20th Conference of the International Society for Music Information Retrieval

Generating Music with GANs

An Overview and Case Studies

November 4th, 2019

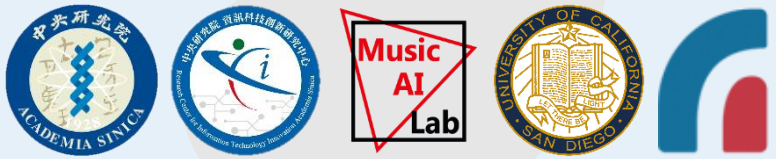
salu133445.github.io/ismir2019tutorial/

Hao-Wen Dong

UC San Diego & Academia Sinica

Yi-Hsuan Yang

Taiwan AI Labs & Academia Sinica



Outline

Section 1

Overview of music generation research

Section 2

Introduction to GANs

Coding session 1: GAN for images

◁ break

Section 3

Case studies of GAN-based systems (I)

Coding session 2: GAN for piano rolls

◁ break

Case studies of GAN-based systems (II)

Section 4

Current limitations

Future research directions

salu133445.github.io/ismir2019tutorial/

About us

Hao-Wen Dong

- *Ph.D. student*, UC San Diego (2019-)
- *Research intern*, Yamaha Corporation (2019)
- *Research assistant*, Academia Sinica (2017-2019)
- First author of MuseGAN & BMuseGAN

Yi-Hsuan Yang

- *Chief Music Scientist*, Taiwan AI Labs (2019-)
- *Research professor*, Academia Sinica (2011-)
- *Ph.D.*, National Taiwan University (2006-2010)
- Associate Editor of IEEE TMM and TAFFC (2017-2019)
- Program Chair of ISMIR @ Taipei, Taiwan (2014)
- Tutorial speaker of ISMIR (2012 last time)

About the Music and AI Lab @ Sinica

- **About Academia Sinica**
  - National academy of Taiwan, founded in 1928 (not a university)
  - About 1,000 full, associate, assistant research professors
- **About Music and AI Lab**
  - https://musicai.citi.sinica.edu.tw/
  - Since Sep 2011
  - Members
    - PI [me]
    - research assistants
    - PhD/master students
  - 3 AAAI full papers + 3 IJCAI full papers in 2018 and 2019

中央研究院
ACADEMIA SINICA

中央研究院
資訊科技創新研究中心
Research Center for Information Technology Innovation
Academia Sinica

About the Music Team @ Taiwan AI Labs

- **About Taiwan AI Labs**
  - https://ailabs.tw/
  - Privately-funded research organization (like openAI), founded in 2017
  - Three main research area: 1) HCI, 2) medicine, 3) smart city
  - 100+ employees (late 2019)

- **About the "Yating" Music AI team**
  - Members
    - scientist [me]
    - ML engineers (for models)
    - musicians
    - program manager
    - software engineers (for frontend/backend)

AI Labs.tw
台灣人工智慧實驗室

Outline

Section 1

Overview of music generation research

Section 2

Introduction to GANs

Section 3

Case studies of GAN-based systems

Section 4

Current limitations

Future research directions

From a music production view

- **Composing / songwriting**
  - Melody
  - Chords
  - Lyrics
- **Arranging**
  - Instrumentation
  - Structure
- **Mixing**
  - Timbres/tones
  - Balancing

Use cases of music AI

- **Make musicians' life easier**
  - Inspire ideas
  - Suggest continuations, accompaniments, lyrics, or drum loops
  - Suggest mixing presets
- **Empower everyone to make music**
  - Democratization of music creation
- **Create copyright free music for videos or games**
- **Music education** (e.g., auto-accompaniment)

