PAT 498/598 (Winter 2025)

Music & AI

Lecture 15: Pianoroll-based Music Generation

Instructor: Hao-Wen Dong

Homework 5: AI Song Contest

- Please listen to the **ten finalists of AI Song Contest 2024**
- **Read the about pages** by clicking the cover arts
- **Answer the following questions** (in 5-10 sentences each)
  - **Which is your favorite song**?
  - Following Q1, **what did they do well**?
  - Following Q1, **what can be improved**?
  - Based on the ten finalists, **what tasks are easy** for current AI in music production?
  - Based on the ten finalists, **what tasks are difficult** for current AI in music production?

Homework 5: AI Song Contest

- Instructions will be released on the course website
- Please submit your work to Gradescope
- Due at **11:59pm ET** on **March 14**
- Late submissions: **1 point deducted per day**
- No late submission is allowed a week after the due date

Project

- **Open-ended group project** (group size: 2–3)
  - **Building a new AI music tool or Exploring creative & artistic use of AI tools**
- **Milestones**
  - **Pitch**: Mar 19
  - **Presentation**: Apr 21
  - **Final report**: Apr 28
- Due at **11:59pm ET** on the date specified
- **No late submissions!** Submit your work early and update it later.

Project Pitch

- Brief 10-min presentation
  - **Team member introduction**
  - **Topic**: What do you want to work on?
  - **Topic**: Who is the target audience/user/customer/reader?
  - **Methodology**: How are you going to approach it?
  - **Methodology**: What are the tools (programming languages, platforms, plugins, hardware, etc.) that you'll be using?
  - **Expected results**: What are the expected deliverables (e.g., an instrument, a plugin, a web/mobile app, a standalone software, an installation, a performance, a composition)?
  - **Planning**: What are the milestones? What do you expect to achieve by the end of February and March?

Project Pitch

- Send me an email with the following info by **11:59 PM ET on March 19**
  - **Names and U-M IDs of all team members**
  - **Topic**: What do you want to work on?
  - **Topic**: Who is the target audience/user/customer/reader?
  - **Methodology**: How are you going to approach it?
  - **Methodology**: What are the tools (programming languages, platforms, plugins, hardware, etc.) that you'll be using?
  - **Expected results**: What are the expected deliverables (e.g., an instrument, a plugin, a web/mobile app, a standalone software, an installation, a performance, a composition)?
  - **Planning**: What are the milestones? What do you expect to achieve by the end of February and March?

(Recap) AI Song Contest

- Annual international competition showcasing the **creative potential of human–AI co-creativity in the songwriting process**

aisongcontest.com

(Recap) Yaboi Hanoi – Entering Demons & Gods (2022)

youtu.be/PbrRoR3nEVw

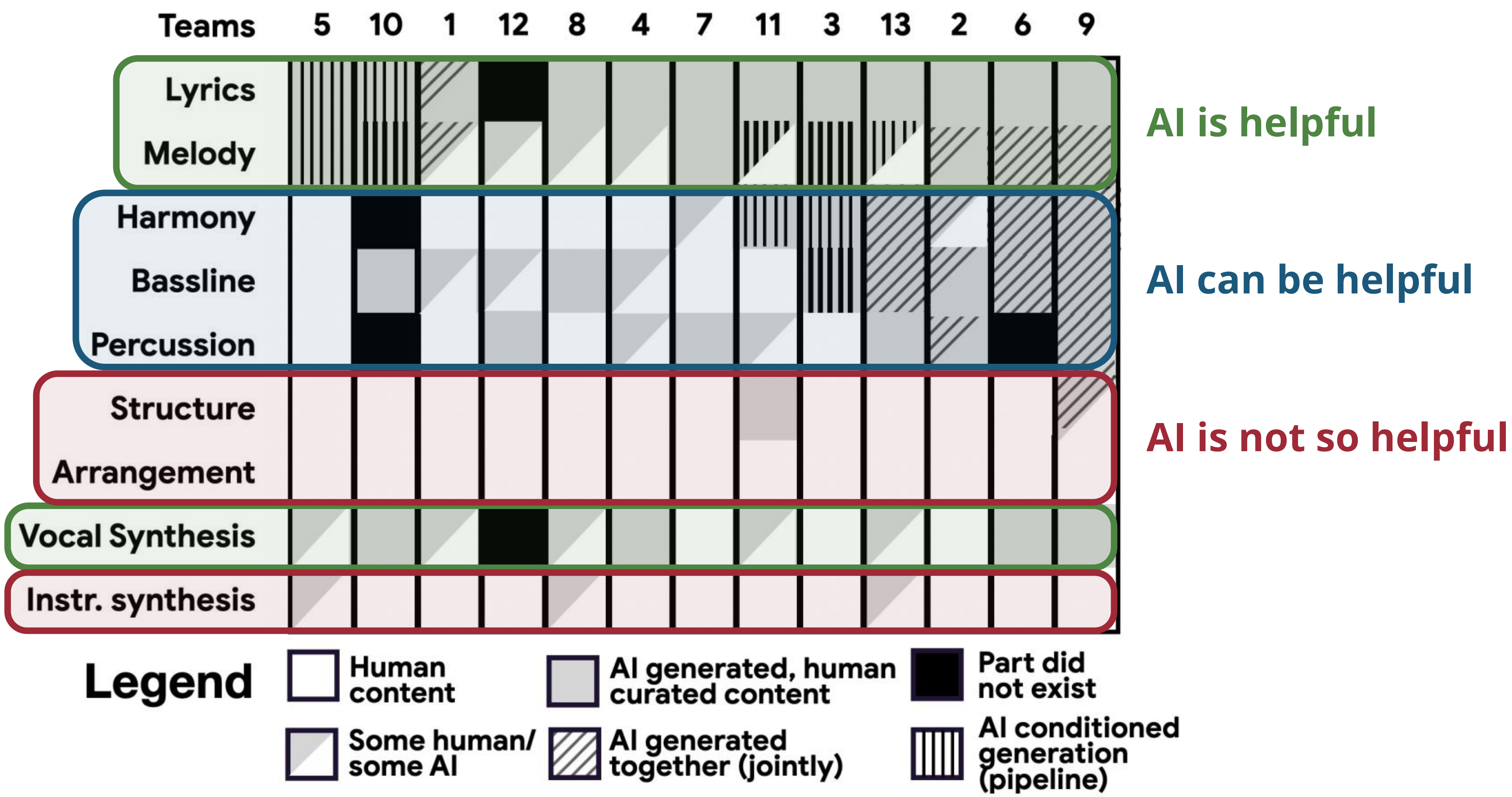
soundcloud.com/yaboihanoi/enter-demons-and-gods

(Recap) The Making of Entering Demons & Gods (2022)

“**It was like a saxophonist trained in classical Thai motifs, who played a special ‘Thai Edition’ saxophone with Phi Nai tunings, had joined the musical conversation.** The same was true with the trumpet model and the ขลุ่ย ‘Khlui’ - a flute from Thai, Laos and Cambodian repertoire. I could assemble a **transcultural ensemble** to expand the sonic palette of Thai motifs, whilst adhering to underlying tunings and idiomatic inflections like never before.”

lamtharnhantrakul.github.io/enter-demons-and-gods/

(Recap) How can AI Augment Human Creativity?



(Source: Huang et al., 2020)

Cheng-Zhi Anna Huang, Hendrik Vincent Koops, Ed Newton-Rex, Monica Dinculescu, and Carrie J. Cai, "AI Song Contest: Human-AI Co-Creation in Songwriting," *ISMIR*, 2020.

(Recap) Four Paradigms of Music Generation

Symbolic music generation

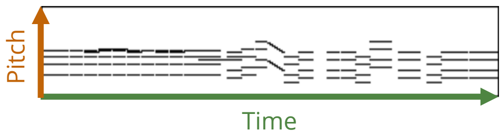
- **Text-based**
- **Image-based**

Audio-domain music generation

- **Time series-based**
- **Image-based**

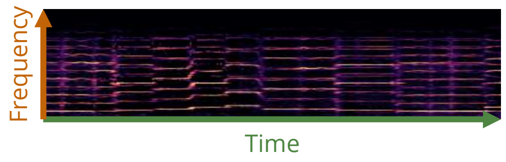
```
Program_change_0,
Note_on_60, Time_shift_2, Note_off_60,
Note_on_60, Time_shift_2, Note_off_60,
Note_on_76, Time_shift_2, Note_off_67,
Note_on_67, Time_shift_2, Note_off_67,
...
```

MIDI



Piano roll

Waveform



Spectrogram

Today, we also have many latent-space based systems!

(Recap) Topics of Symbolic Music Generation

Unconditional

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

Today's topic!

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 → sheet music
- Video → sheet music
- X → sheet music

(Recap) Two Paradigms of Symbolic Music Generation

Text-based

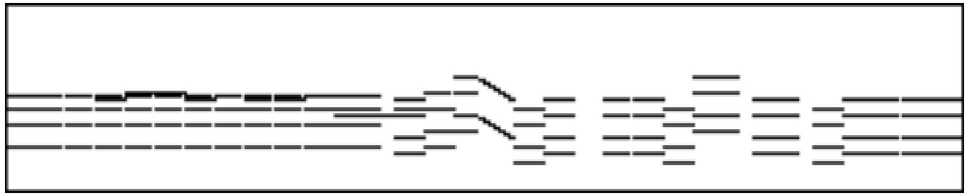
- Treat music like **text**
- Sharing models with **natural language processing (NLP)**
  - RNNs, LSTMs, Transformers, etc.

Today's topic!

```
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, ...
```

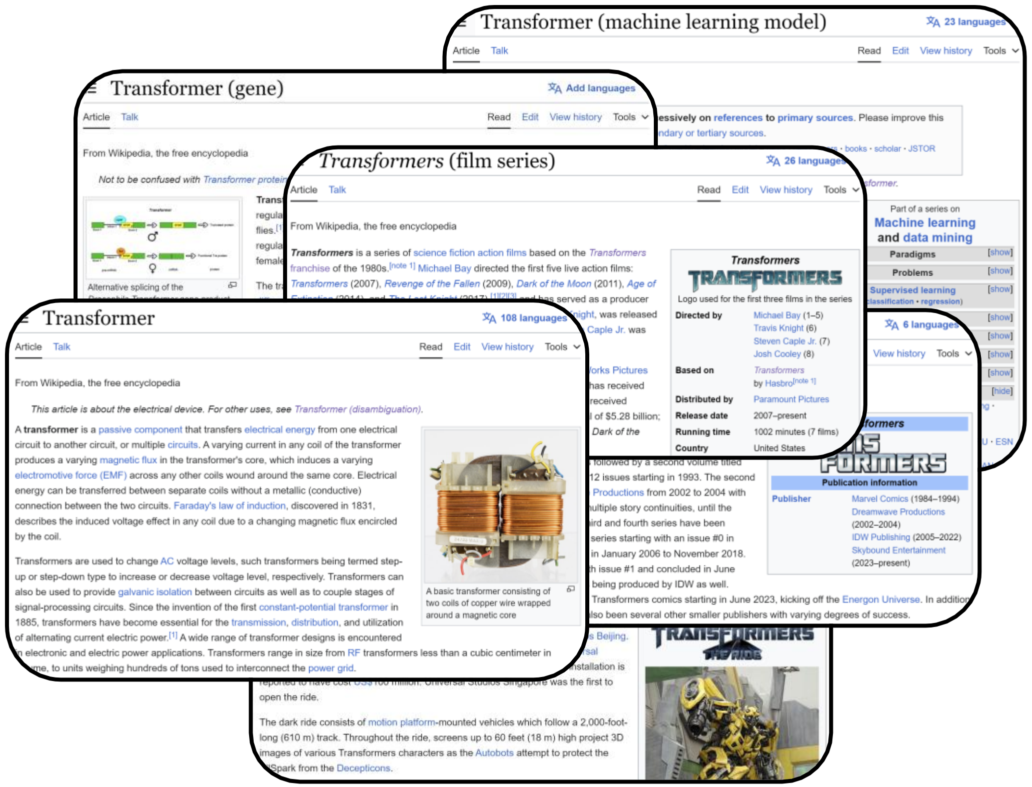
Image-based

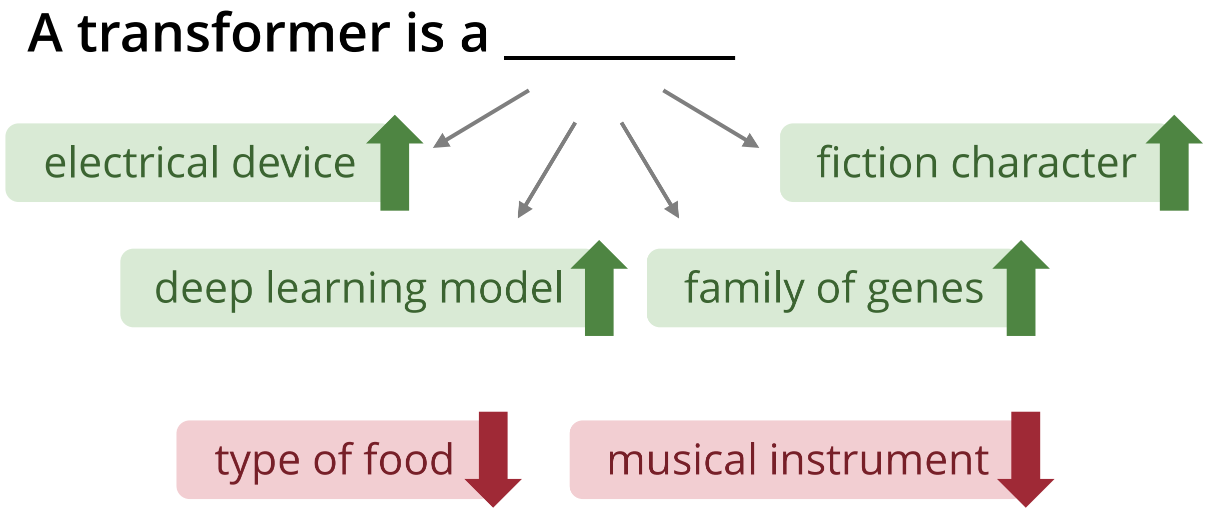
- Treat music like **images**
- Sharing models with **computer vision (CV)**
  - GANs, VAEs, diffusion models, etc.



(Recap) Language Models

- Predicting the next word given the past sequence of words





(Recap) Language Models (Mathematically)

- A class of machine learning models that learn the next word probability

$$P(\,x_i \mid \underbrace{x_1, x_2, \dots, x_{i-1}}\,)$$

Next word — Previous words

$P($ electrical $|$ A transformer is a $)$ ↑

$P($ character $|$ A transformer is a $)$ ↑

$P($ gene $|$ A transformer is a $)$ ↑

$P($ model $|$ A transformer is a $)$ ↑

$P($ food $|$ A transformer is a $)$ ↓

$P($ musical $|$ A transformer is a $)$ ↓

(Recap) Language Models – Generation

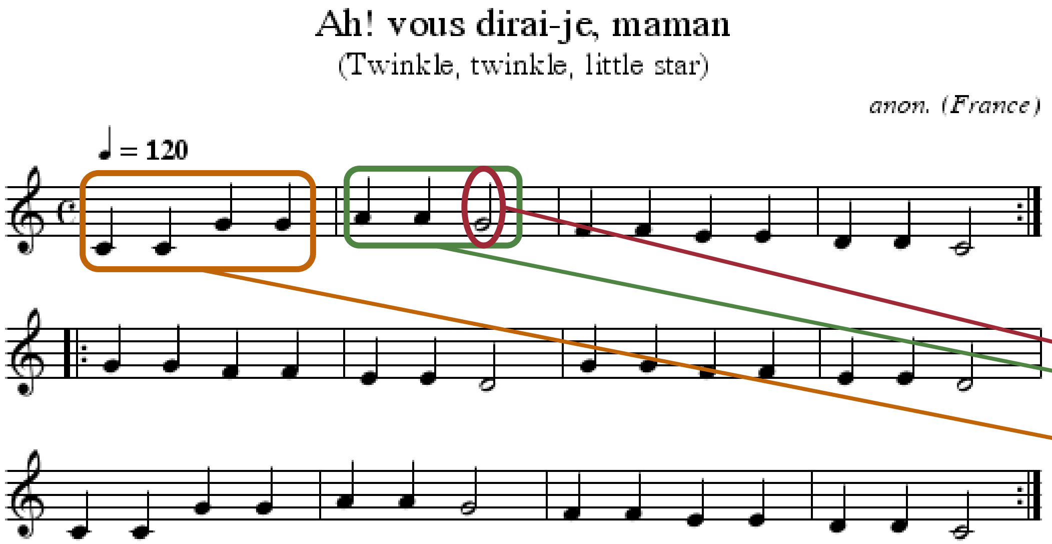
- How do we generate a new sentence using a trained language model?

A transformer is a	→	Model	→	deep
A transformer is a deep	→	Model	→	learning
A transformer is a deep learning	→	Model	→	model
A transformer is a deep learning model	→	Model	→	introduced
A transformer is a deep learning model introduced	→	Model	→	in
A transformer is a deep learning model introduced in	→	Model	→	2017

(Recap) Designing a Machine-readable Music Language

- How can we "represent" music in a way that machines understand?
  - Musical representation is a key component of a music generation system
- Why not using sheet music "images" directly?
  - Machines still have a hard time reading sheet music
  - A challenging task known as "optical music recognition" (OMR)
- Examples:
  - ABC notation
  - MIDI

(Recap) An Example of ABC Notation

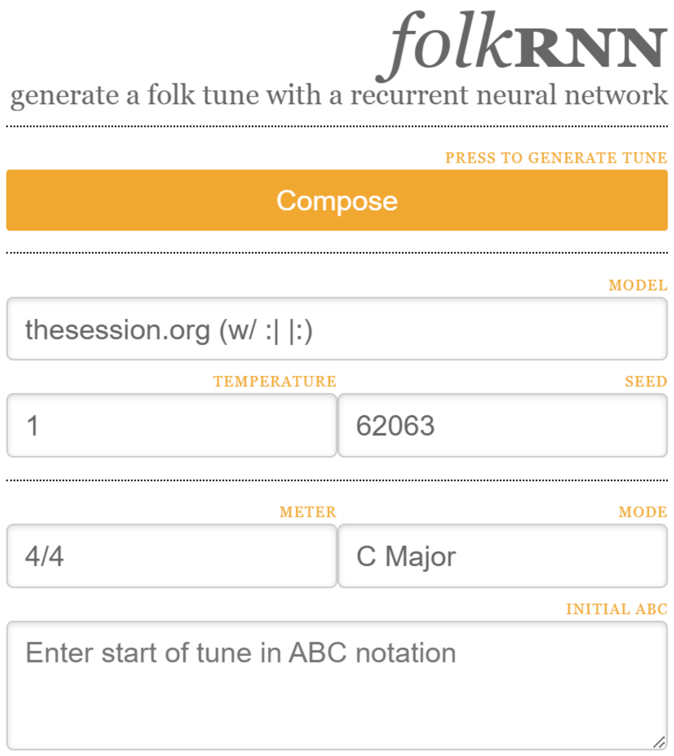



```
X:571
T:Ah! vous dirai-je, maman
T:(Twinkle, twinkle, little star)
C:anon.
O:France
R:Nursery song
M:C
L:1/4
Q:120
K:C
CCGG|AAG2|FFEE|DDC2:|
|:GGFF|EED2|GGFF|EED2|
CCGG|AAG2|FFEE|DDC2:|
```

Metadata (X: through R:)
M:C Meter
L:1/4 Unit note length (temporal resolution)
Q:120 Tempo
K:C Key

(Recap) Example System: Folk RNN (Sturm et al., 2015)

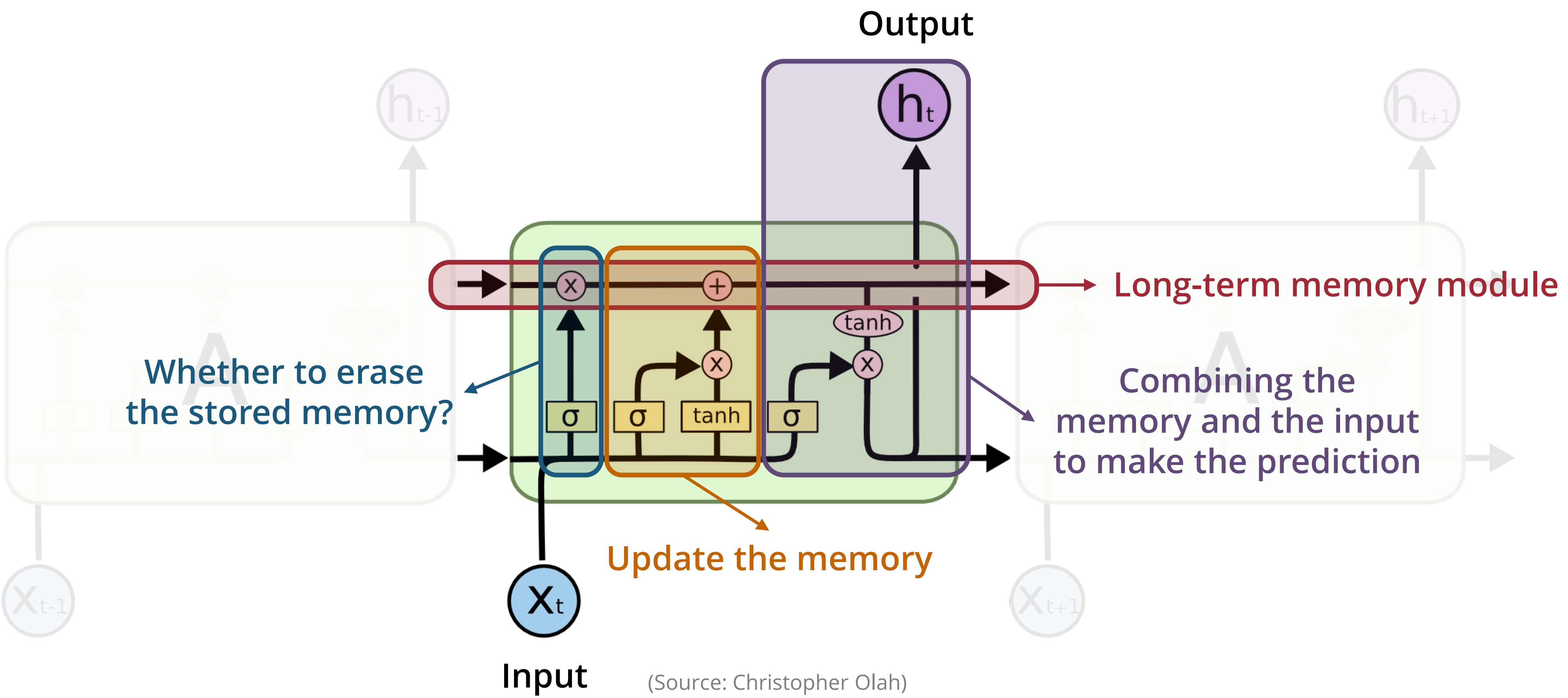
- Data
  - Collections of folk tunes
- Representation
  - ABC notation without metadata
- Model
  - LSTM (long short-term memory)
  - Working on the character level



folkrnn.org

Bob L. Sturm, Joao Felipe Santos, and Iryna Korshunova, "Folk Music Style Modelling by Recurrent Neural Networks with Long Short Term Memory Units," *ISMIR Late-Breaking Demos*, 2015.

