PAT 463/563 (Fall 2026)

Music & AI

Lecture 15: Piano Roll-based Music Generation

Instructor: Hao-Wen (Herman) Dong

UNIVERSITY OF MICHIGAN

Two Paradigms of Symbolic Music Generation

Text-based

- Treat music like **text**
- Sharing models with **natural language processing (NLP)**
  - RNNs, LSTMs, Transformers, etc.

```
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, ...
```

Image-based

- Treat music like **images**
- Sharing models with **computer vision (CV)** & **computer graphics (CG)**
  - GANs, VAEs, diffusion models, etc.

Today's topic!

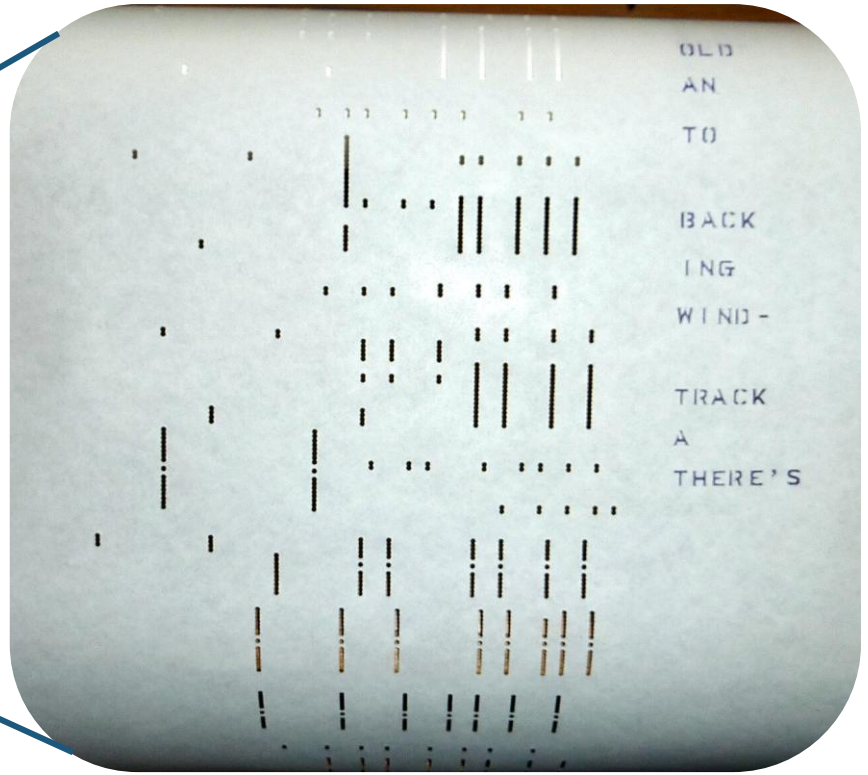


Piano Roll Representation

Piano Rolls



(Source: Draconichiaro)



(Source: Tangerineduel)

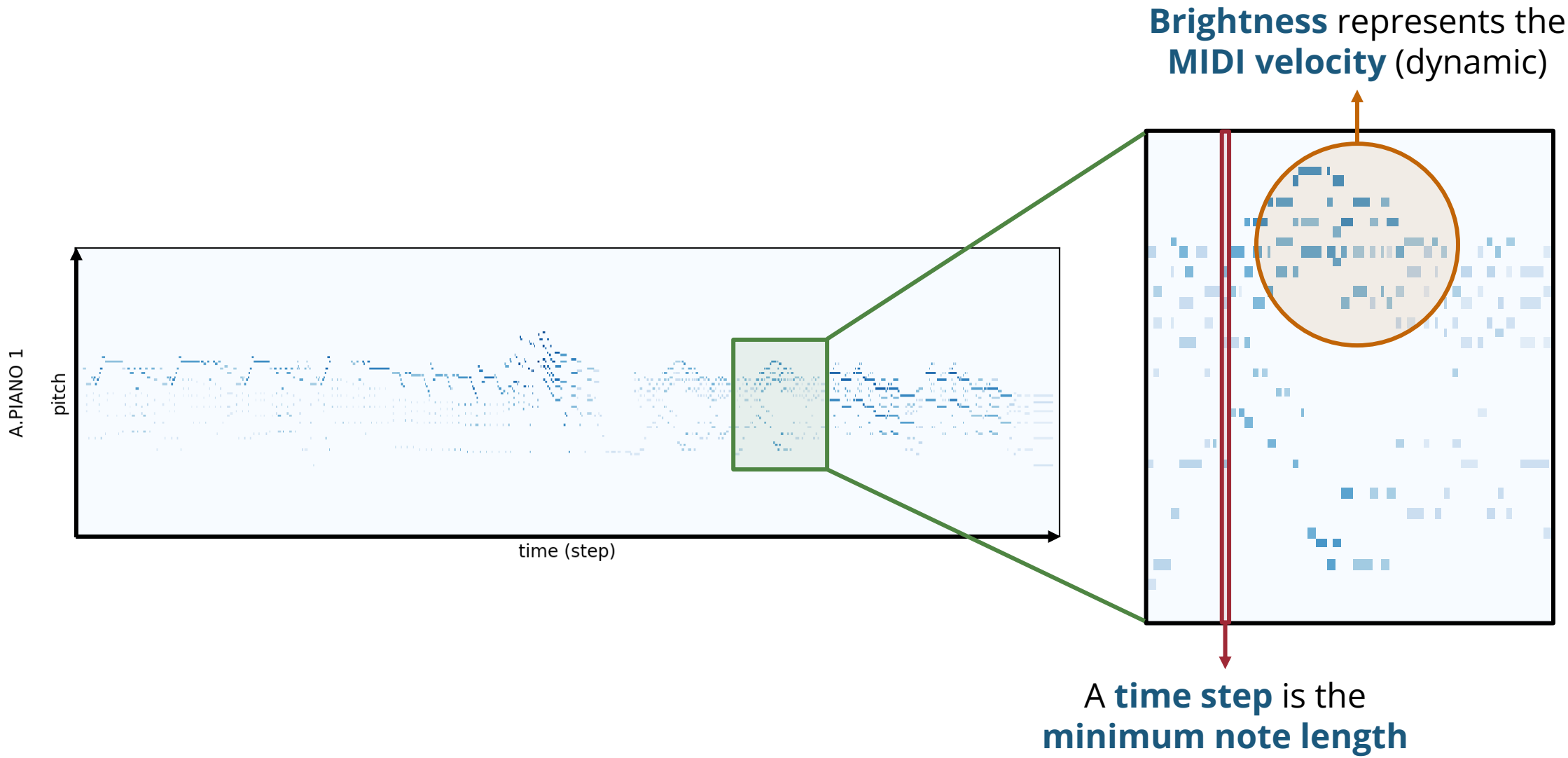
Draconichiaro, CC By-SA 4.0, via Wikimedia Commons
Tangerineduel, CC By-SA 4.0, via Wikimedia Commons

Player Pianos



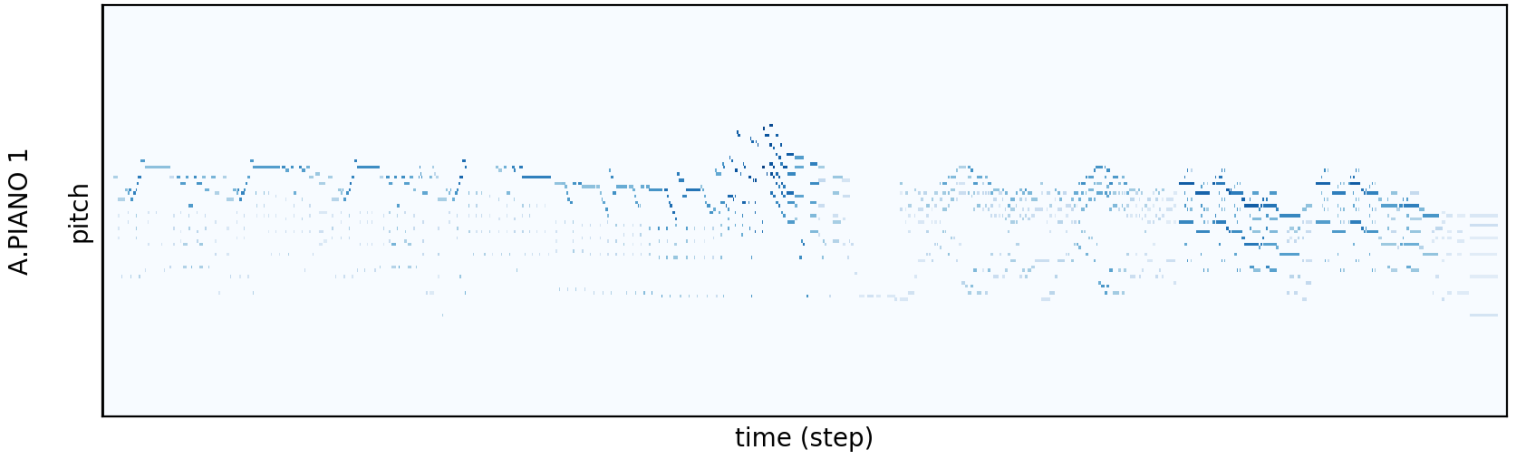
youtu.be/07krQ661fok

Piano Roll Representation

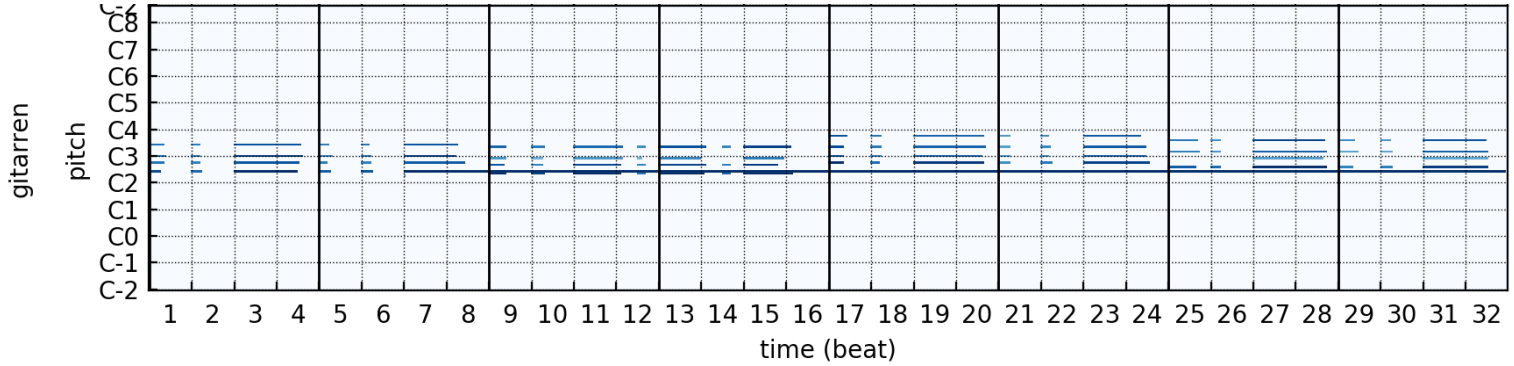


Piano Roll Representation

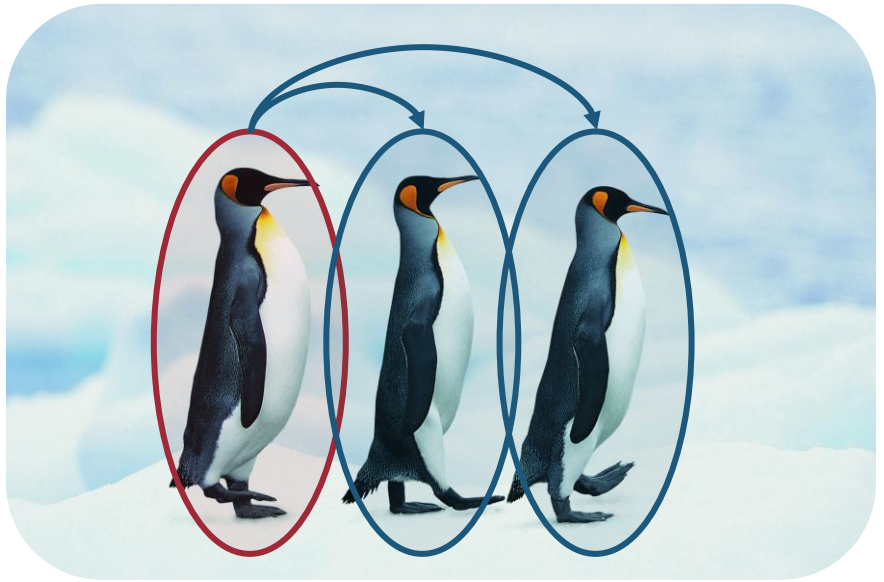
With expressive timing



Without expressive timing



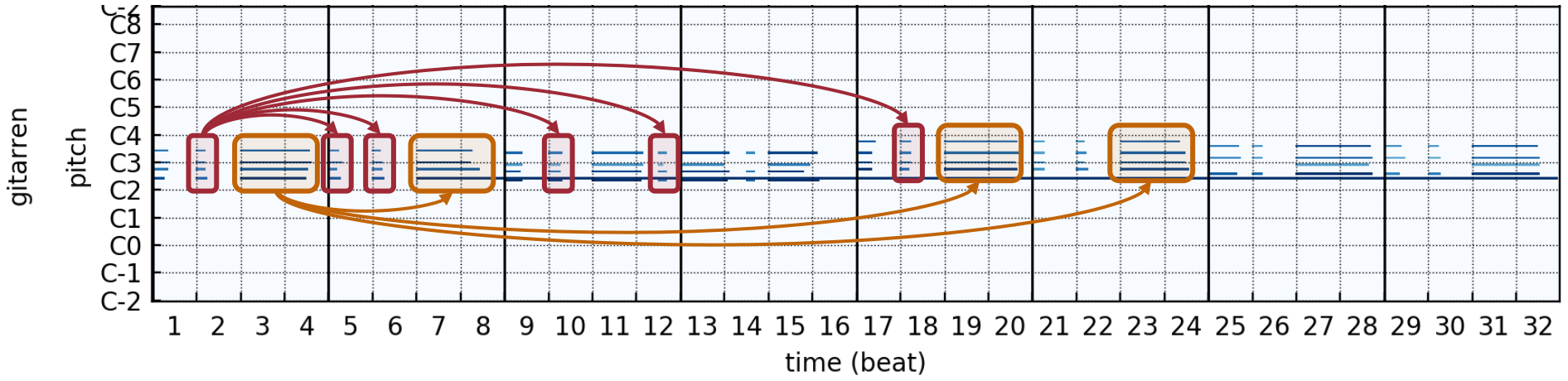
Reusable Pattern Detectors



Reusable Pattern Detectors



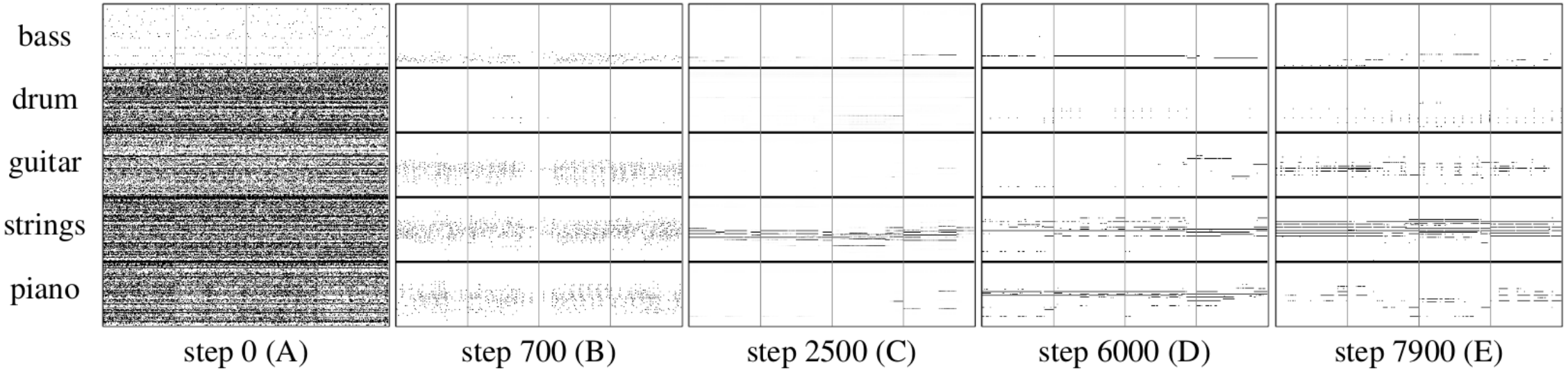
Why Piano Rolls?



Many musical patterns like melodies, chords, scales and arpeggios are translational invariant in the temporal and pitch axes

MuseGAN: A GAN for Pianorolls (Dong et al., 2018)

The generator improves over time **So does the discriminator!**



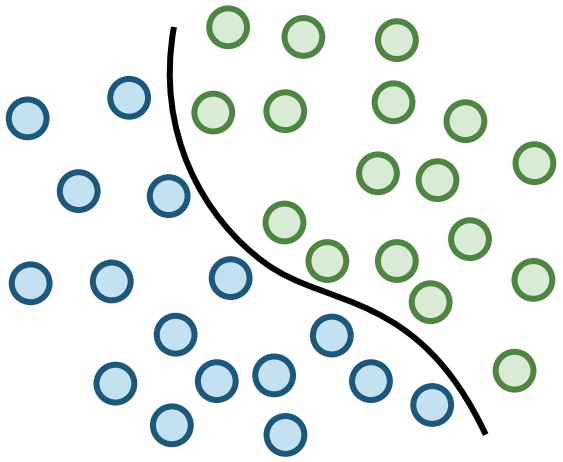
(Source: Dong et al., 2018)

Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang, "MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment," *AAAI*, 2018.

Generative Adversarial Nets (GANs)

Discriminative vs Generative Models

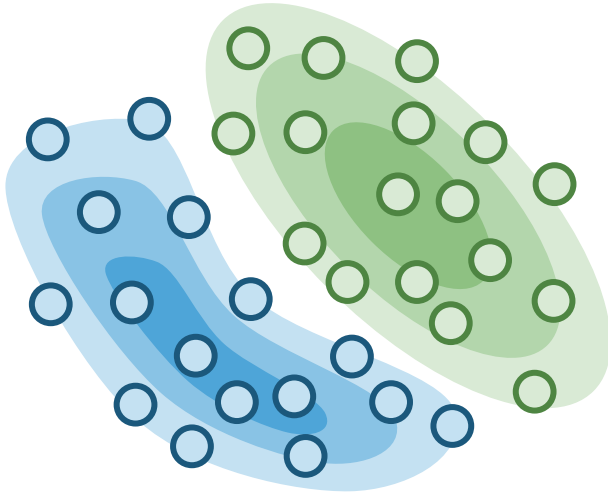
Discriminative



Discriminative models learn the decision boundary

$P(y|x)$

Generative



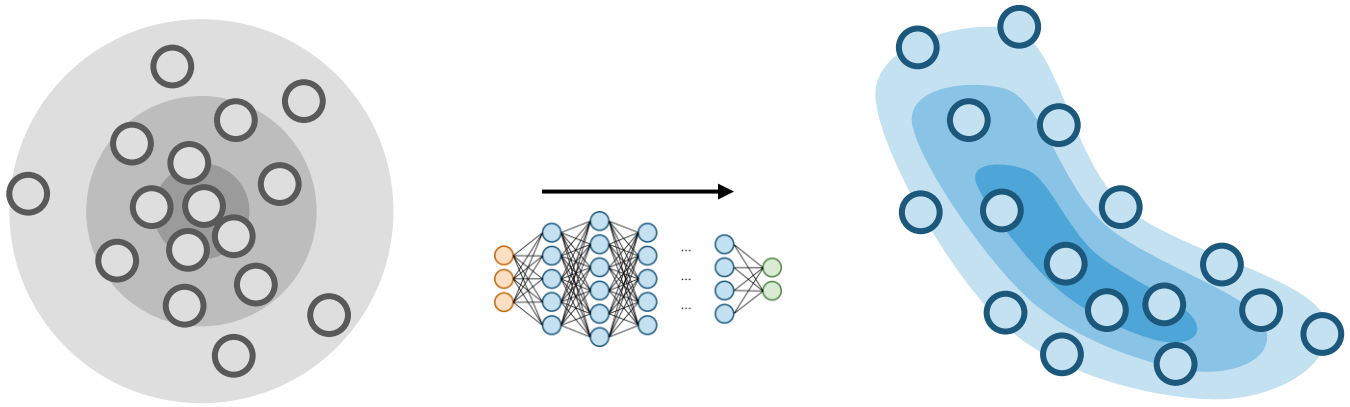
Generative models learn the underlying distribution

