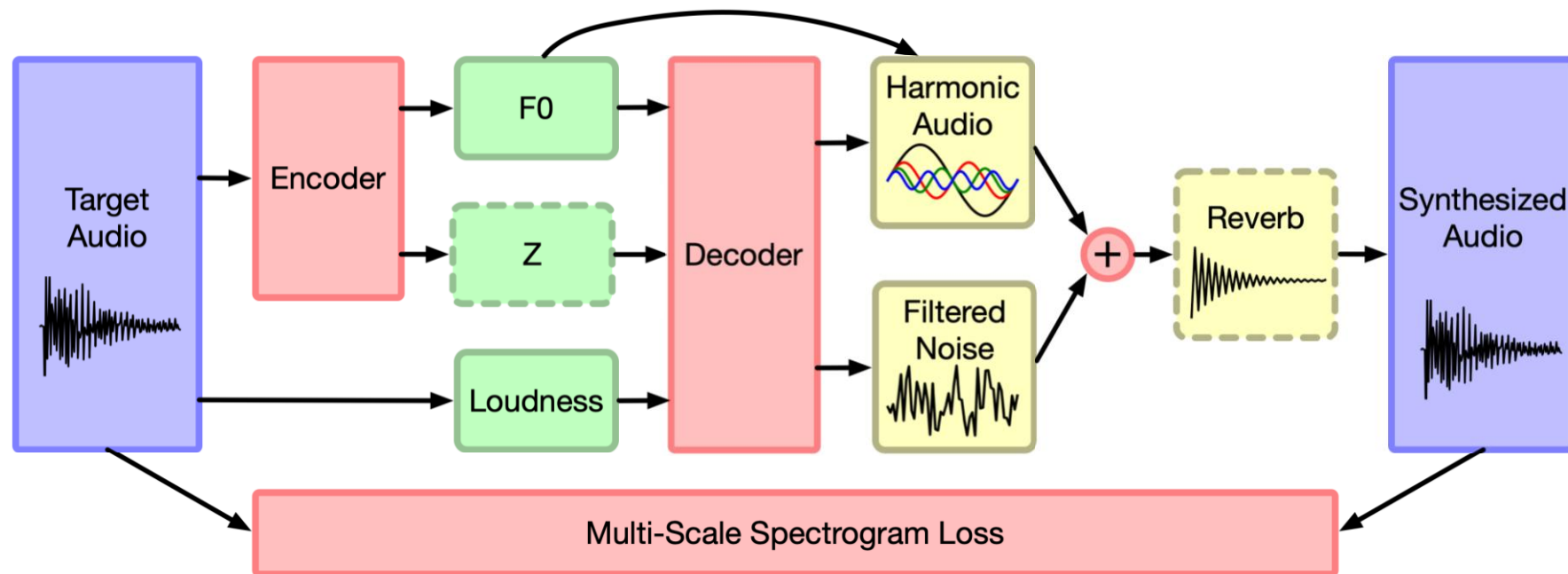
# RAVE: Real-time Audio Synthesis (Caillon & Esling, 2022)



[youtu.be/dMZs04TzxUI](https://youtu.be/dMZs04TzxUI)

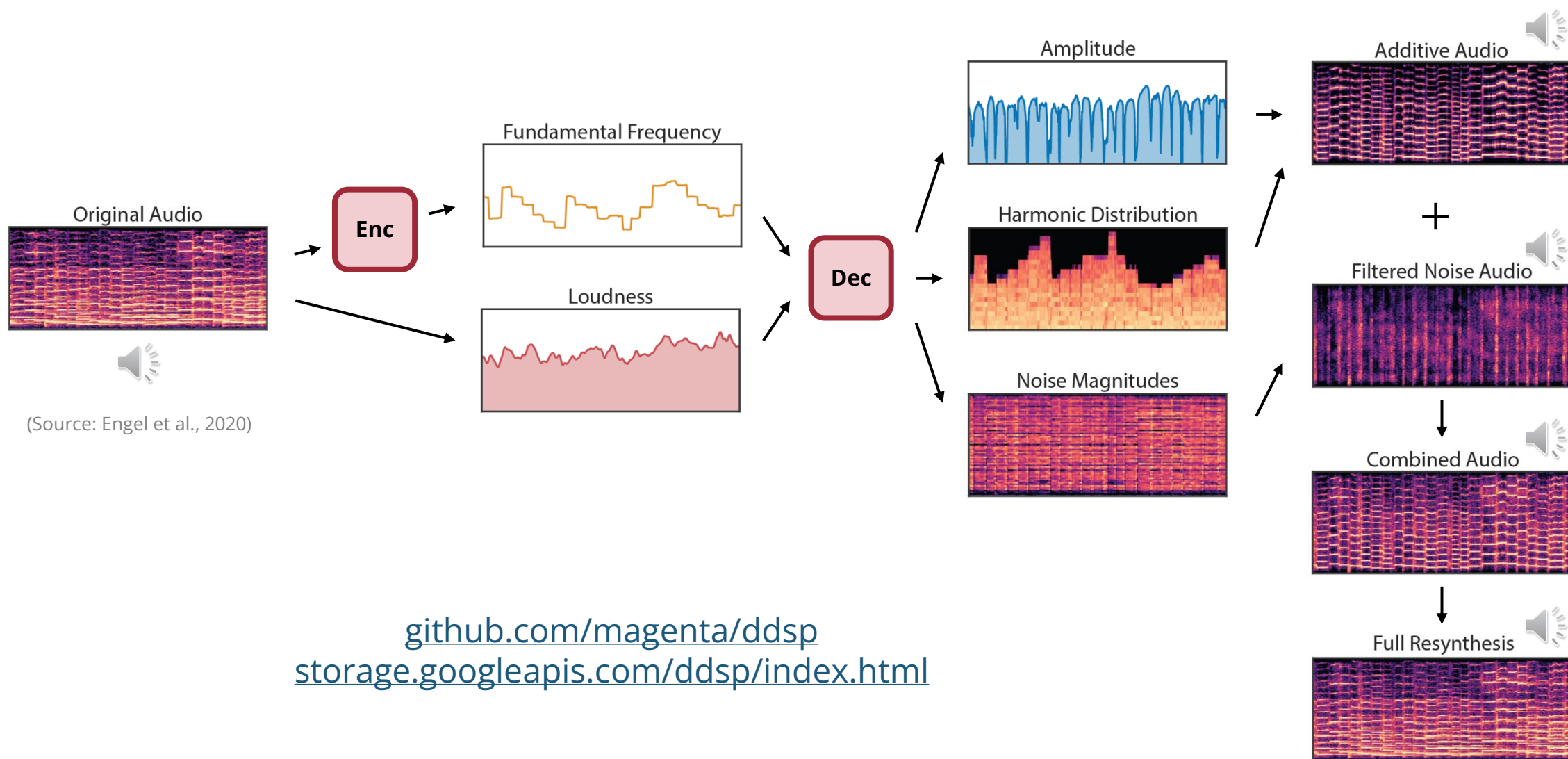
# Differentiable DSP

# Differentiable DSP (DDSP) (Engel et al., 2020)



(Source: Engel et al., 2020)

# Differentiable DSP (DDSP) (Engel et al., 2020)



# Entering Demons & Gods by Yaboi Hanoi (2022)



[youtu.be/PbrRoR3nEVw](https://youtu.be/PbrRoR3nEVw)