(Recap) Demystifying LSTMs (Hochreiter & Schmidhuber, 1997)



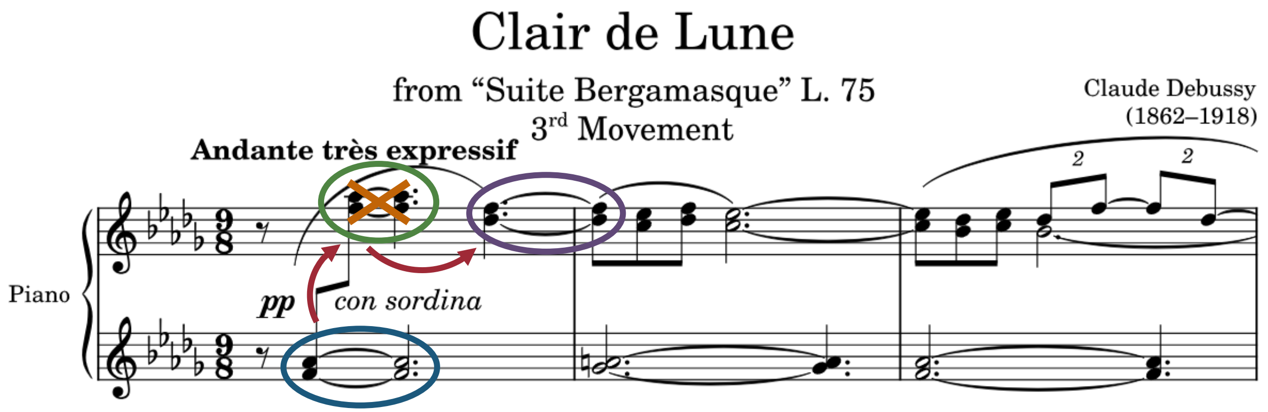
(Source: Christopher Olah)

colah.github.io/posts/2015-08-Understanding-LSTMs/

Sepp Hochreiter and Jürgen Schmidhuber, "Long Short-Term Memory," *Neural Computation*, 9(8):1735-1780, 1997.

(Recap) Representing Polyphonic Music

- We can now handle music with multi-pitch at the same time
  - In the literature, "polyphonic" & "multi-pitch" are often used interchangeably




```
Note_on_65, Note_on_68, Time_shift_eighth_note, Note_on_77, Note_on_80,
Time_shift_half_note, Note_off_77, Note_off_80, Note_on_73, Note_on_77,
Time_shift_dotted_quarter_note, Note_off_65, Note_off_68, ...
```

(Recap) Example: Performance RNN (Oore et al., 2020)

- Data
  - Yamaha e-Piano Competition dataset (MAESTRO)
- Representation
  - 128 Note-On events
  - 128 Note-Off events
  - 125 Time-Shift events (8ms–1s)
  - 32 Set-Velocity events — Handle dynamics
- Model
  - LSTM

Examples of generated music

Ian Simon and Sageev Oore, "Performance RNN: Generating Music with Expressive Timing and Dynamics," *Magenta Blog*, June 29, 2017.
Sageev Oore, Ian Simon, Sander Dieleman, Douglas Eck, and Karen Simonyan, "This Time with Feeling: Learning Expressive Musical Performance", *Neural Computing and Applications*, 32, 2020.

(Recap) Example: A.I. Duet (Mann et al., 2016)

youtu.be/0ZE1bfPtvZo

experiments.withgoogle.com/ai/ai-duet/view/

github.com/googlecreativelab/aiexperiments-ai-duet

(Recap) Example: Music Transformer (Huang et al., 2019)

- Data
  - Yamaha e-Piano Competition dataset (MAESTRO)
- Representation
  - 128 Note-On events
  - 128 Note-Off events
  - 100 Time-Shift events (10ms–1s)
  - 32 Set-Velocity events
- Model
  - Transformer

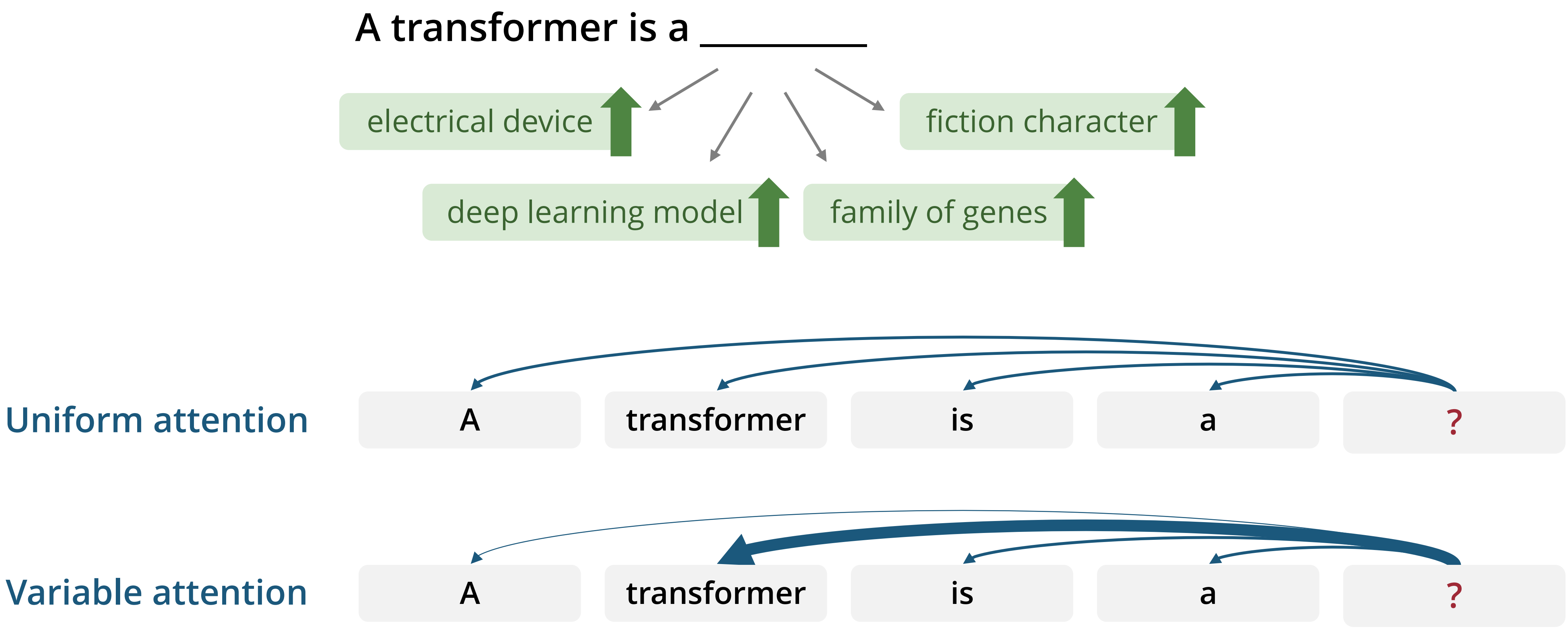
Almost the same representation as PerformanceRNN

Handle dynamics

Examples of generated music

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck, "Music Transformer: Generating Music with Long-Term Structure," *ICLR*, 2019.

(Recap) Self-attention Mechanism (Cheng et al., 2016)

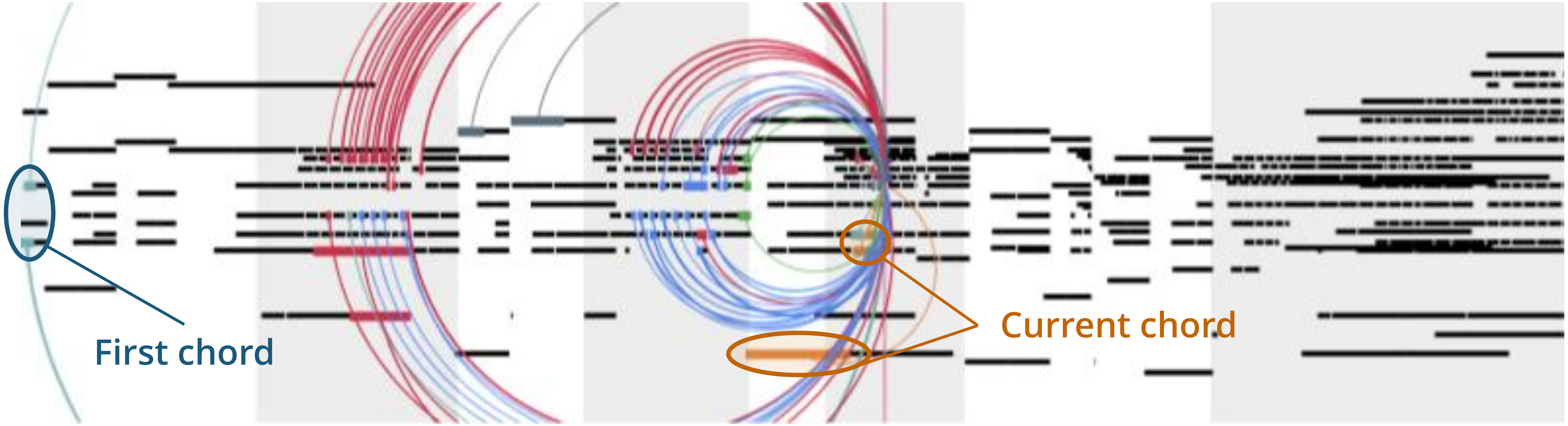


Transformers learn what to attend to from big data!

Jianpeng Cheng, Li Dong, and Mirella Lapata, "Long Short-Term Memory-Networks for Machine Reading," *EMNLP*, 2016.

(Recap) Visualizing Musical Self-attention

(Each color represents an attention head)



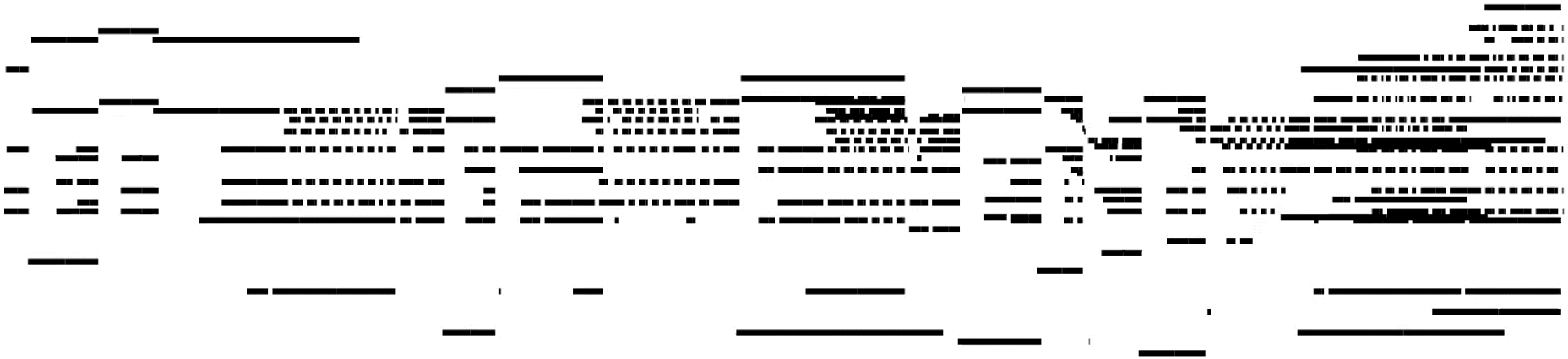
(Source: Huang et al., 2018)

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck, "Music Transformer: Generating Music with Long-Term Structure," *ICLR*, 2019.

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck, "Music Transformer: Generating Music with Long-Term Structure," *Magenta Blog*, December 13, 2018.

(Recap) Visualizing Musical Self-attention

(Each color represents an attention head)



(Source: Huang et al., 2018)

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck, "Music Transformer: Generating Music with Long-Term Structure," *ICLR*, 2019.
Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck, "Music Transformer: Generating Music with Long-Term Structure," *Magenta Blog*, December 13, 2018.

(Recap) Example: MuseNet (Payne et al., 2019)

- **Data**: ClassicalArchives + BitMidi + MAESTRO
- **Representation**: "**instrument:velocity:pitch**"
  - Time shifts in real time (sec)
- **Model**: Transformer

```
bach piano_strings start tempo90
piano:v72:G1 piano:v72:G2 piano:v72:B4
piano:v72:D4 violin:v80:G4 piano:v72:G4
piano:v72:B5 piano:v72:D5 wait:12
piano:v0:B5 wait:5 piano:v72:D5 wait:12
...
```

Example of generated music

Christine Payne, "MuseNet," *OpenAI*, 2019.

(Recap) Example: Multitrack Music Transformer (Dong et al., 2023)

- **Data**: Symbolic Orchestral Database (SOD)
- **Representation**: "(**beat**, **position**, **pitch**, **duration**, **instrument**)"
  - No time shift events **Why?**
- **Model**: Multi-dimensional Transformer

```
(0, 0,  0,  0,  0,  0) | Start of song
(1, 0,  0,  0,  0, 15) | Instrument: accordion
(1, 0,  0,  0,  0, 36) | Instrument: trombone
(1, 0,  0,  0,  0, 39) | Instrument: brasses
(2, 0,  0,  0,  0,  0) | Start of notes
(3, 1,  1, 41, 15, 36) | Note: beat=1, position=1,  pitch=E2, duration=48, instrument=trombone
(3, 1,  1, 65,  4, 39) | Note: beat=1, position=1,  pitch=E4, duration=12, instrument=brasses
(3, 1,  1, 65, 17, 15) | Note: beat=1, position=1,  pitch=E4, duration=72, instrument=accordion
(3, 1,  1, 68,  4, 39) | Note: beat=1, position=1,  pitch=G4, duration=12, instrument=brasses
(3, 1,  1, 68, 17, 15) | Note: beat=1, position=1,  pitch=G4, duration=72, instrument=accordion
(3, 1,  1, 73, 17, 15) | Note: beat=1, position=1,  pitch=C5, duration=72, instrument=accordion
(3, 1, 13, 68,  4, 39) | Note: beat=1, position=13, pitch=G4, duration=12, instrument=brasses
(3, 1, 13, 73,  4, 39) | Note: beat=1, position=13, pitch=C5, duration=12, instrument=brasses
(3, 2,  1, 73, 12, 39) | Note: beat=2, position=1,  pitch=C5, duration=36, instrument=brasses
(3, 2,  1, 77, 12, 39) | Note: beat=2, position=1,  pitch=E5, duration=36, instrument=brasses
          ...          | ...
(4, 0,  0,  0,  0,  0) | End of song
```

(Source: Dong et al., 2023)

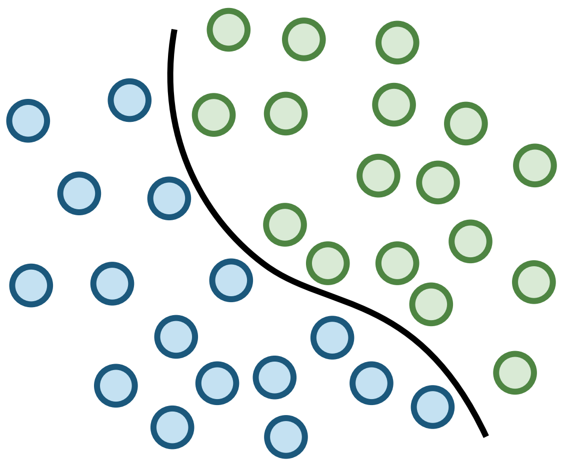
Example of generated music

Hao-Wen Dong, Ke Chen, Shlomo Dubnov, Julian McAuley, and Taylor Berg-Kirkpatrick, "Multitrack Music Transformer," *ICASSP*, 2023.

Generative Adversarial Nets (GANs)

Discriminative vs Generative Models

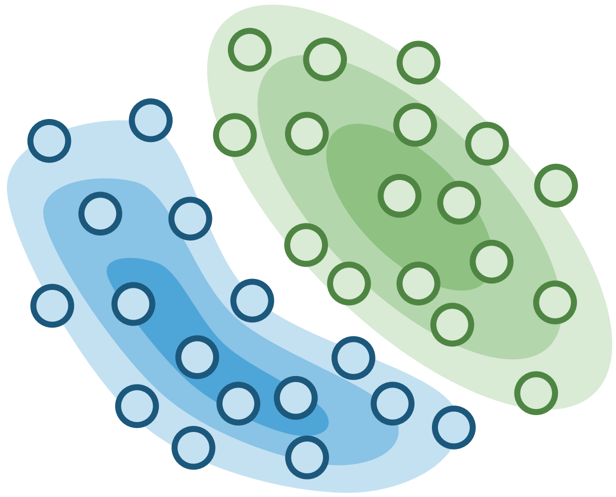
Discriminative



Discriminative models learn the decision boundary

$P(y|x)$

Generative

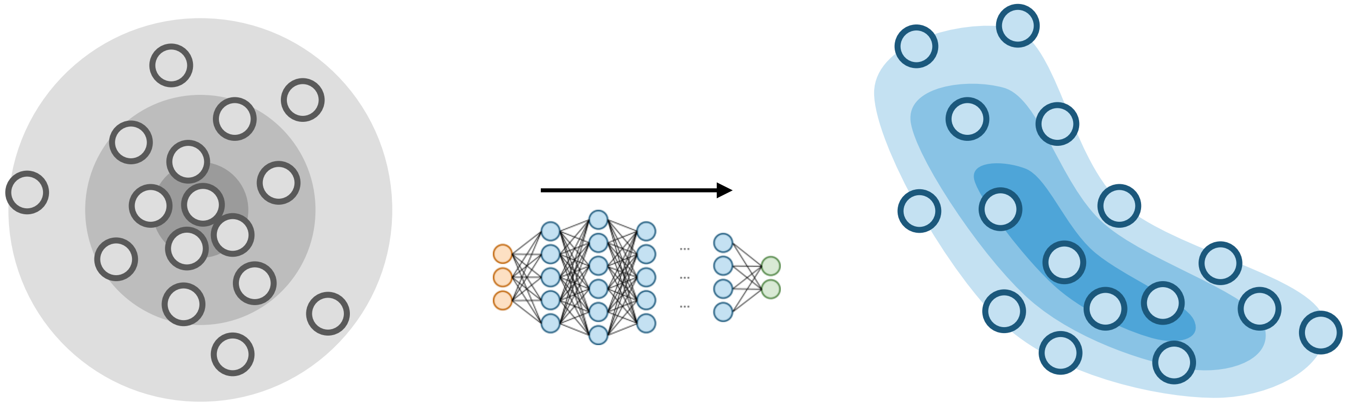


Generative models learn the underlying distribution

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

Generating Data from a Random Distribution

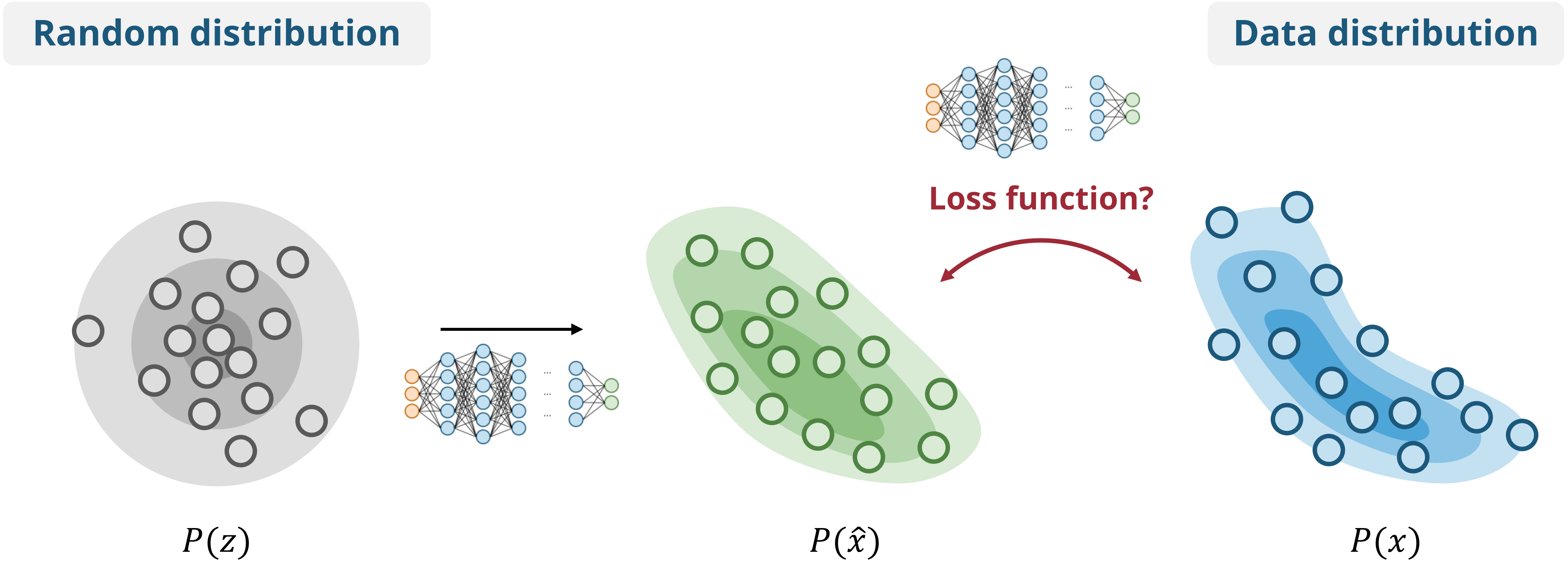
Random distribution **Data distribution**