$P(x)$ or $P(x|y)$

Generating Data from a Random Distribution

Random distribution

Data distribution



$P(z)$

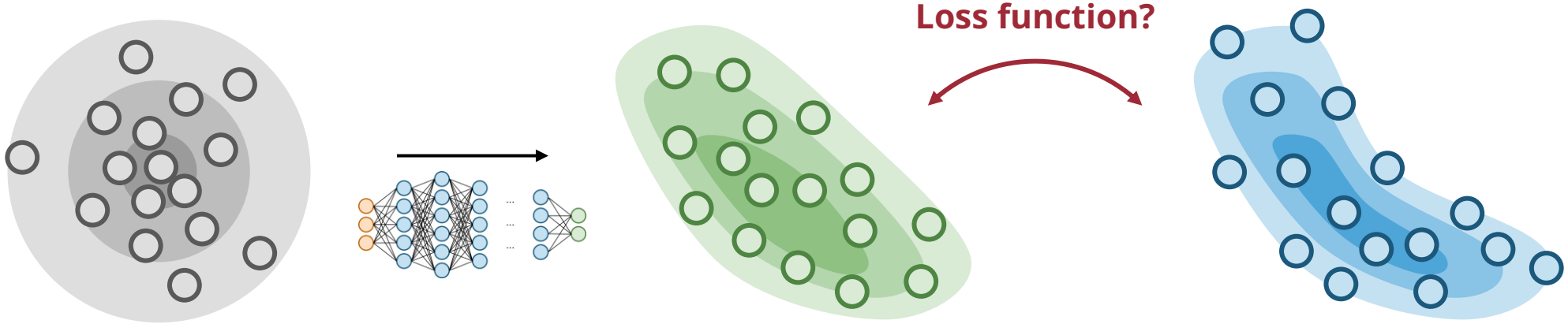
$P(x)$

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

A Loss Function for Distributions

Random distribution

Data distribution

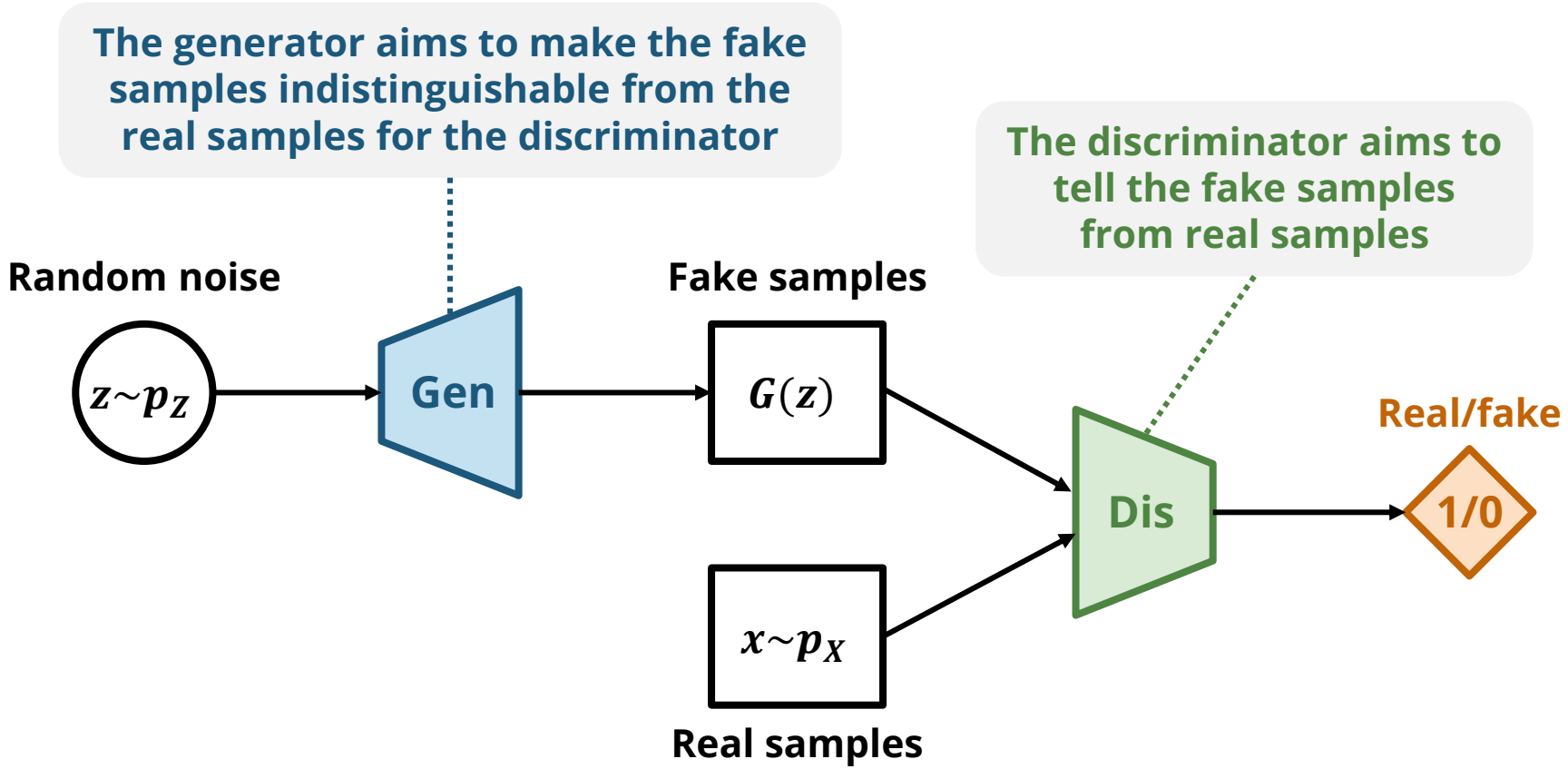


$P(z)$ $P(\hat{x})$ $P(x)$

Unfortunately, no easy way to measure the difference from samples

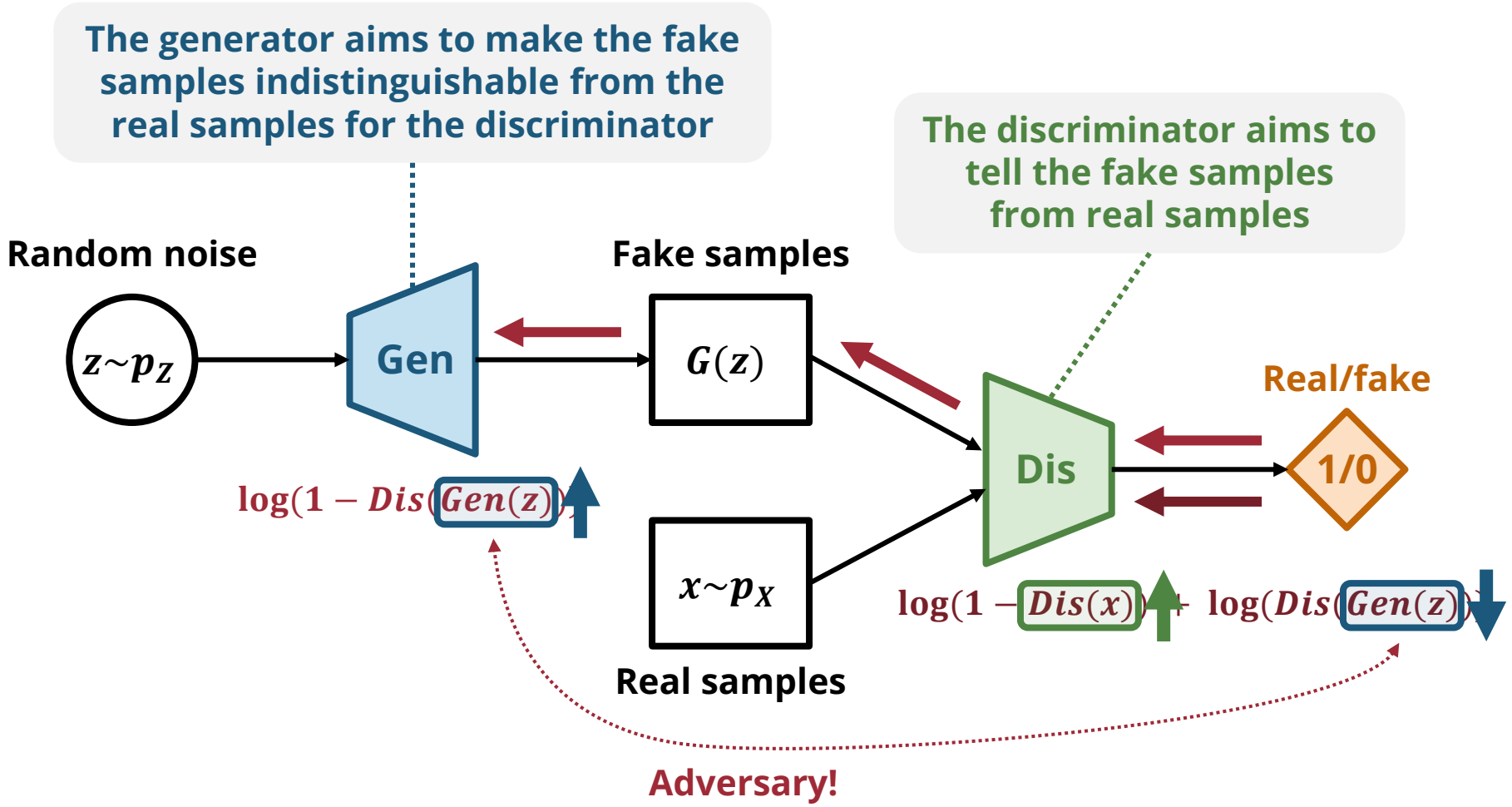
But what about another neural network!?

Generative Adversarial Nets (GANs) (Goodfellow et al., 2014)



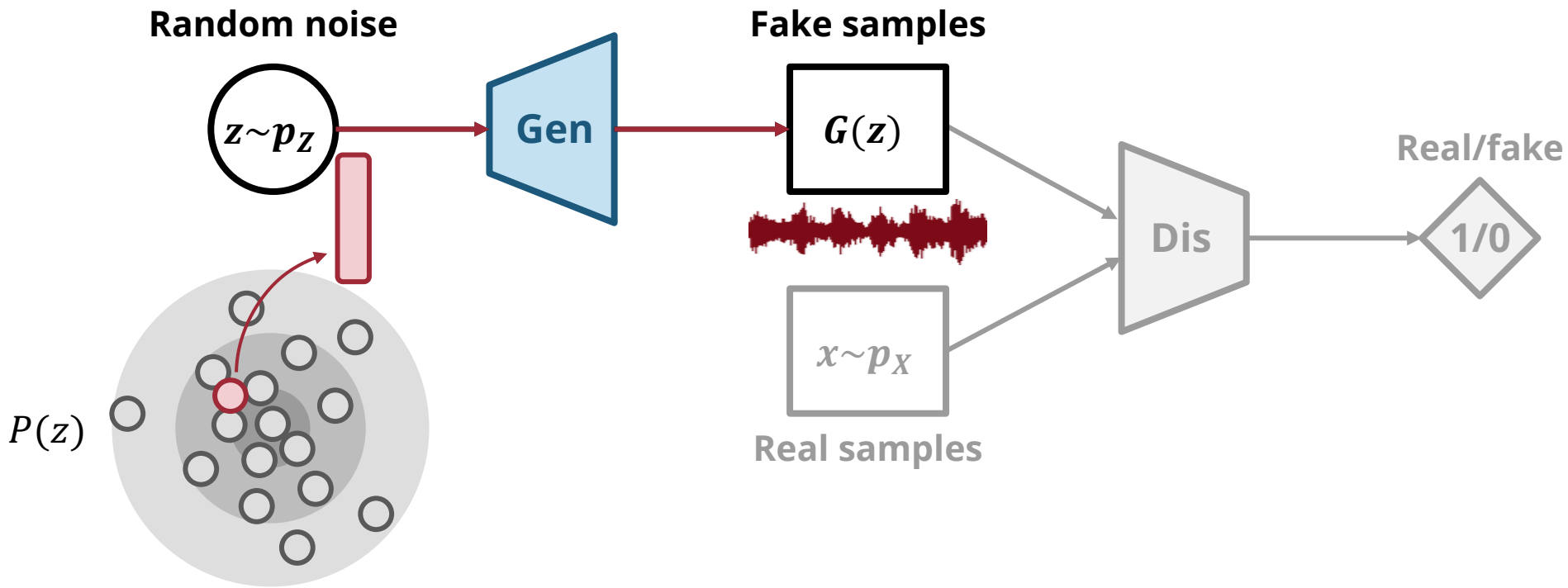
Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative Adversarial Networks," *NeurIPS*, 2014.

Generative Adversarial Nets (GANs): Training



Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative Adversarial Networks," *NeurIPS*, 2014.

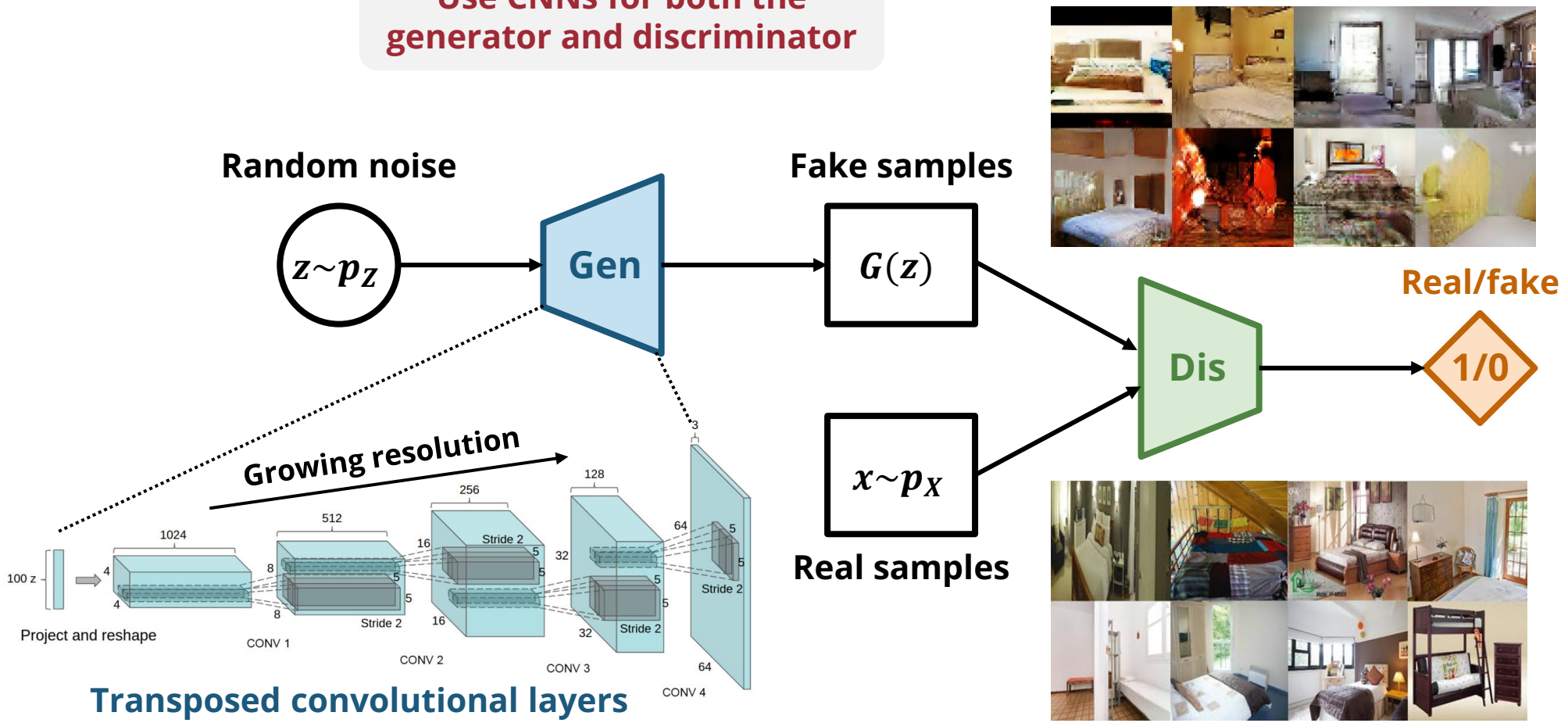
Generative Adversarial Nets (GANs): Generation



Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative Adversarial Networks," *NeurIPS*, 2014.

Deep Convolutional GANs (DCGANs) (Radford et al., 2014)

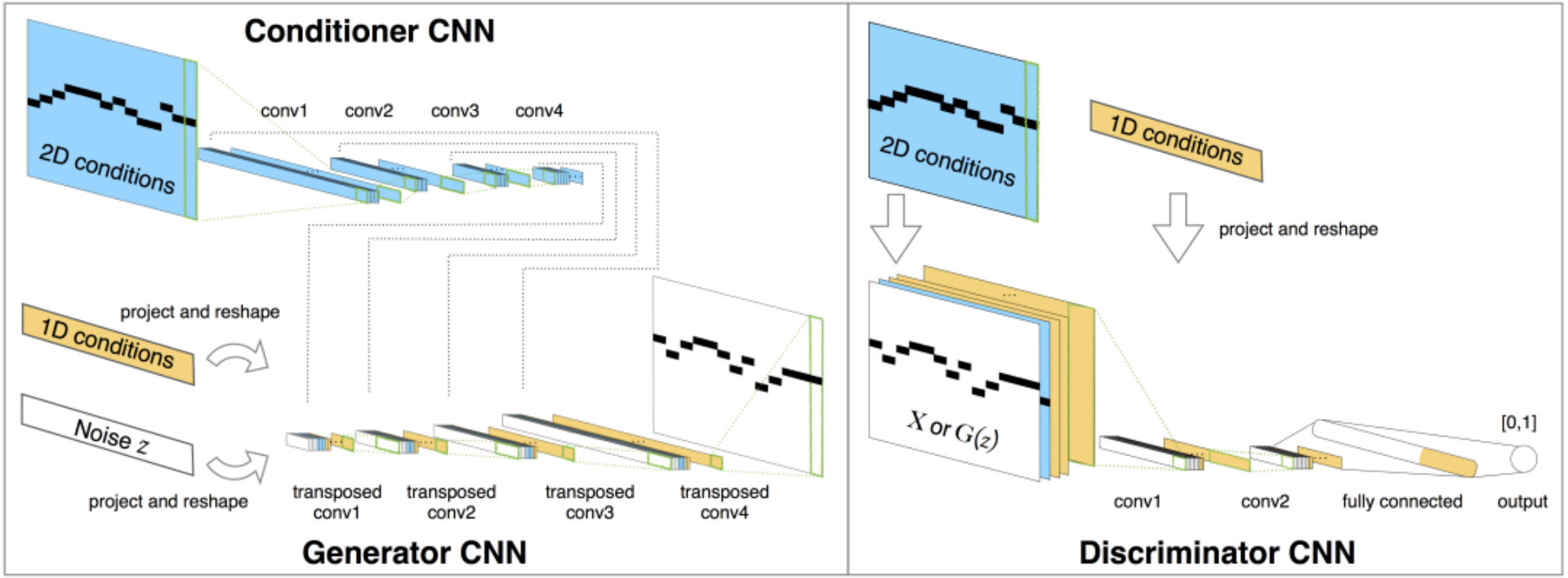
Use CNNs for both the generator and discriminator



Alec Radford, Luke Metz, and Soumith Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *ICLR*, 2016.

Generating Music using GANs

MidiNet (Yang et al., 2017)

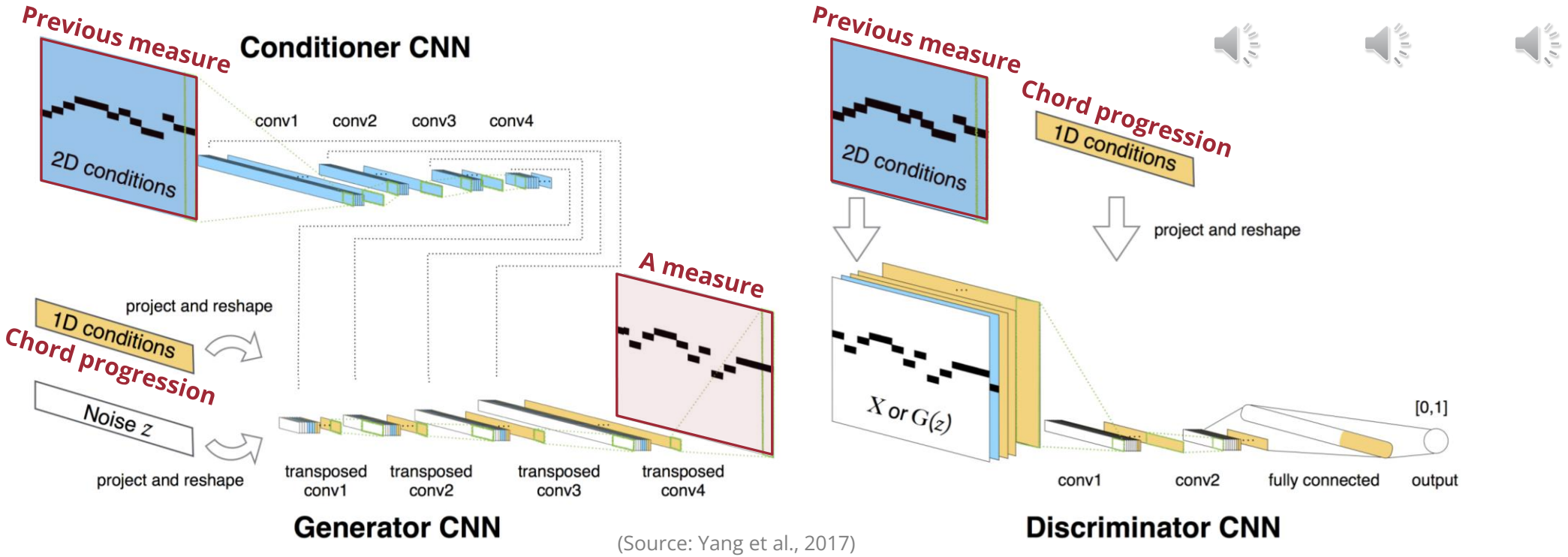


(Source: Yang et al., 2017)

Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang, "MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation," *ISMIR*, 2017.

MidiNet (Yang et al., 2017)

Examples of generated music

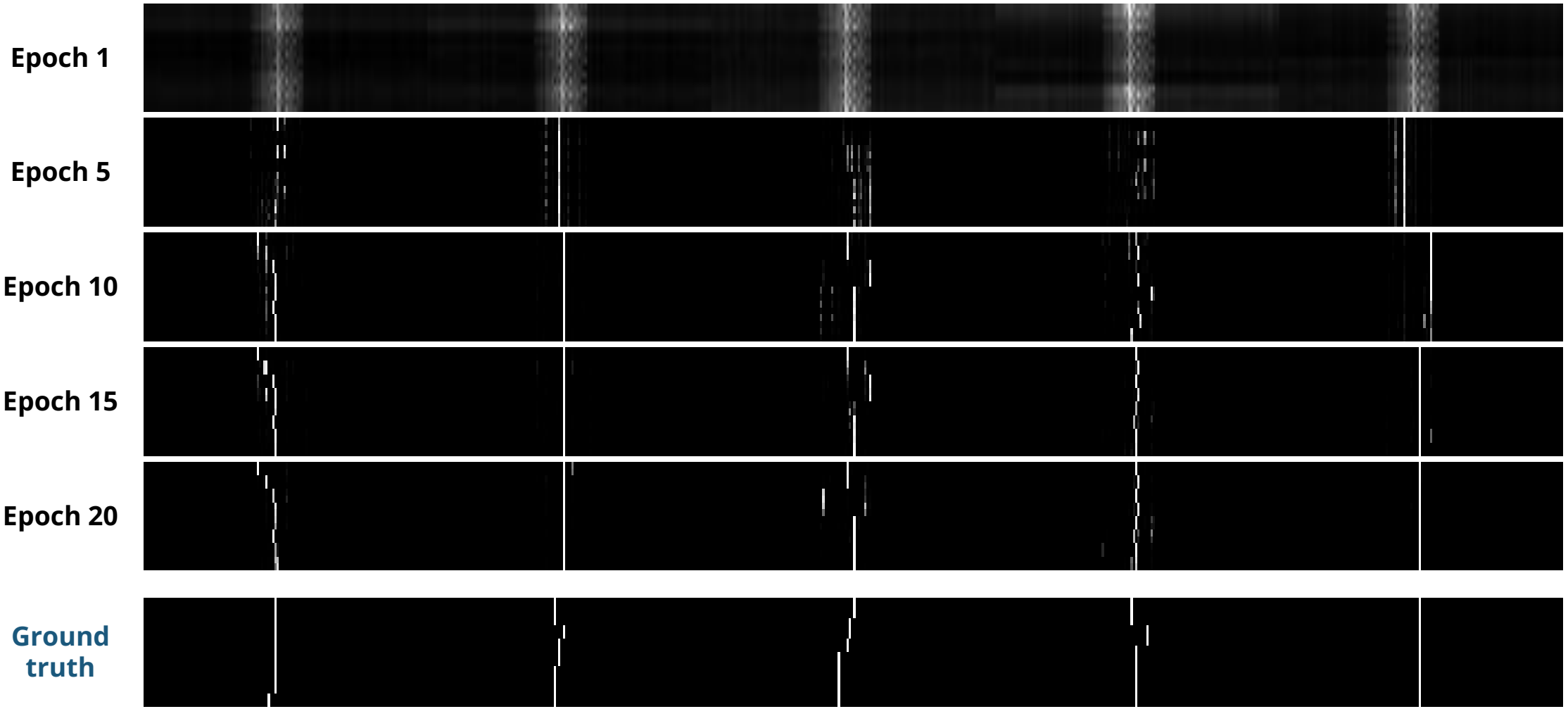


(Source: Yang et al., 2017)

Generate the next measure given the previous one (measure by measure)

Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang, "MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation," *ISMIR*, 2017.

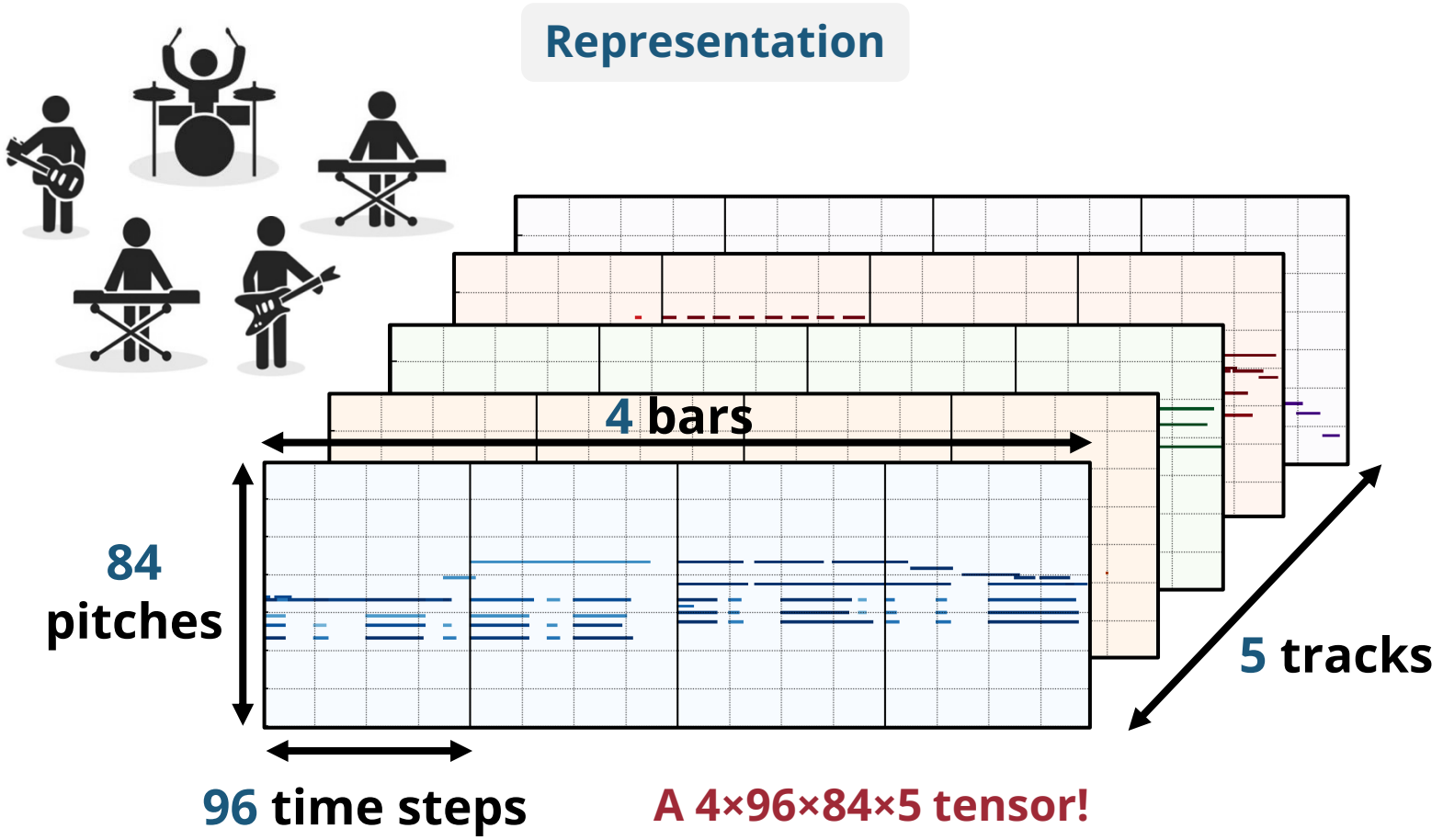
MidiNet (Yang et al., 2017)

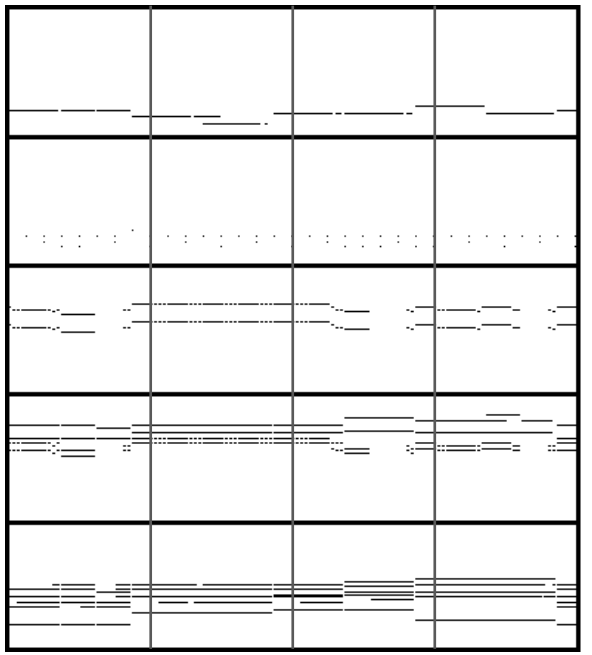


(Source: Yang et al., 2017)

Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang, "MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation," *ISMIR*, 2017.