[soundcloud.com/yaboihanoi/enter-demons-and-gods](https://soundcloud.com/yaboihanoi/enter-demons-and-gods)

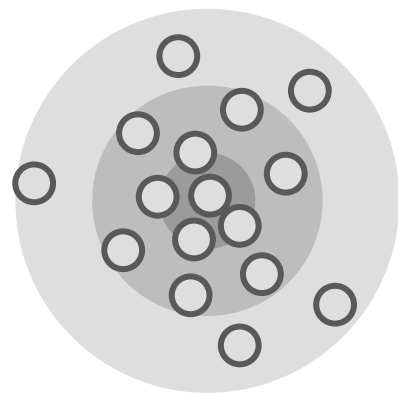


# Recap

# Deep Latent Variable Models

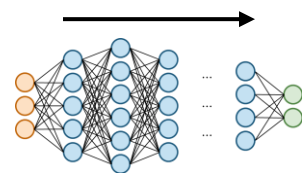
- **Intuition:** Learn to map a known distribution to the data distribution

Known distribution

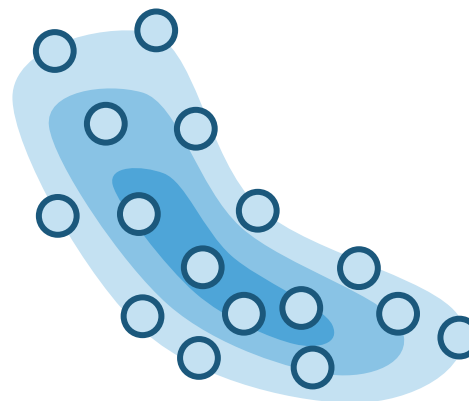