$P(z)$ $P(x)$

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

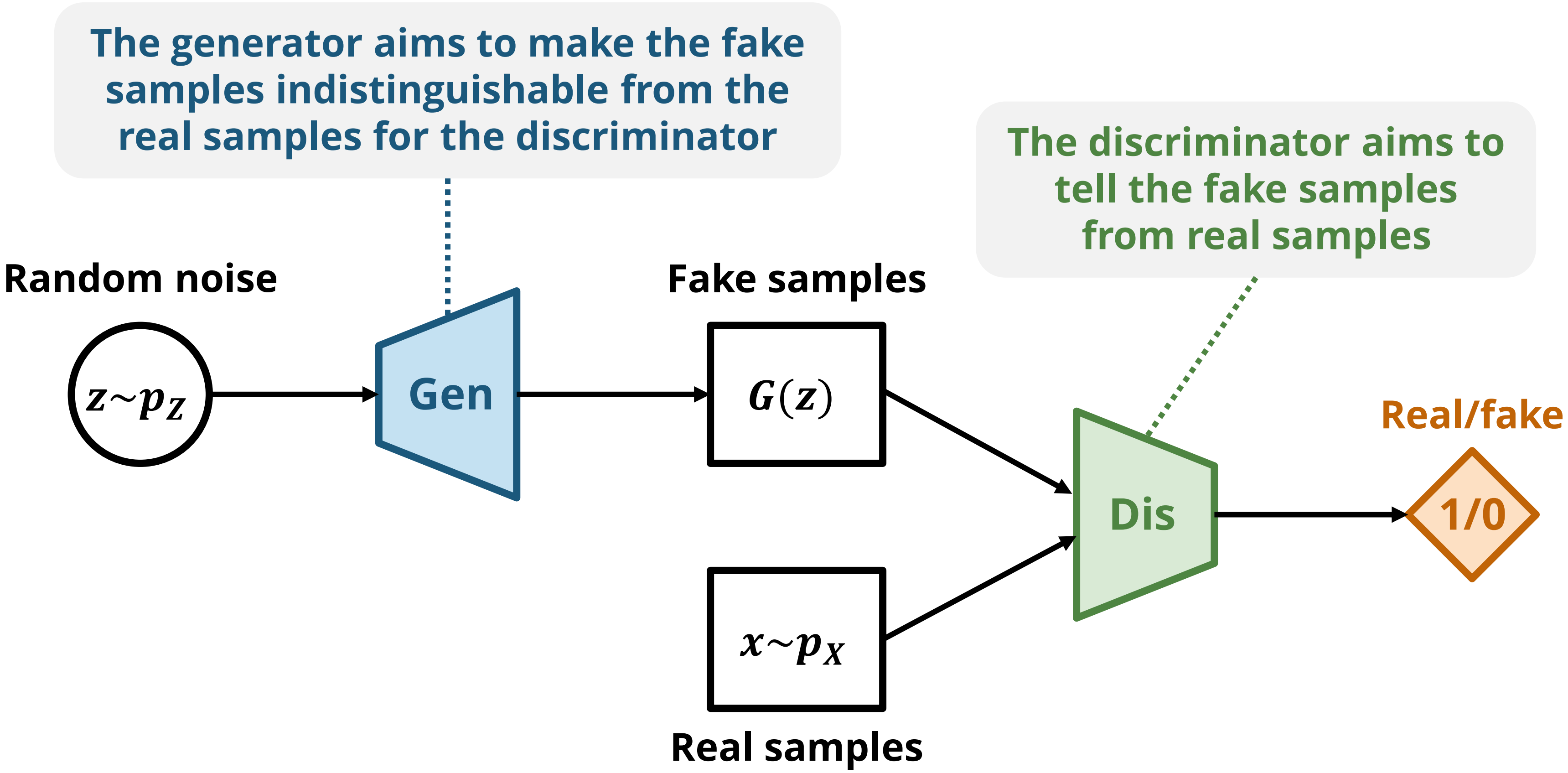
A Loss Function for Distributions



Unfortunately, no easy way to measure the difference between two distributions

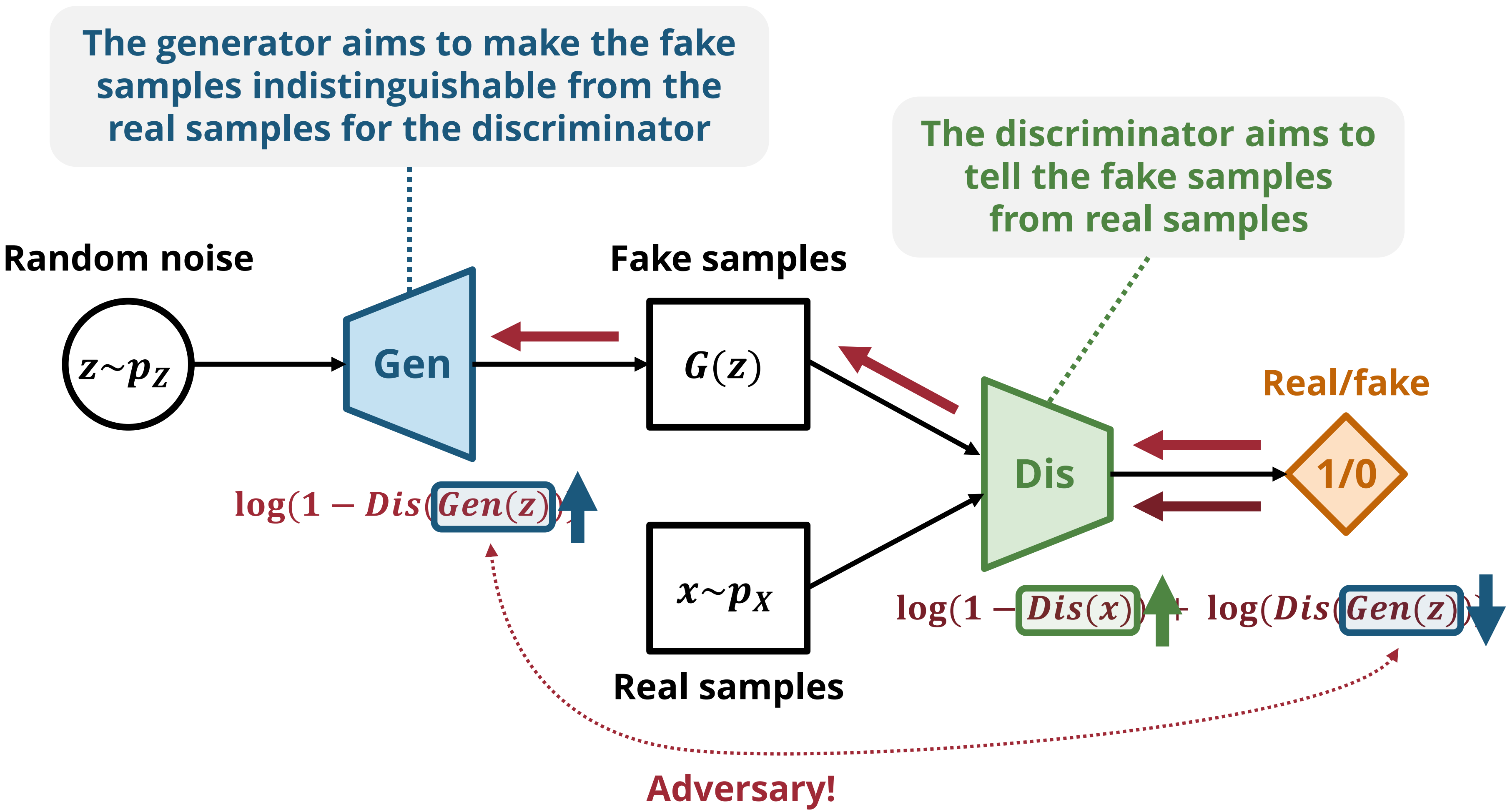
But what about another neural network!?

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



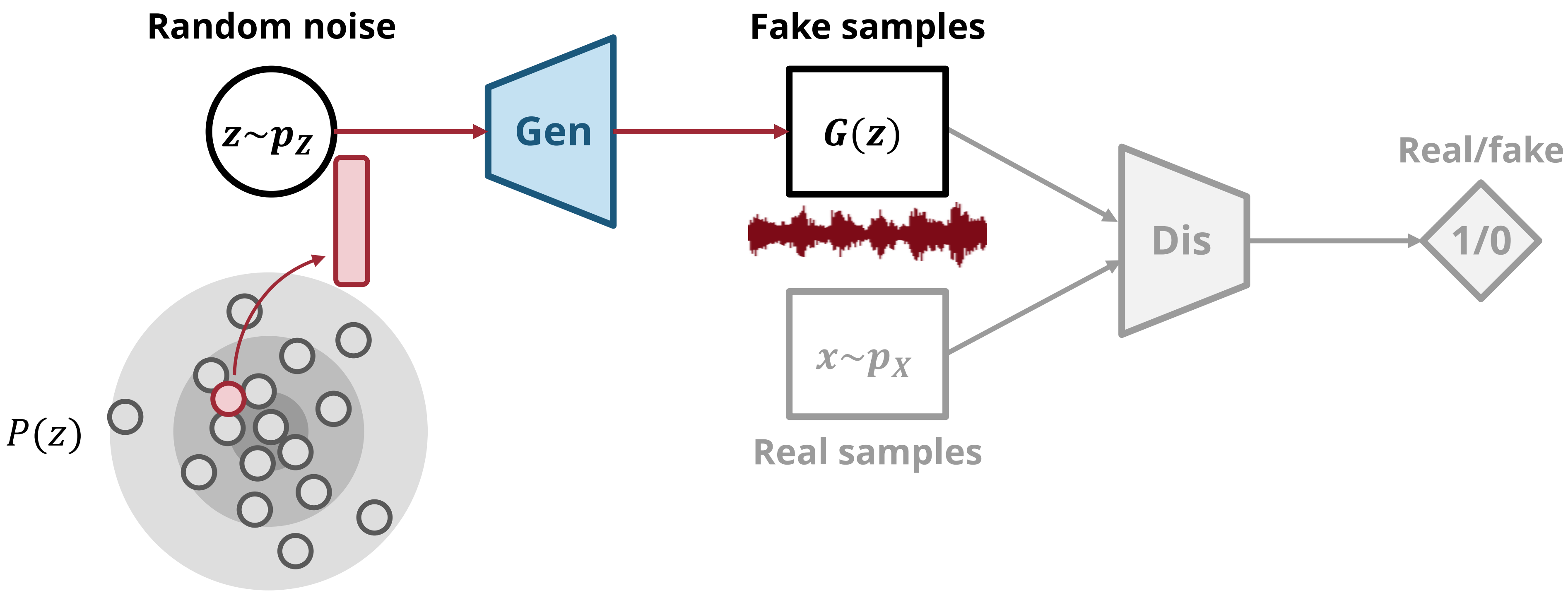
Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative Adversarial Networks," *NeurIPS*, 2014.

Generative Adversarial Nets (GANs) – Training



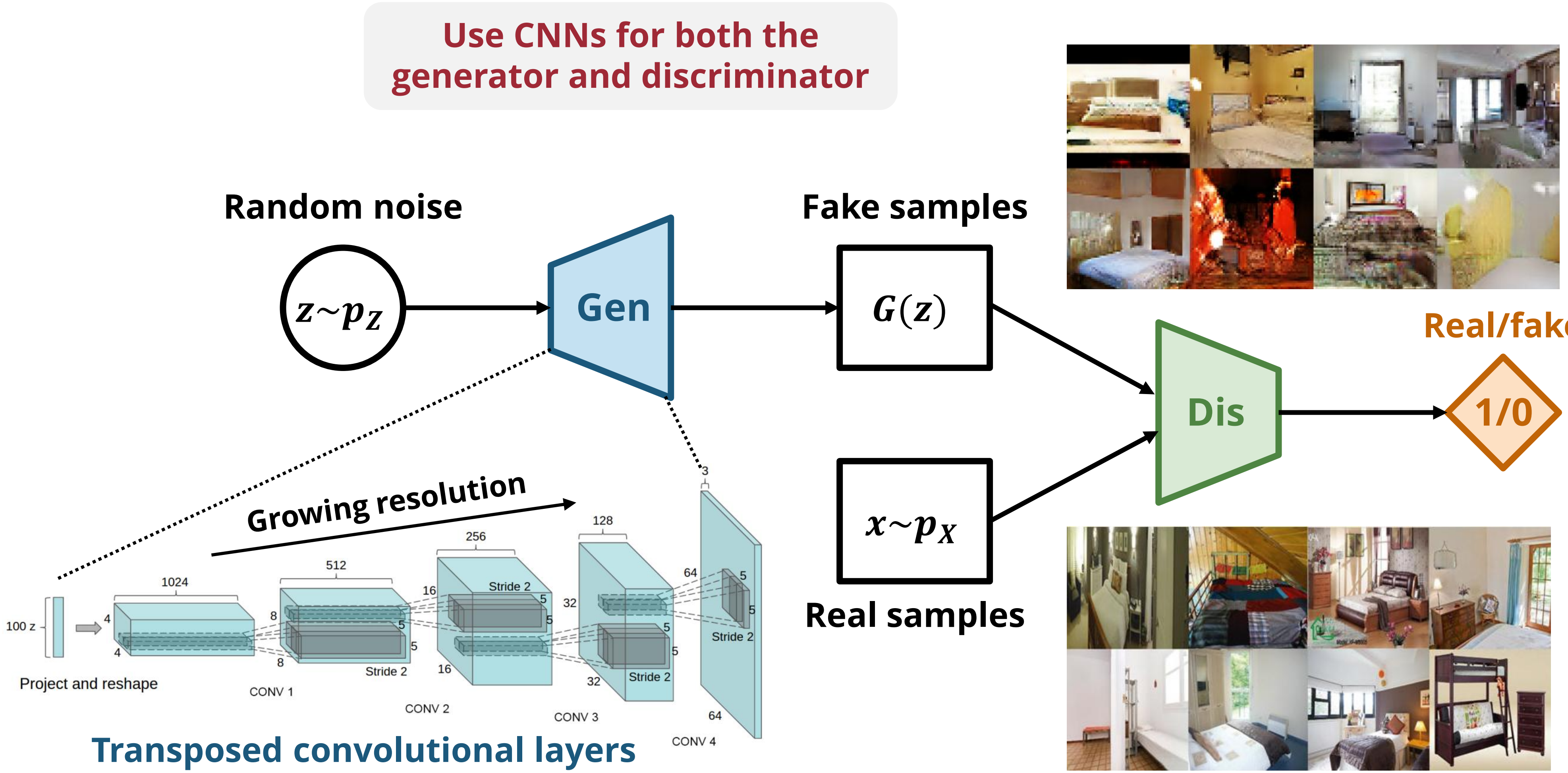
Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative Adversarial Networks," *NeurIPS*, 2014.

Generative Adversarial Nets (GANs) – Generation



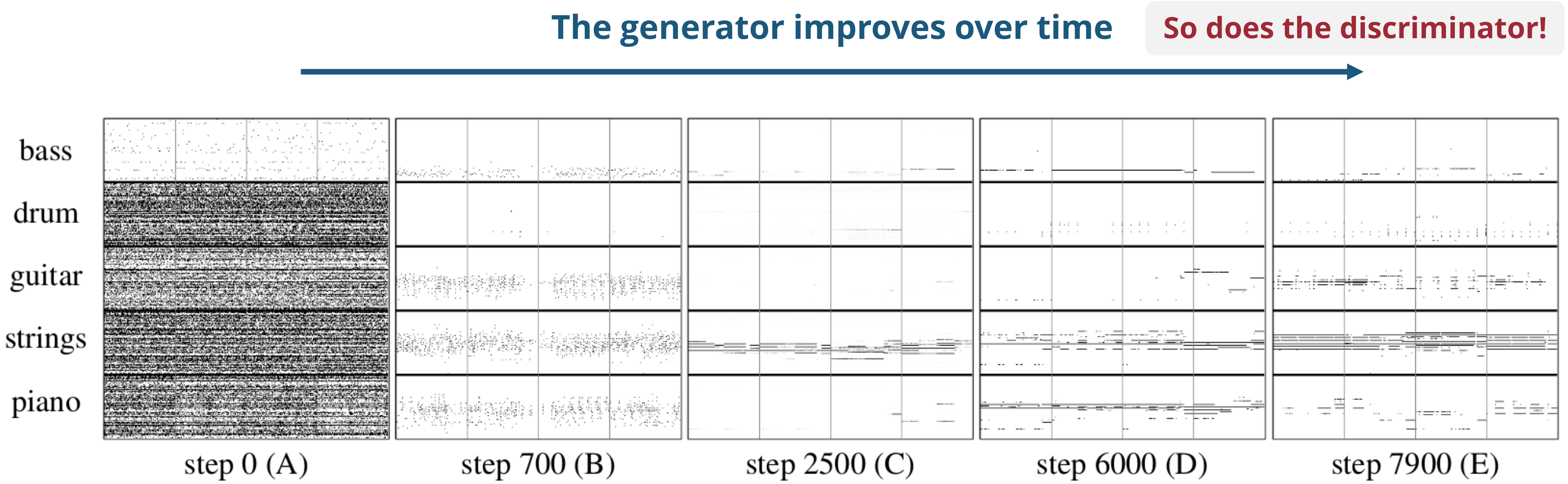
Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative Adversarial Networks," *NeurIPS*, 2014.

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



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

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

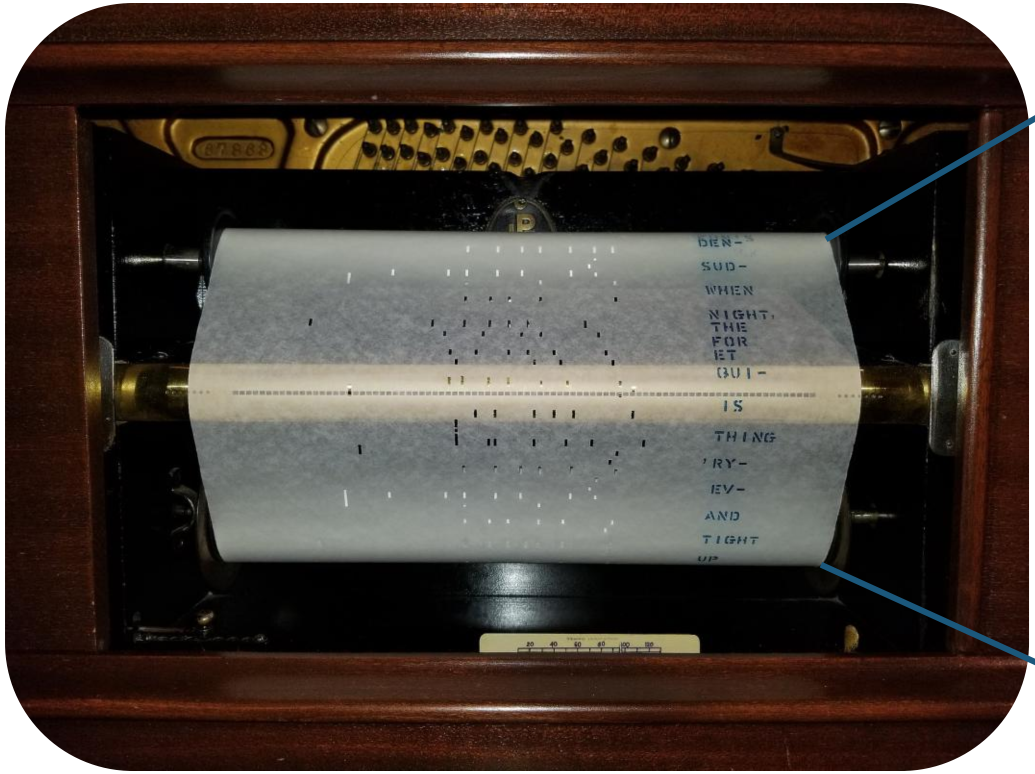


(Source: Dong et al., 2018)

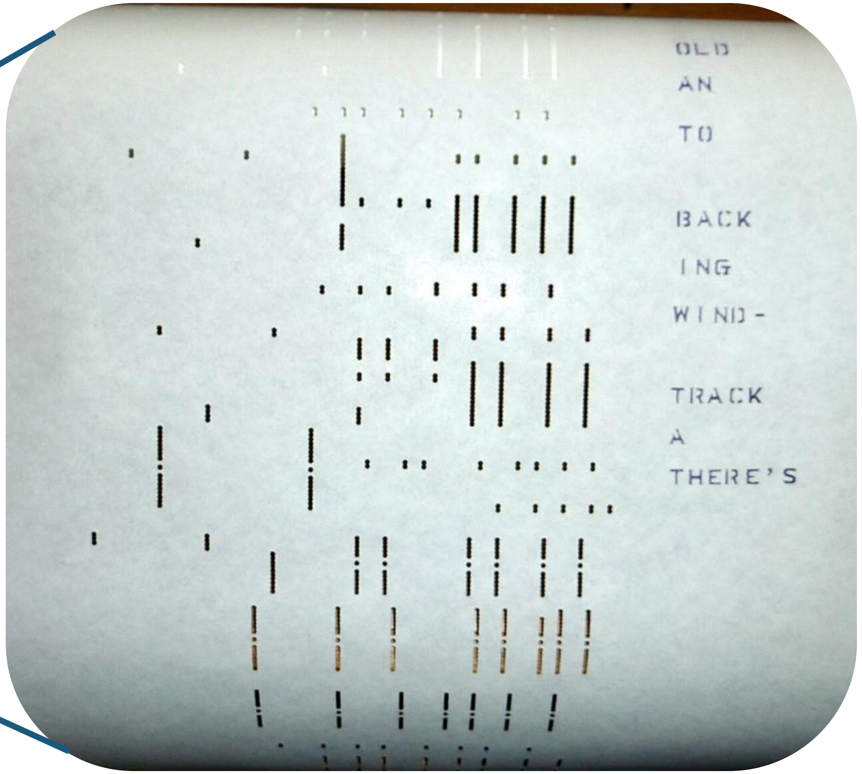
Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang, "MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment," *AAAI*, 2018.