Companies involved in automatic music generation

Google AI
magenta

IBM Watson Beat AI

amper

Jukedeck

AIVA

Spotify®

FLOW MACHINES

Sony CSL

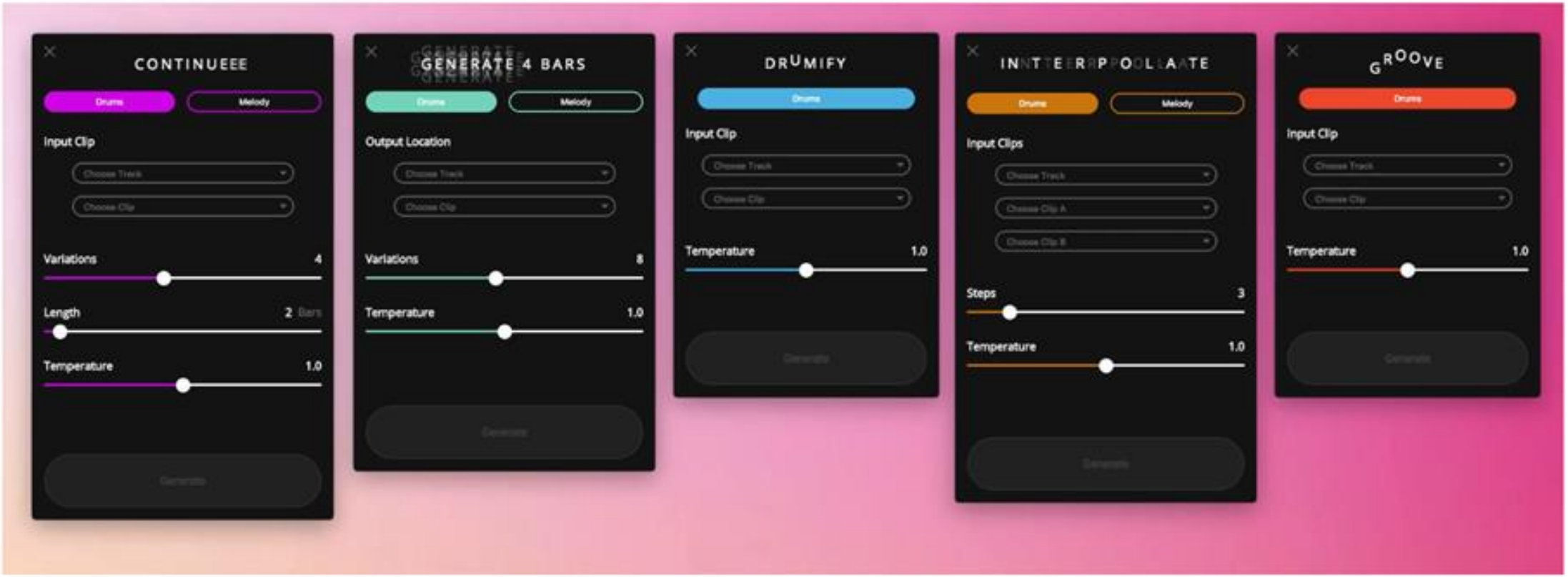
AI Labs.tw
台灣人工智慧實驗室

... and more!