$P(z)$

$P(x | z)$



Data distribution

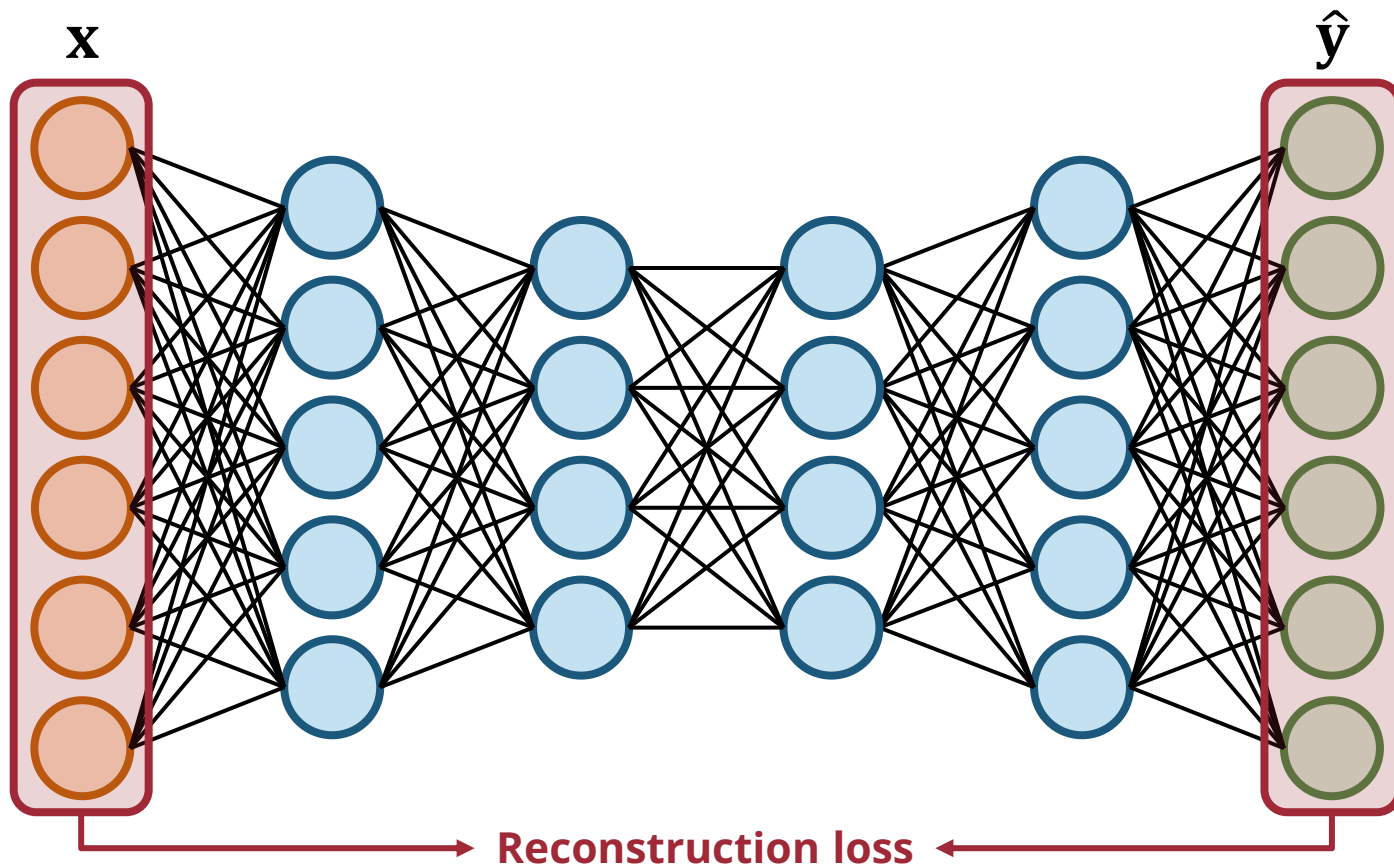


$P(x)$

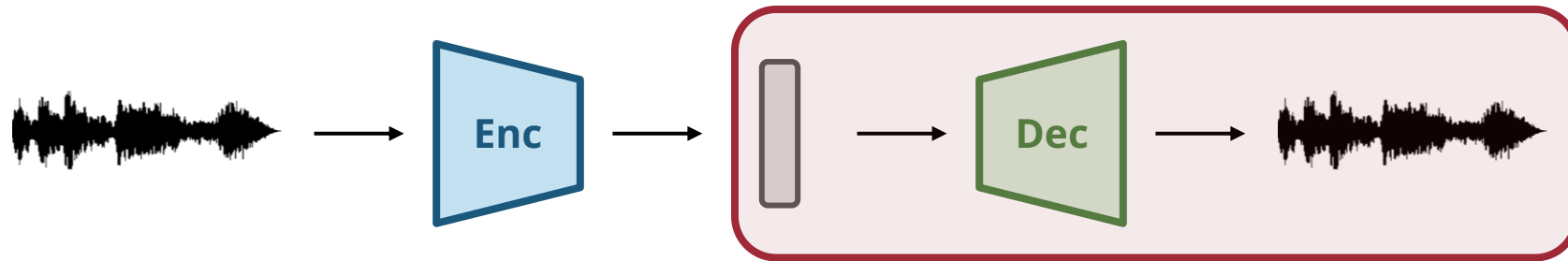
$$P(x) = P(z) P(x | z)$$

# Autoencoders

- A neural network where the **input and output are the same**



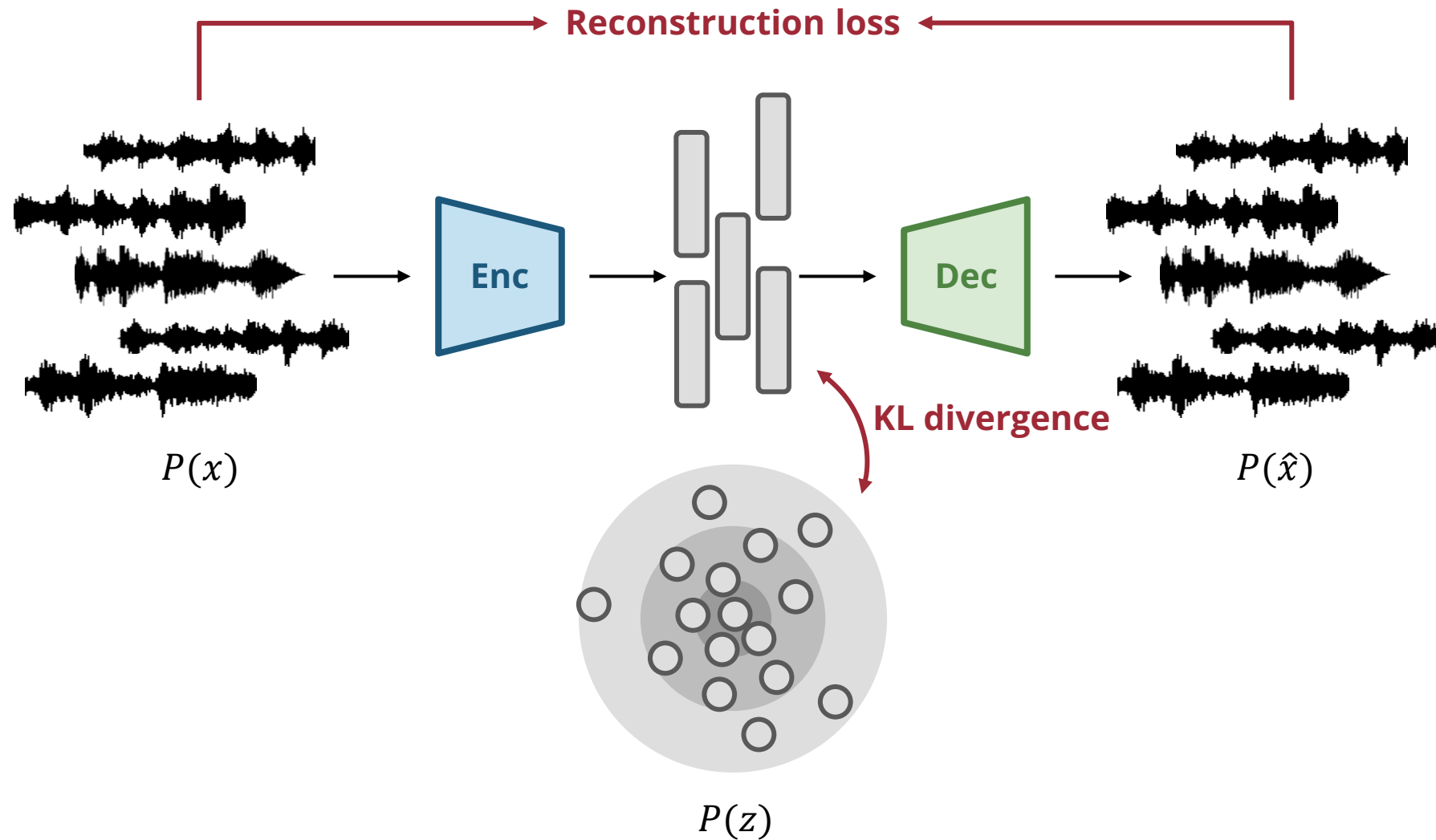
# Autoencoder



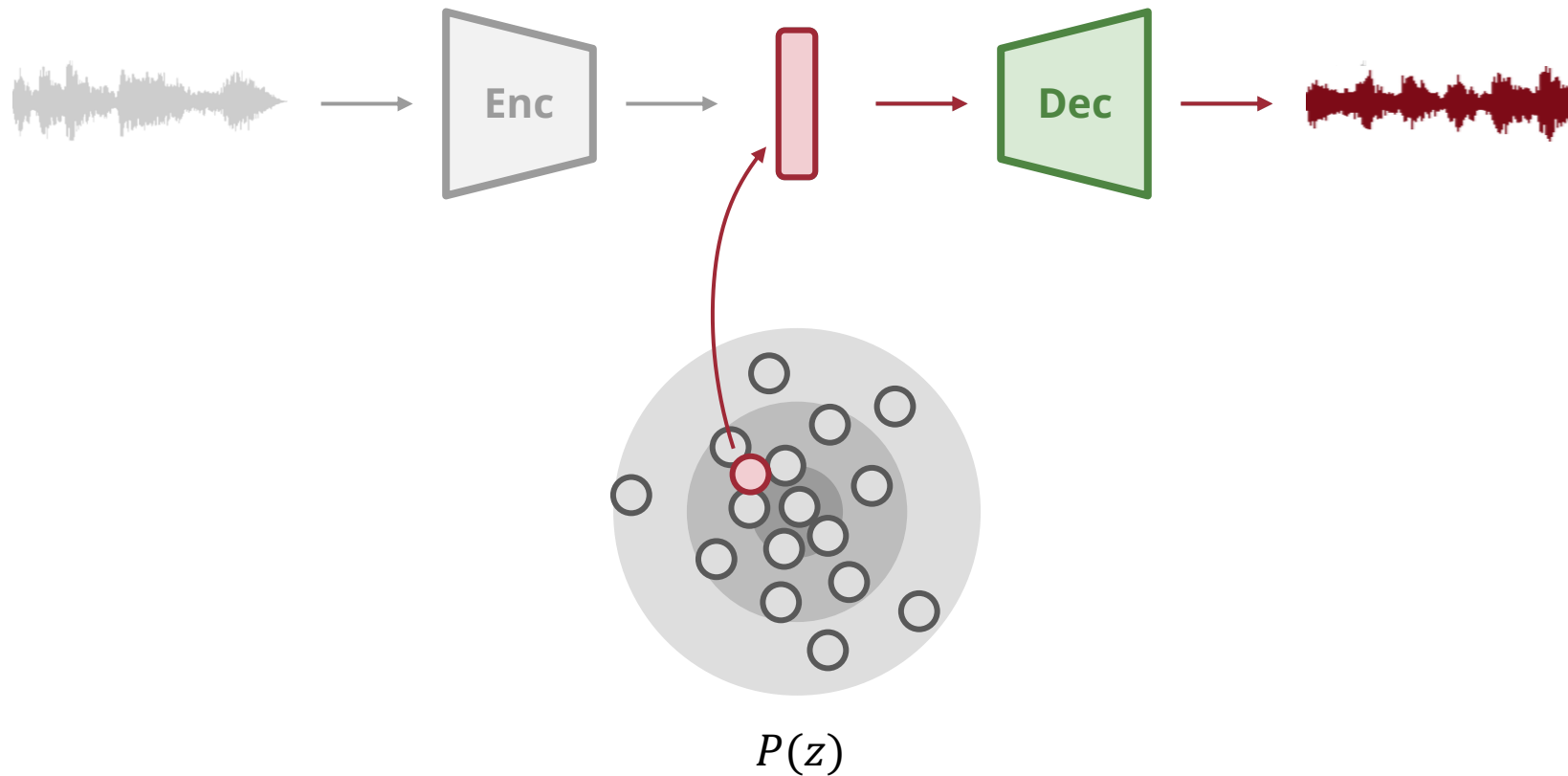