Piano Roll Representation

Piano Rolls



(Source: Draconichiaro)

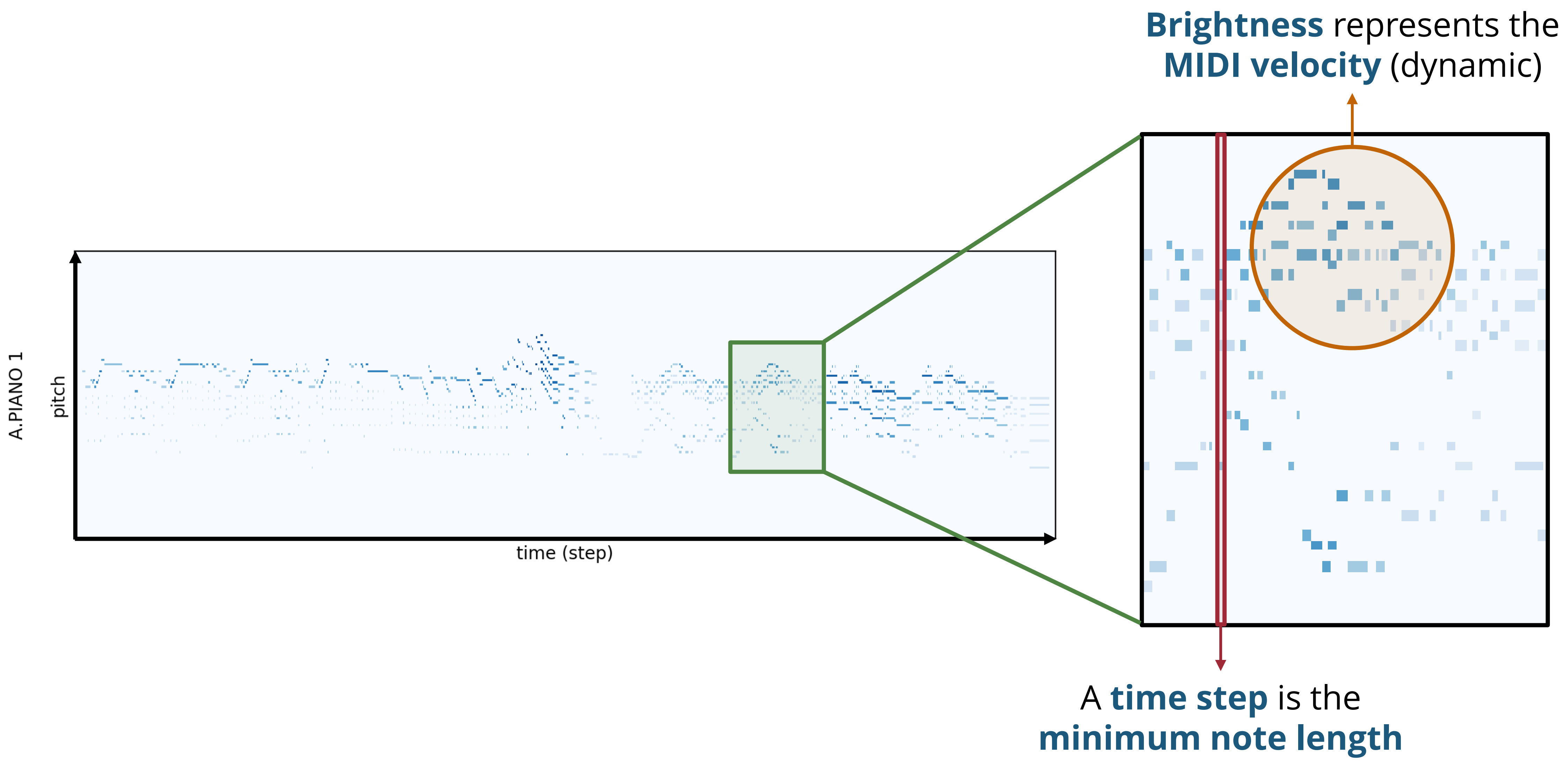


(Source: Tangerineduel)

Player Pianos

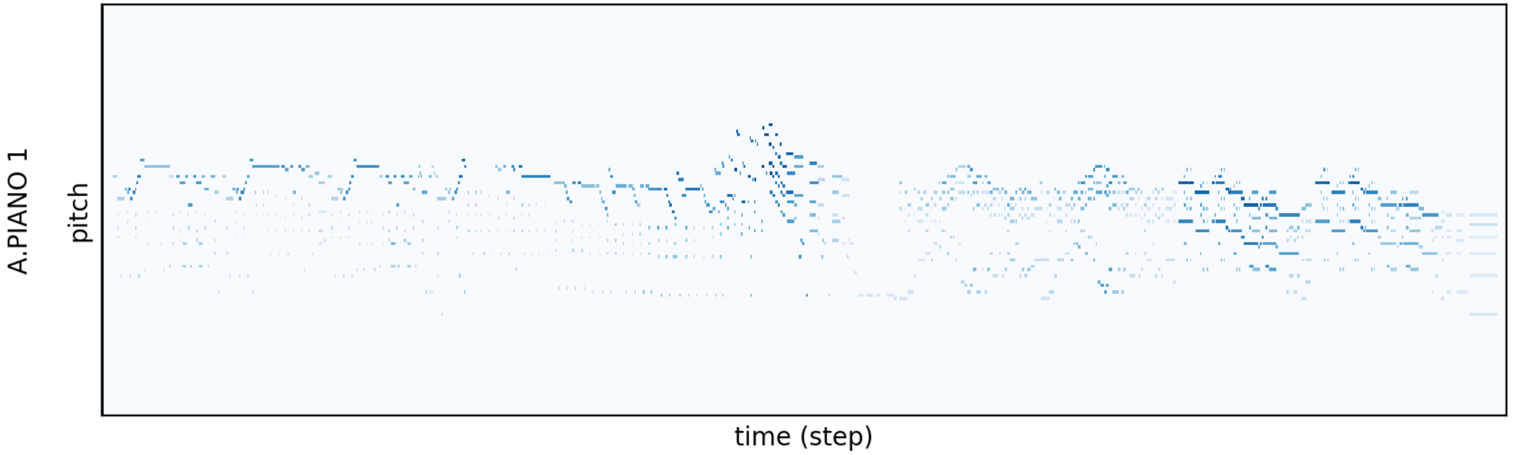
youtu.be/07krQ661fok

Piano Roll Representation

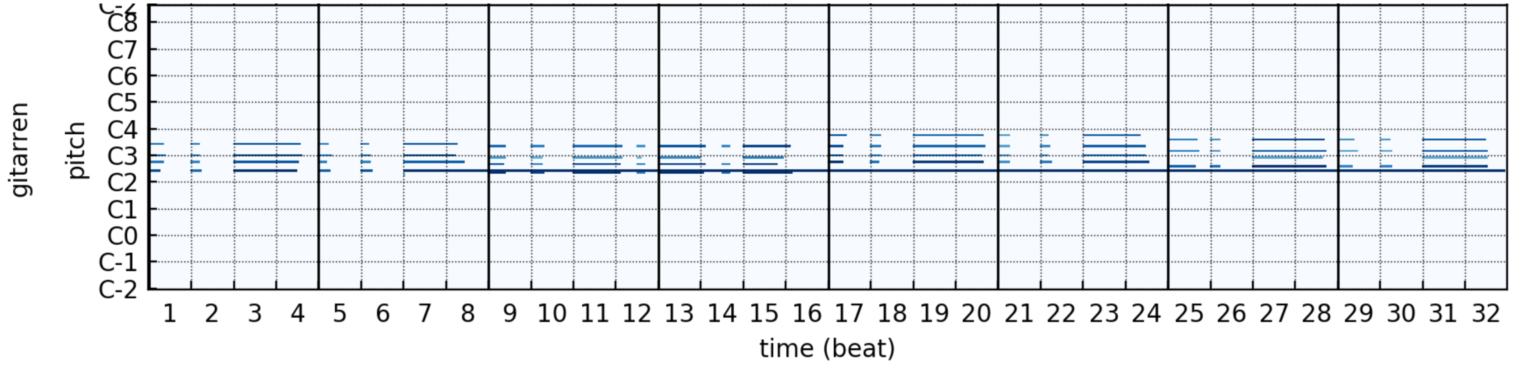


Piano Roll Representation

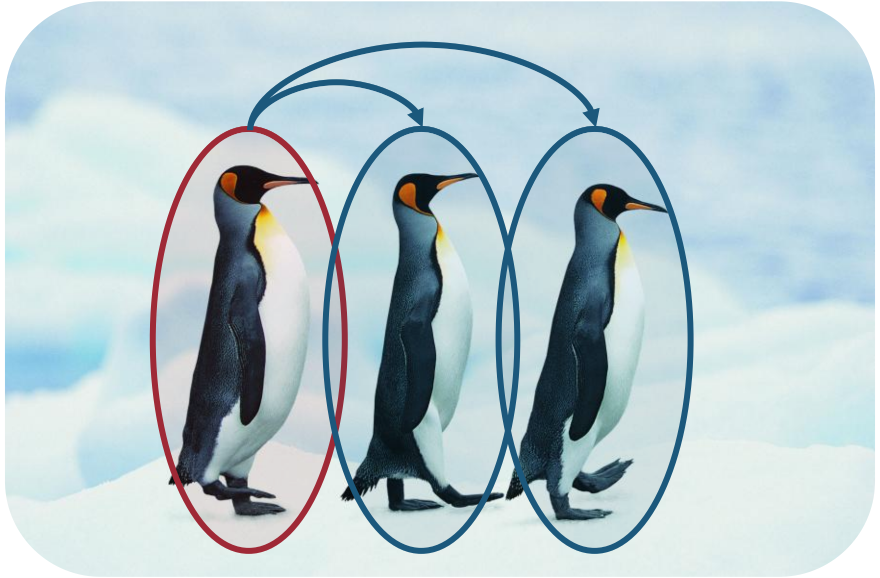
With expressive timing



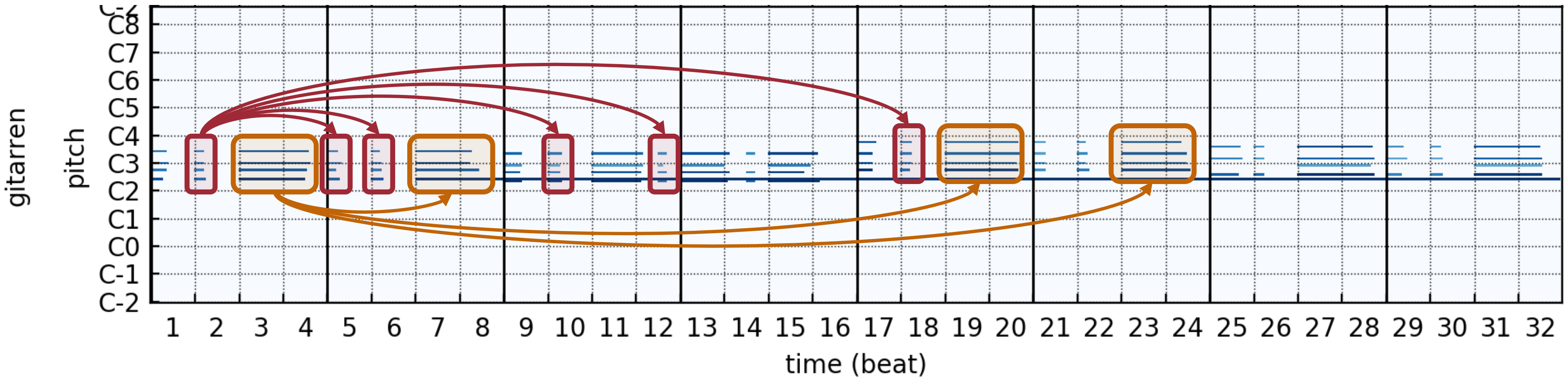
Without expressive timing



(Recap) Reusable Pattern Detectors



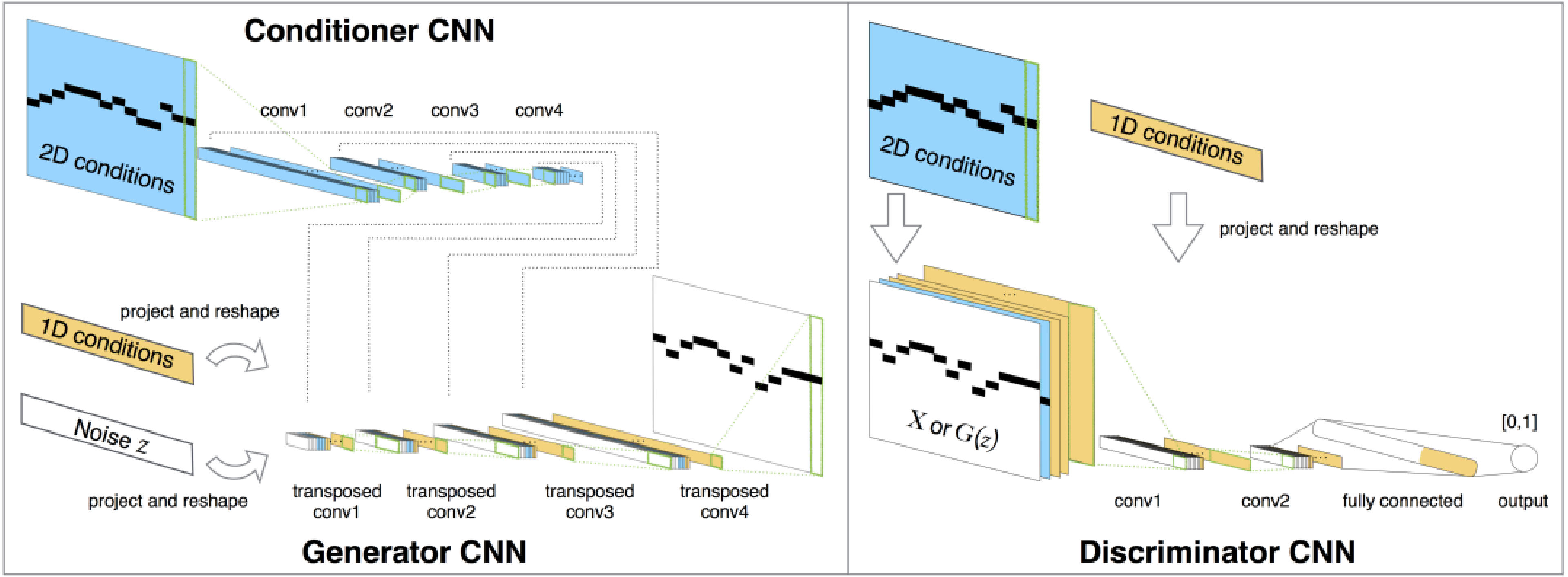
Why Piano Rolls?



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

Music Generation using GANs

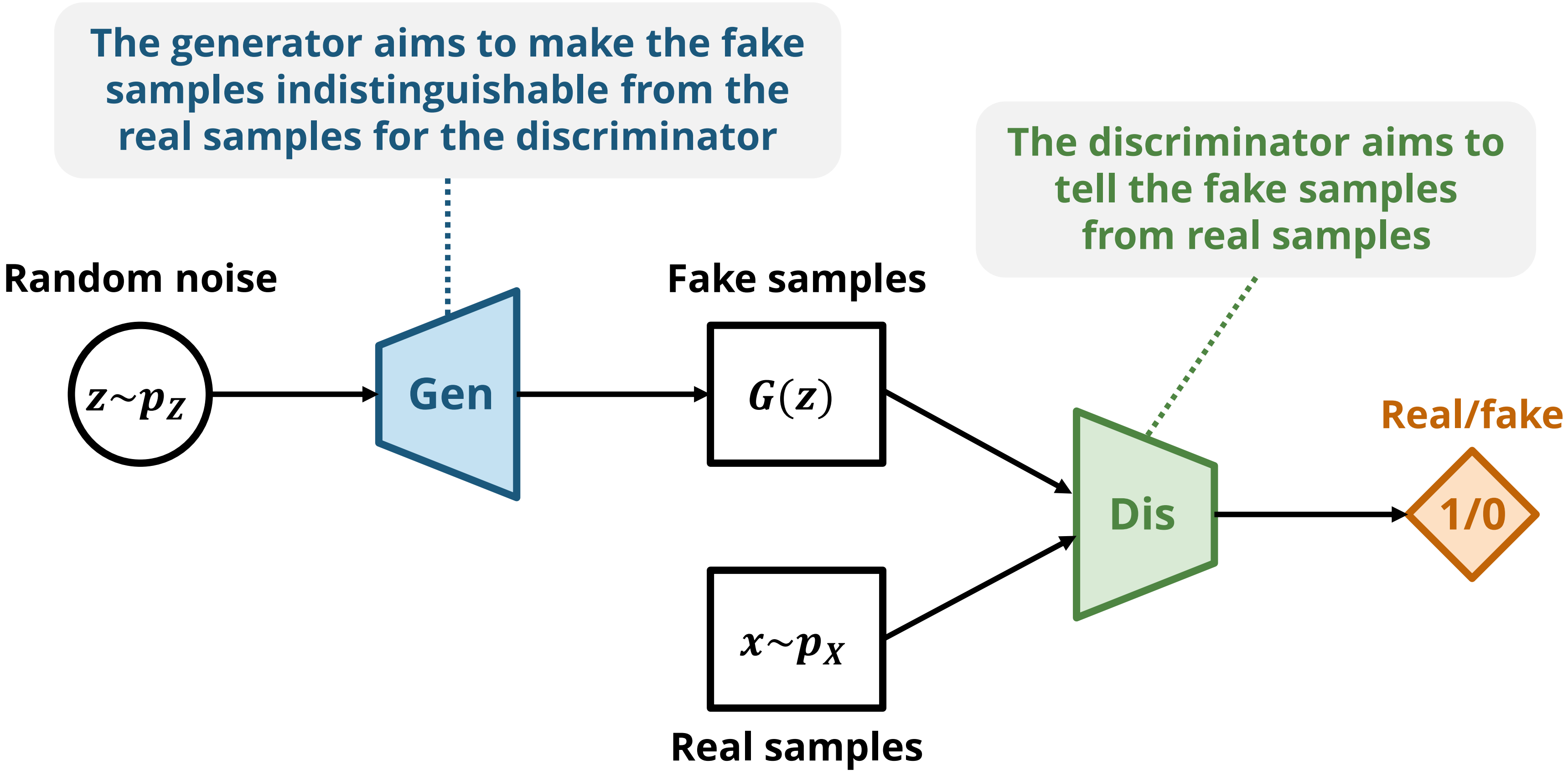
Example: MidiNet (Yang et al., 2017)



(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.

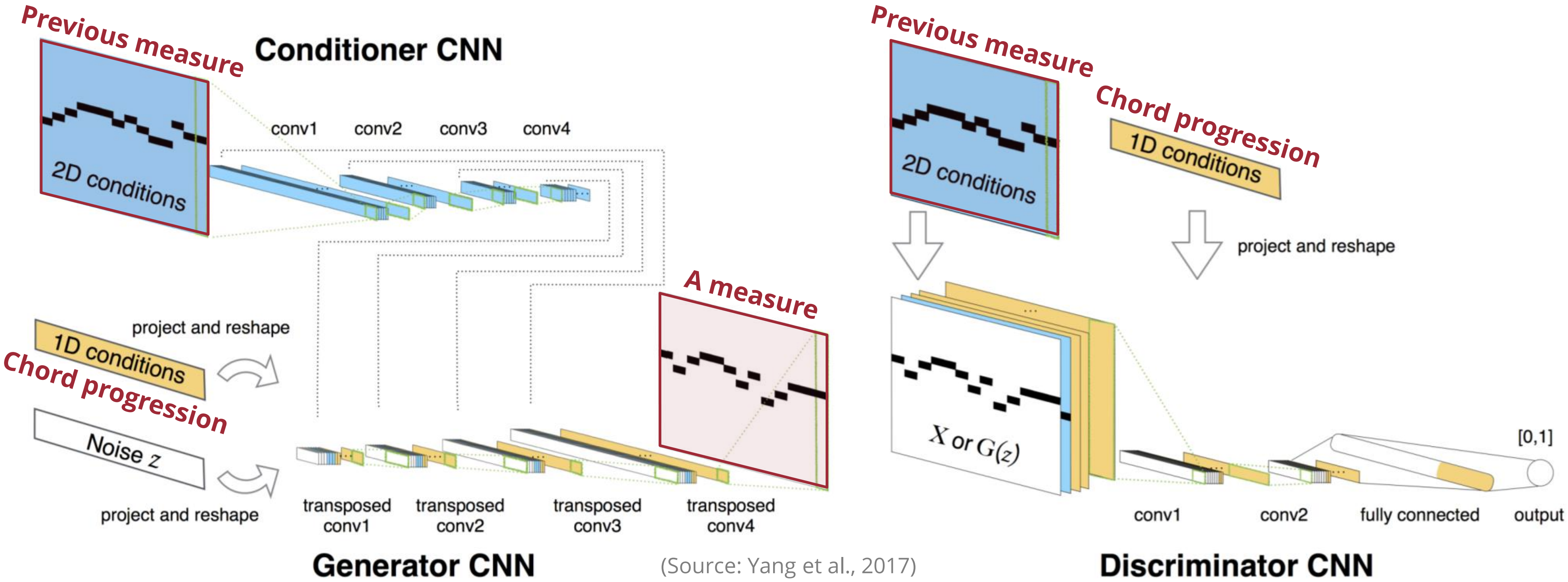
(Recap) Generative Adversarial Nets (GANs) (Goodfellow et al., 2014)



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.