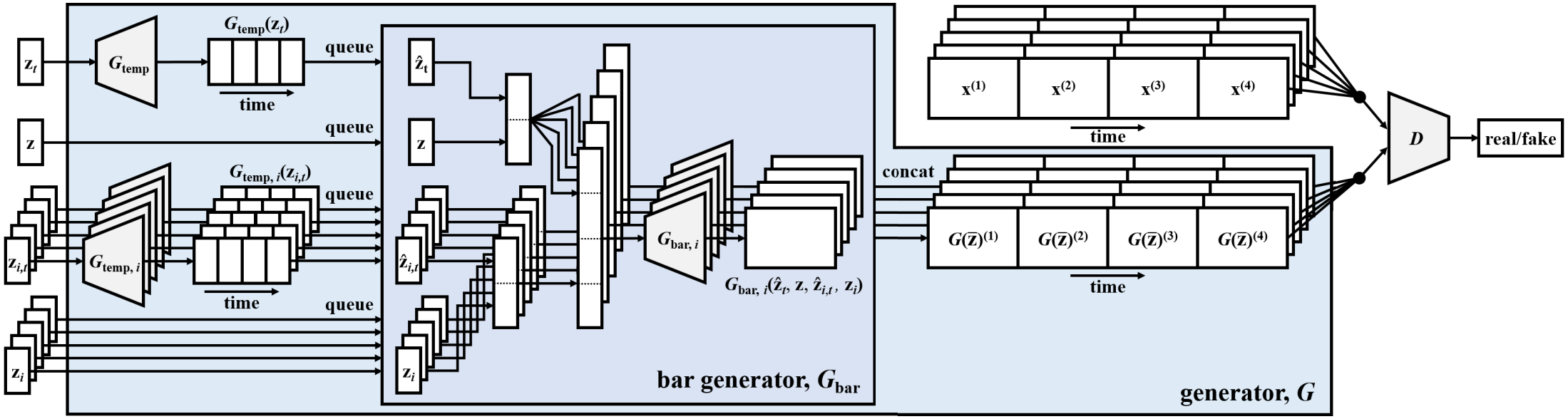
MuseGAN (Dong et al., 2018)





Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang, "MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment," *AAAI*, 2018.

MuseGAN (Dong et al., 2018)



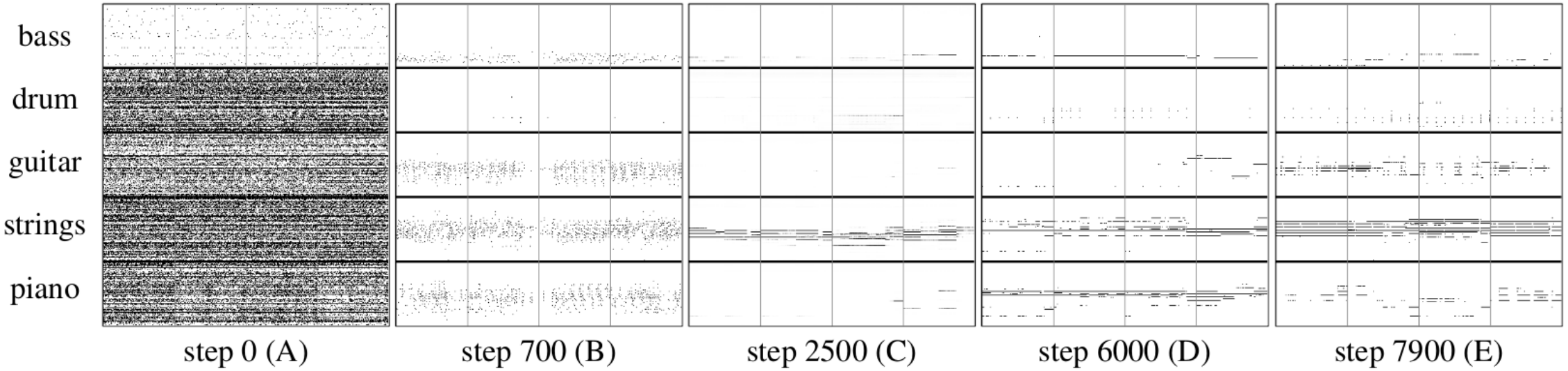
(Source: Dong et al., 2018)

Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang, "MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment," *AAAI*, 2018.

MuseGAN (Dong et al., 2018)

Examples of generated music

The generator improves over time



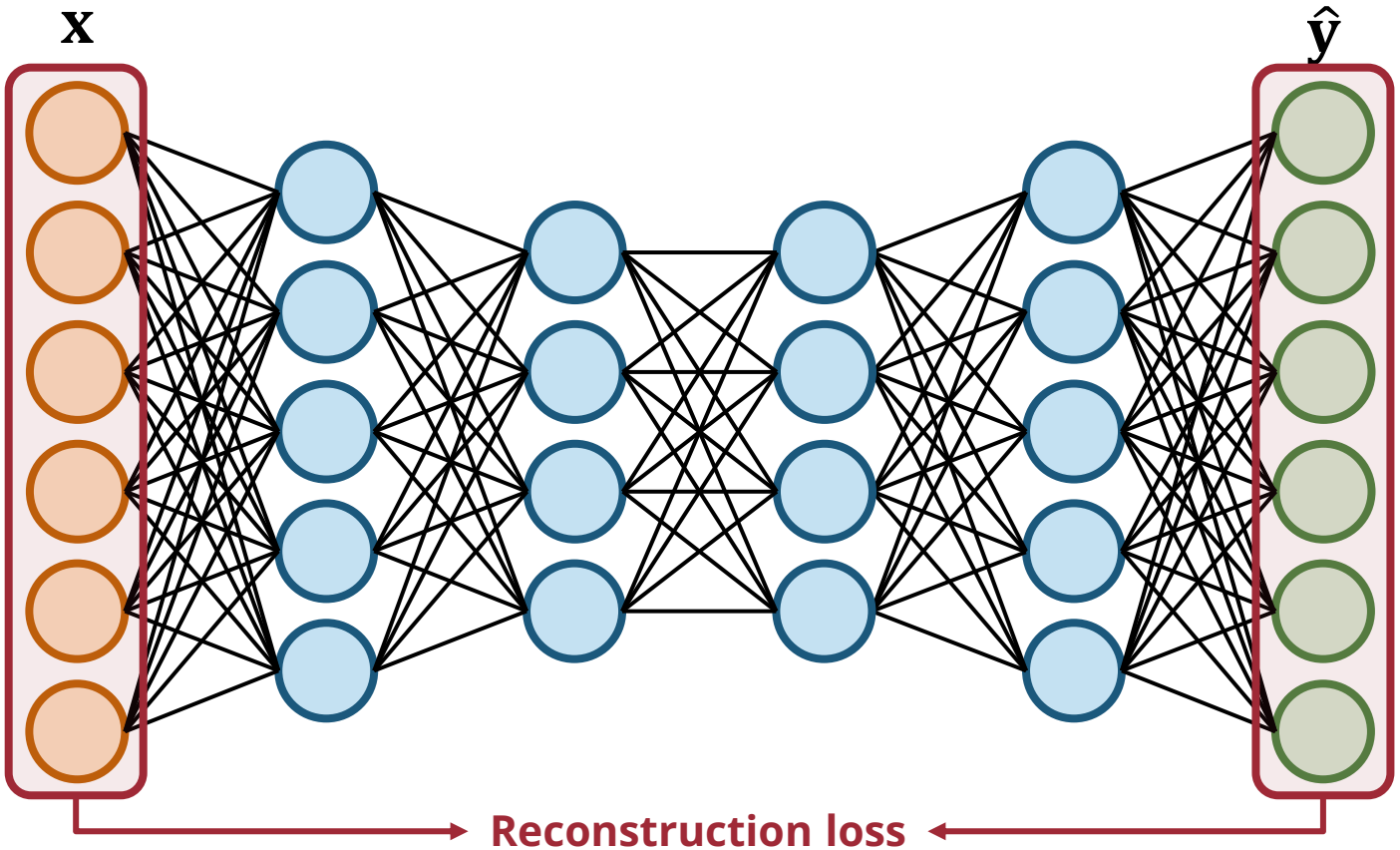
(Source: Dong et al., 2018)

Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang, "MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment," *AAAI*, 2018.

Diffusion Models

Autoencoders

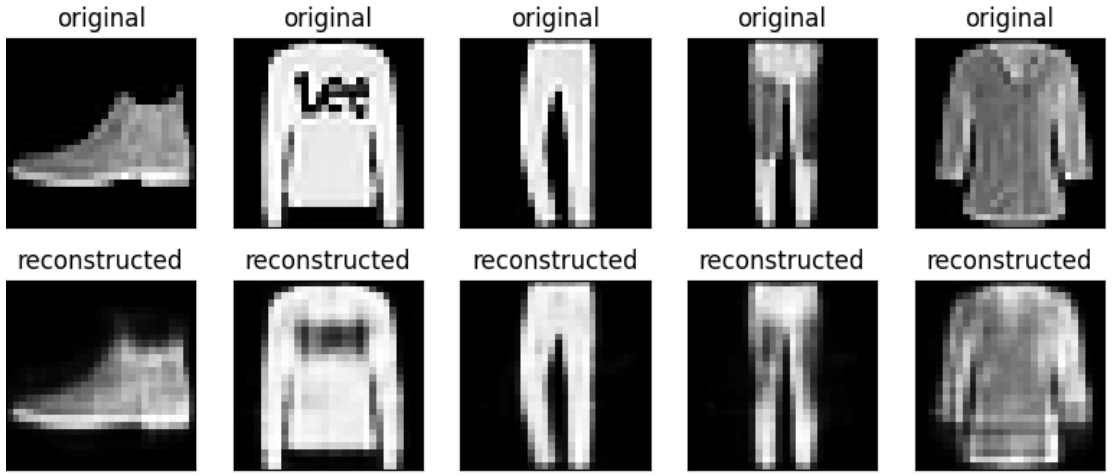
- A neural network where the **input and output are the same**



Autoencoders: Reconstruction Examples

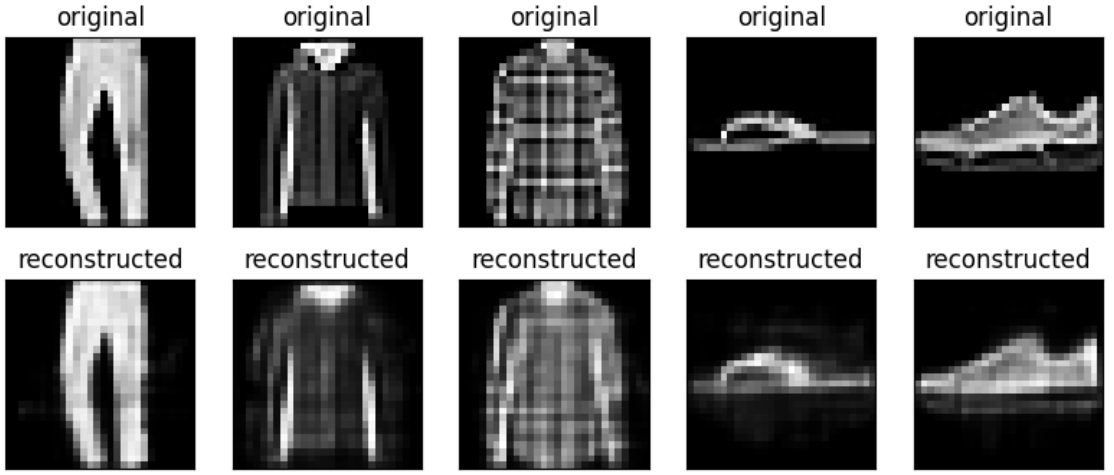
Original

Reconstructed



Original

Reconstructed



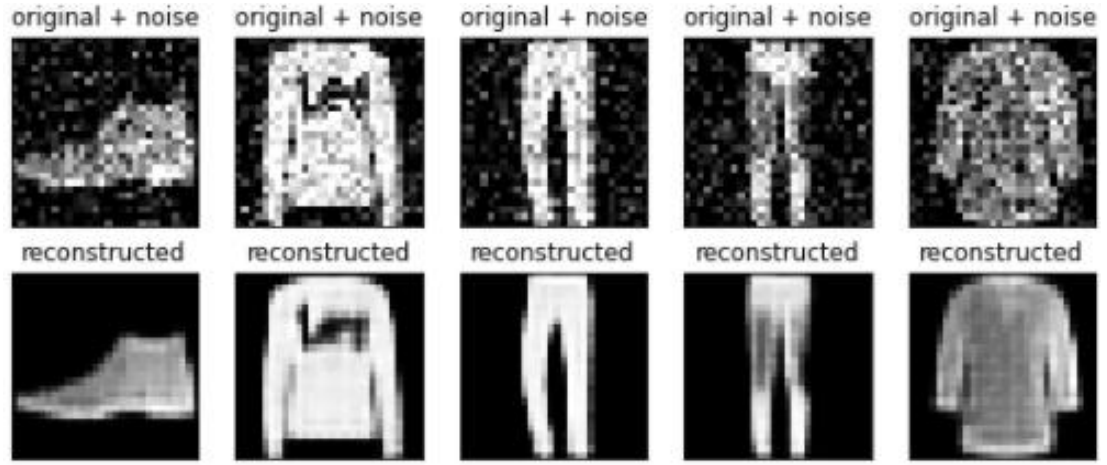
(Source: tensorflow.org)

tensorflow.org/tutorials/generative/autoencoder

Denoising Autoencoders (Pascal et al., 2008)

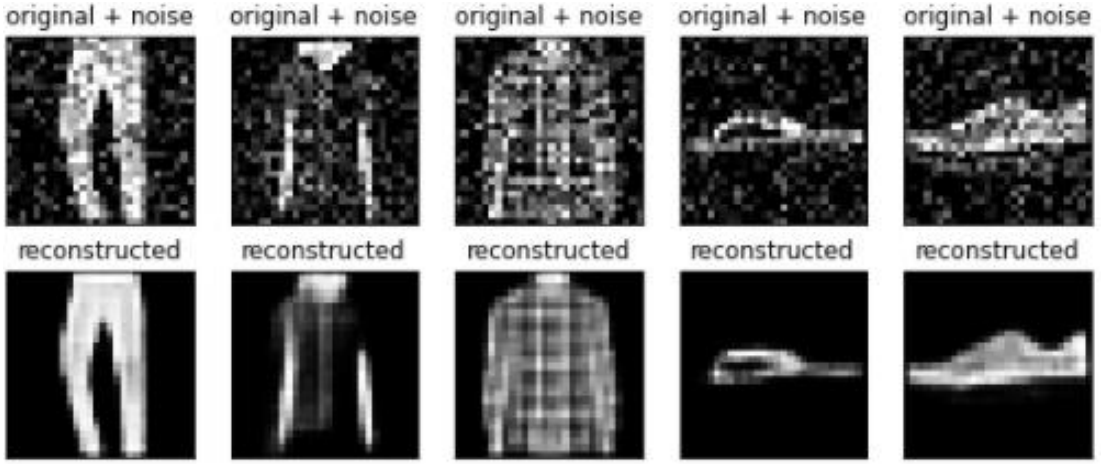
Noisy

Reconstructed



Noisy

Reconstructed



Denoising autoencoders learn to reconstruct noisy inputs

(Source: tensorflow.org)

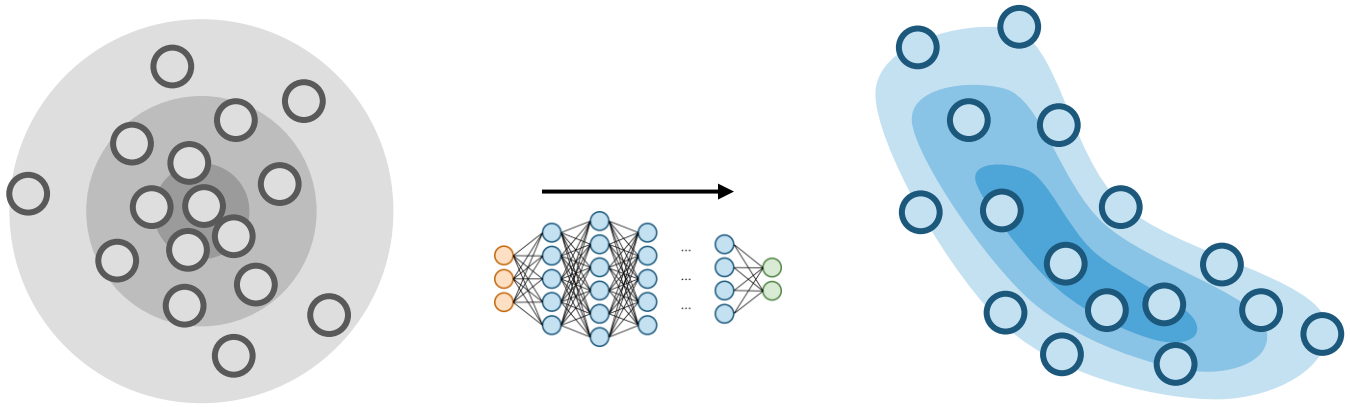
tensorflow.org/tutorials/generative/autoencoder

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol, "Extracting and Composing Robust Features with Denoising Autoencoders," *ICML*, 2008.
Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion," *PMLR*, 11(110):3371-2408, 2010.

Generating Data from a Random Distribution

Random distribution

Data distribution



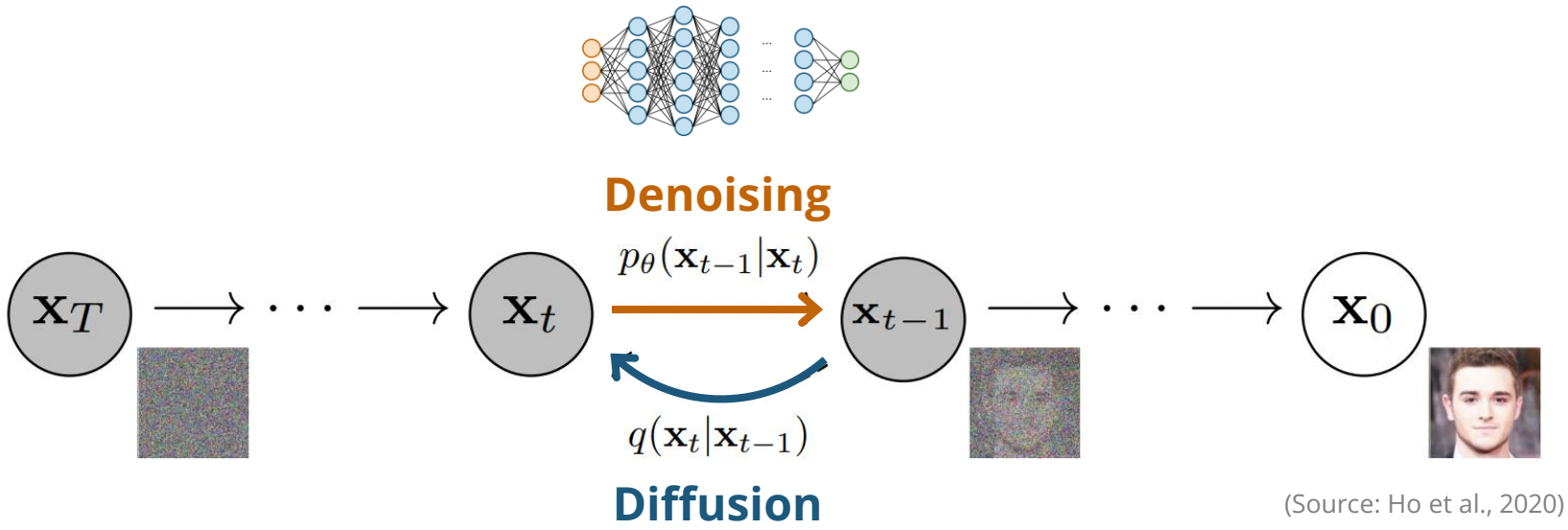
$P(z)$

$P(x)$

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

Diffusion Models (Ho et al., 2020)

- **Intuition**: Many denoising autoencoders stacked together

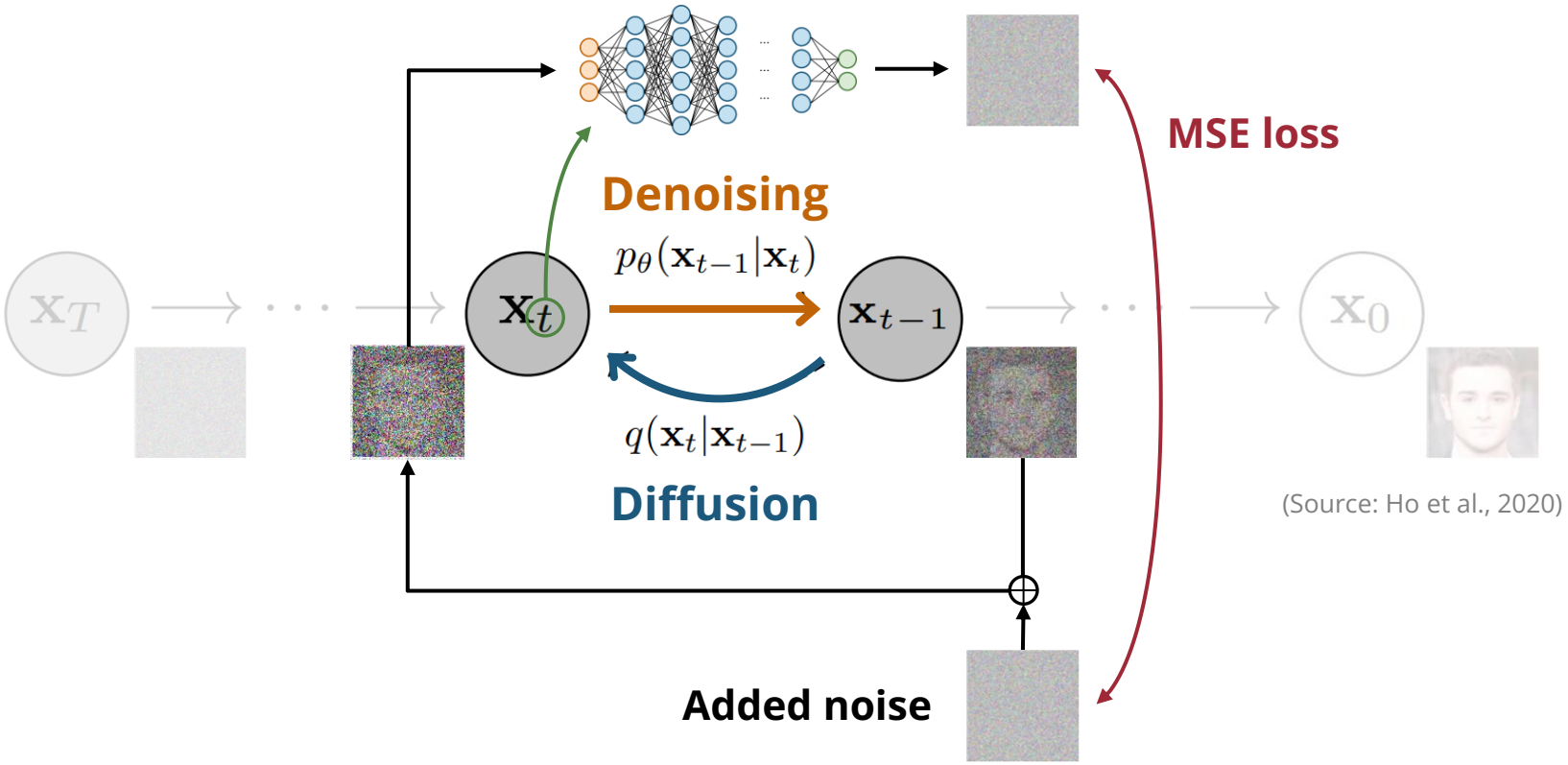


(Source: Ho et al., 2020)

Jonathan Ho, Ajay Jain, and Pieter Abbeel, "Denoising Diffusion Probabilistic Models," *NeurIPS*, 2020.