**Isn't this like a generative model?**

**What exactly is a generative model?**

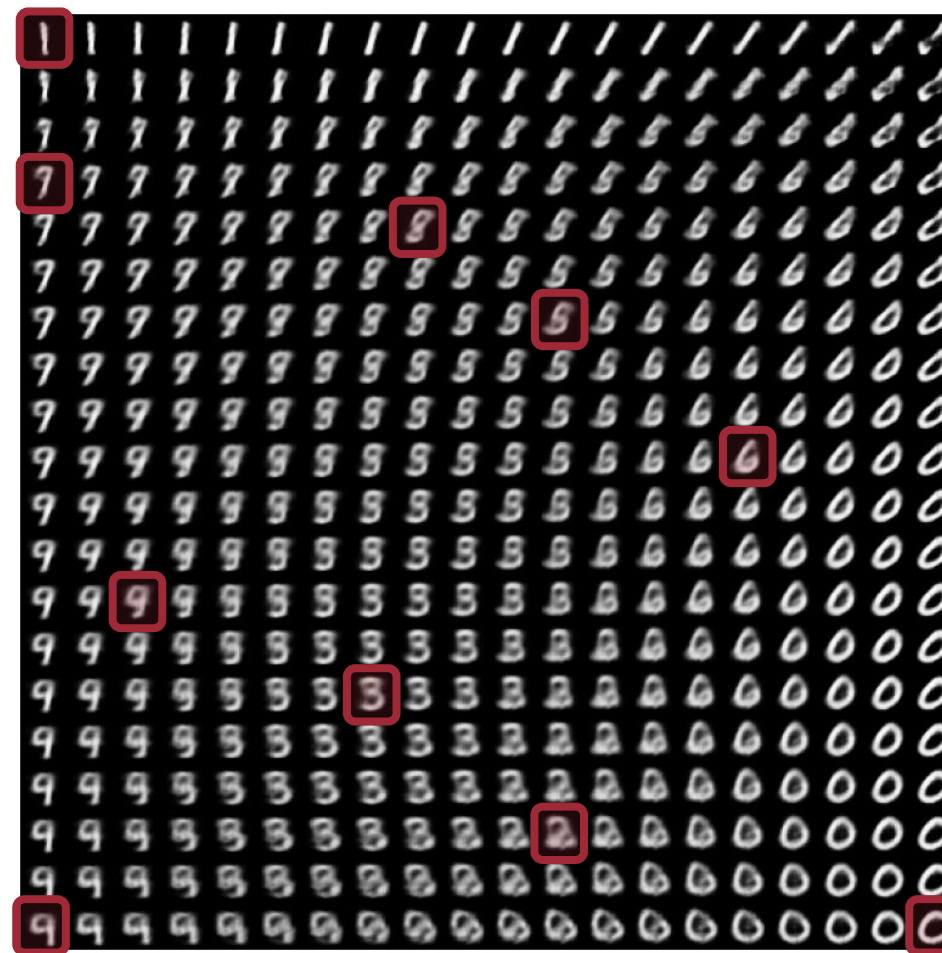
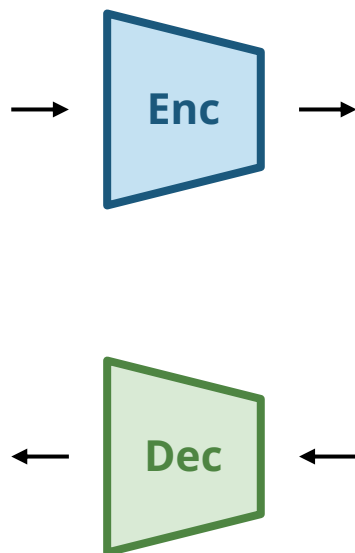
# Variational Autoencoder (VAE): Training



# Variational Autoencoder (VAE): Generation



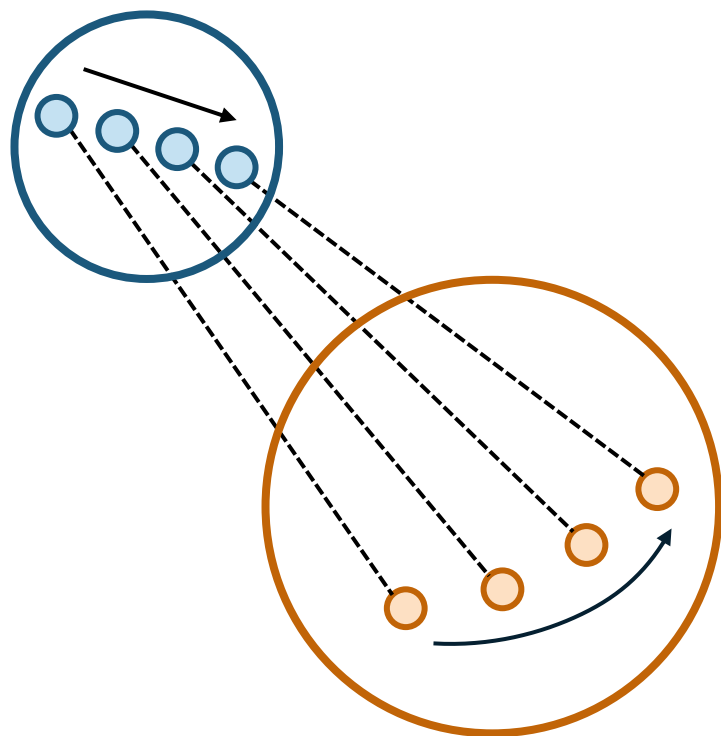
# What does a VAE learn?



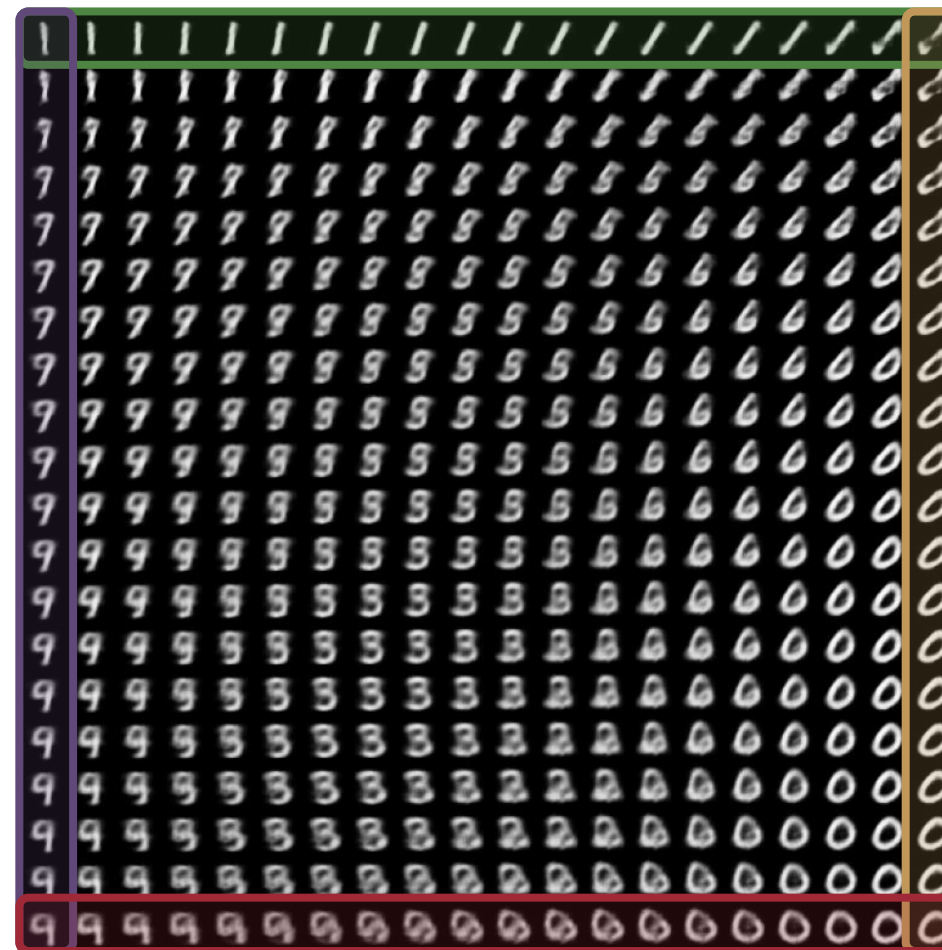
(Source: tensorflow.org)