Example: MidiNet (Yang et al., 2017)

Examples of generated music

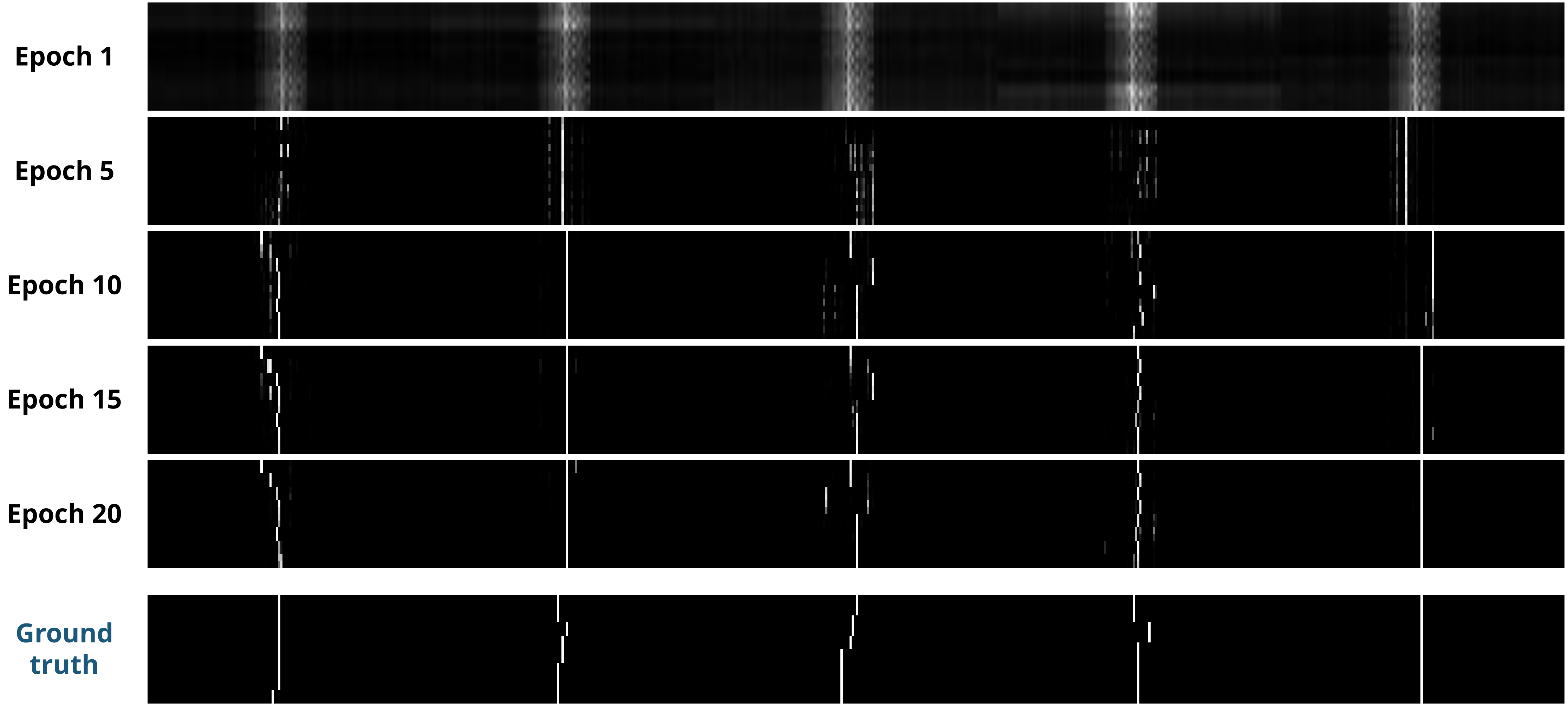


(Source: Yang et al., 2017)

MidiNet generates music measure-by-measure by conditioning on the last measure generated

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

Example: MidiNet (Yang et al., 2017)

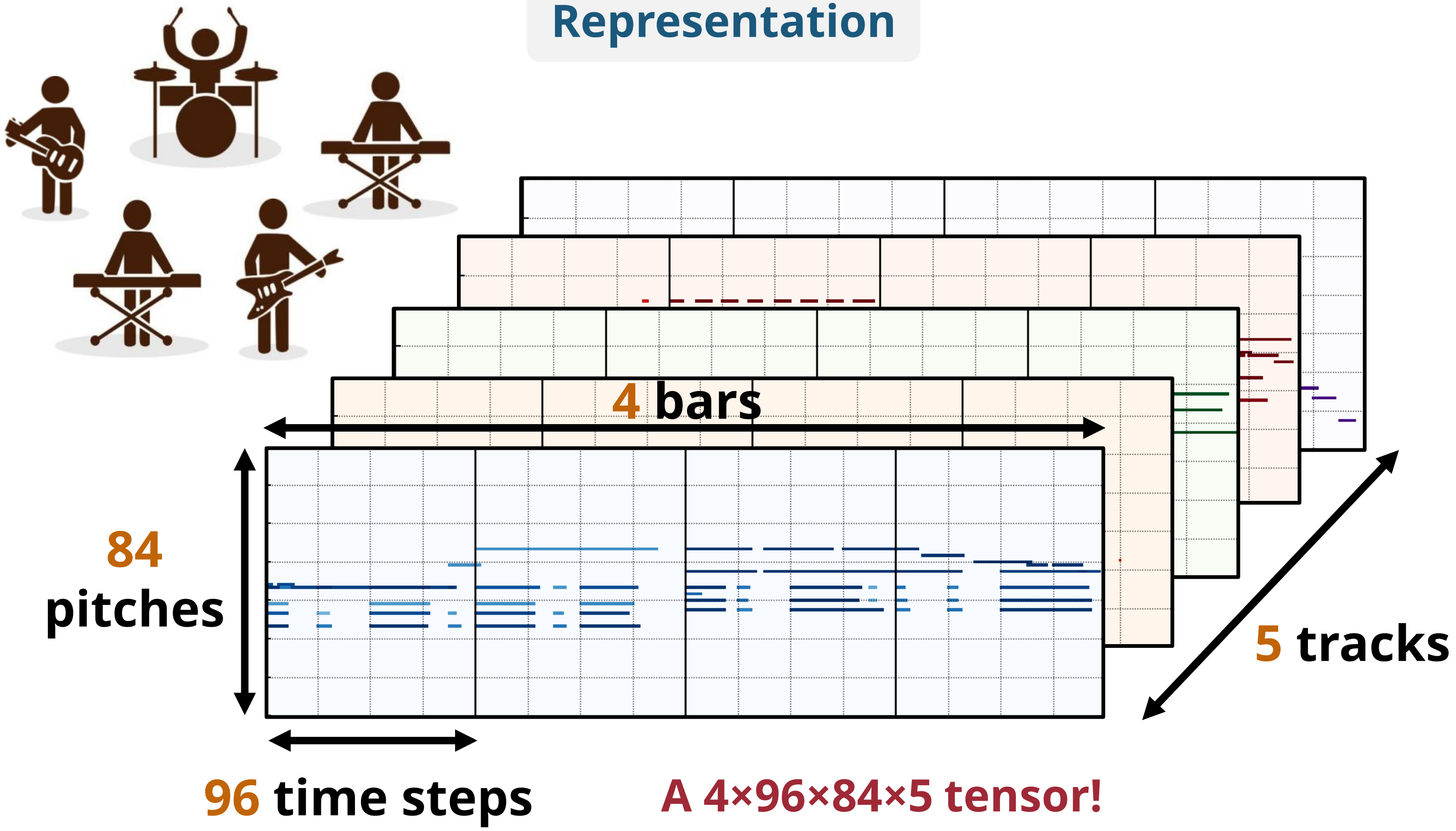


(Source: Yang et al., 2017)

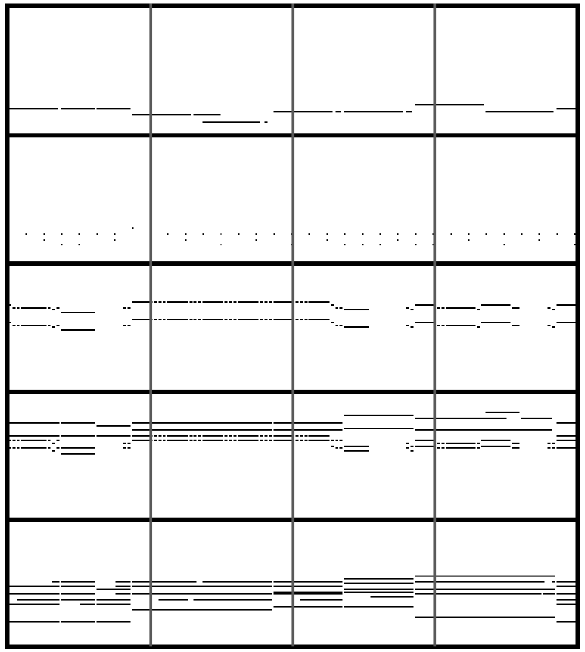
Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang, "MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation," *ISMIR*, 2017.

Example: MuseGAN (Dong et al., 2018)

Representation

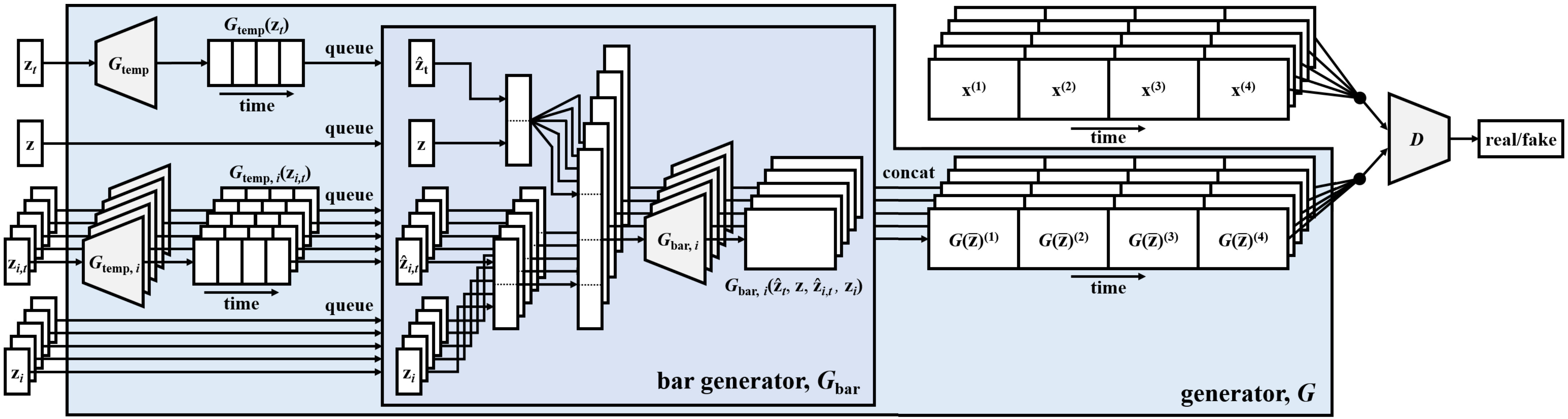


A training sample



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.

Example: MuseGAN (Dong et al., 2018)

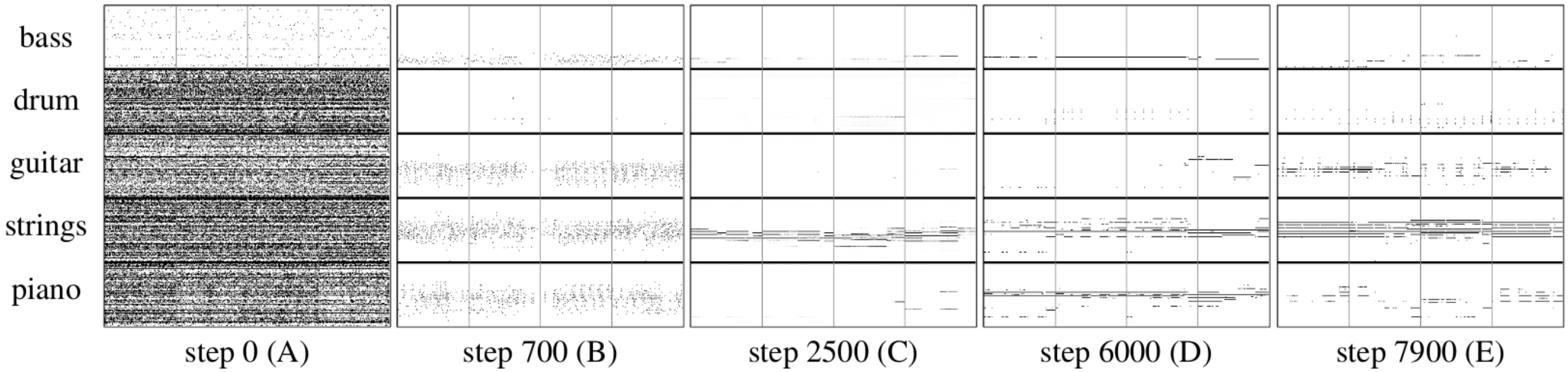


(Source: Dong et al., 2018)

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.

Example: MuseGAN (Dong et al., 2018)

Examples of generated music



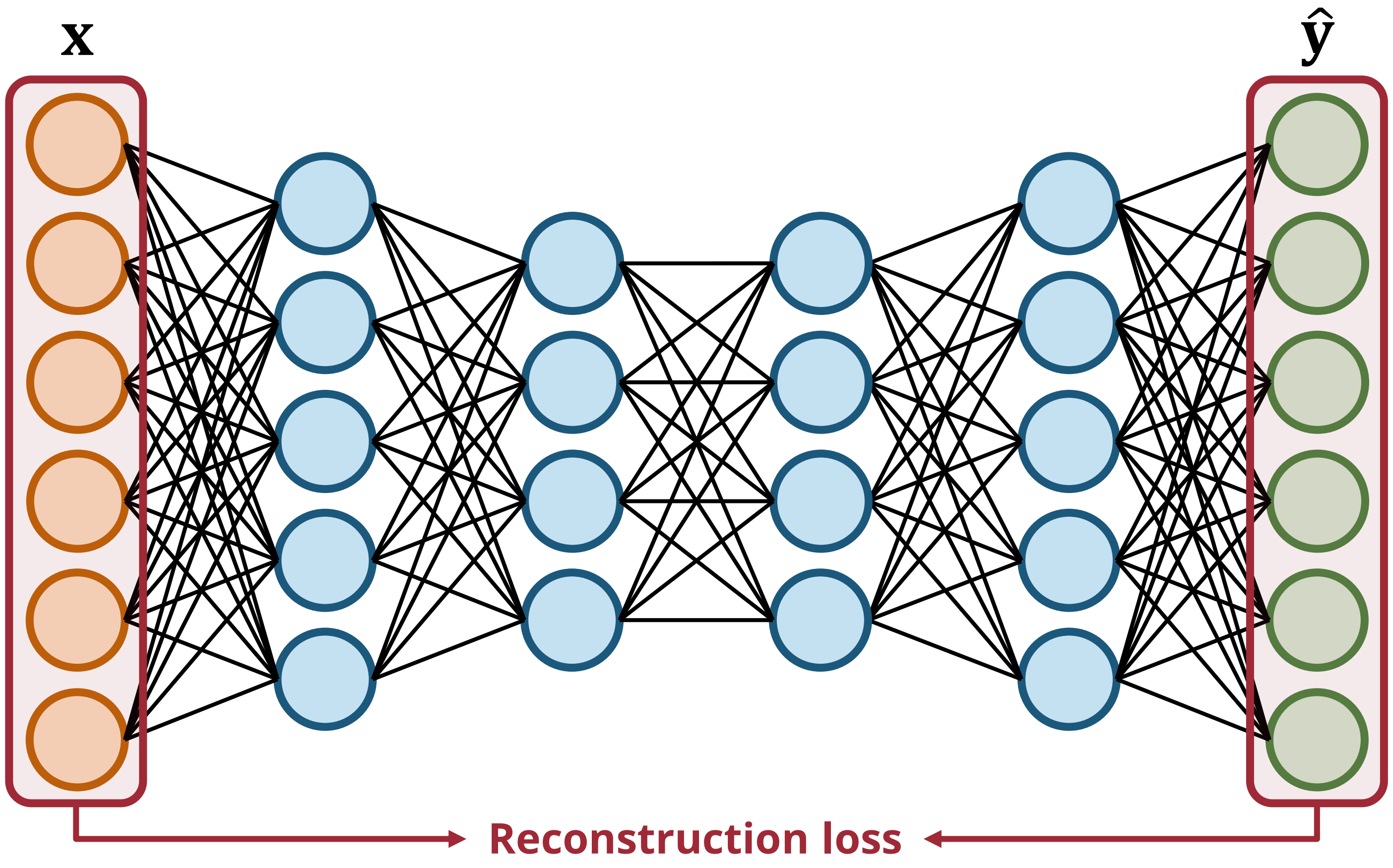
(Source: Dong et al., 2018)

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.

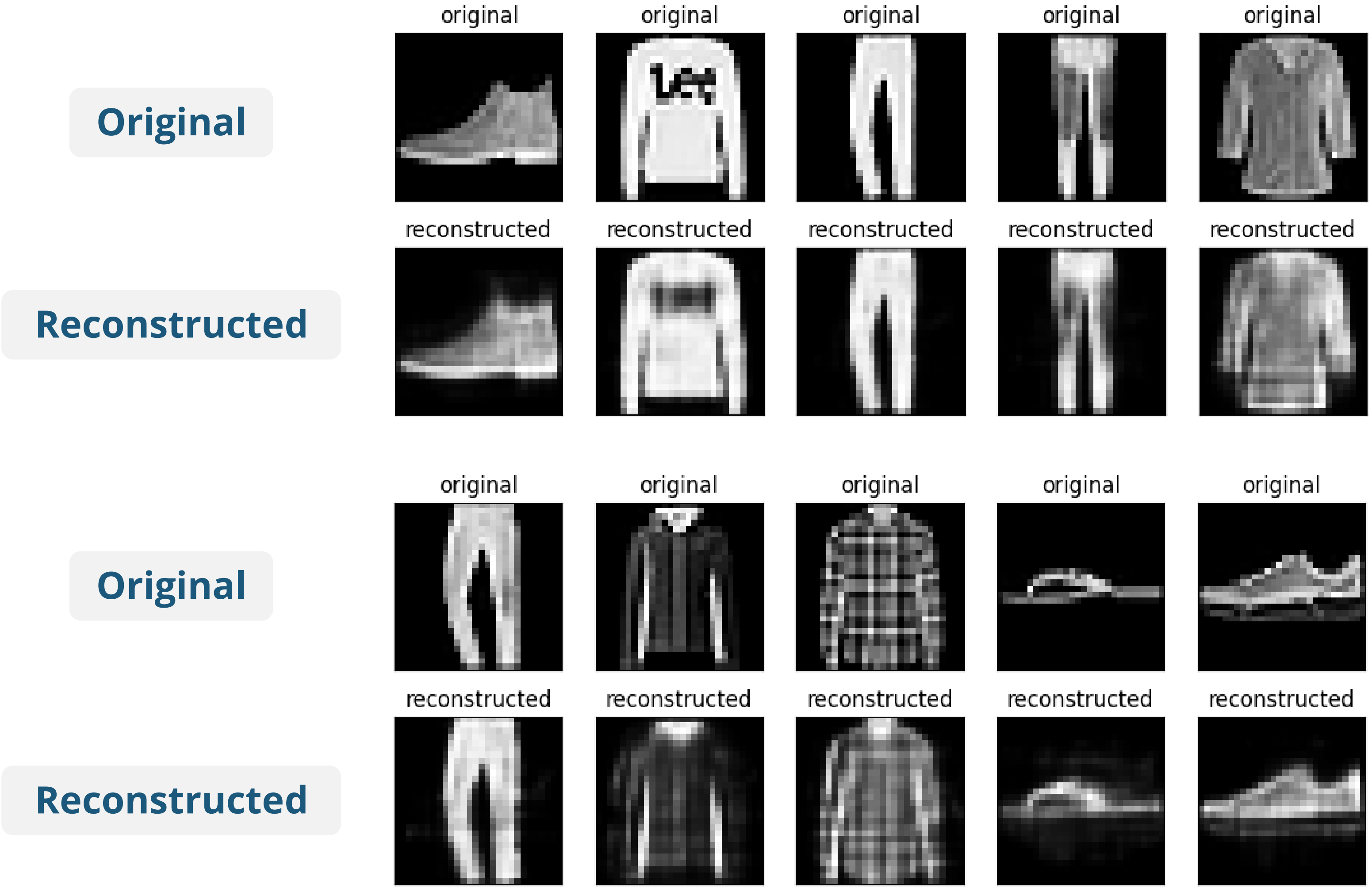
Diffusion Models

Autoencoders

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



Autoencoders – Reconstruction Examples



(Source: tensorflow.org)

tensorflow.org/tutorials/generative/autoencoder

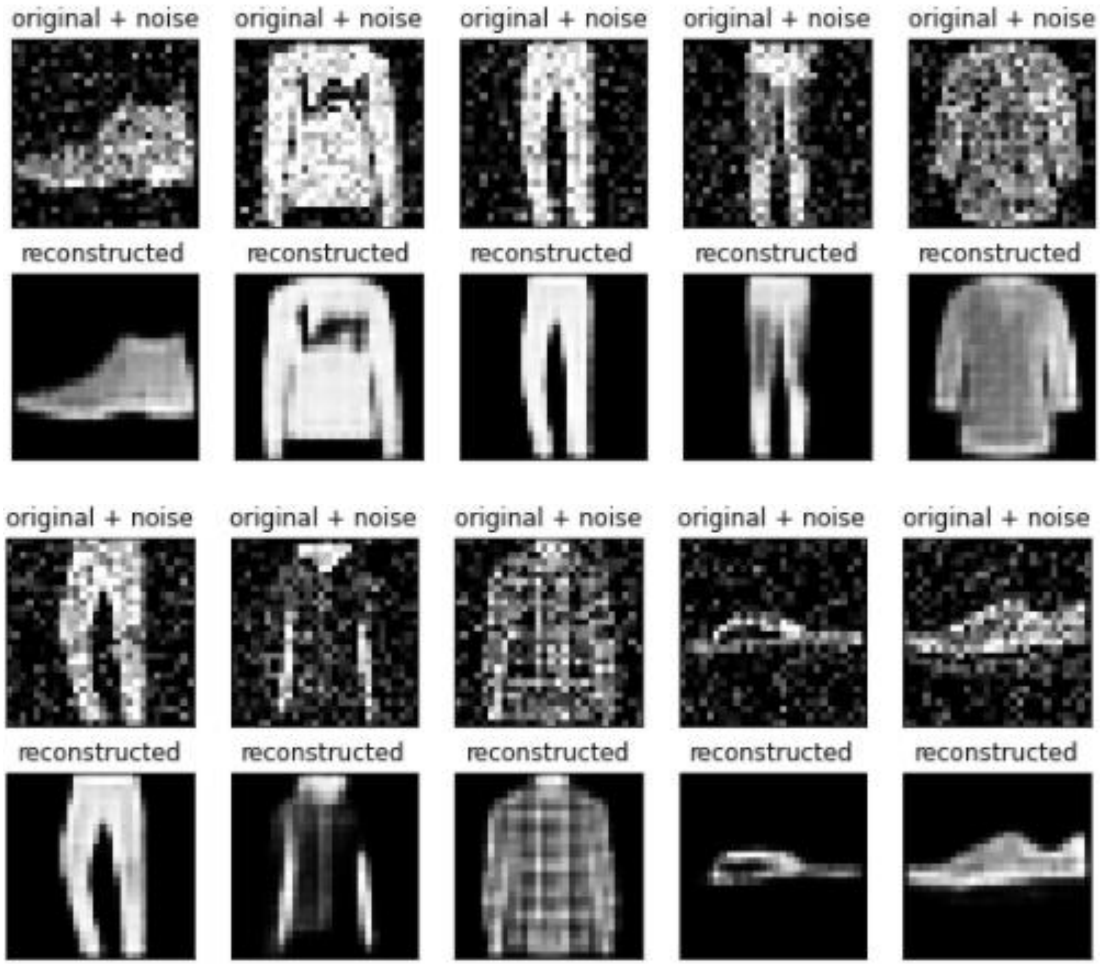
Denoising Autoencoders (Pascal et al., 2008)