Diffusion Models – Training

- **Intuition**: Many denoising autoencoders stacked together

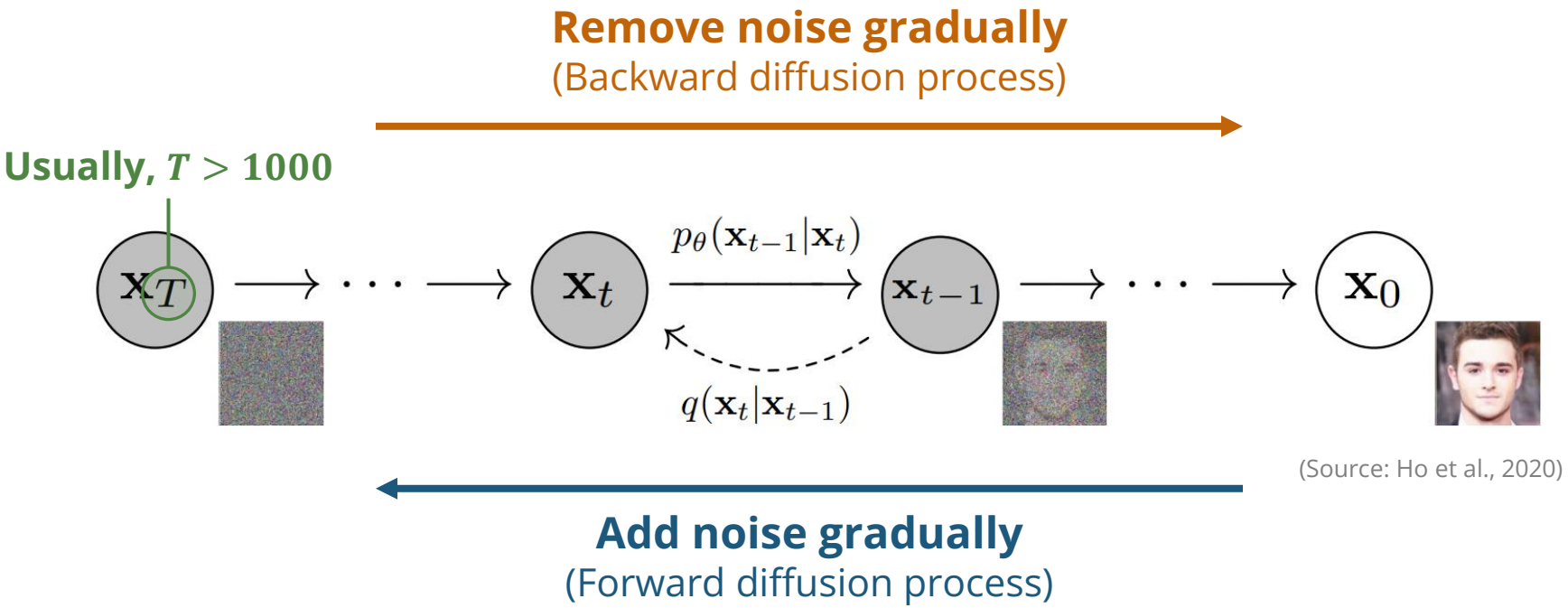


(Source: Ho et al., 2020)

Jonathan Ho, Ajay Jain, and Pieter Abbeel, "Denoising Diffusion Probabilistic Models," *NeurIPS*, 2020.

Diffusion Models (Ho et al., 2020)

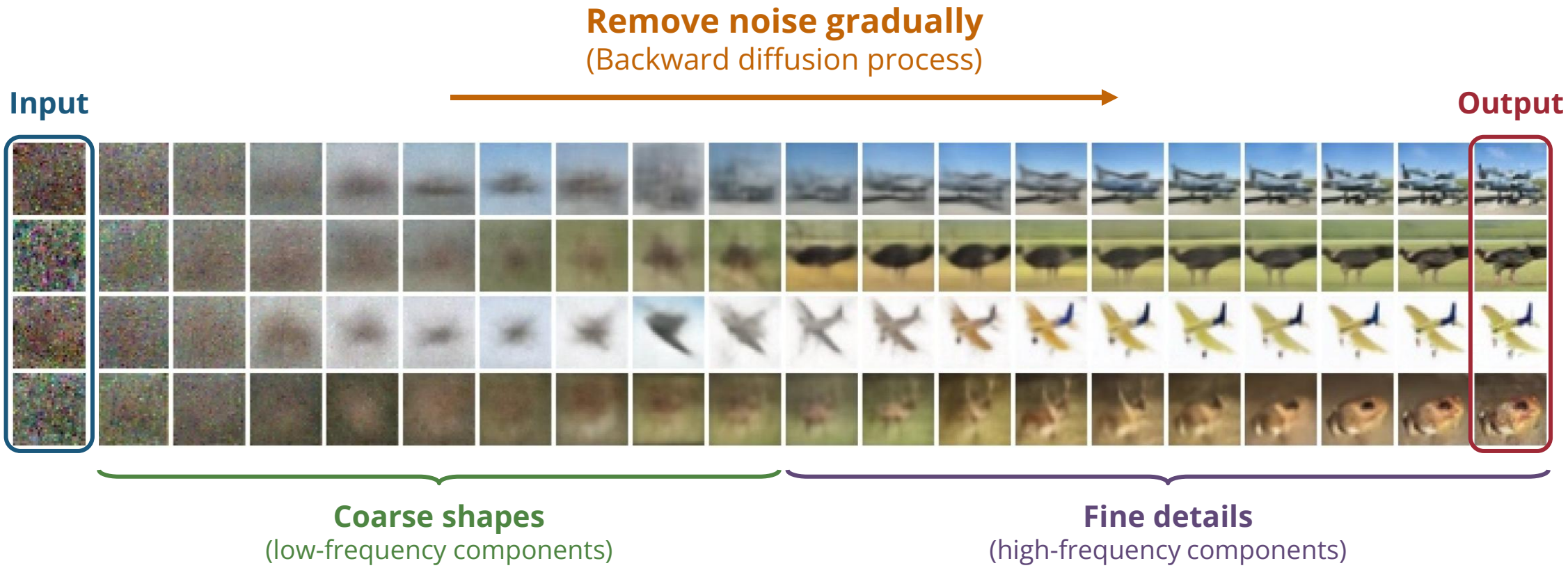
- **Intuition**: Many denoising autoencoders stacked together



(Source: Ho et al., 2020)

Jonathan Ho, Ajay Jain, and Pieter Abbeel, "Denoising Diffusion Probabilistic Models," *NeurIPS*, 2020.

Diffusion Models – Generation

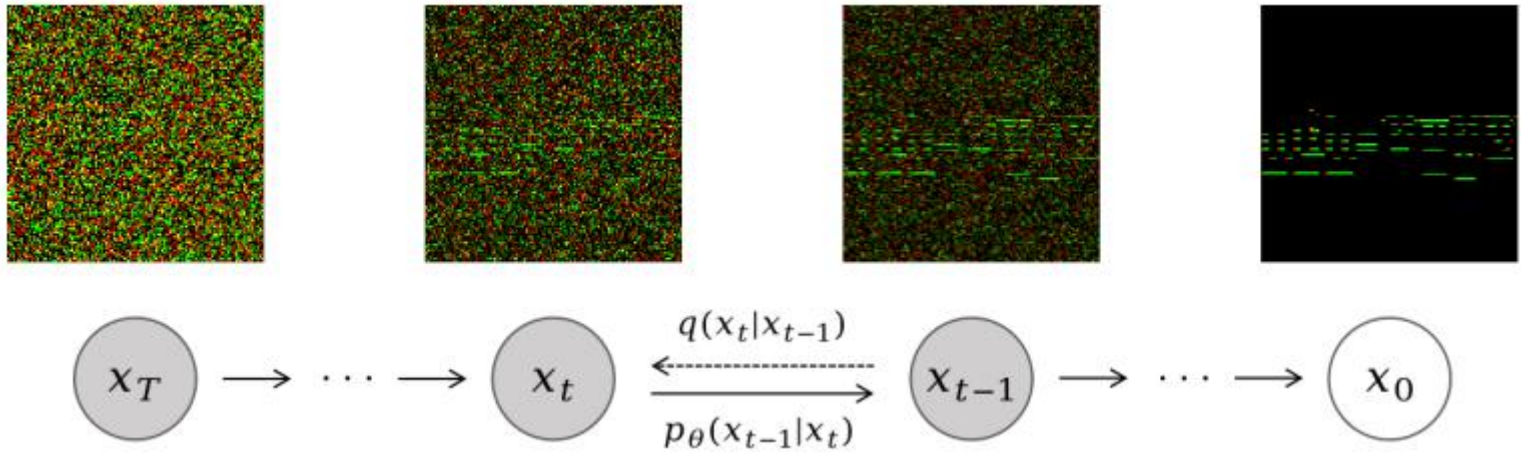


(Source: Ho et al., 2020)

Jonathan Ho, Ajay Jain, and Pieter Abbeel, "Denoising Diffusion Probabilistic Models," *NeurIPS*, 2020.

Generating Music using Diffusion Models

Polyffusion (Min et al., 2023)

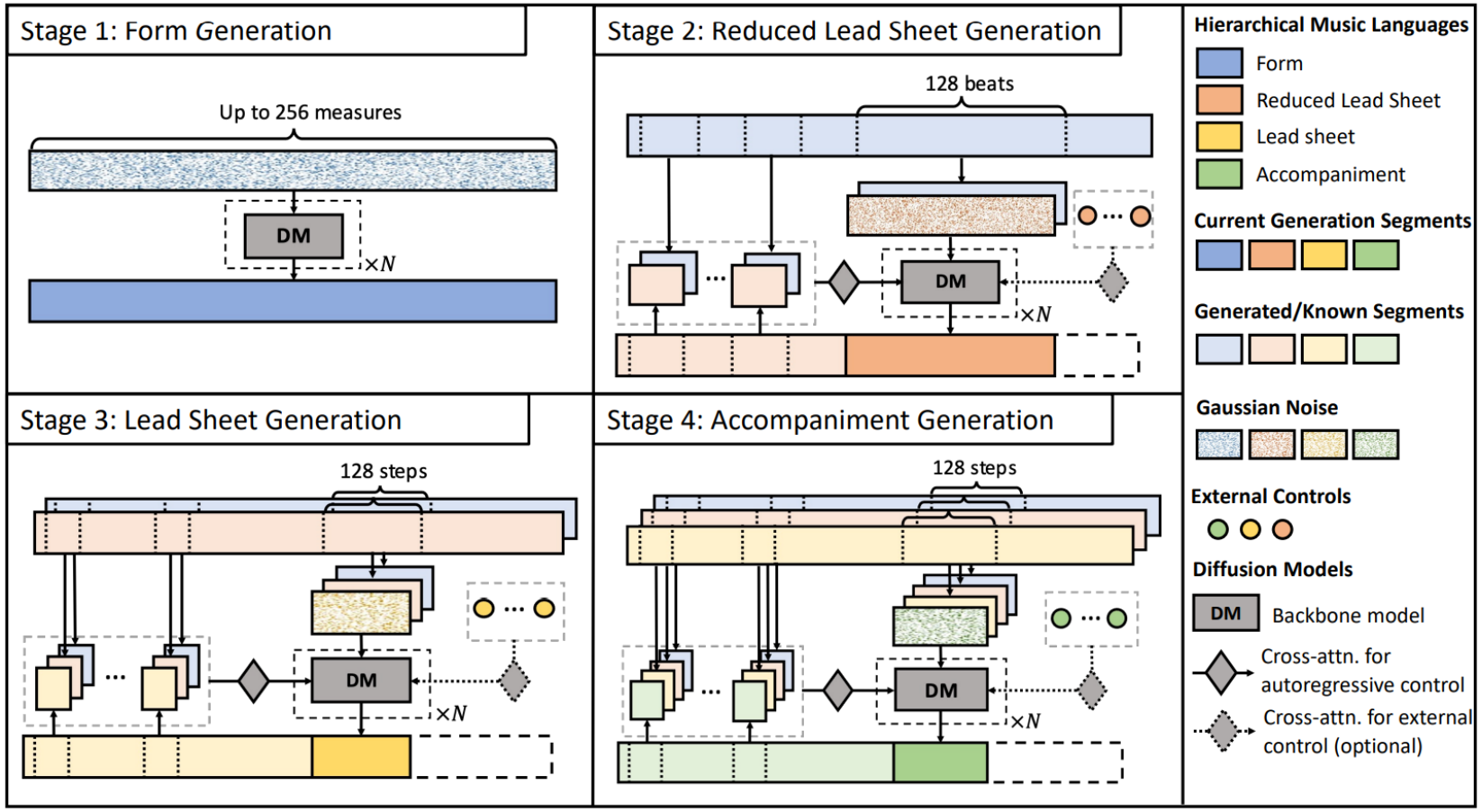


(Source: Min et al., 2023)

polyffusion.github.io

Lejun Min, Junyan Jiang, Gus Xia, and Jingwei Zhao, "Polyffusion: A Diffusion Model for Polyphonic Score Generation with Internal and External Controls," *ISMIR*, 2023.

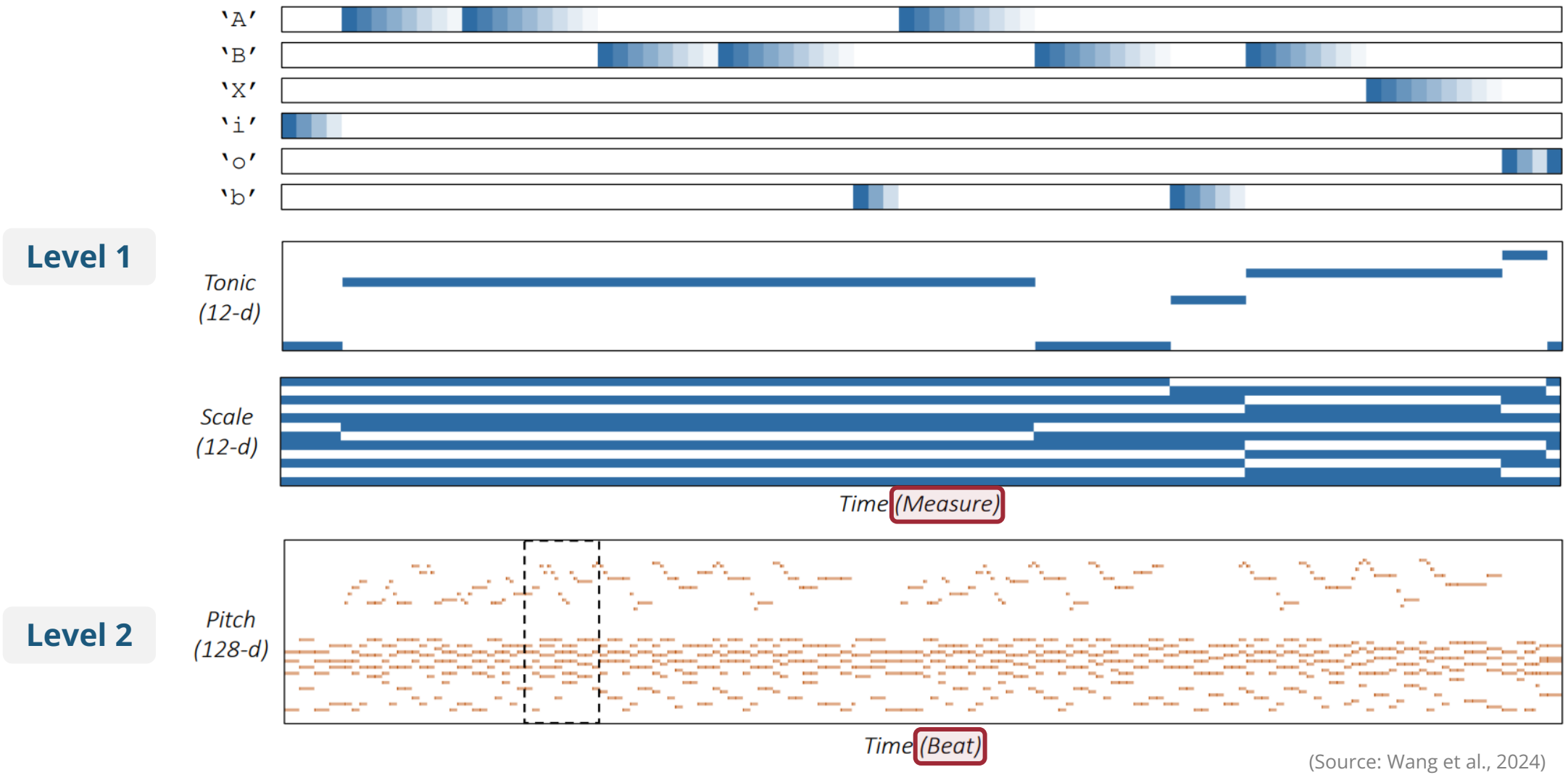
Cascaded Diffusion Models (Wang et al., 2024)



(Source: Wang et al., 2024)

Ziyu Wang, Lejun Min, and Gus Xia, "Whole-Song Hierarchical Generation of Symbolic Music Using Cascaded Diffusion Models," *ICLR*, 2024.

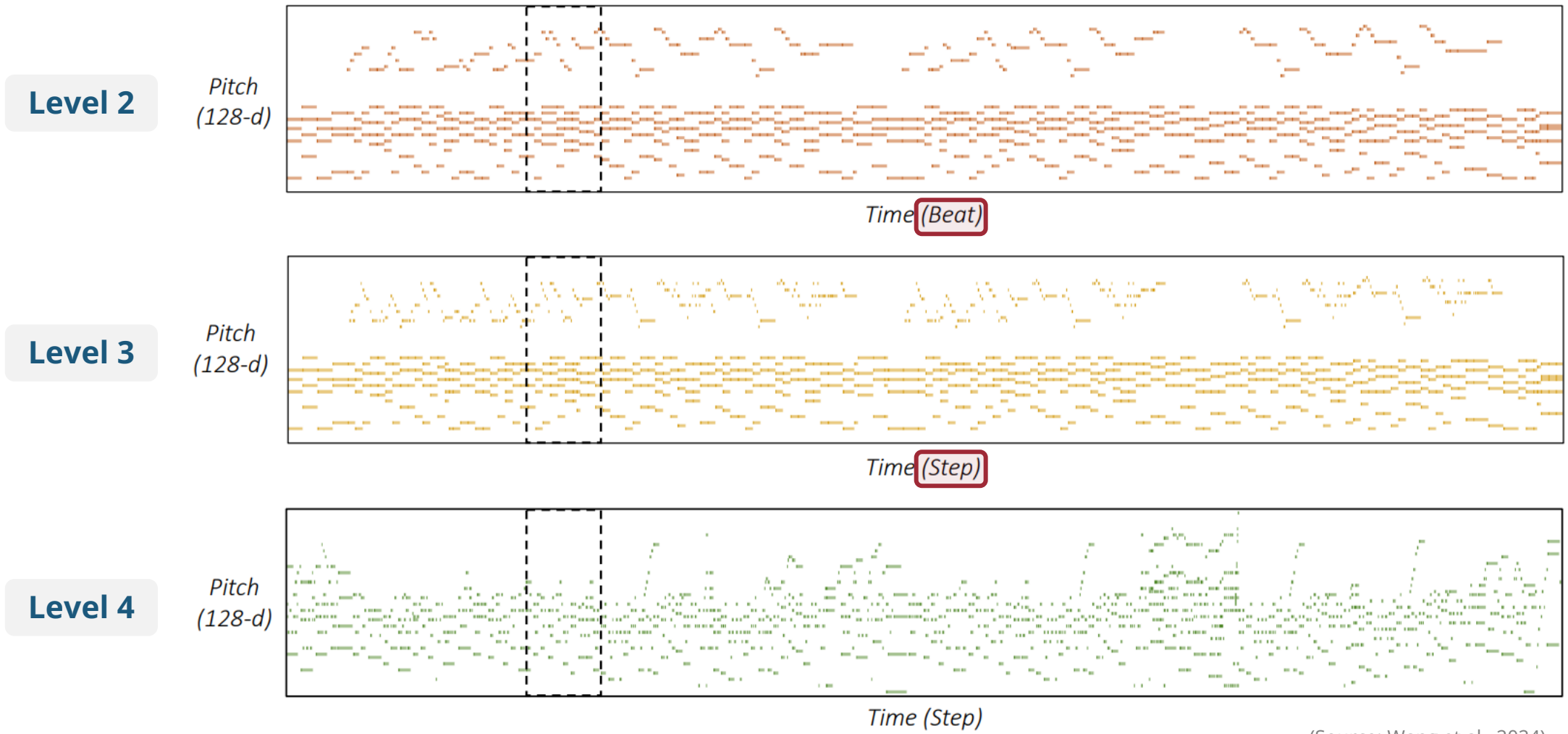
Cascaded Diffusion Models (Wang et al., 2024)



(Source: Wang et al., 2024)

Ziyu Wang, Lejun Min, and Gus Xia, "Whole-Song Hierarchical Generation of Symbolic Music Using Cascaded Diffusion Models," *ICLR*, 2024.

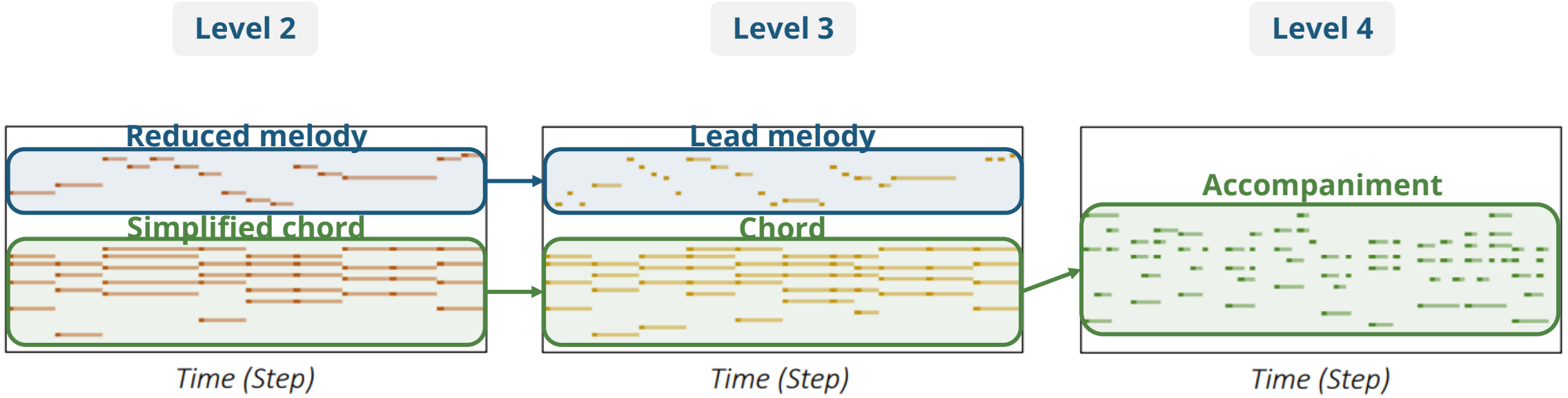
Cascaded Diffusion Models (Wang et al., 2024)



(Source: Wang et al., 2024)

Ziyu Wang, Lejun Min, and Gus Xia, "Whole-Song Hierarchical Generation of Symbolic Music Using Cascaded Diffusion Models," *ICLR*, 2024.