# Latent Space Interpolation of a VAE

Latent space

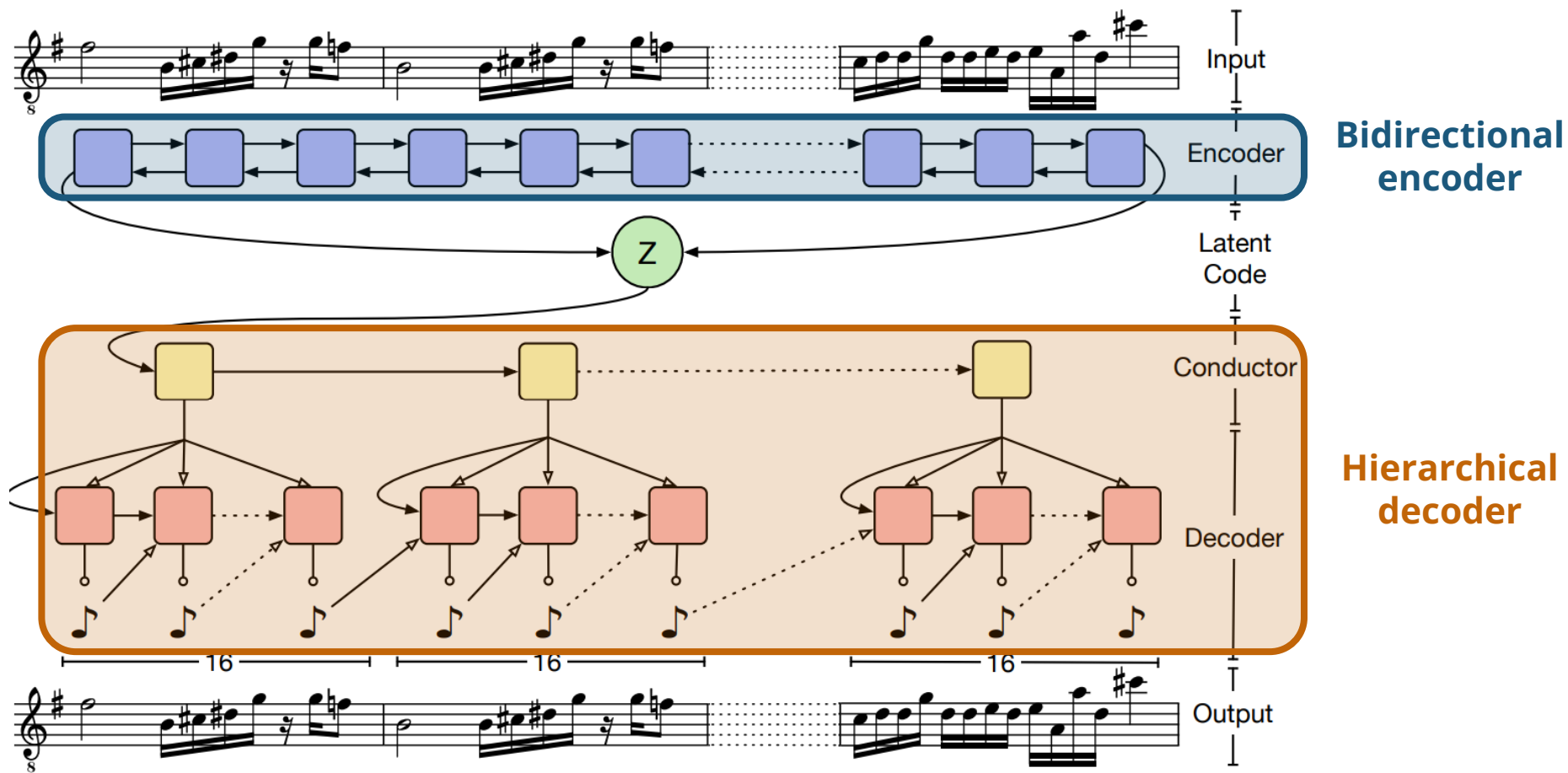


Data space



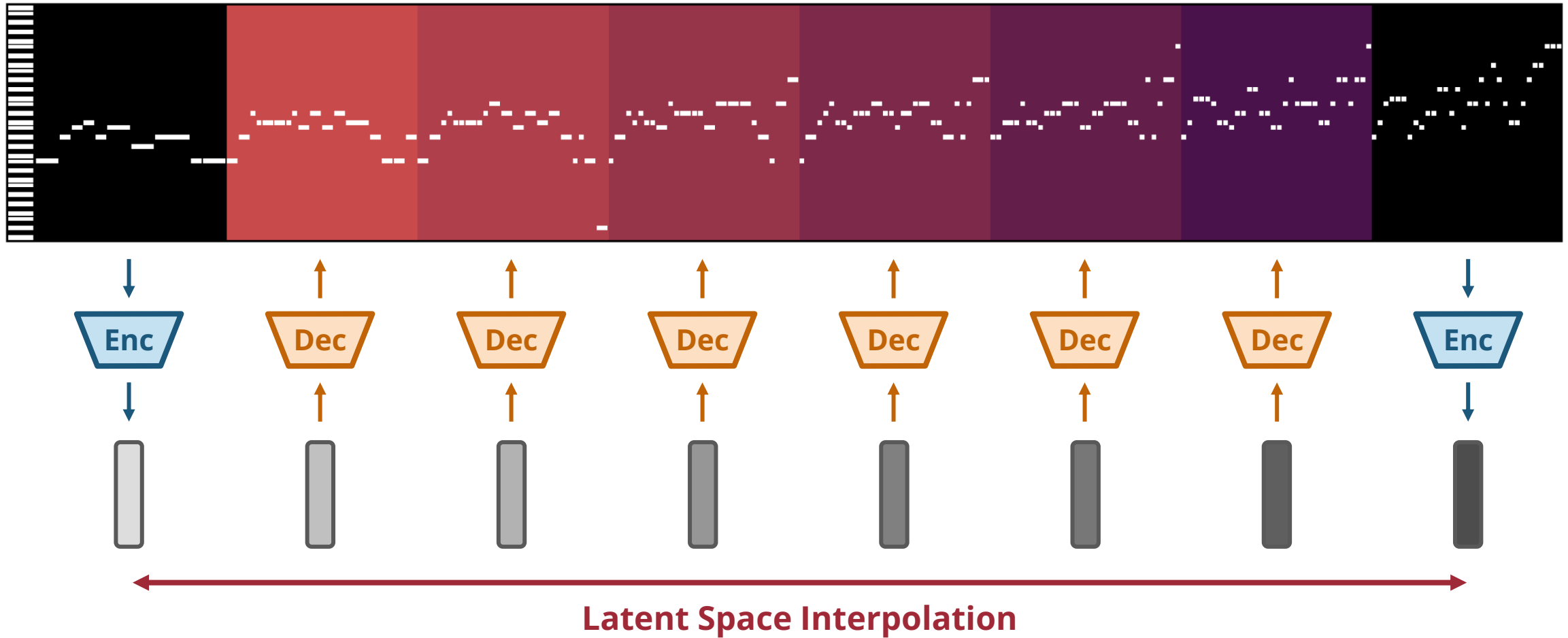
(Source: tensorflow.org)