Noisy

Reconstructed

Noisy

Reconstructed



(Source: tensorflow.org)

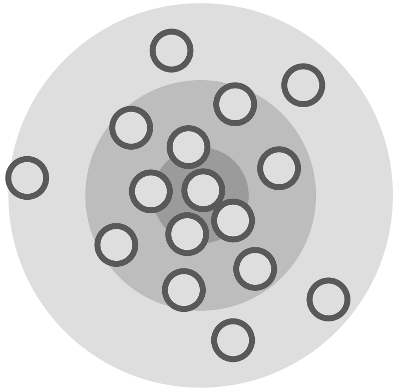
Denoising autoencoders learn to reconstruct noisy inputs

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.

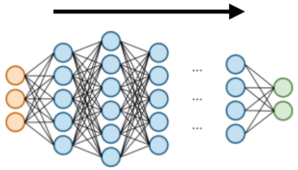
(Recap) Generating Data from a Random Distribution

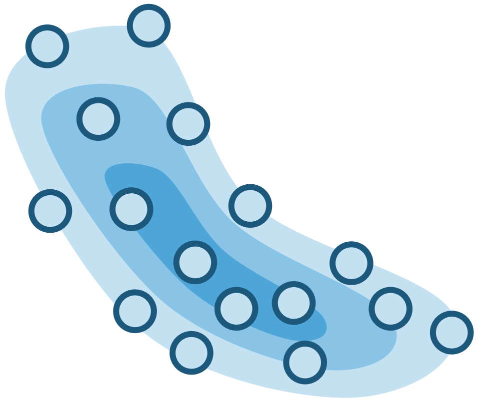
Random distribution

Data distribution



$P(z)$



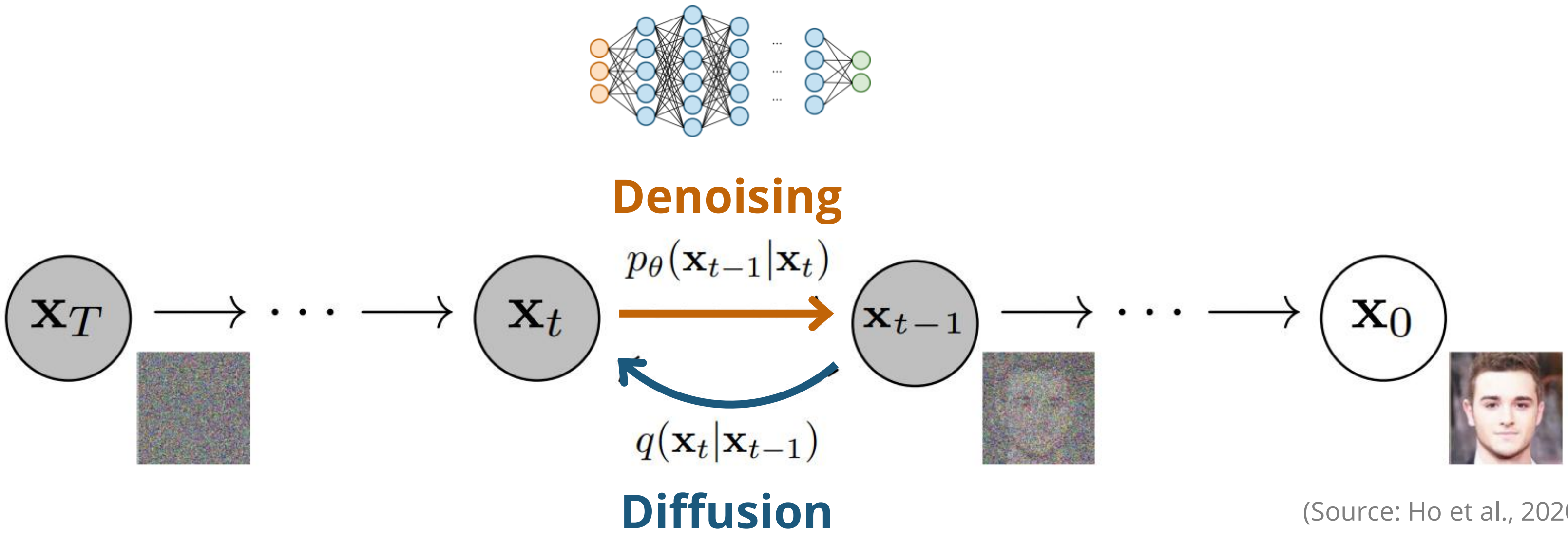


$P(x)$

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

Diffusion Models (Ho et al., 2020)

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

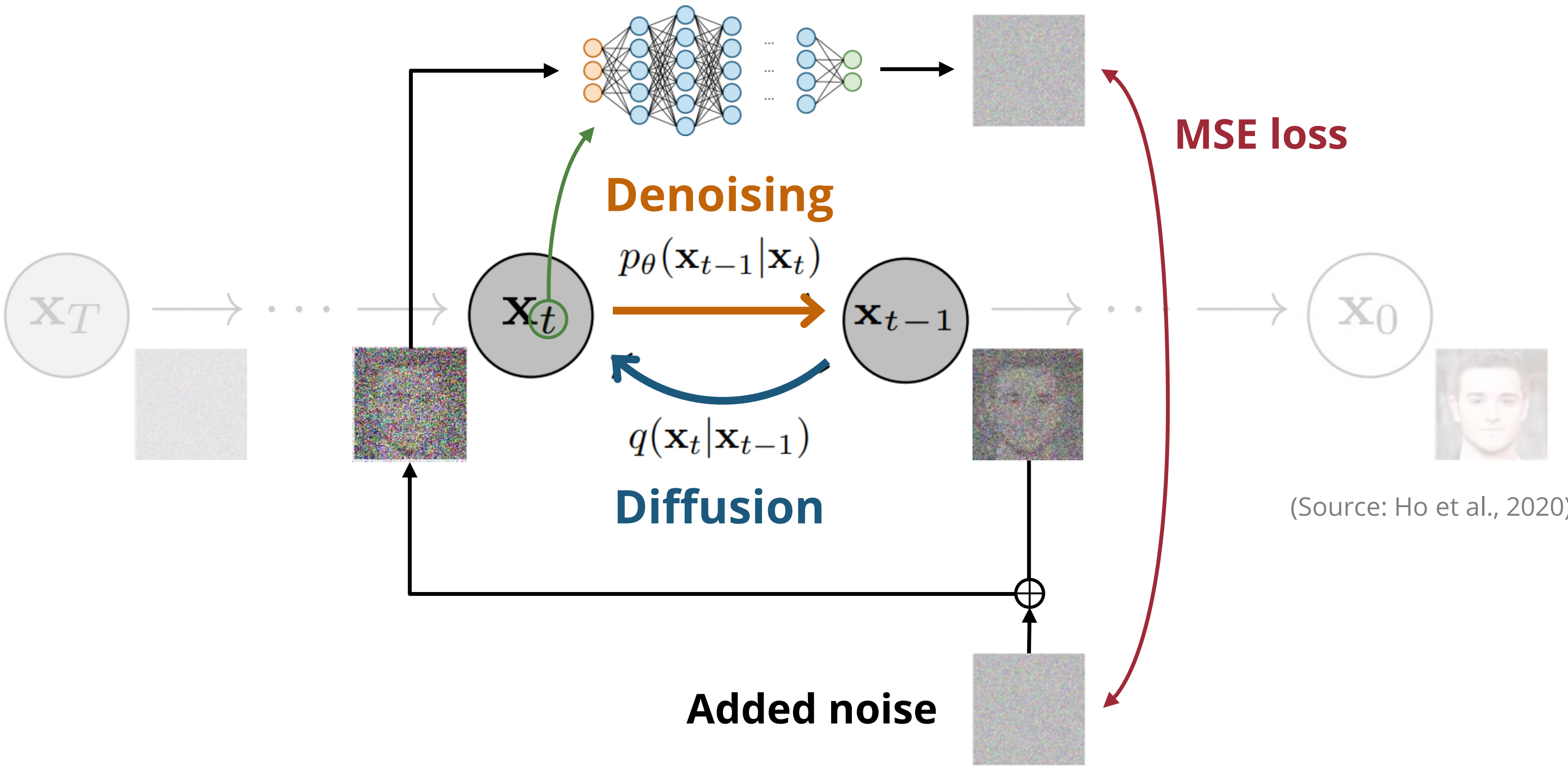


(Source: Ho et al., 2020)

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

Diffusion Models – Training

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

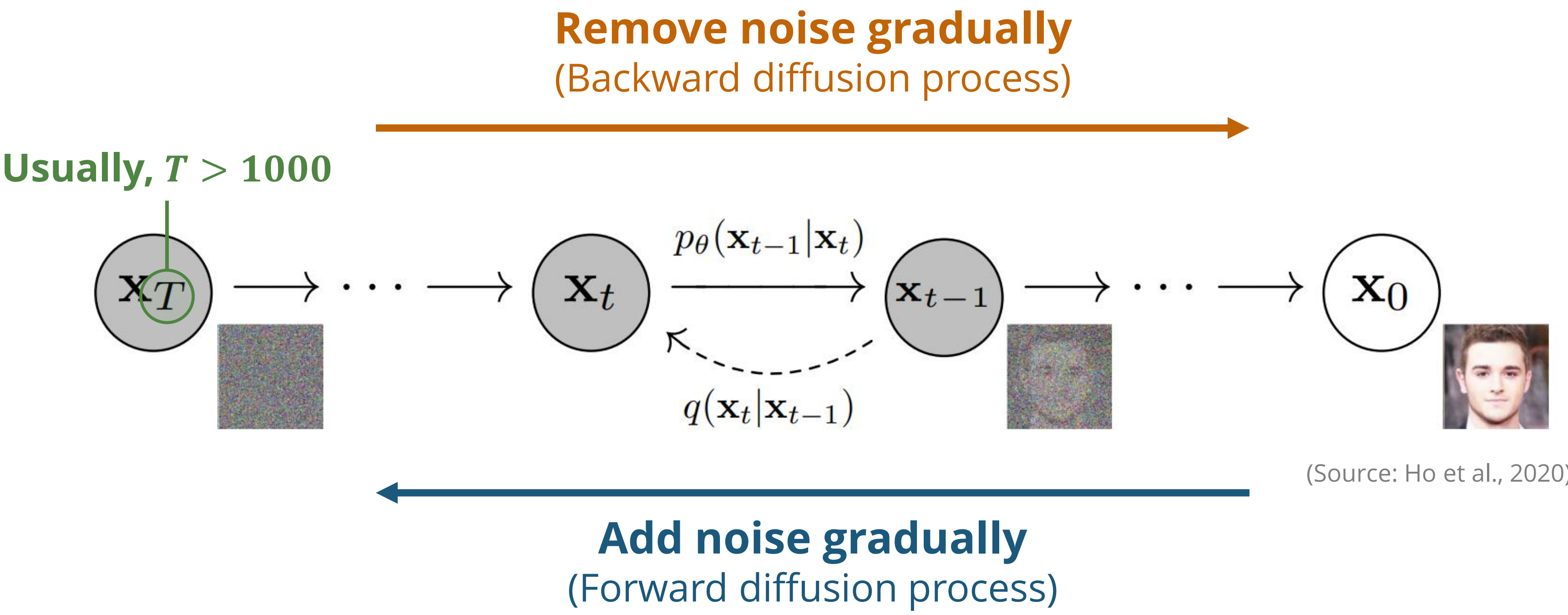


(Source: Ho et al., 2020)

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

Diffusion Models (Ho et al., 2020)

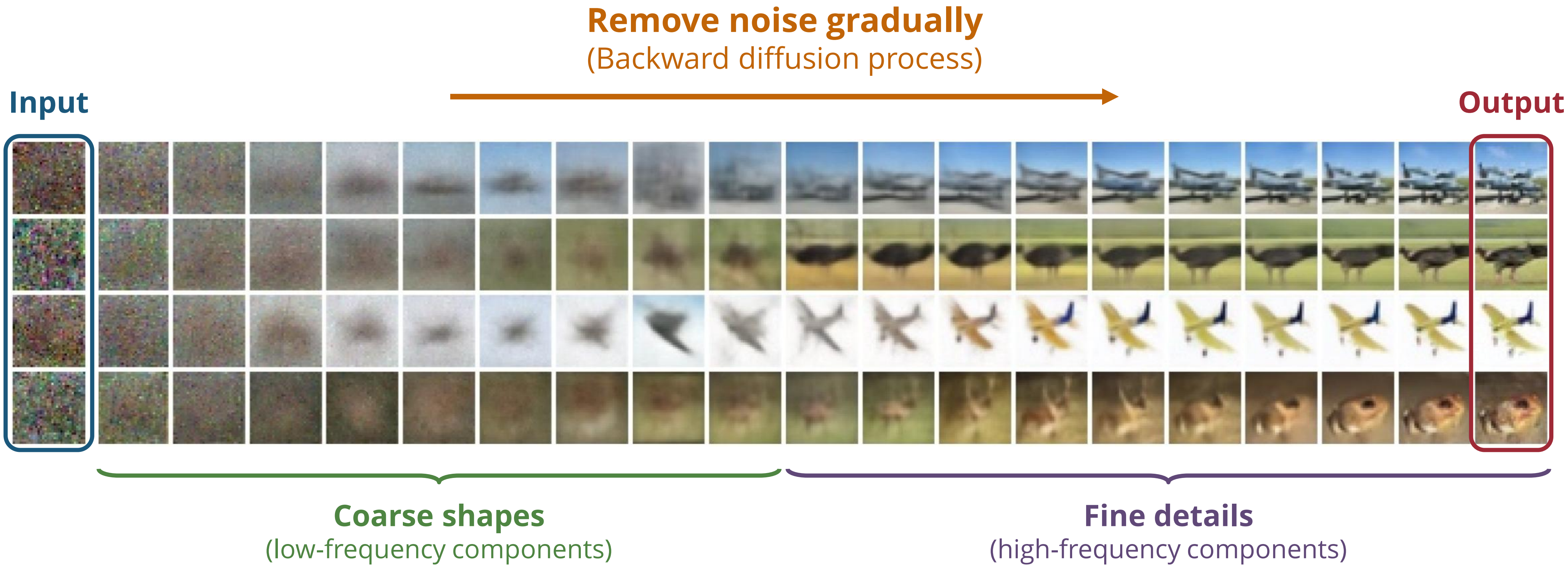
- **Intuition**: Many denoising autoencoders stacked together



(Source: Ho et al., 2020)

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

Diffusion Models – Generation

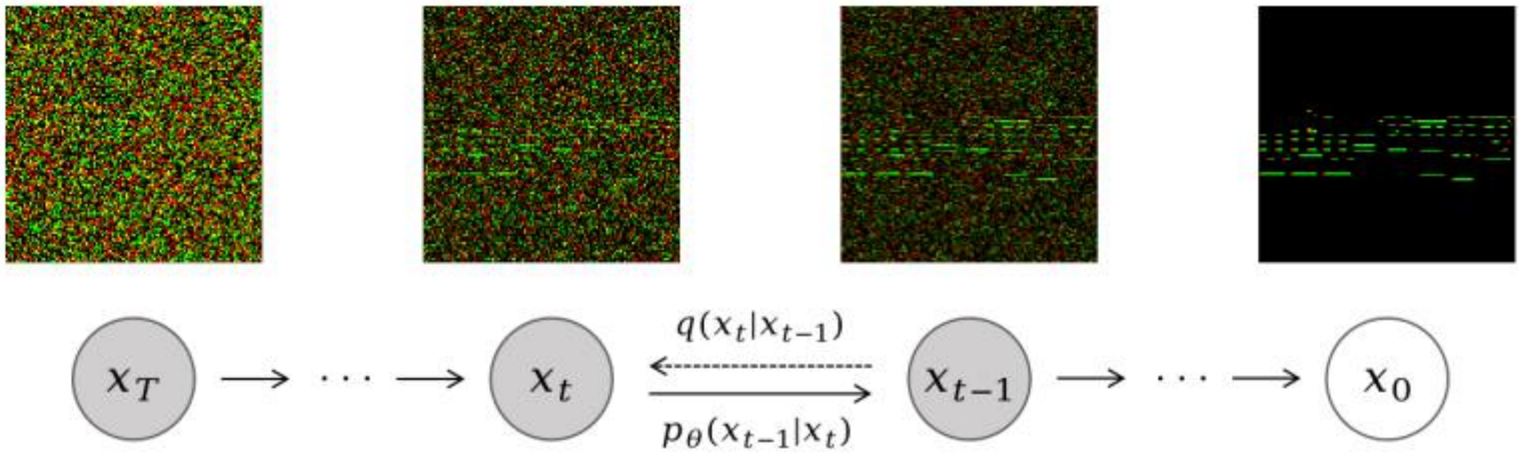


(Source: Ho et al., 2020)

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

Music Generation using Diffusion Models

Example: Polyffusion (Min et al., 2023)

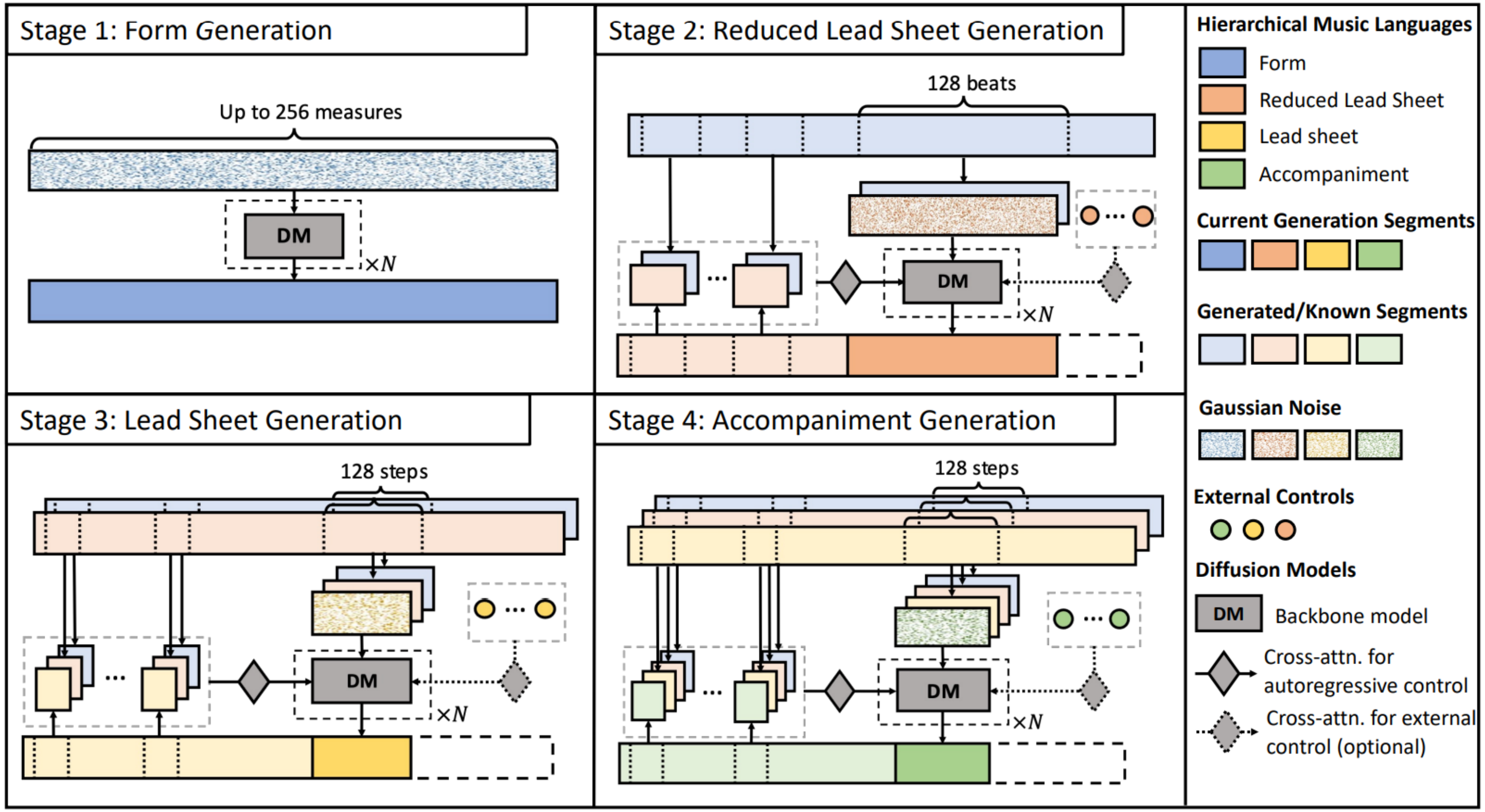


(Source: Min et al., 2023)

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.