Cascaded Diffusion Models (Wang et al., 2024)



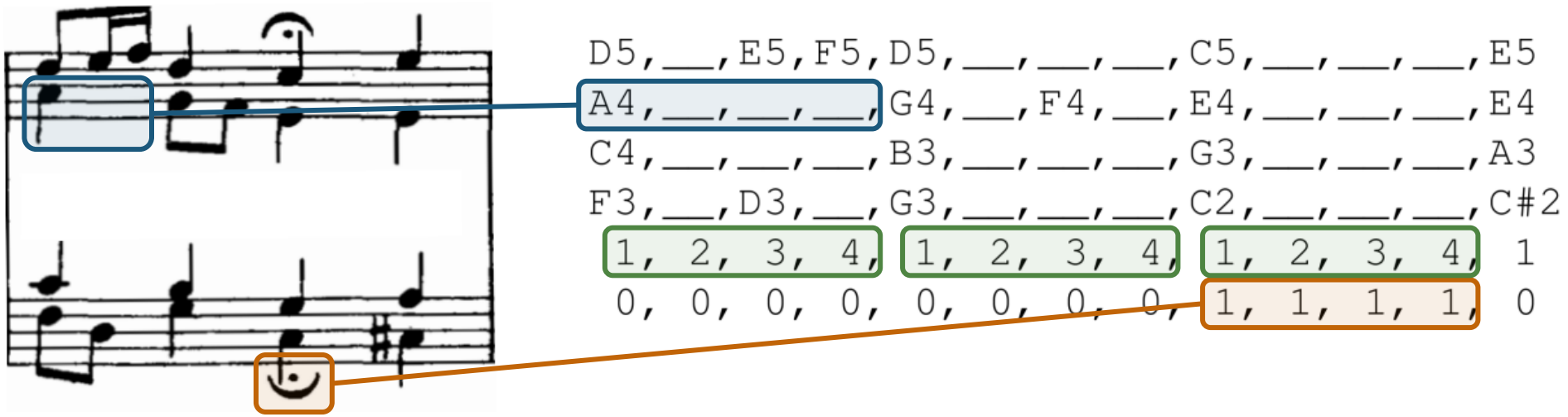
(Source: Wang et al., 2024)

wholesonggen.github.io

Ziyu Wang, Lejun Min, and Gus Xia, "Whole-Song Hierarchical Generation of Symbolic Music Using Cascaded Diffusion Models," *ICLR*, 2024.

Music Infilling Models

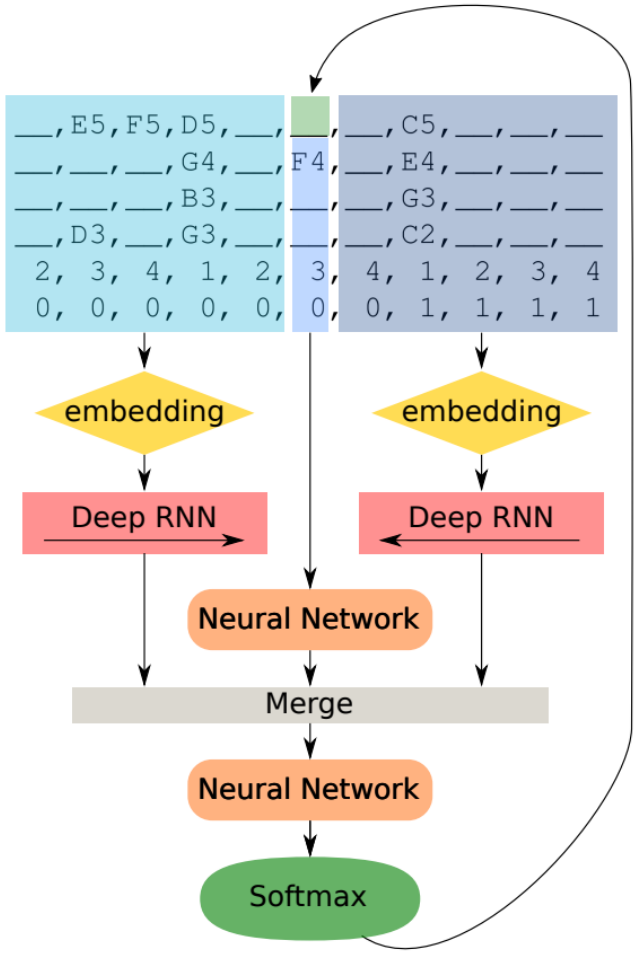
DeepBach (Hadjeres et al., 2017)



(Source: Hadjeres et al., 2017)

Gaëtan Hadjeres, François Pachet, and Frank Nielsen, "DeepBach: a Steerable Model for Bach Chorales Generation," *ICML*, 2017.

DeepBach (Hadjeres et al., 2017)
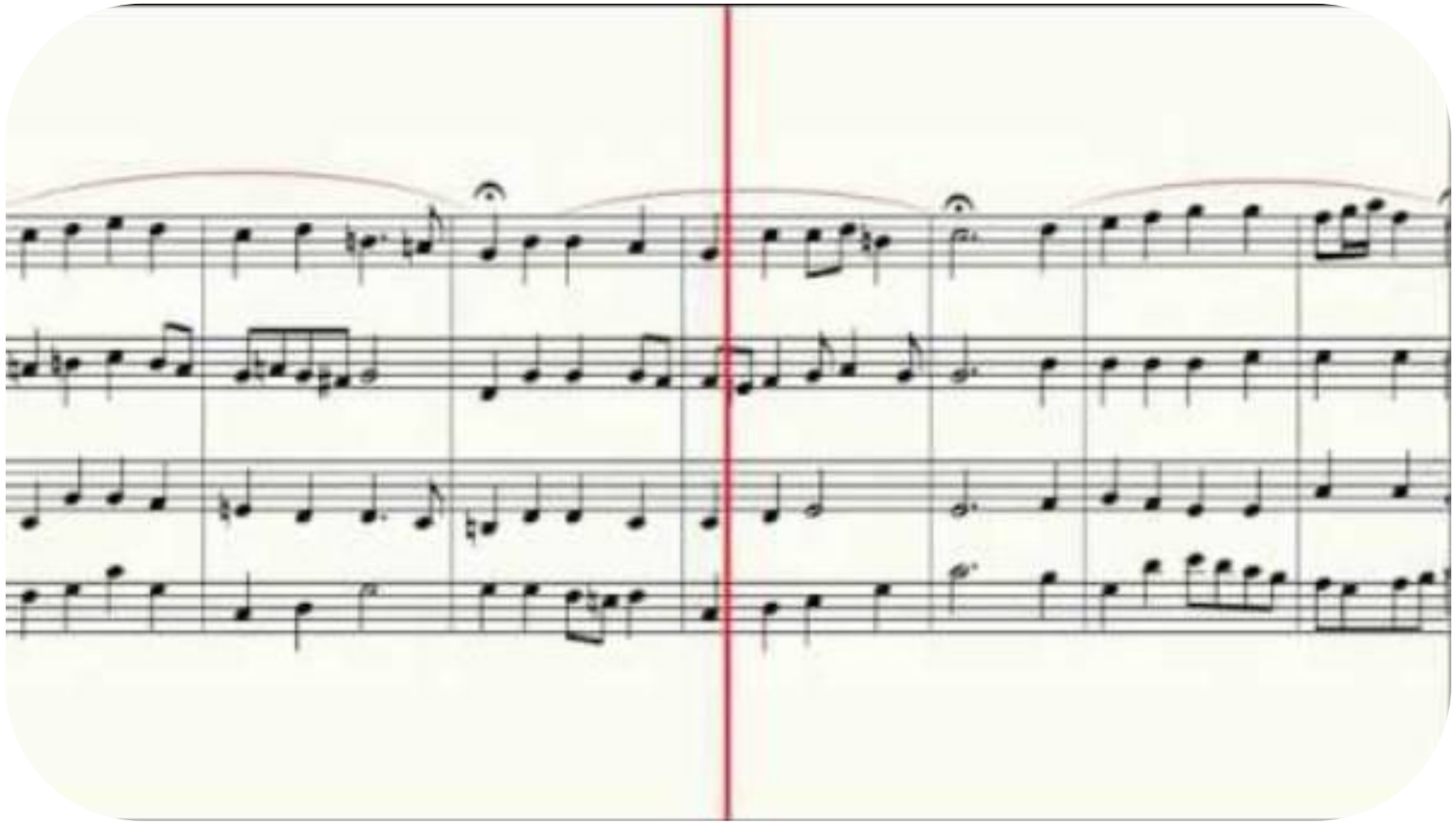


(Source: Hadjeres et al., 2017)

Algorithm 1 Pseudo-Gibbs sampling

Input: Chorale length L, metadata $\mathcal{M}$ containing lists of length L, probability distributions (p_1, p_2, p_3, p_4), maximum number of iterations M
Create four lists $\mathcal{V} = (\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3, \mathcal{V}_4)$ of length L
{The lists are initialized with random notes drawn from the ranges of the corresponding voices (sampled uniformly or from the marginal distributions of the notes)}
for m from 1 to M **do**
    Choose voice i uniformly between 1 and 4
    Choose time t uniformly between 1 and L
    Re-sample $\mathcal{V}_i^t$ from $p_i(\mathcal{V}_i^t | \mathcal{V}_{\setminus i,t}, \mathcal{M}, \theta_i)$
end for
Output: $\mathcal{V} = (\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3, \mathcal{V}_4)$

Gaëtan Hadjeres, François Pachet, and Frank Nielsen, "DeepBach: a Steerable Model for Bach Chorales Generation," *ICML*, 2017.

DeepBach: Reharmonization Examples (Hadjeres et al., 2017)

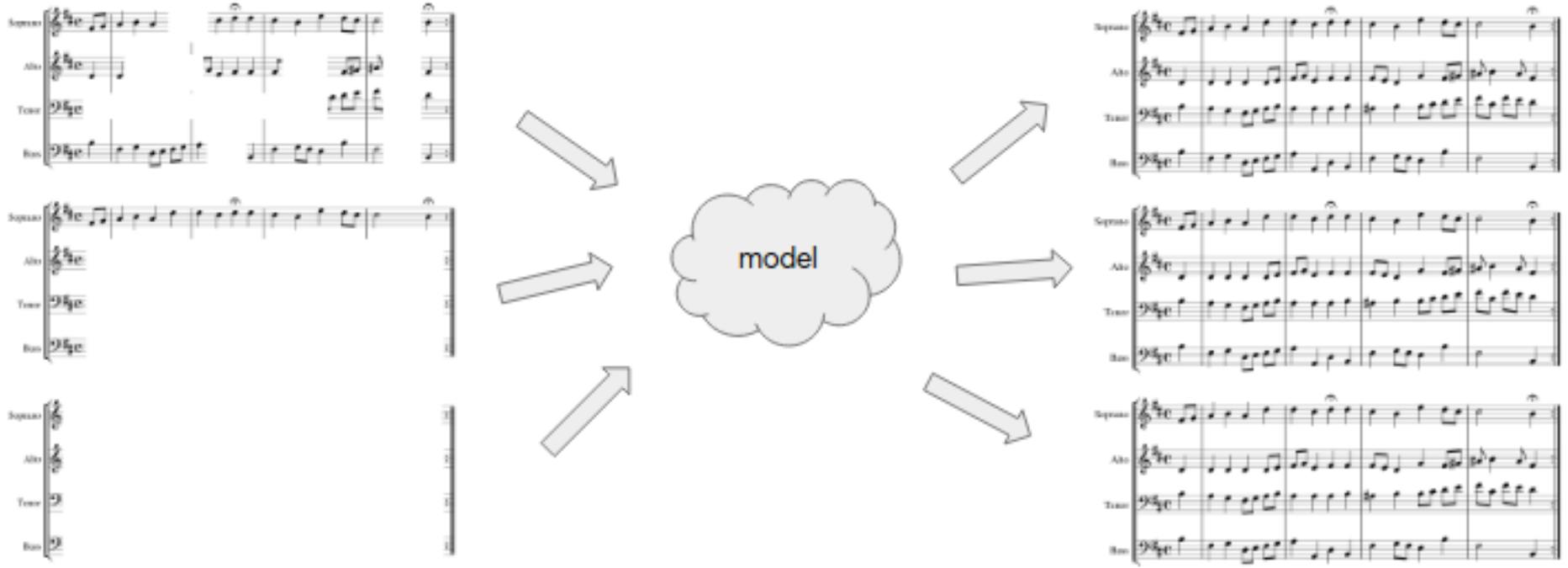


youtu.be/QiBM7-5hA6o

Gaëtan Hadjeres, François Pachet, and Frank Nielsen, "DeepBach: a Steerable Model for Bach Chorales Generation," *ICML*, 2017.

Coconet (Huang et al., 2017)

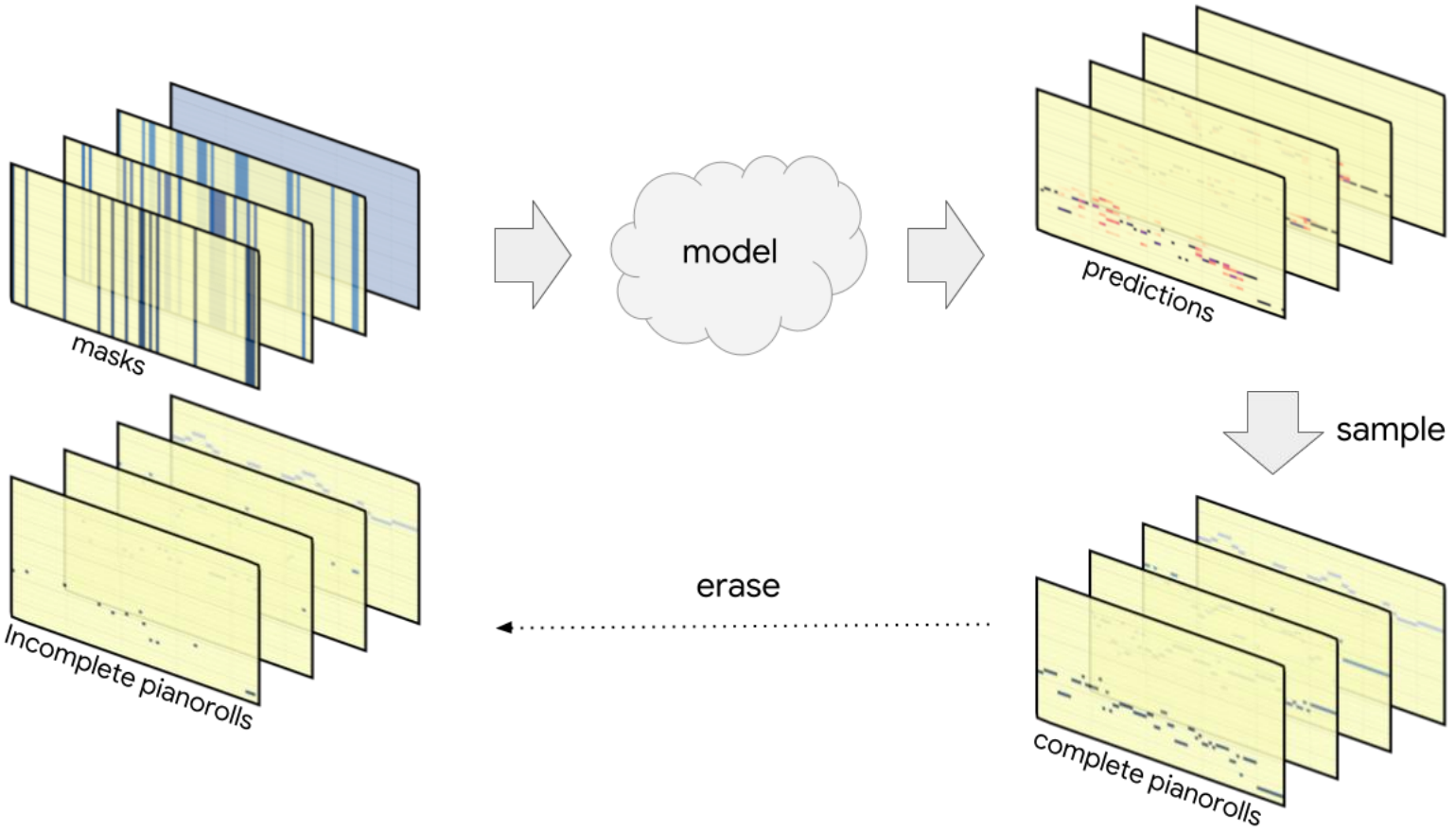
- Based on Orderless NADE (Uria et al, 2014)



(Source: Huang et al., 2019)

Benigno Uria, Iain Murray, and Hugo Larochelle, "A Deep and Tractable Density Estimator," *ICML*, 2014.
Cheng-Zhi Anna Huang, Tim Cooijmans, Adam Roberts, Aaron Courville, and Douglas Eck, "Counterpoint by Convolution," *ISMIR*, 2017.
Cheng-Zhi Anna Huang, Tim Cooijmans, Monica Dinculescu, Adam Roberts, and Curtis Hawthorne, "Coconet: the ML model behind today's Bach Doodle," *Magenta Blog*, 2019.

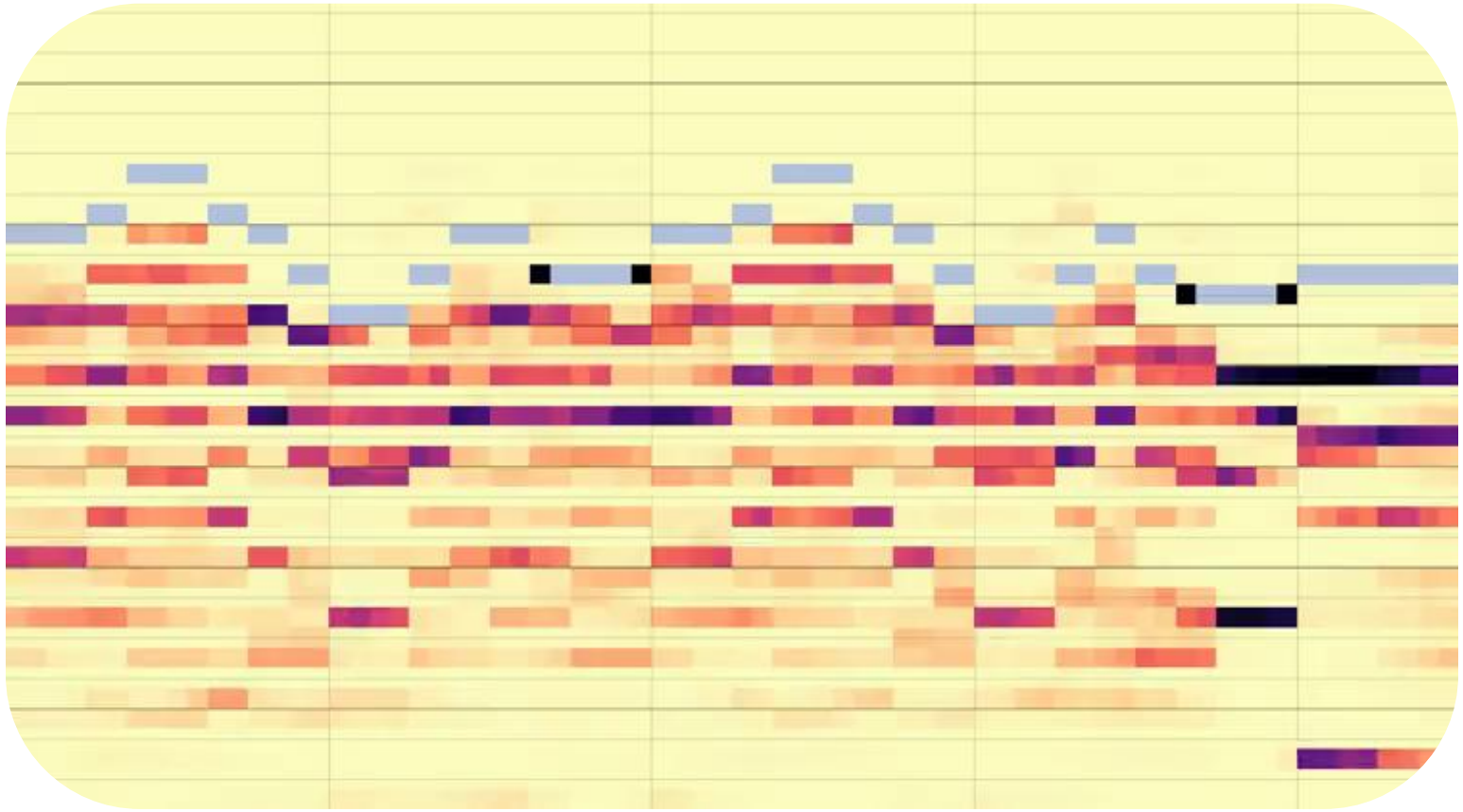
Coconet (Huang et al., 2017)



(Source: Huang et al., 2019)

Cheng-Zhi Anna Huang, Tim Cooijmans, Adam Roberts, Aaron Courville, and Douglas Eck, "Counterpoint by Convolution," *ISMIR*, 2017.
Cheng-Zhi Anna Huang, Tim Cooijmans, Monica Dinculescu, Adam Roberts, and Curtis Hawthorne, "Coconet: the ML model behind today's Bach Doodle," *Magenta Blog*, 2019.

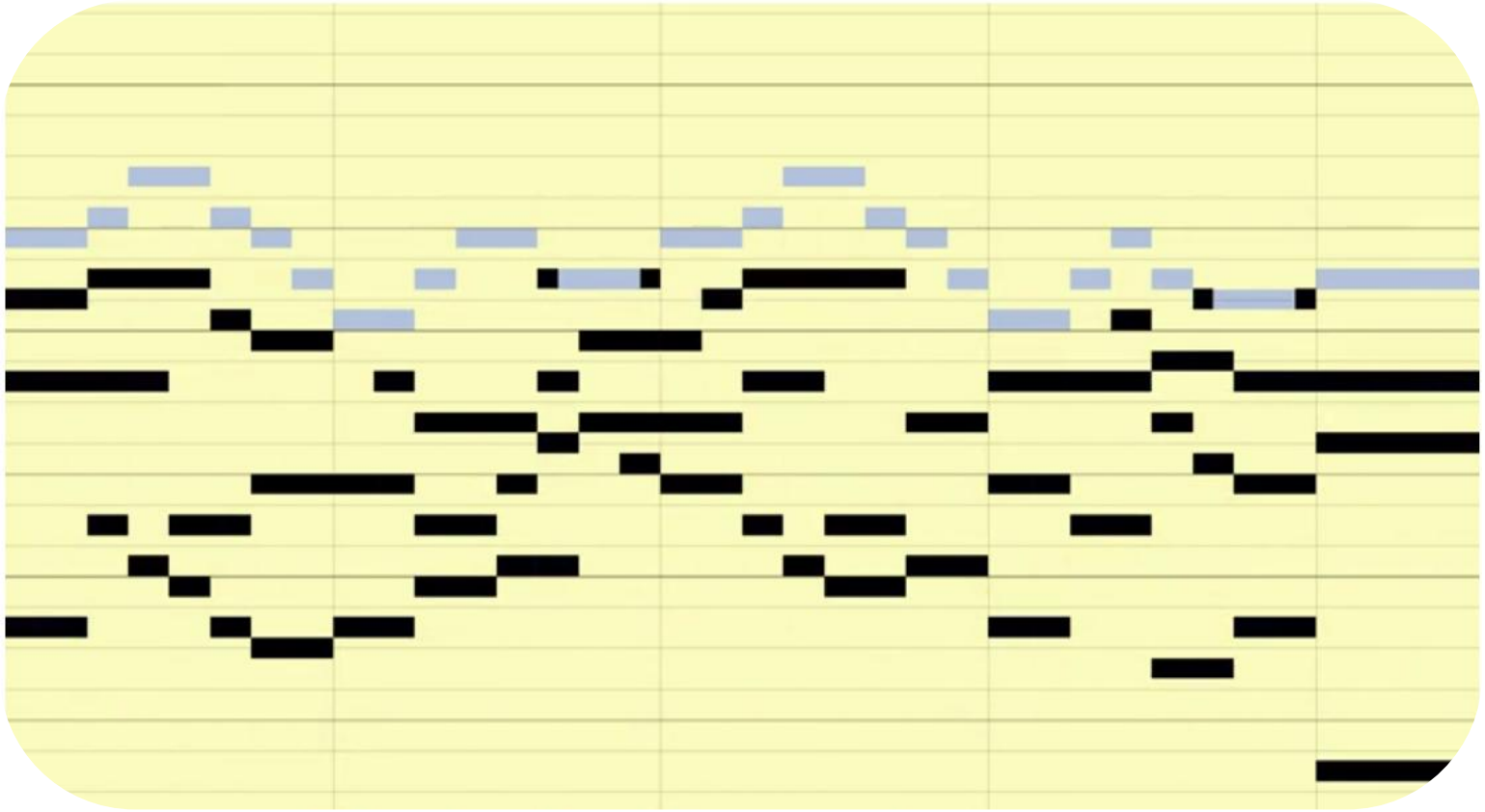
Coconet (Huang et al., 2017)



(Source: Huang et al., 2017)

Cheng-Zhi Anna Huang, Tim Cooijmans, Adam Roberts, Aaron Courville, and Douglas Eck, "Counterpoint by Convolution," *ISMIR*, 2017.