# MusicVAE: A VAE for Symbolic Music (Roberts et al., 2018)



(Source: Roberts et al., 2018)

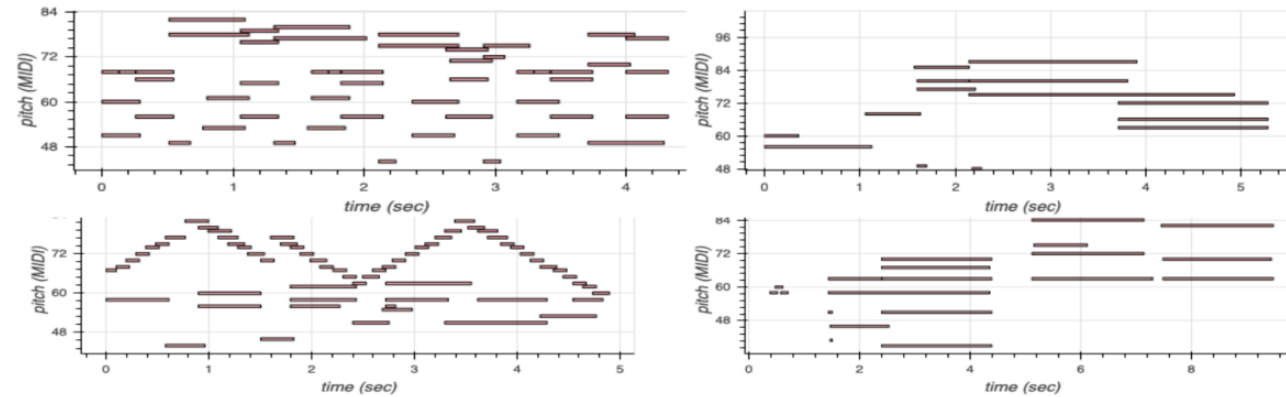
# Latent Space Interpolation for MusicVAE (Roberts et al., 2018)



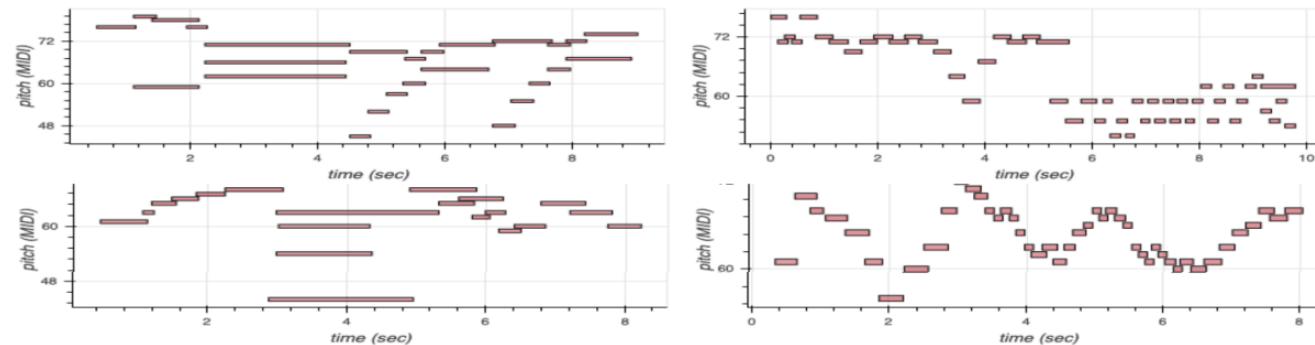
(Source: Roberts et al., 2018)

# Music FaderNet (Tan & Herremans, 2020)

## High Arousal → Low Arousal



## Low Arousal → High Arousal



(Source: Tan & Herremans, 2020)

[music-fadernets.github.io](https://music-fadernets.github.io)

# Music SketchNet (Chen et al., 2020)

The diagram illustrates the Music SketchNet architecture across three stages: Past Context, Generation, and Future Context. It features four staves:

- Original:** The target musical piece.
- Control Pitch:** A sequence of chords and triplets that guide the pitch of the generated music. Chords shown include {Ab5, Db6, Eb6, Gb6}, {C6, Eb6, Db6, F6, Db6}, {F6, Gb6, Ab6, Ab6, F6}, and {Db6, F6, Ab6, Bb6, Db6}.
- Control Rhythm:** A sequence of rhythmic patterns, represented by pink bars, that guide the rhythm of the generated music.
- Control Both:** A sequence of chords and triplets that guide both pitch and rhythm. A grey bar labeled "No Sketch" indicates a period where no sketch is provided.

Vertical lines separate the Past Context, Generation, and Future Context sections.