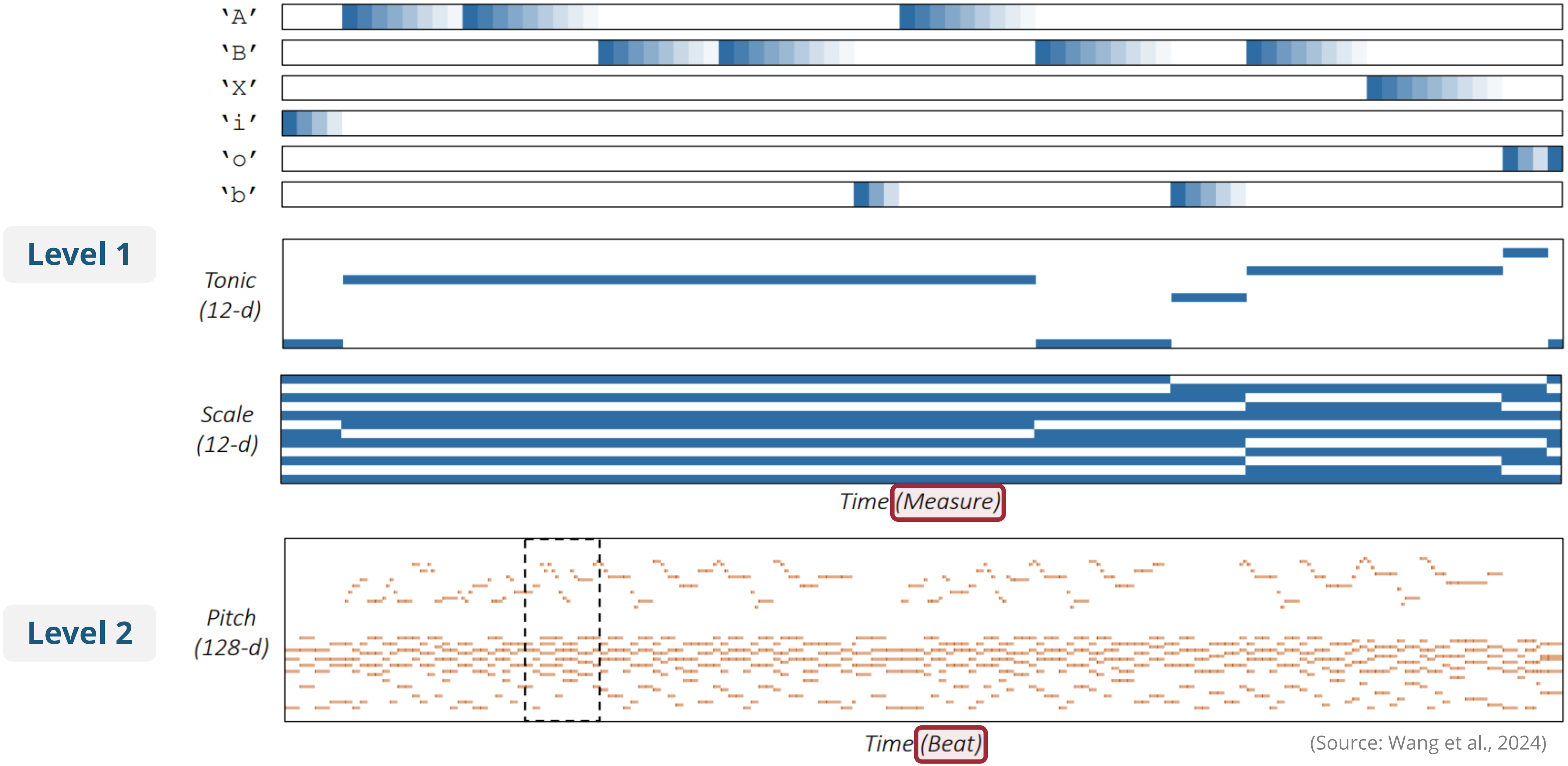
Example: Cascaded Diffusion Models (Wang et al., 2024)



(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.

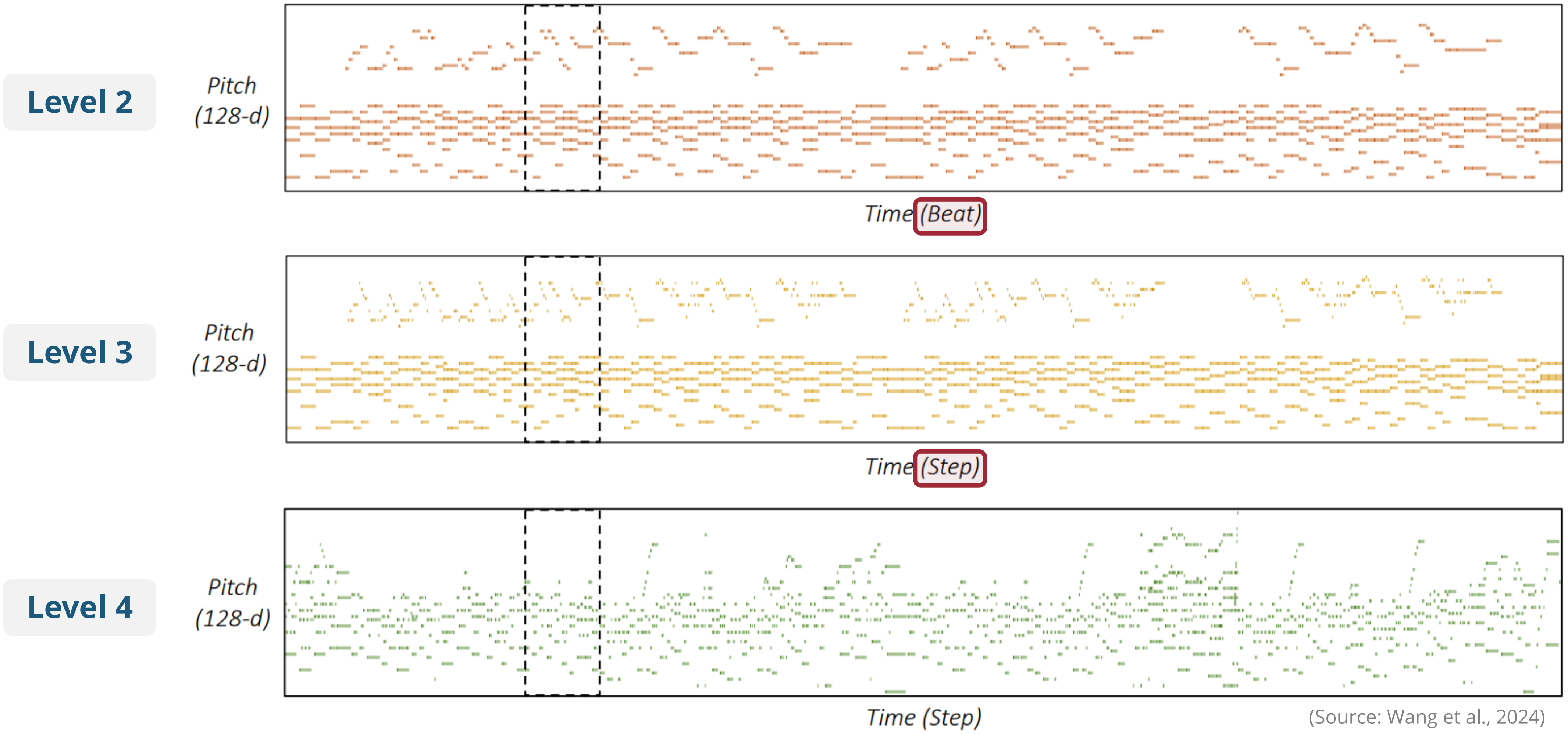
Example: Cascaded Diffusion Models (Wang et al., 2024)



(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.

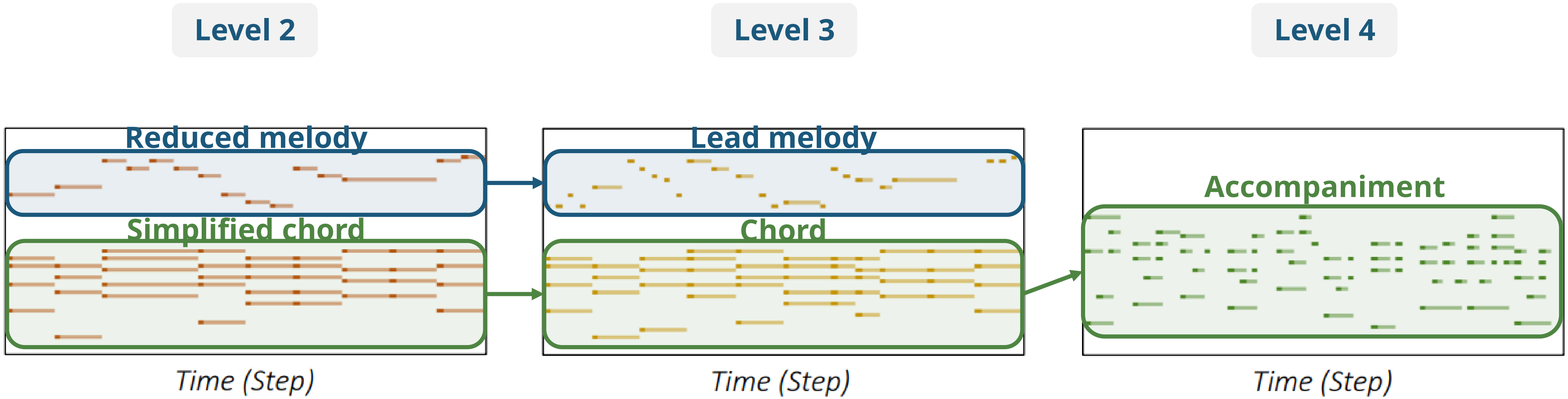
Example: Cascaded Diffusion Models (Wang et al., 2024)



(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.

Example: Cascaded Diffusion Models (Wang et al., 2024)



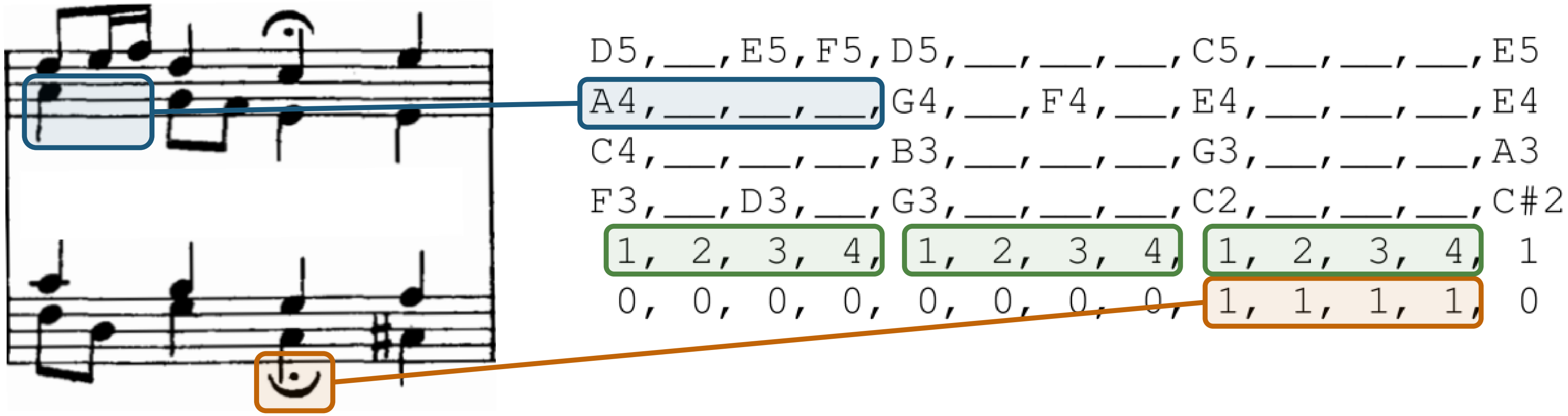
(Source: Wang et al., 2024)

wholesonggen.github.io

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

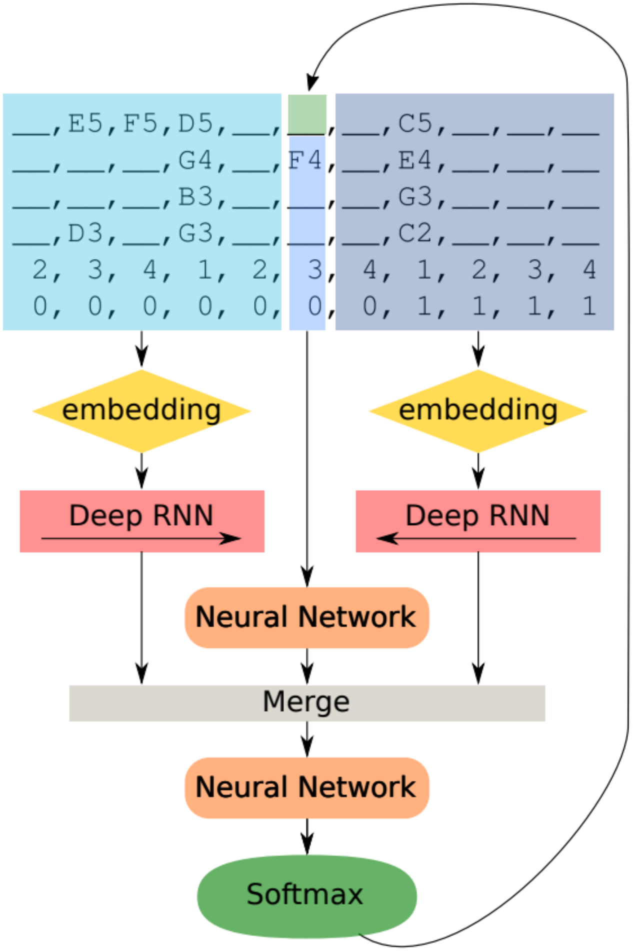
Music Infilling Models

Example: DeepBach (Hadjeres et al., 2017)



(Source: Hadjeres et al., 2017)

Example: 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.

Example: DeepBach (Hadjeres et al., 2017)

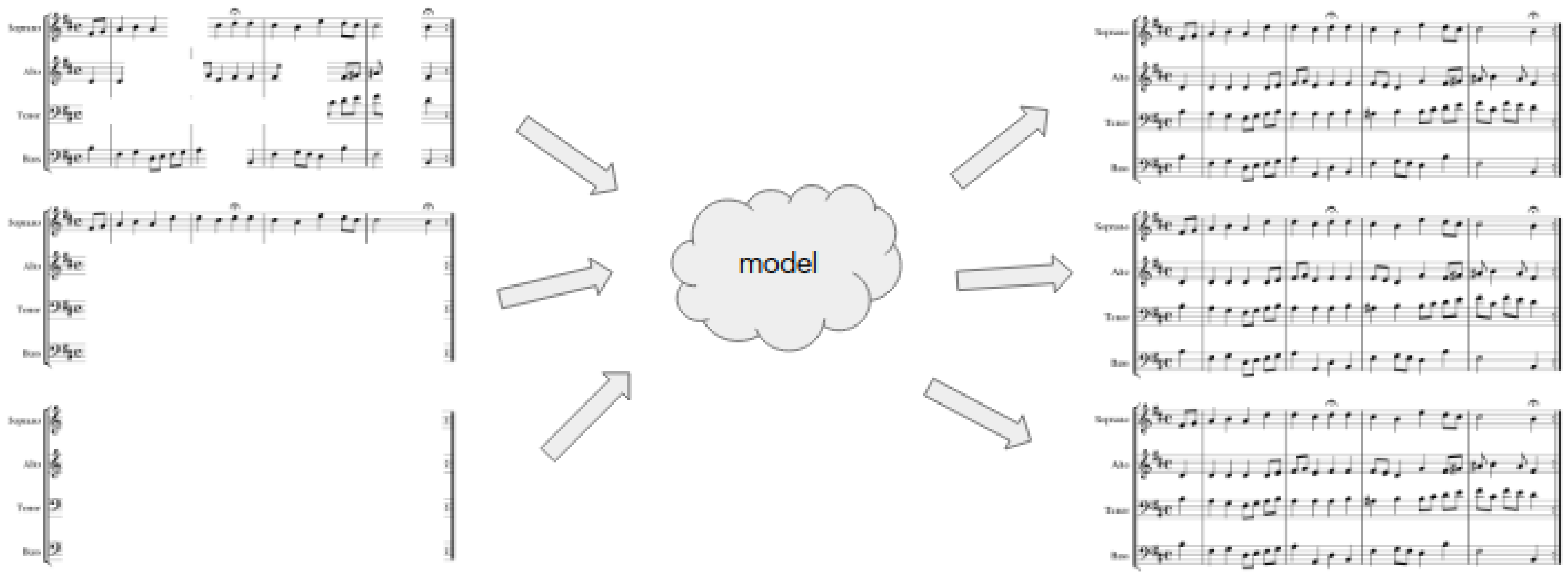
Reharmonization example

youtu.be/QiBM7-5hA6o

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

Example: Coconet (Huang et al., 2017)

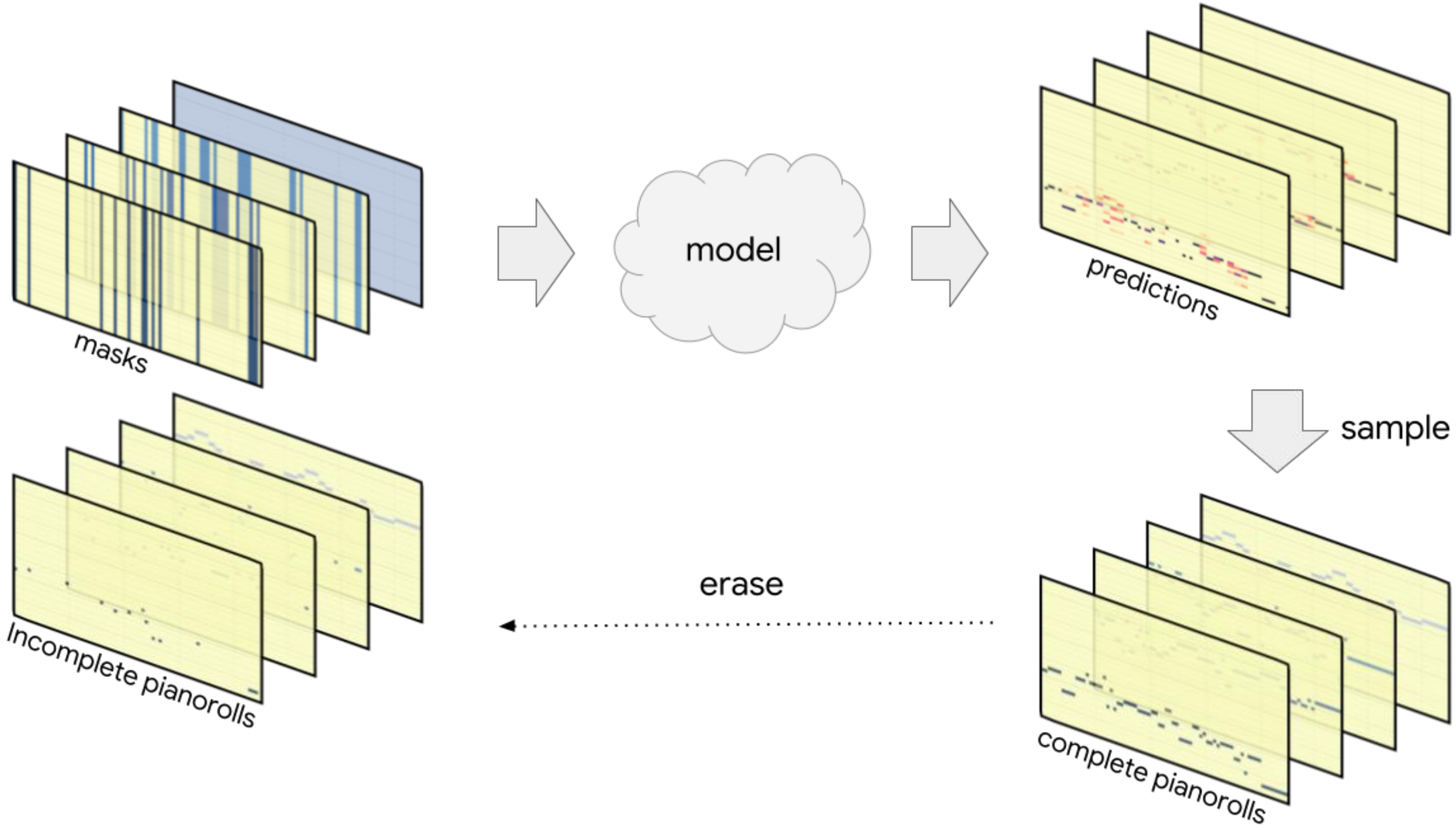
- Based on Orderless NADE (Uria et al, 2014)



(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.