Coconet (Huang et al., 2017)



(Source: Huang et al., 2017)

Cheng-Zhi Anna Huang, Tim Cooijmans, Adam Roberts, Aaron Courville, and Douglas Eck, "Counterpoint by Convolution," *ISMIR*, 2017.

JS Bach Doodle with Coconet (2019)

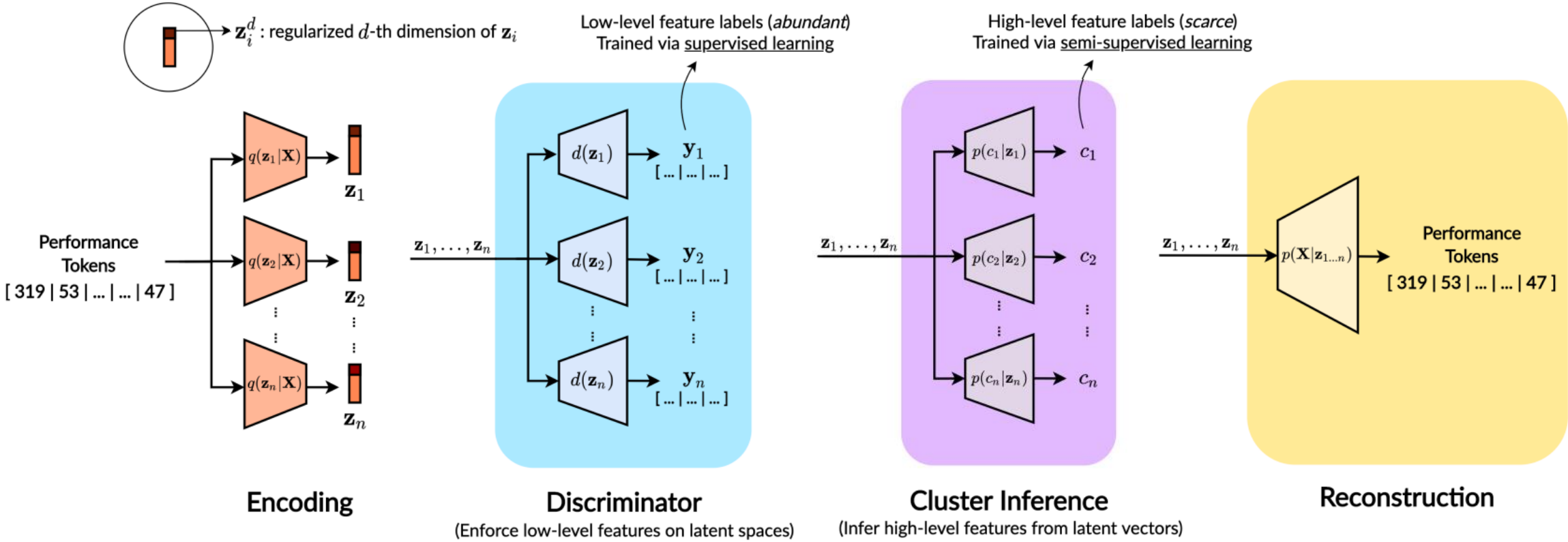


youtu.be/XBfYPp6KF2g & magenta.tensorflow.org/coconet

doodles.google/doodle/celebrating-johann-sebastian-bach/

Controllable Music Generation

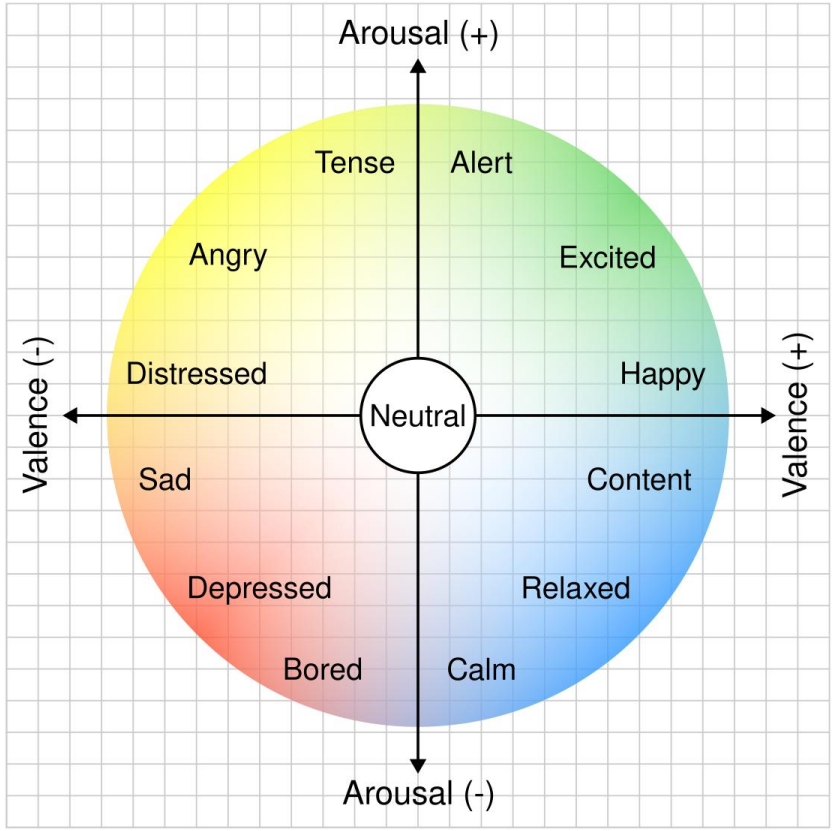
Music FaderNet (Tan & Herremans, 2020)



(Source: Tan & Herremeans, 2020)

Hao Hao Tan and Dorien Herremans, "Music FaderNets: Controllable Music Generation Based On High-Level Features via Low-Level Feature Modelling," ISMIR, 2020.

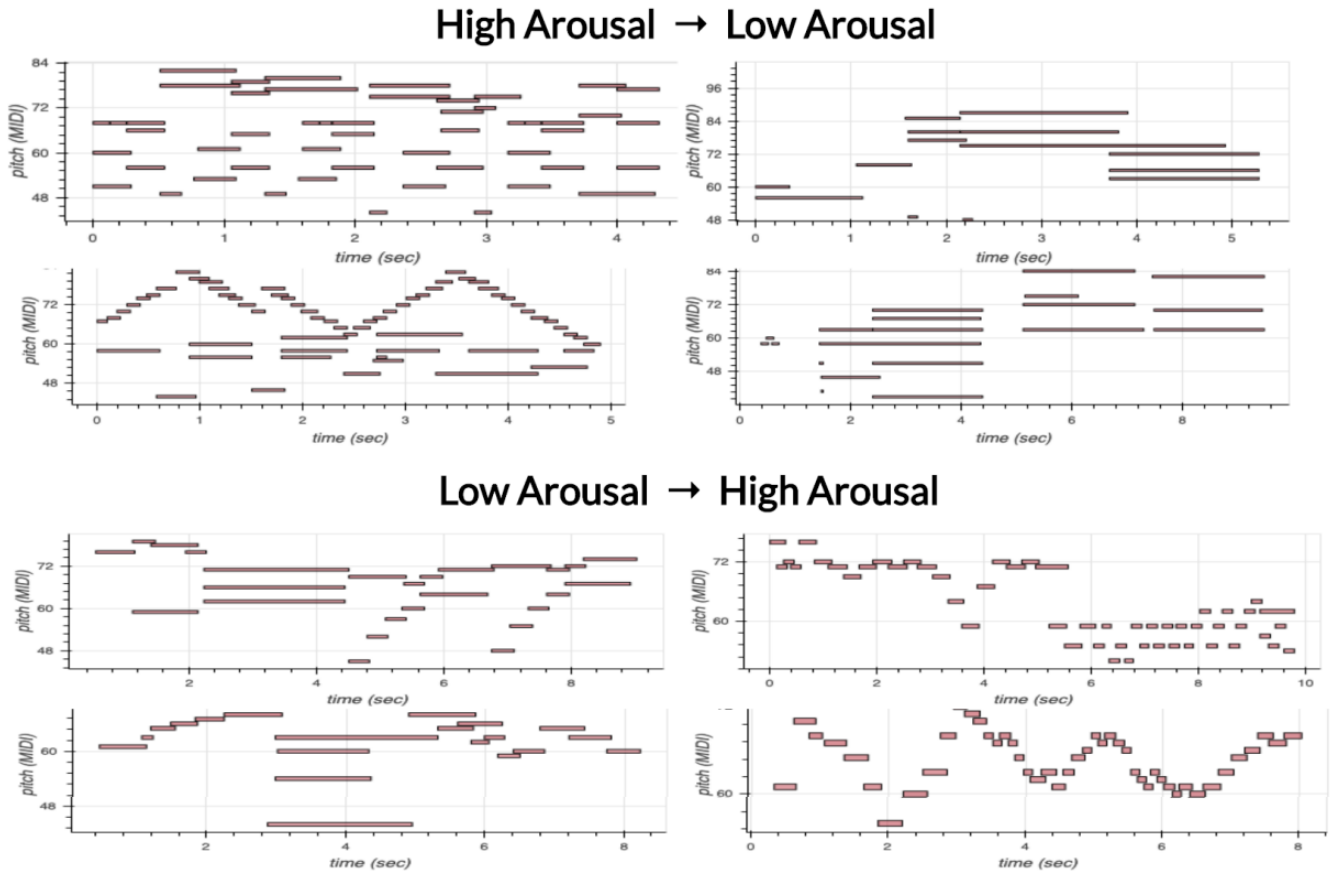
Valence-Arousal Model for Emotion



(Source: mrAnmol)

Draconichiaro, CC By-SA 4.0, via Wikimedia Commons

Music FaderNet (Tan & Herremans, 2020)

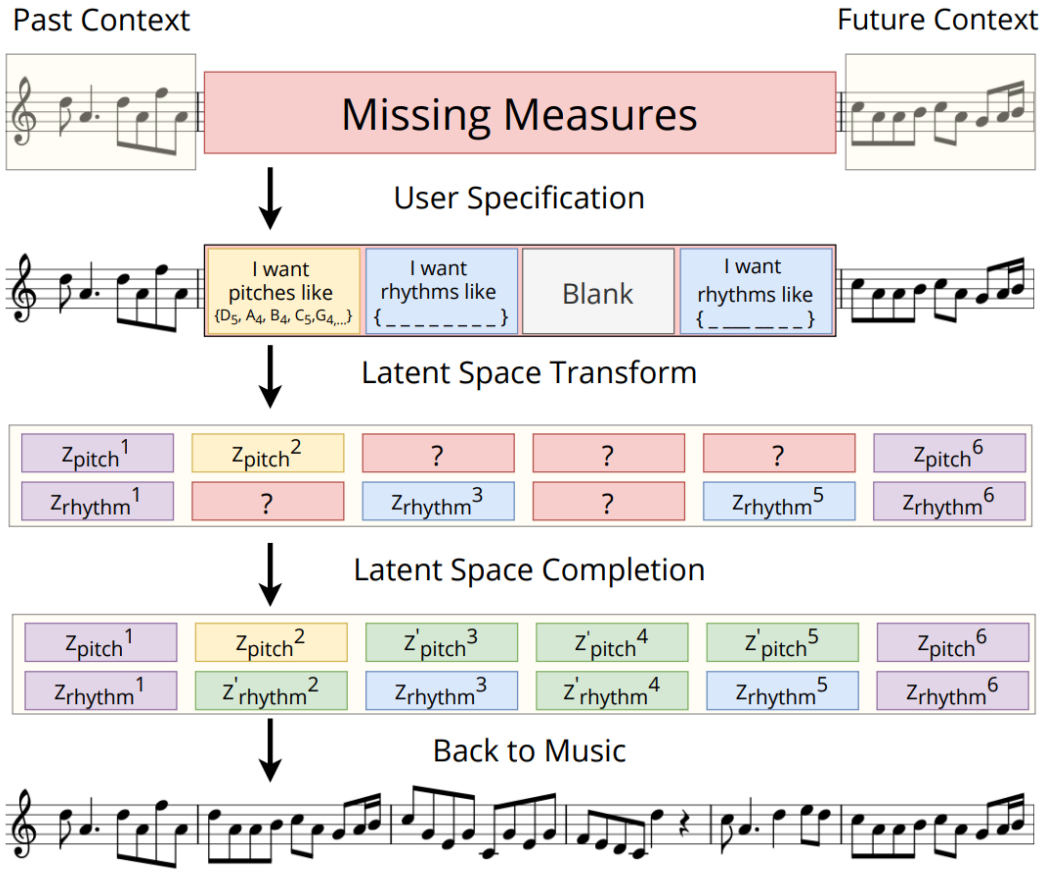


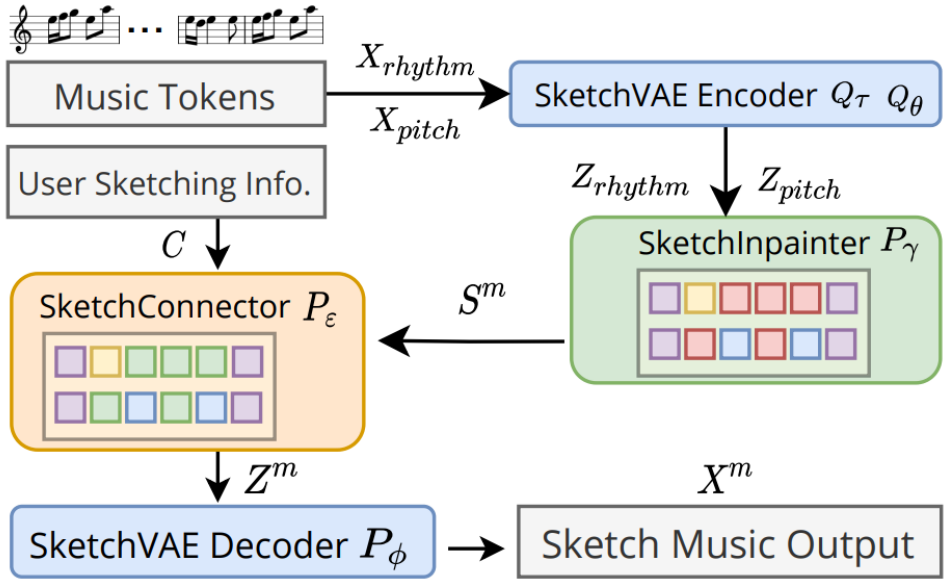
(Source: Tan & Herremeans, 2020)

music-fadernets.github.io

Hao Hao Tan and Dorien Herremans, "Music FaderNets: Controllable Music Generation Based On High-Level Features via Low-Level Feature Modelling," ISMIR, 2020.

Music SketchNet (Chen et al., 2020)

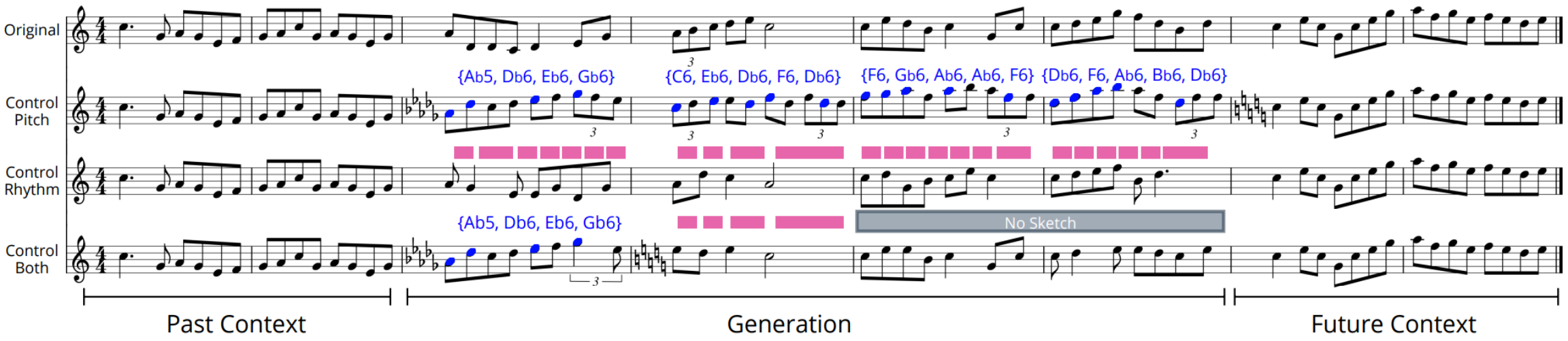




(Source: Chen et al., 2020)

Ke Chen, Cheng-i Wang, Taylor Berg-Kirkpatrick, and Shlomo Dubnov, "Music FaderNets: Controllable Music Generation Based On High-Level Features via Low-Level Feature Modelling," *ISMIR*, 2020.

Music SketchNet (Chen et al., 2020)



(Source: Chen et al., 2020)

Ke Chen, Cheng-i Wang, Taylor Berg-Kirkpatrick, and Shlomo Dubnov, "Music FaderNets: Controllable Music Generation Based On High-Level Features via Low-Level Feature Modelling," *ISMIR*, 2020.

Topics of Symbolic Music Generation

Unconditional

Symbolic music generation

- ∅ → melody
- ∅ → lead sheet (Melody & chords)
- ∅ → sheet music

Conditional

Automatic arrangement

- Melody → lead sheet
- Melody → multitrack
- Lead sheet → multitrack
- Solo → multitrack
- Multitrack → simple version

Performance rendering

- Sheet music → performance

Improvisation systems

- Performance → performance

Multimodal

X-to-music generation

- Text → music
- Video → music
- Gestures → music
- Motions → music
- Brain waves → music
- X → music

Recap

Two Paradigms of Symbolic Music Generation

Text-based

- Treat music like **text**
- Sharing models with **natural language processing (NLP)**
  - RNNs, LSTMs, Transformers, etc.

```
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, ...
```

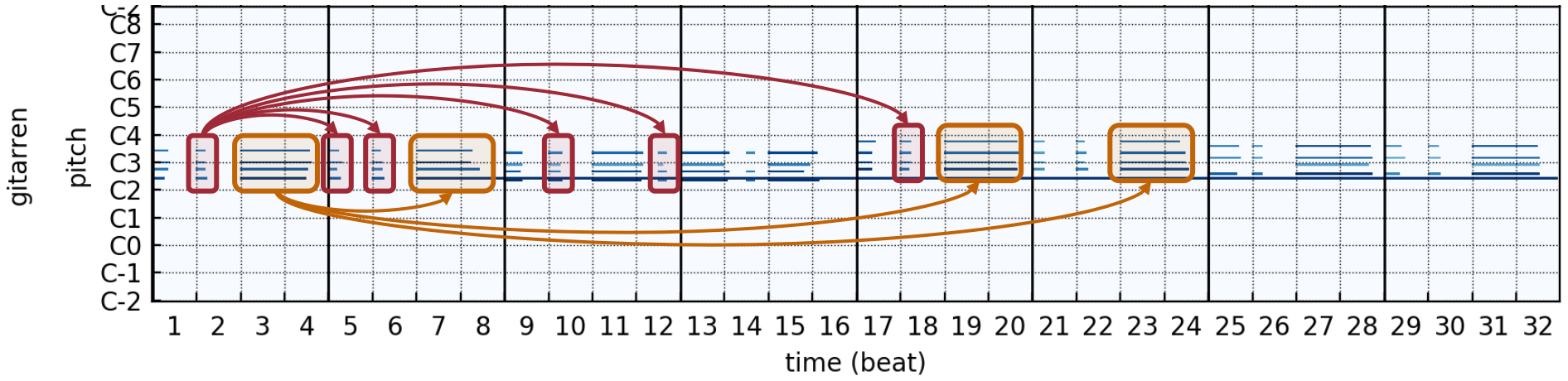
Image-based

- Treat music like **images**
- Sharing models with **computer vision (CV)** & **computer graphics (CG)**
  - GANs, VAEs, diffusion models, etc.

Today's topic!

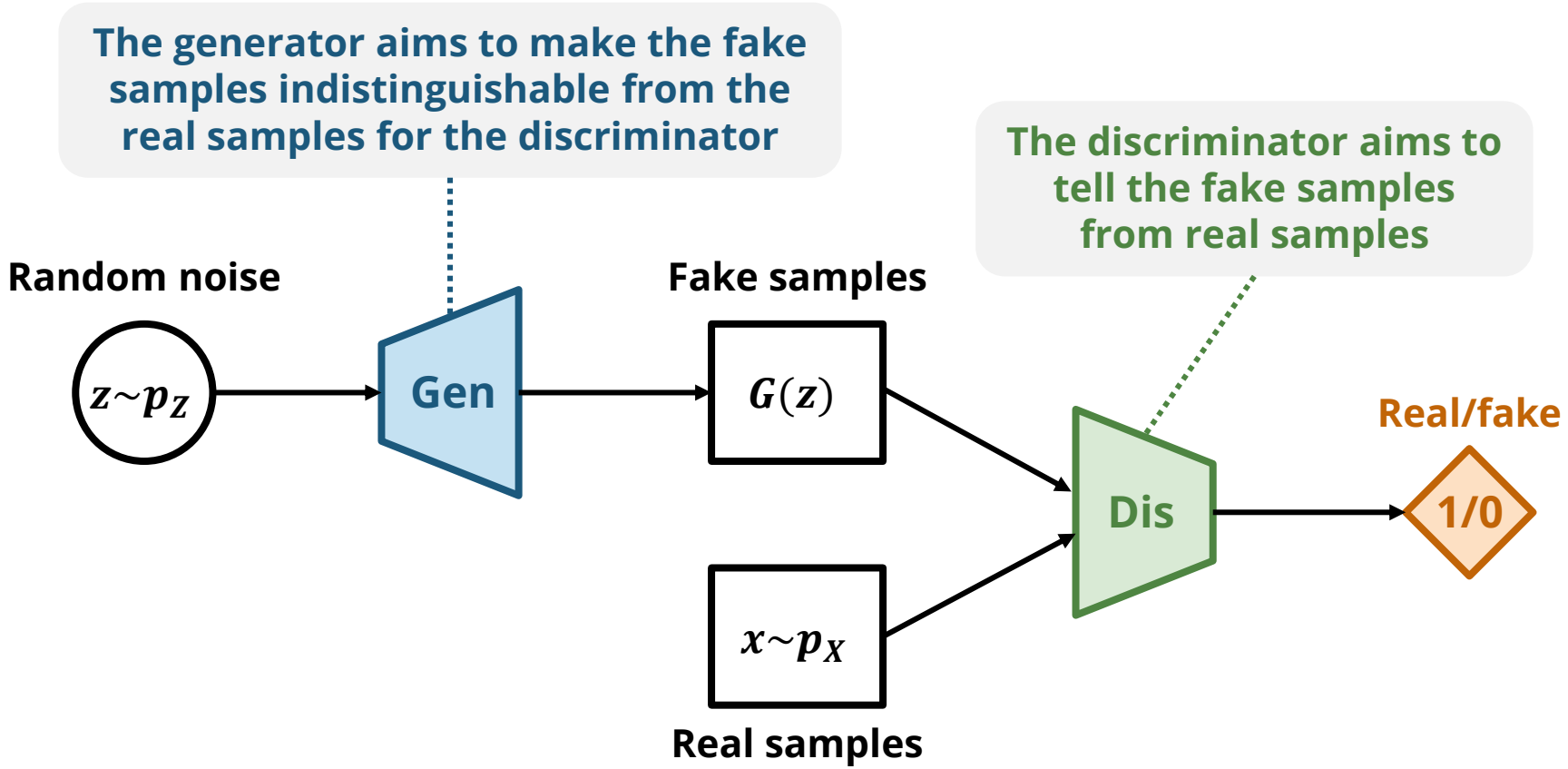


Why Piano Rolls?