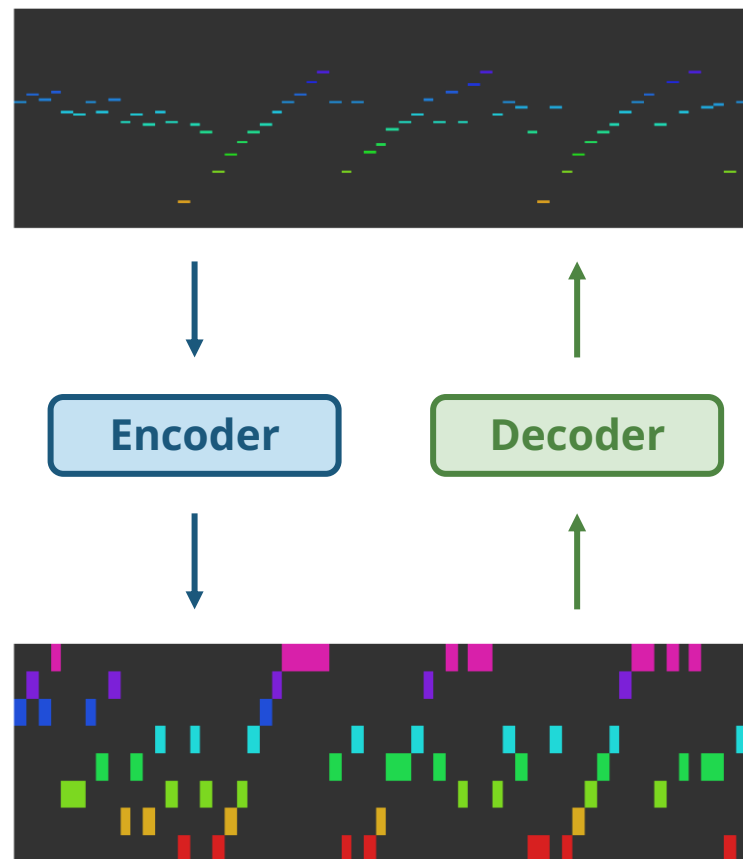
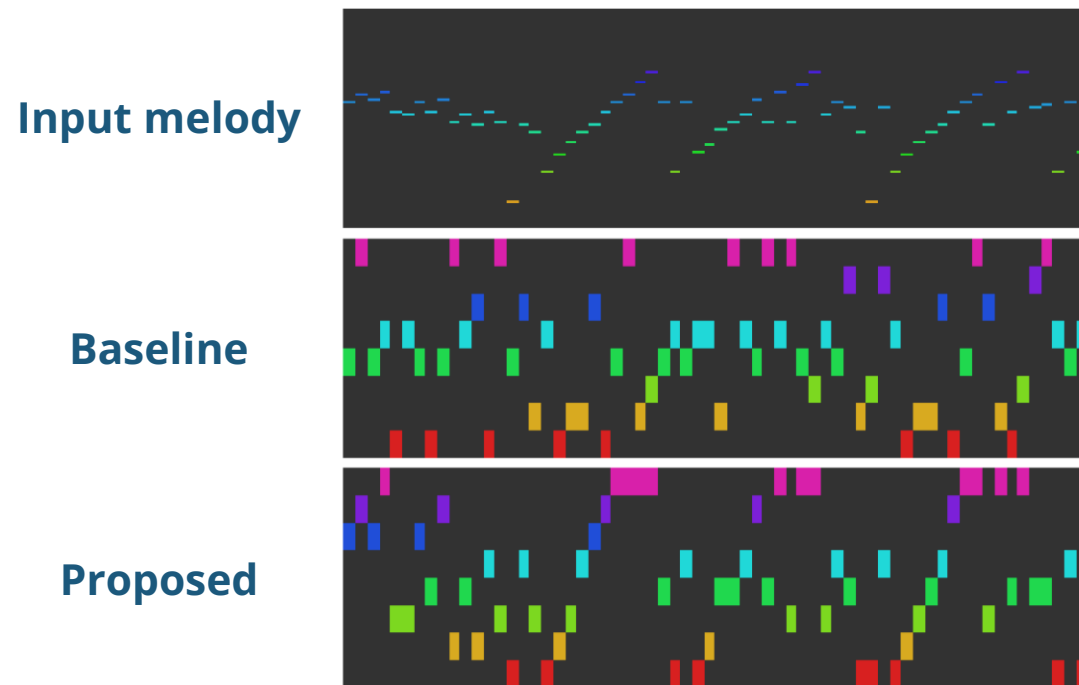
(Source: Chen et al., 2020)

# Piano Genie (Donahue et al., 2019)



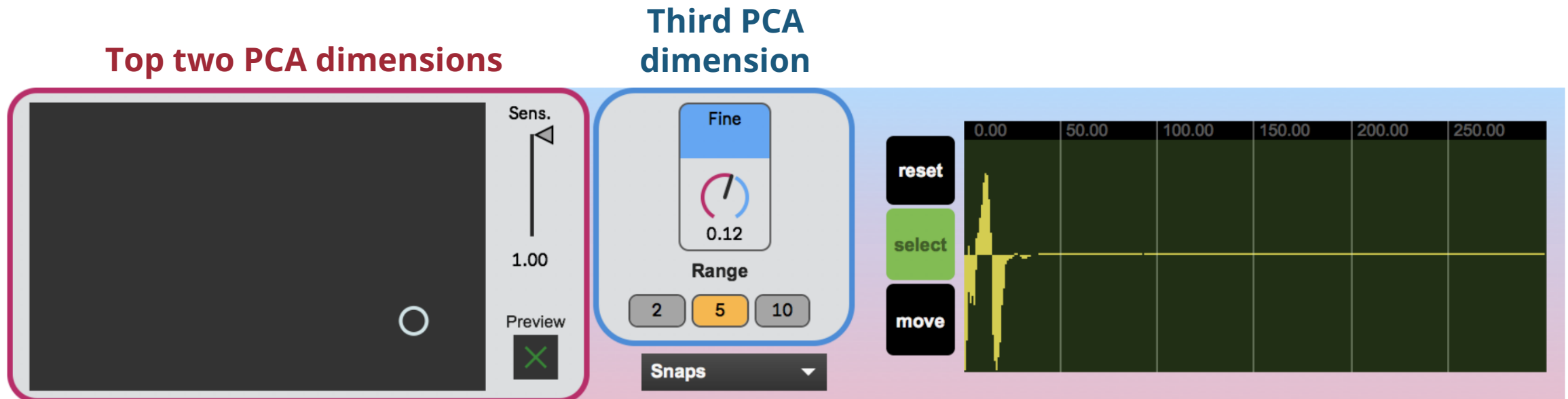
[youtu.be/YRb0XAnUpIk](https://youtu.be/YRb0XAnUpIk) & [magenta.tensorflow.org/pianogenie](https://magenta.tensorflow.org/pianogenie)

# Piano Genie (Donahue et al., 2019)



(Source: Donahue et al., 2019)

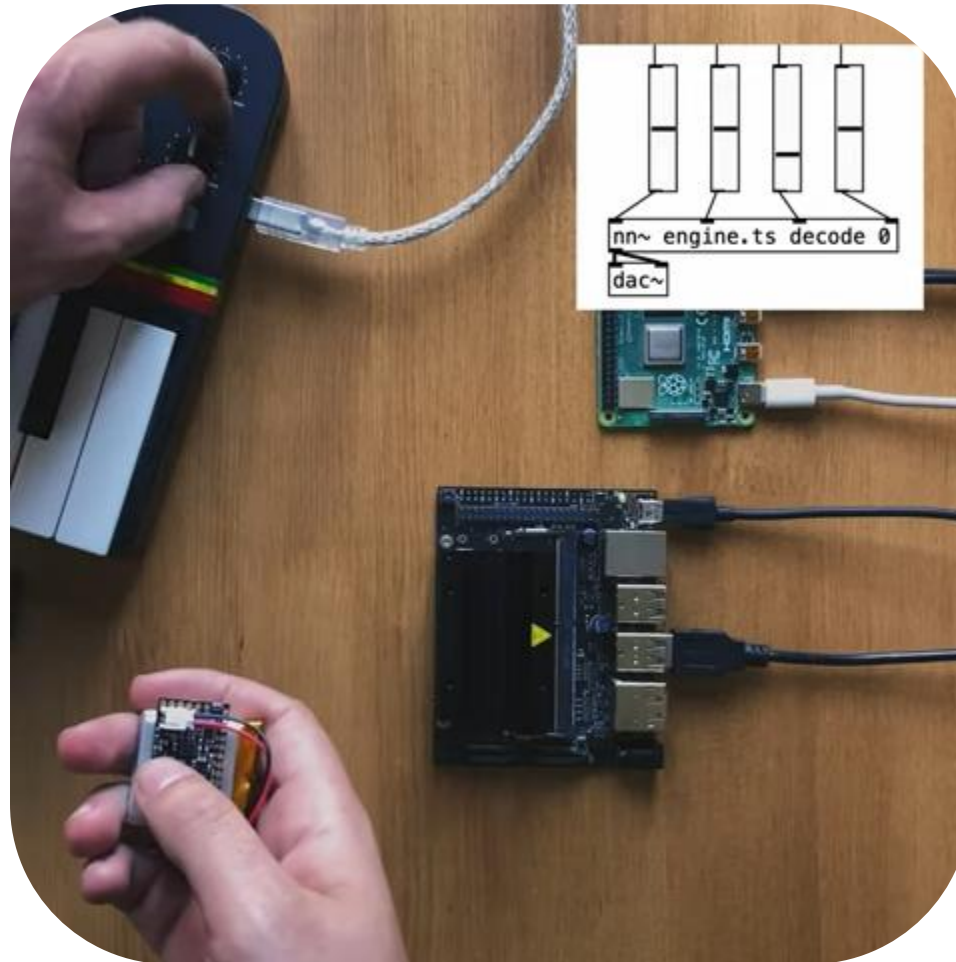
# Neural Drum Machine (Aouameur et al., 2019)