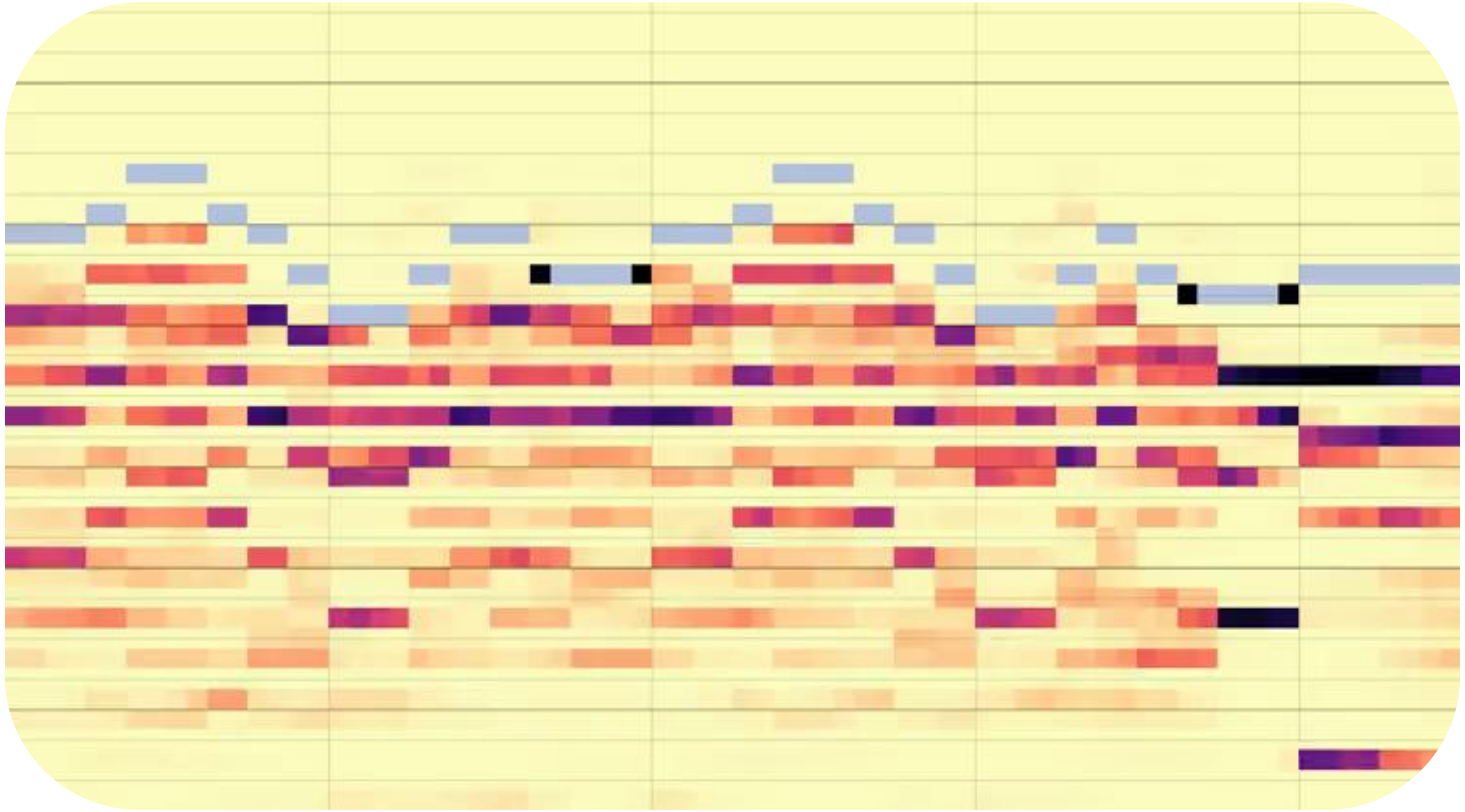
Example: Coconet (Huang et al., 2017)



(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.

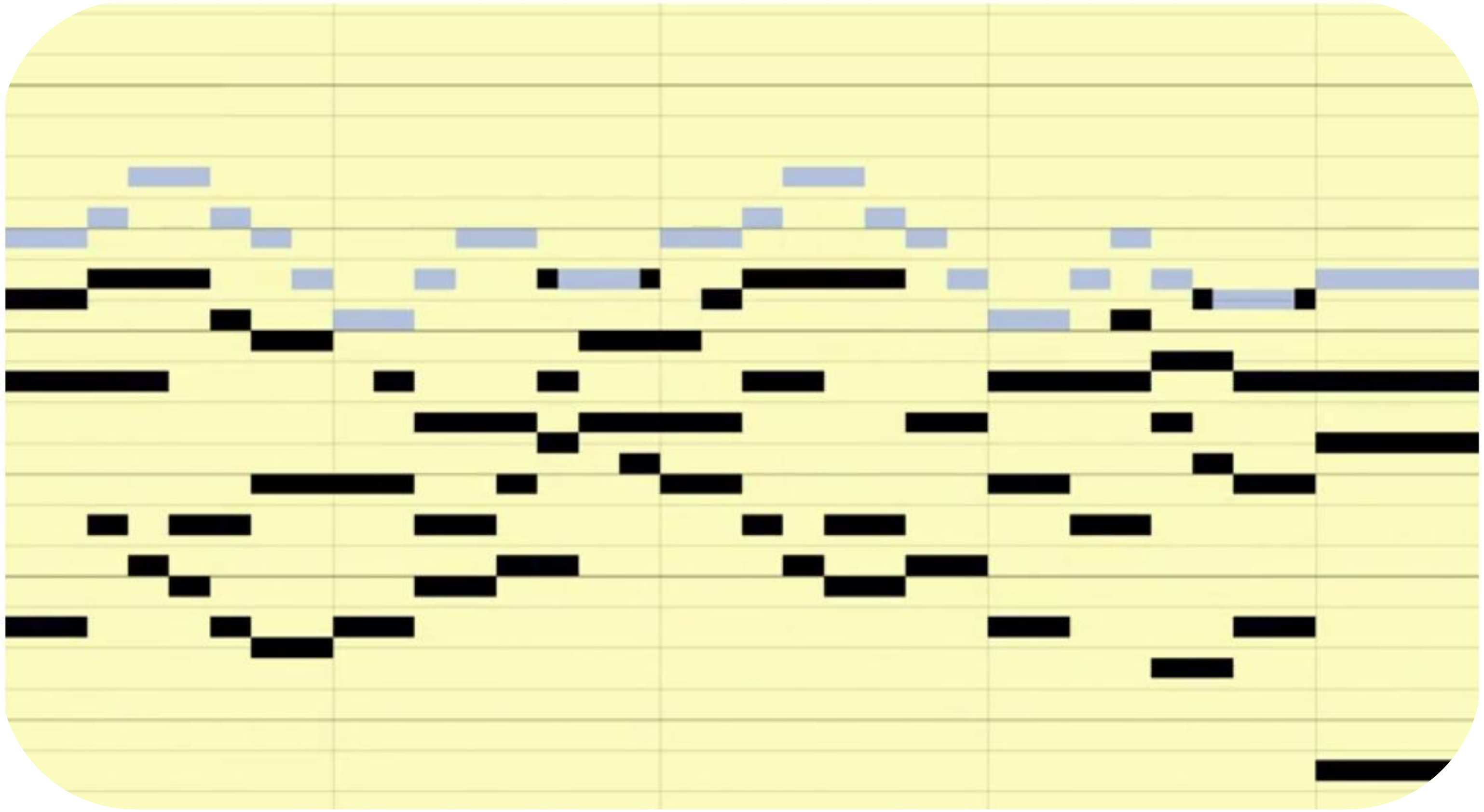
Example: Coconet (Huang et al., 2017)



(Source: Huang et al., 2017)

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

Example: 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.

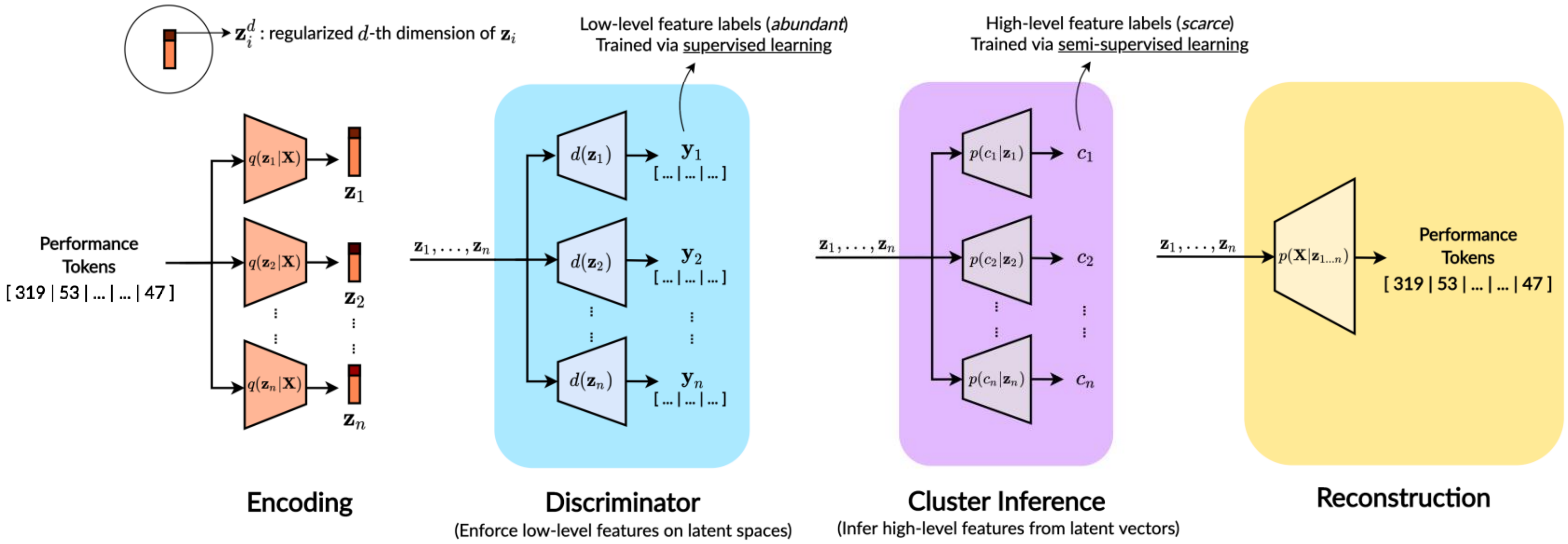
Example: JS Bach Doodle (2019)

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

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

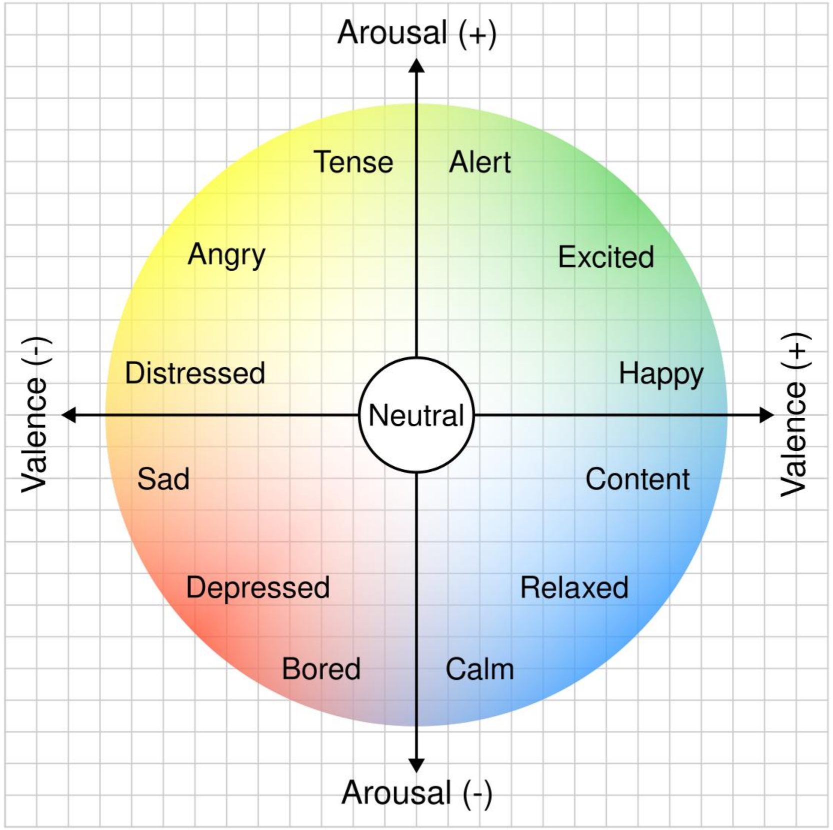
Controllable Music Generation

Example: Music FaderNet (Tan & Herremans, 2020)



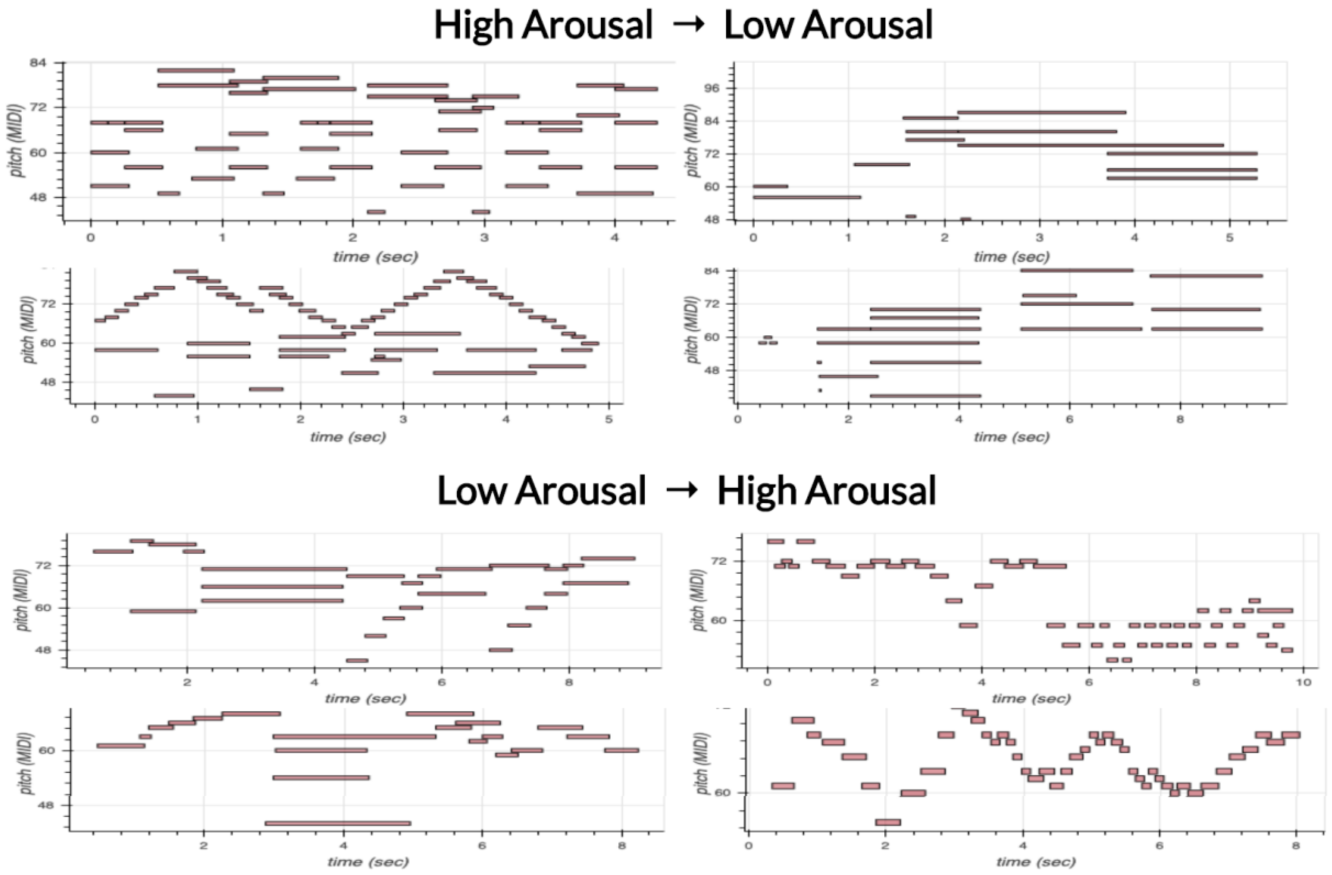
(Source: Tan & Herremeans, 2020)

Valence-Arousal Model for Emotion



(Source: mrAnmol)

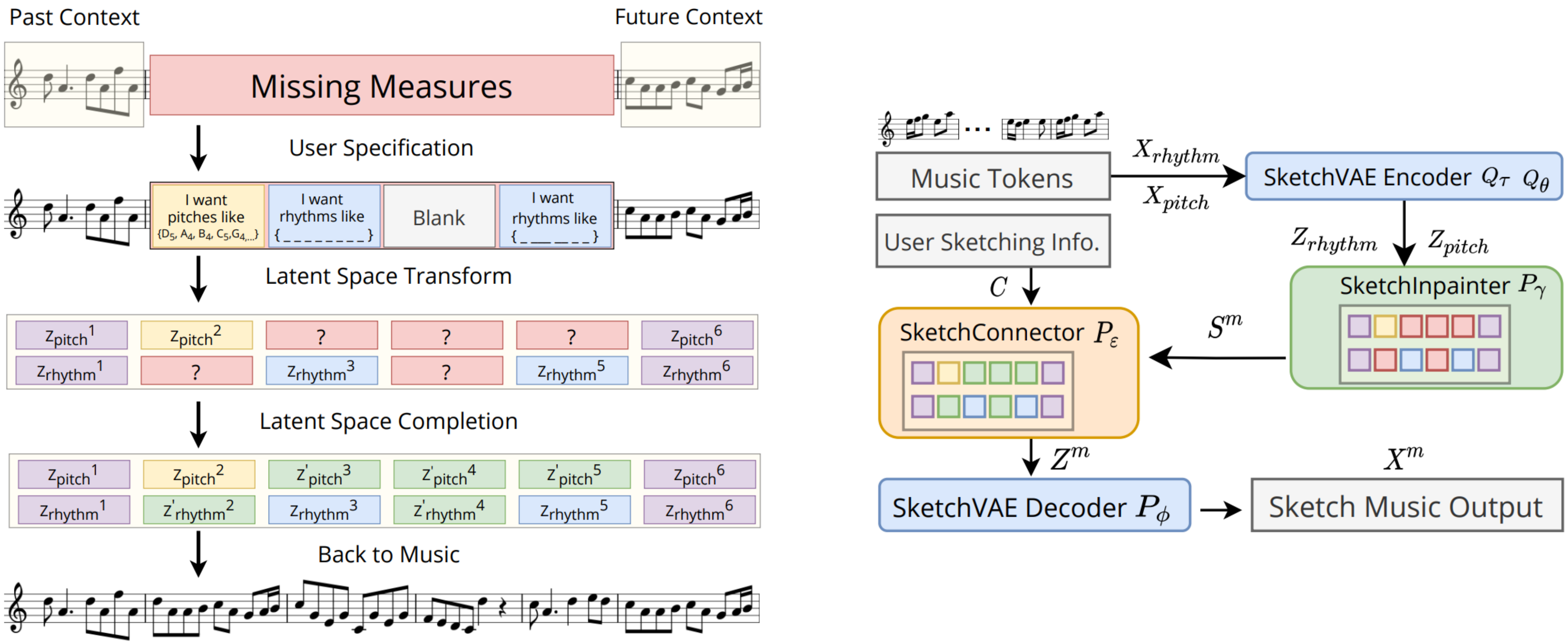
Example: Music FaderNet (Tan & Herremans, 2020)



(Source: Tan & Herremeans, 2020)

music-fadernets.github.io

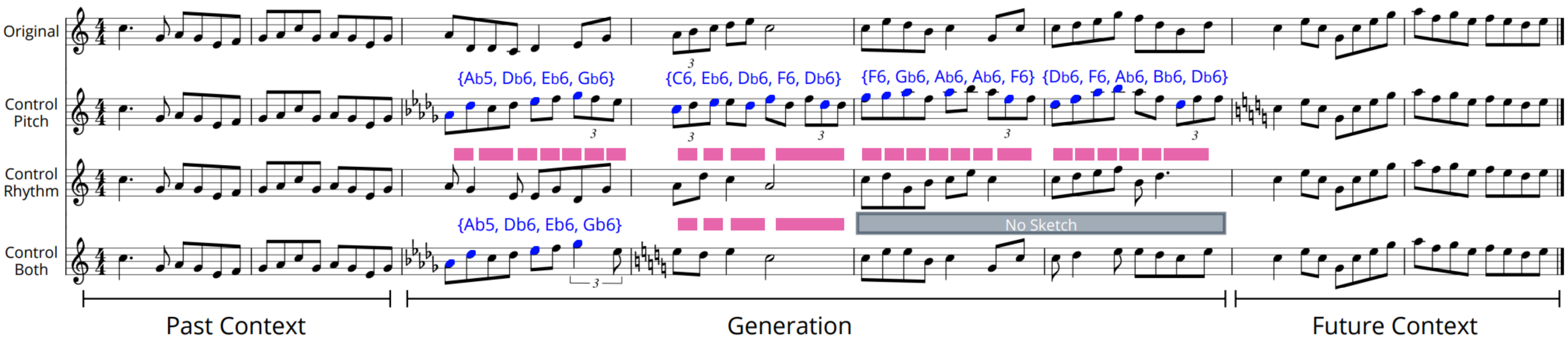
Example: Music SketchNet (Chen et al., 2020)



(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.

Example: 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.