Many musical patterns like melodies, chords, scales and arpeggios are translational invariant in the temporal and pitch axes

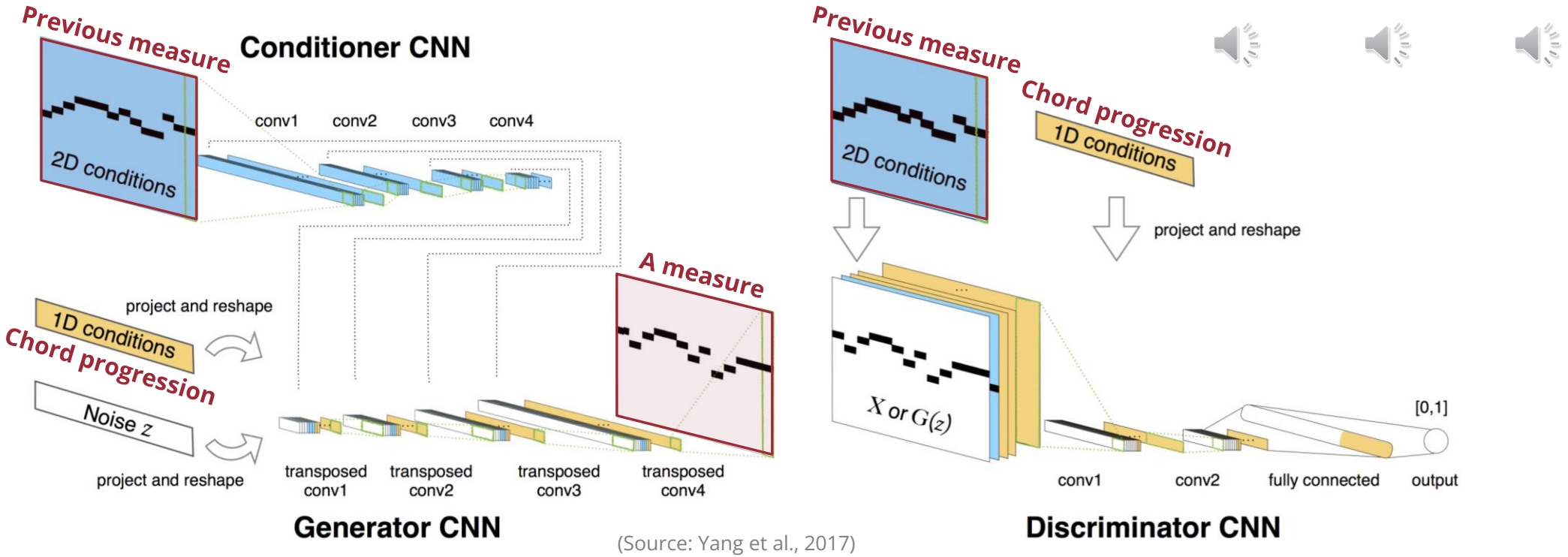
Generative Adversarial Nets (GANs) (Goodfellow et al., 2014)



Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative Adversarial Networks," *NeurIPS*, 2014.

MidiNet (Yang et al., 2017)

Examples of generated music

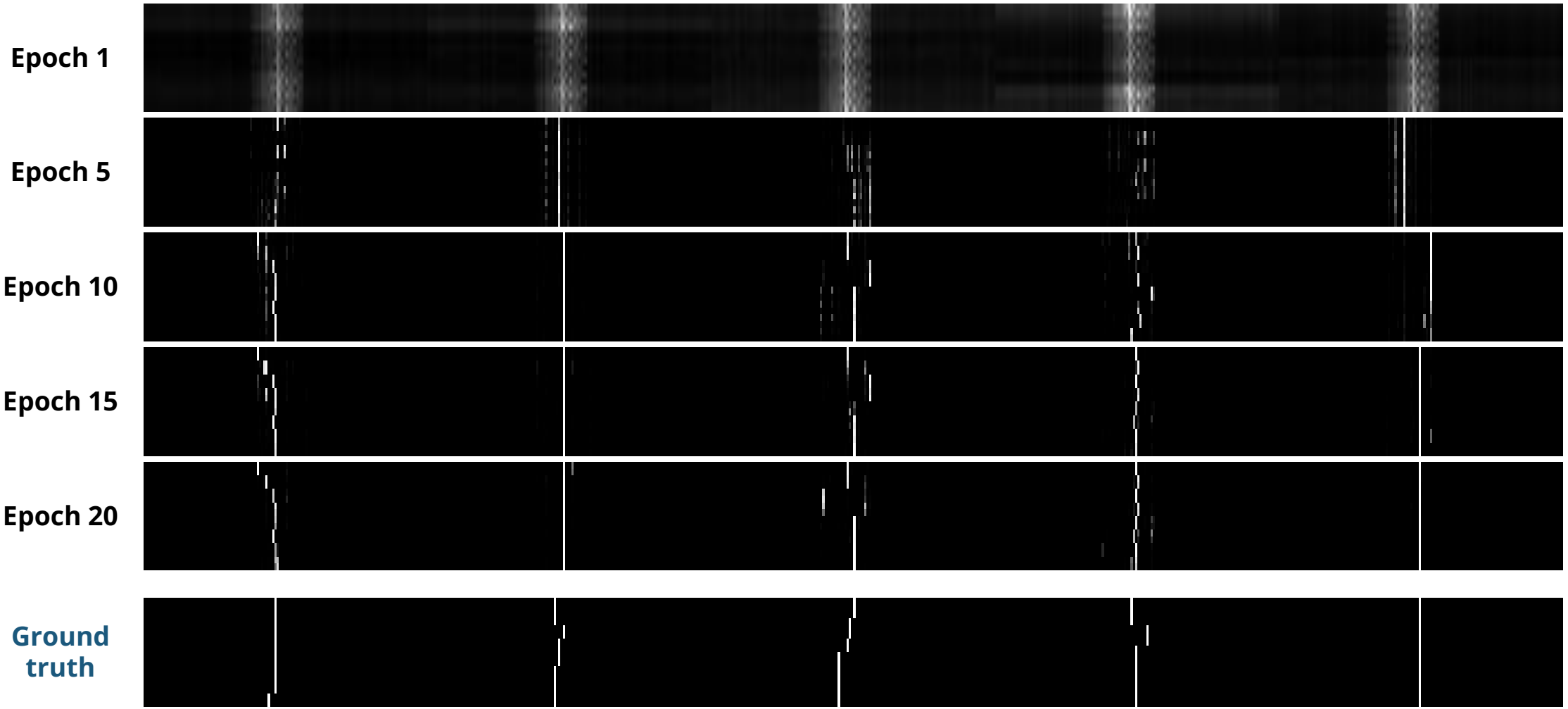


(Source: Yang et al., 2017)

Generate the next measure given the previous one (measure by measure)

Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang, "MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation," *ISMIR*, 2017.

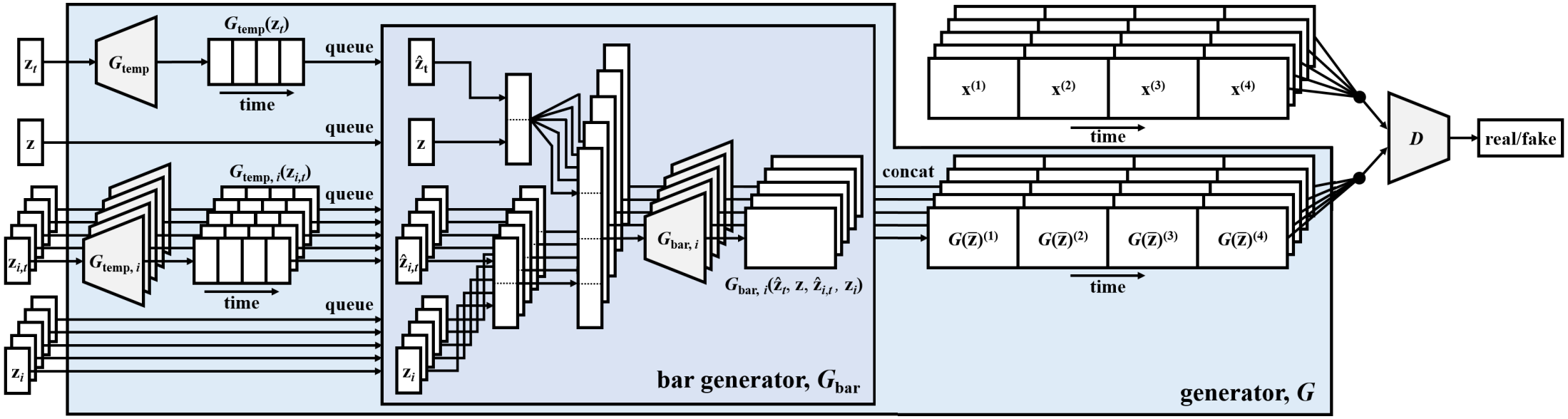
MidiNet (Yang et al., 2017)



(Source: Yang et al., 2017)

Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang, "MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation," *ISMIR*, 2017.

MuseGAN (Dong et al., 2018)



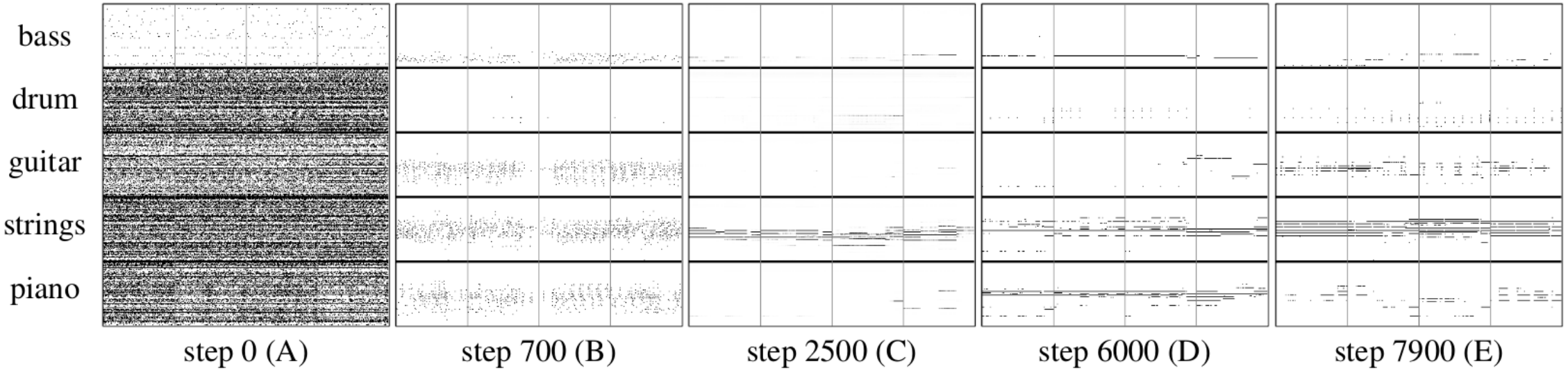
(Source: Dong et al., 2018)

Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang, "MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment," *AAAI*, 2018.

MuseGAN (Dong et al., 2018)

Examples of generated music

The generator improves over time

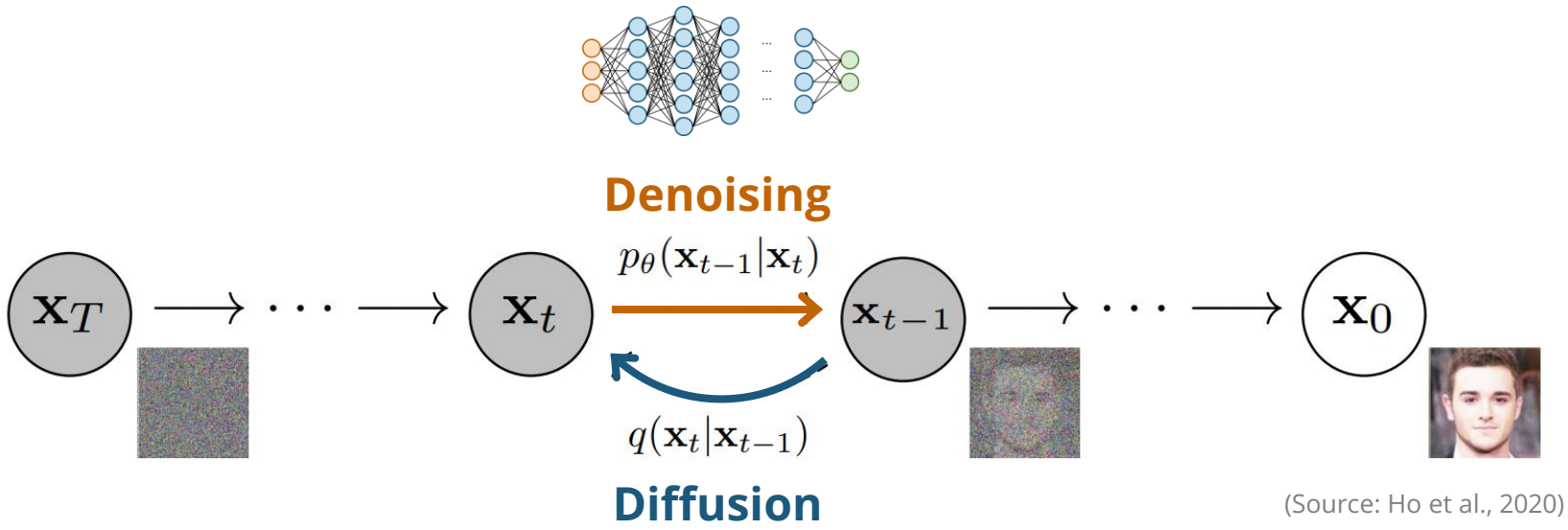


(Source: Dong et al., 2018)

Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang, "MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment," *AAAI*, 2018.

Diffusion Models (Ho et al., 2020)

- **Intuition**: Many denoising autoencoders stacked together

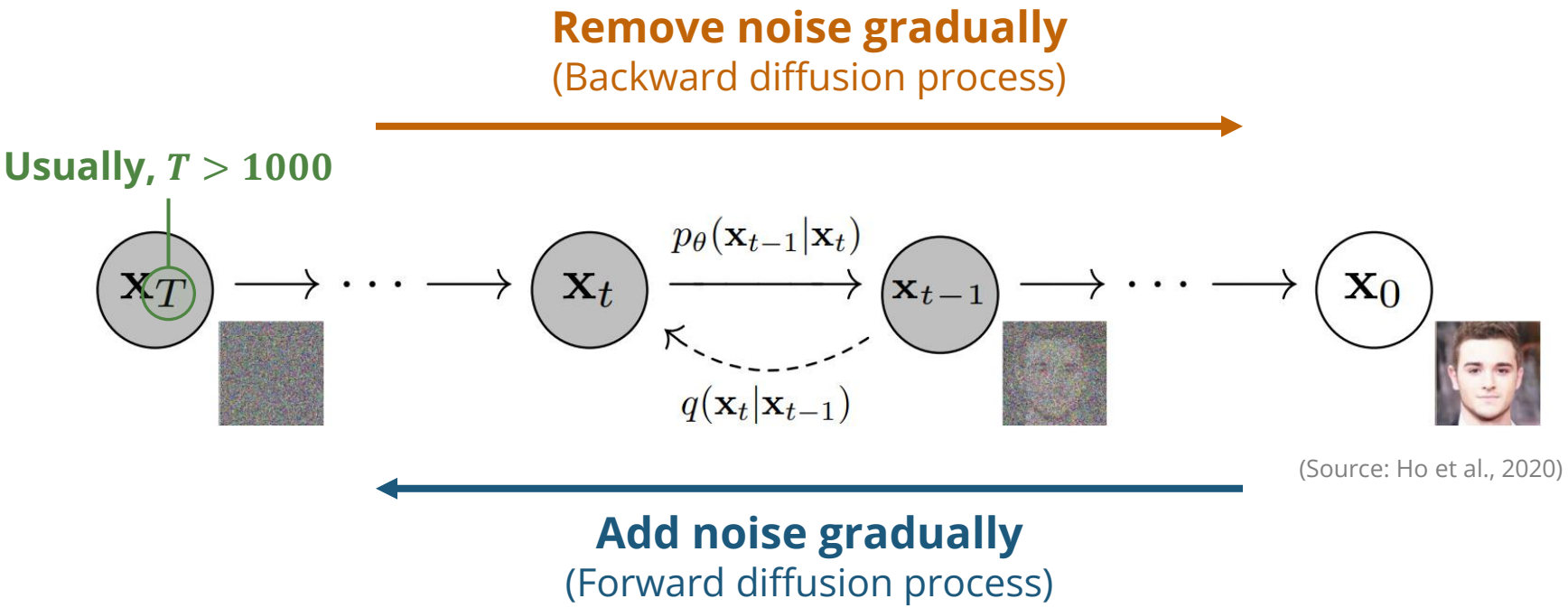


(Source: Ho et al., 2020)

Jonathan Ho, Ajay Jain, and Pieter Abbeel, "Denoising Diffusion Probabilistic Models," *NeurIPS*, 2020.

Diffusion Models (Ho et al., 2020)

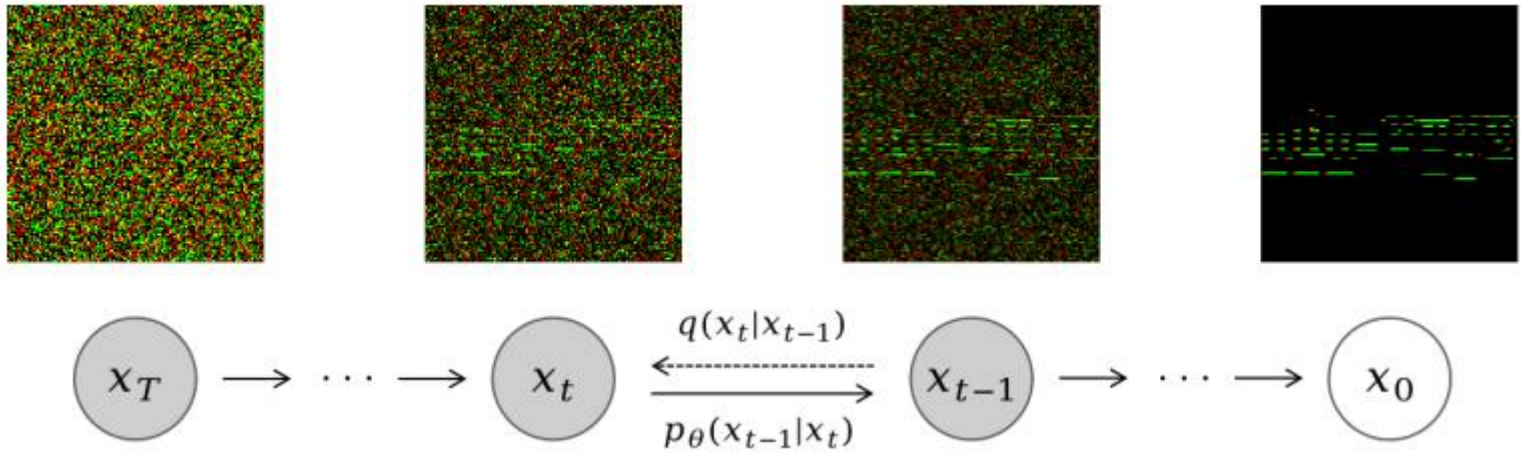
- **Intuition**: Many denoising autoencoders stacked together



(Source: Ho et al., 2020)

Jonathan Ho, Ajay Jain, and Pieter Abbeel, "Denoising Diffusion Probabilistic Models," *NeurIPS*, 2020.

Polyffusion (Min et al., 2023)

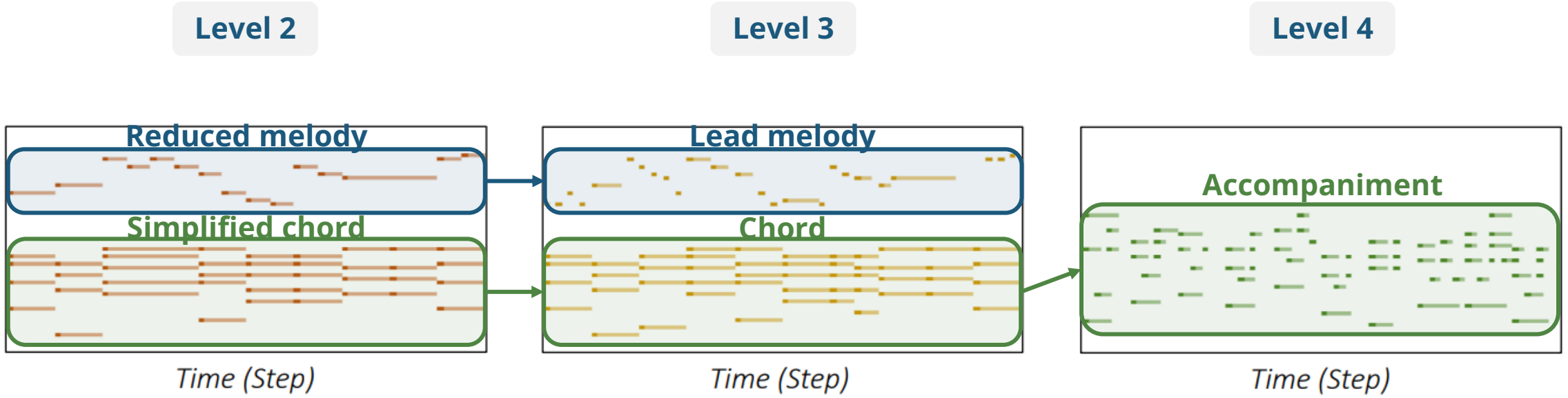


(Source: Min et al., 2023)

polyffusion.github.io

Lejun Min, Junyan Jiang, Gus Xia, and Jingwei Zhao, "Polyffusion: A Diffusion Model for Polyphonic Score Generation with Internal and External Controls," *ISMIR*, 2023.

Cascaded Diffusion Models (Wang et al., 2024)

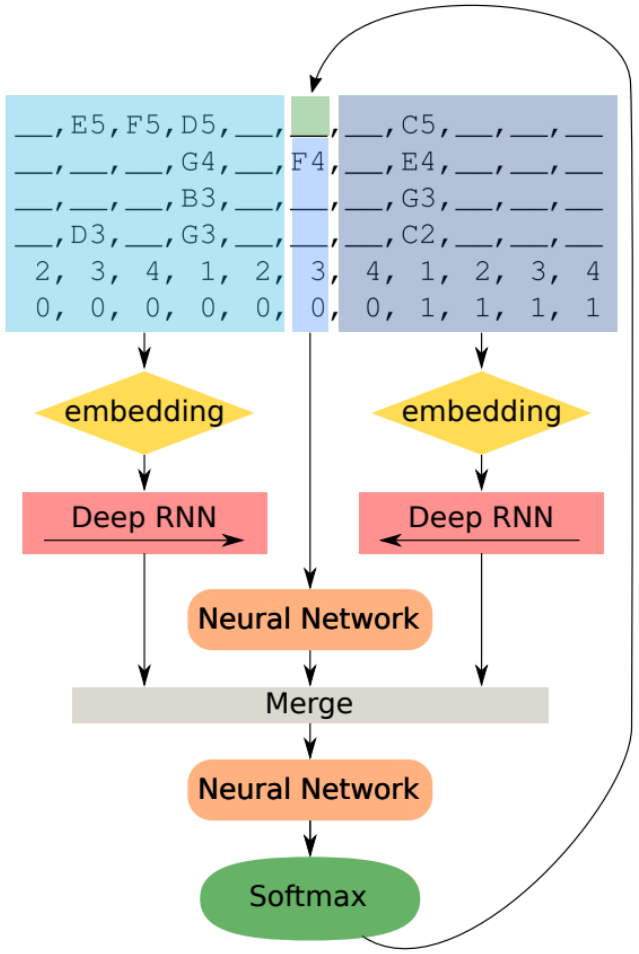


(Source: Wang et al., 2024)

wholesonggen.github.io

Ziyu Wang, Lejun Min, and Gus Xia, "Whole-Song Hierarchical Generation of Symbolic Music Using Cascaded Diffusion Models," *ICLR*, 2024.