(Source: Aouameur et al., 2019)

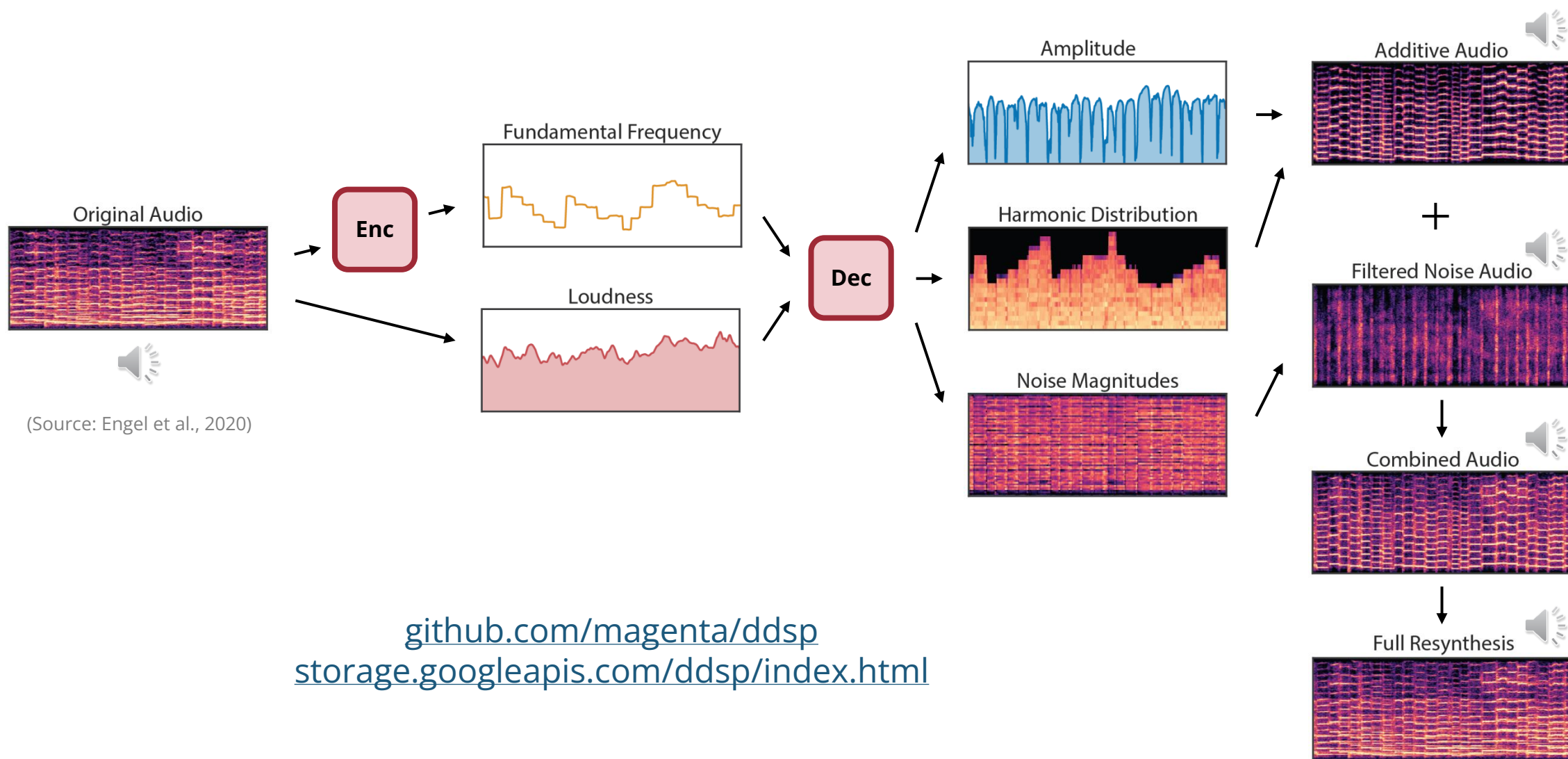
[drive.google.com/file/d/1DDo0\\_KnwkWirCM4t0PT8cp6uotsfuufj/view](https://drive.google.com/file/d/1DDo0_KnwkWirCM4t0PT8cp6uotsfuufj/view)

# RAVE: Real-time Audio Synthesis (Caillon & Esling, 2022)



[youtu.be/jAIRf4nGgYI](https://youtu.be/jAIRf4nGgYI)

# Differentiable DSP (DDSP) (Engel et al., 2020)



# Entering Demons & Gods by Yaboi Hanoi (2022)



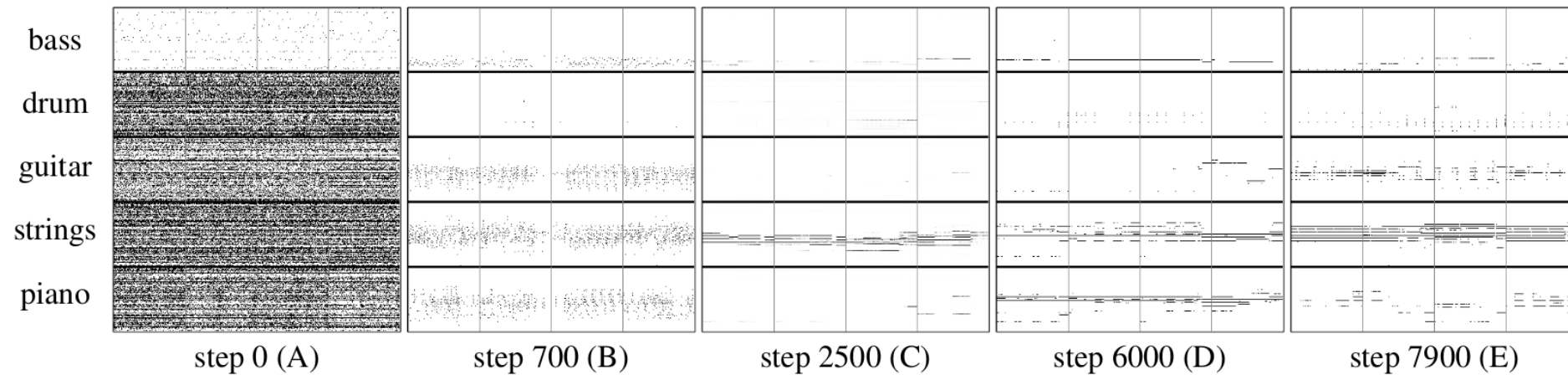
[youtu.be/PbrRoR3nEVw](https://youtu.be/PbrRoR3nEVw)

[soundcloud.com/yaboihanoi/enter-demons-and-gods](https://soundcloud.com/yaboihanoi/enter-demons-and-gods)



## Next Lecture

# Generative Adversarial Nets



(Source: Dong et al., 2018)