DeepBach (Hadjeres et al., 2017)



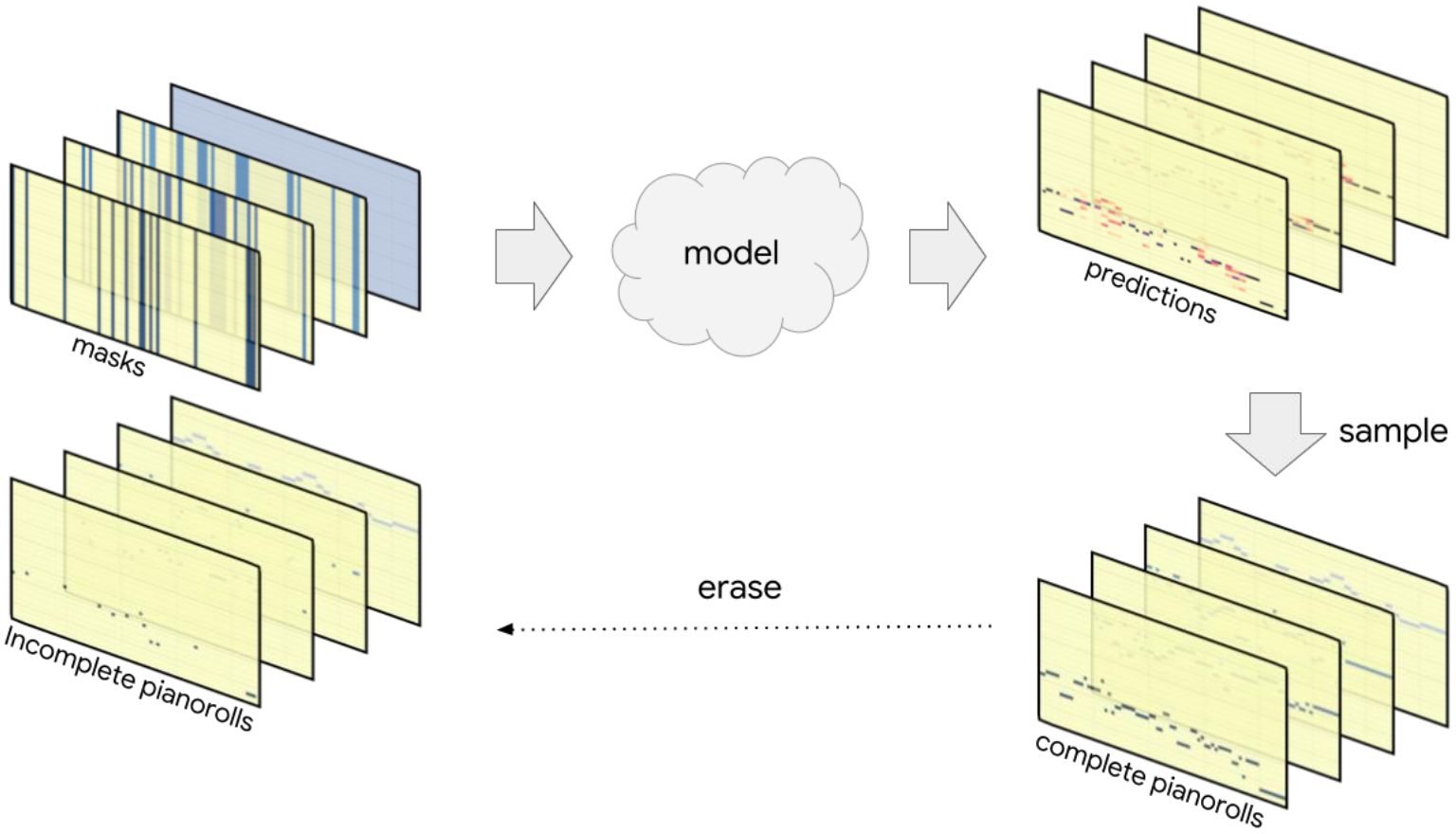
(Source: Hadjeres et al., 2017)

Algorithm 1 Pseudo-Gibbs sampling

1: **Input:** Chorale length L, metadata $\mathcal{M}$ containing lists of length L, probability distributions (p_1, p_2, p_3, p_4), maximum number of iterations M
2: Create four lists $\mathcal{V} = (\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3, \mathcal{V}_4)$ of length L
3: {The lists are initialized with random notes drawn from the ranges of the corresponding voices (sampled uniformly or from the marginal distributions of the notes)}
4: **for** m from 1 to M **do**
5: Choose voice i uniformly between 1 and 4
6: Choose time t uniformly between 1 and L
7: Re-sample $\mathcal{V}_i^t$ from $p_i(\mathcal{V}_i^t | \mathcal{V}_{\setminus i,t}, \mathcal{M}, \theta_i)$
8: **end for**
9: **Output:** $\mathcal{V} = (\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3, \mathcal{V}_4)$

Gaëtan Hadjeres, François Pachet, and Frank Nielsen, "DeepBach: a Steerable Model for Bach Chorales Generation," *ICML*, 2017.

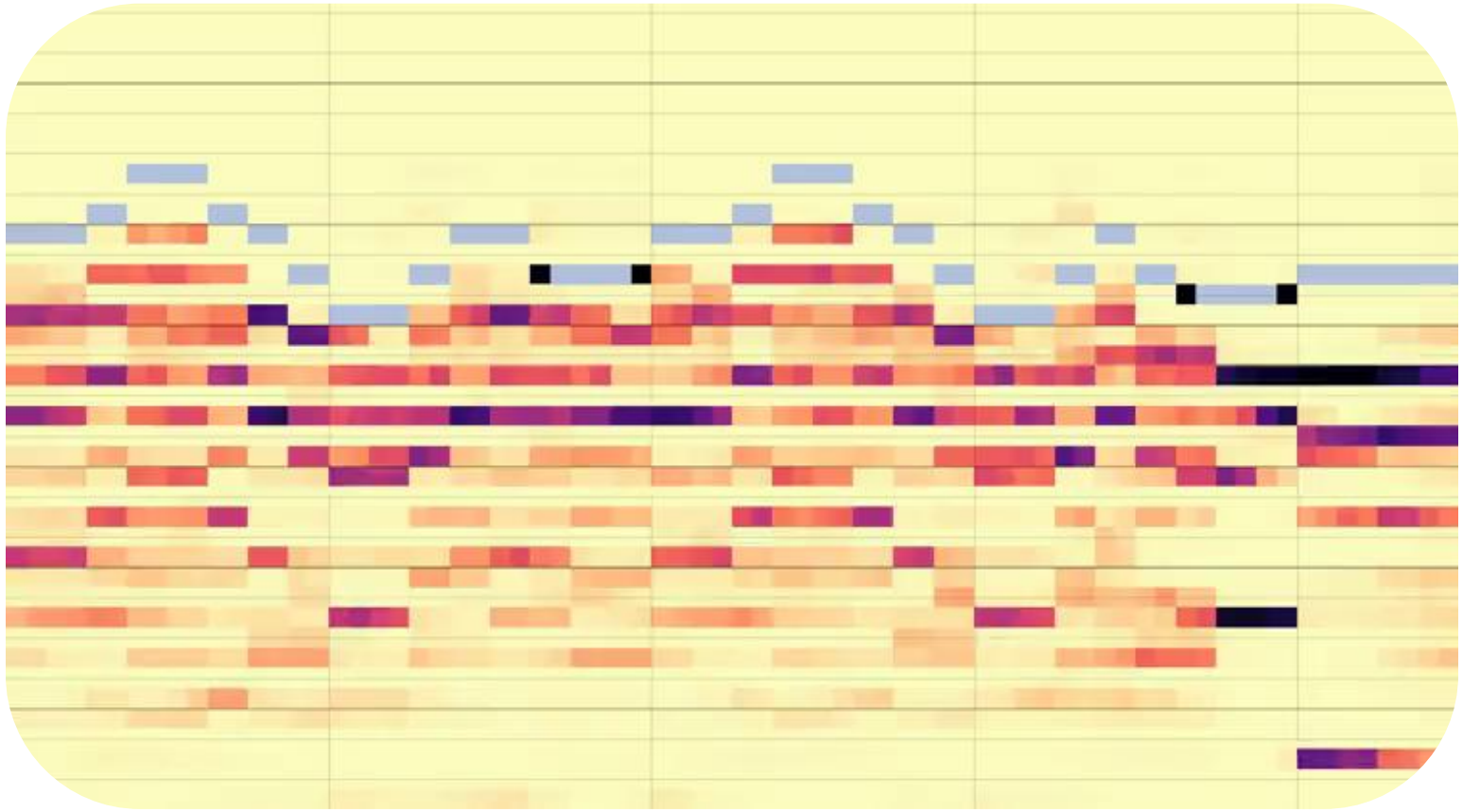
Coconet (Huang et al., 2017)



(Source: Huang et al., 2019)

Cheng-Zhi Anna Huang, Tim Cooijmans, Adam Roberts, Aaron Courville, and Douglas Eck, "Counterpoint by Convolution," *ISMIR*, 2017.
Cheng-Zhi Anna Huang, Tim Cooijmans, Monica Dinculescu, Adam Roberts, and Curtis Hawthorne, "Coconet: the ML model behind today's Bach Doodle," *Magenta Blog*, 2019.

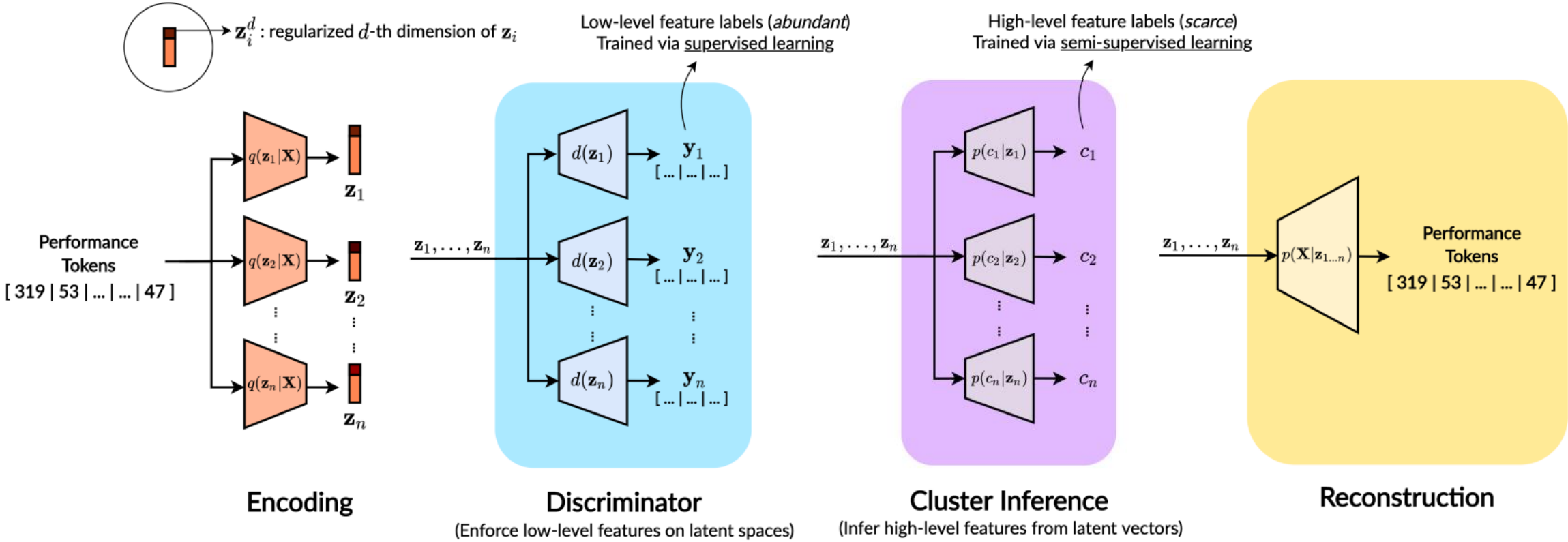
Coconet (Huang et al., 2017)



(Source: Huang et al., 2017)

Cheng-Zhi Anna Huang, Tim Cooijmans, Adam Roberts, Aaron Courville, and Douglas Eck, "Counterpoint by Convolution," *ISMIR*, 2017.

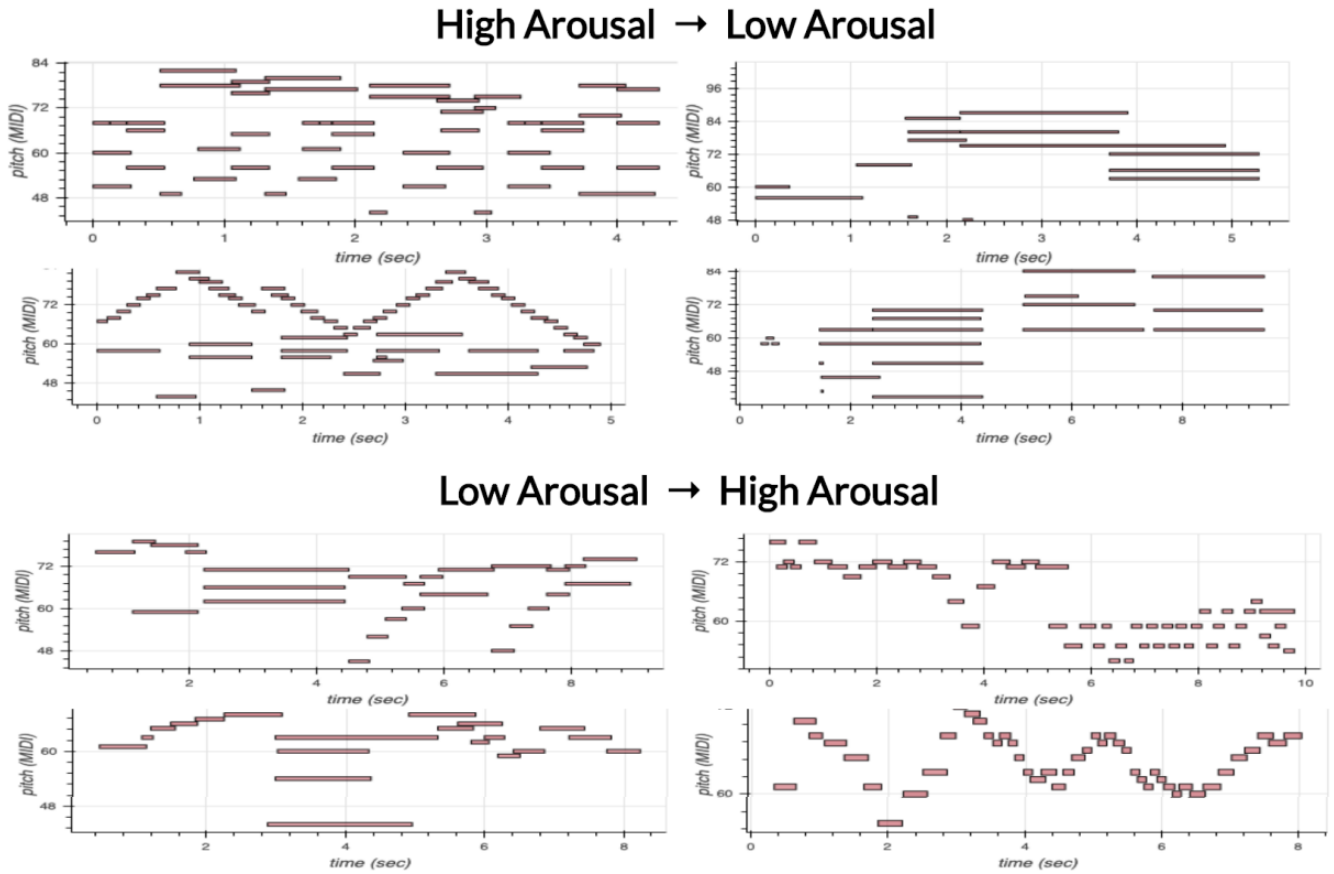
Music FaderNet (Tan & Herremans, 2020)



(Source: Tan & Herremeans, 2020)

Hao Hao Tan and Dorien Herremans, "Music FaderNets: Controllable Music Generation Based On High-Level Features via Low-Level Feature Modelling," ISMIR, 2020.

Music FaderNet (Tan & Herremans, 2020)

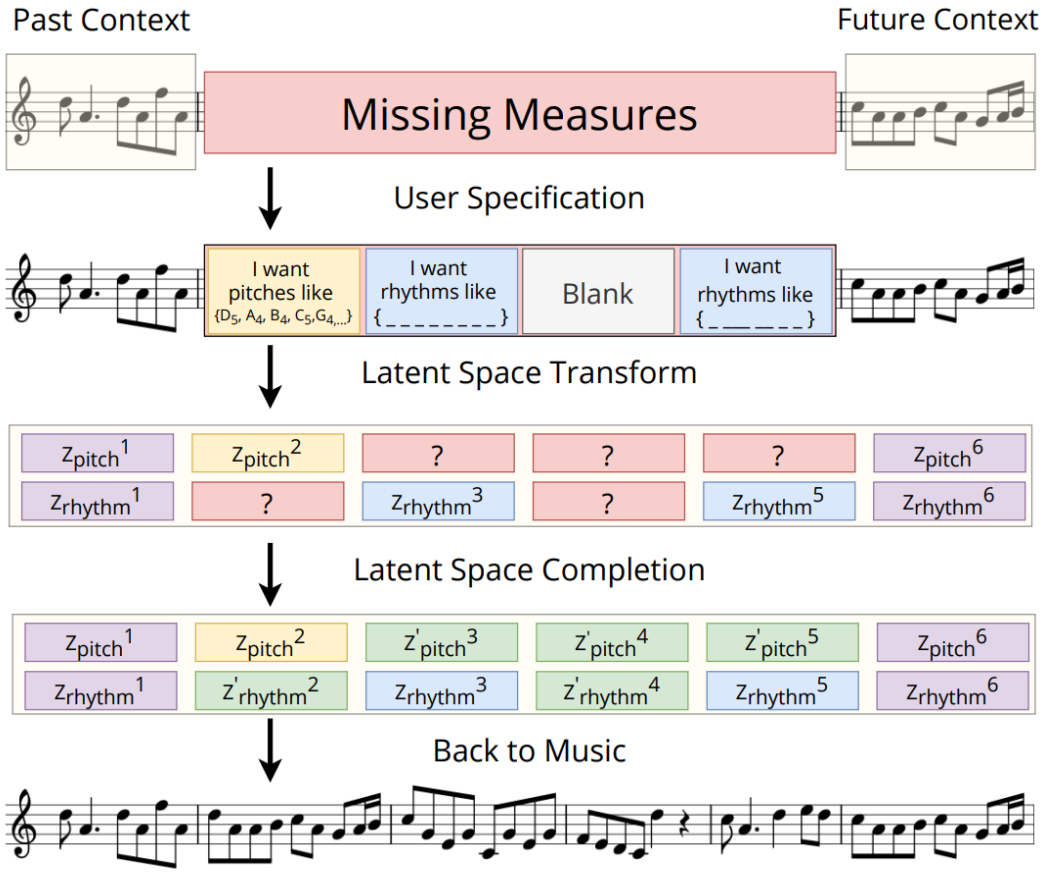


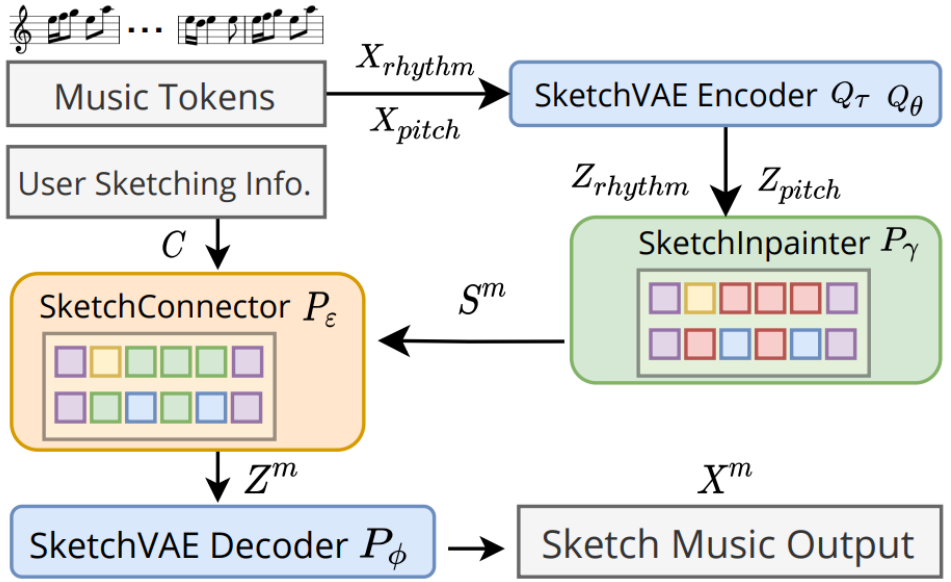
(Source: Tan & Herremeans, 2020)

music-fadernets.github.io

Hao Hao Tan and Dorien Herremans, "Music FaderNets: Controllable Music Generation Based On High-Level Features via Low-Level Feature Modelling," ISMIR, 2020.

Music SketchNet (Chen et al., 2020)

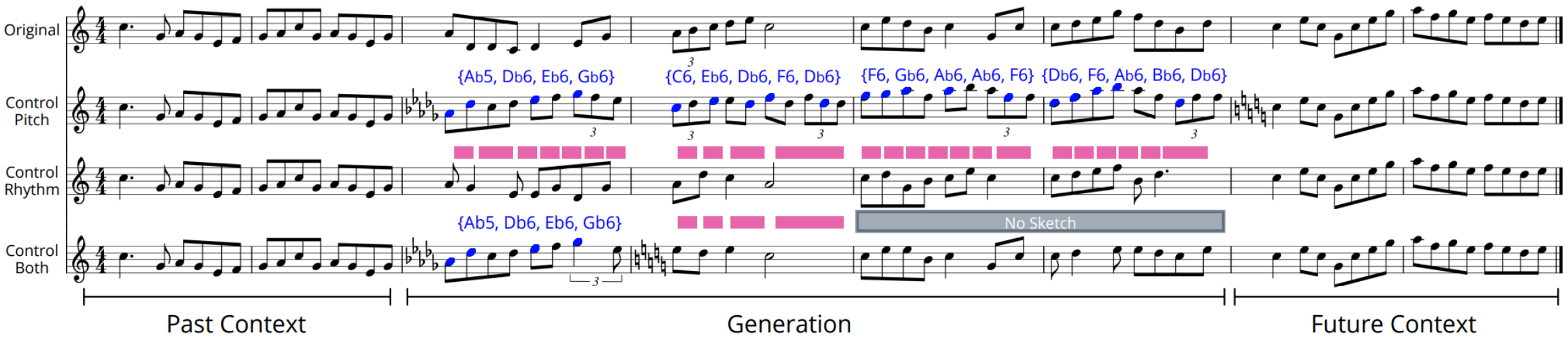




(Source: Chen et al., 2020)

Ke Chen, Cheng-i Wang, Taylor Berg-Kirkpatrick, and Shlomo Dubnov, "Music FaderNets: Controllable Music Generation Based On High-Level Features via Low-Level Feature Modelling," *ISMIR*, 2020.

Music SketchNet (Chen et al., 2020)

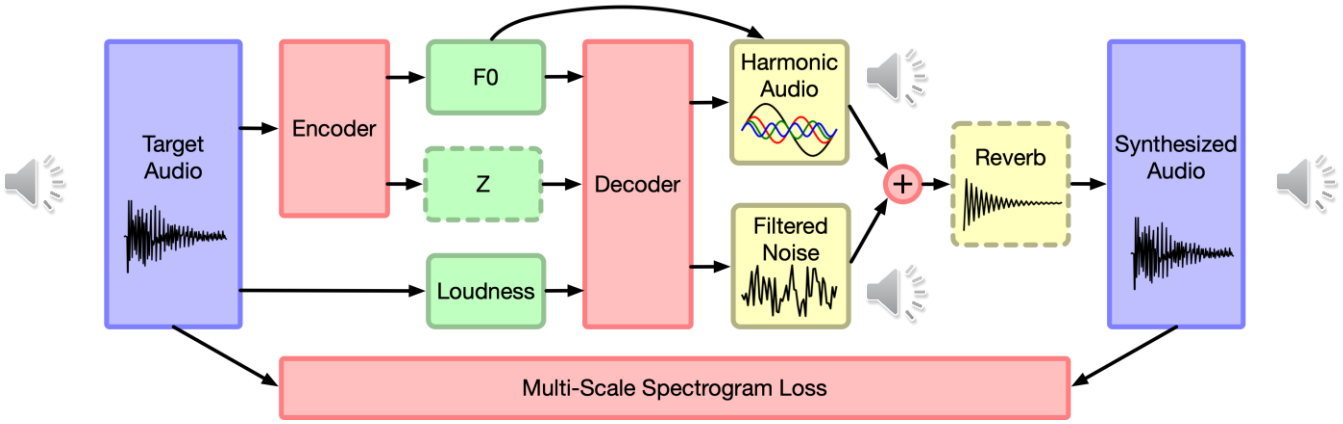


(Source: Chen et al., 2020)

Ke Chen, Cheng-i Wang, Taylor Berg-Kirkpatrick, and Shlomo Dubnov, "Music FaderNets: Controllable Music Generation Based On High-Level Features via Low-Level Feature Modelling," *ISMIR*, 2020.

Next Lecture

Audio-domain Music Generation



(Source: Engel et al., 2020)