Demo: Magenta Studio

Google AI magenta

- https://magenta.tensorflow.org/studio/

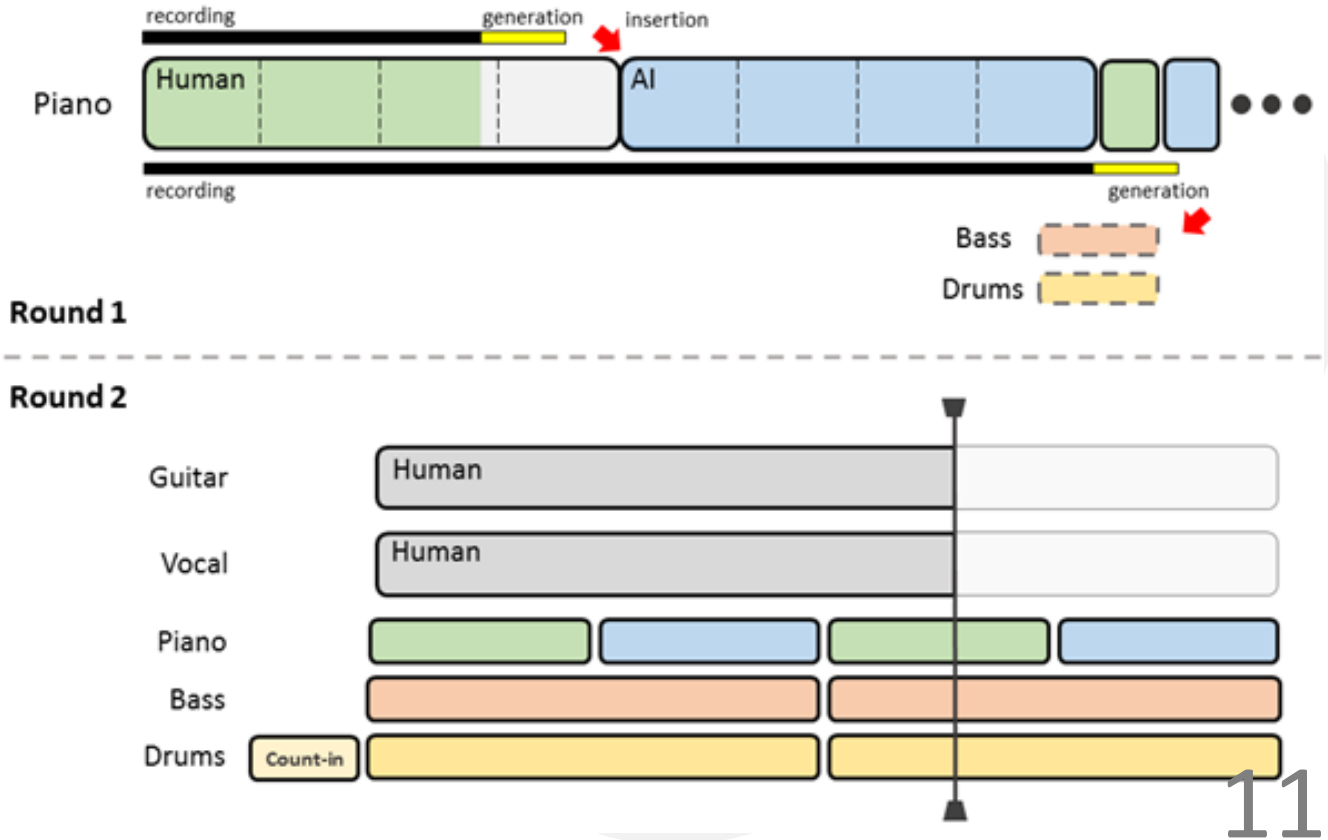


Demo: Jamming with Yating

AI Labs.tw
台灣人工智慧實驗室

- https://www.youtube.com/watch?v=9ZIJrr6lmHg

- Jamming with Yating [1,2]
  - **Input** (by human): piano
  - **Output**: piano + bass + drum



[1] Hsiao et al., "Jamming with Yating: Interactive demonstration of a music composition AI," *ISMIR-LBD* 2019

[2] Yeh et al., "Learning to generate Jazz and Pop piano music from audio via MIR techniques," *ISMIR-LBD* 2019

From a deep learning view

- Input representation
- Model
- Output representation

Models

- Rule based methods
- Concatenation based methods
- Machine learning based methods
  - **VAE**: variational autoencoder
  - **GAN**: generative adversarial network

- **See Section 2: Introduction to GANs**

Why GAN?

- **State-of-the-art model** in:
  - Image generation: **BigGAN** [1]
  - Text-to-speech audio synthesis: **GAN-TTS** [2]
  - Note-level instrument audio synthesis: **GANSynth** [3]
  - Also see ICASSP 2018 tutorial: "**GAN and its applications to signal processing and NLP**" [4]
- Its potential for music generation has not been fully realized
- **Adversarial training has many other applications**
  - For example, source separation [5], domain adapation [6], music transcription [7]

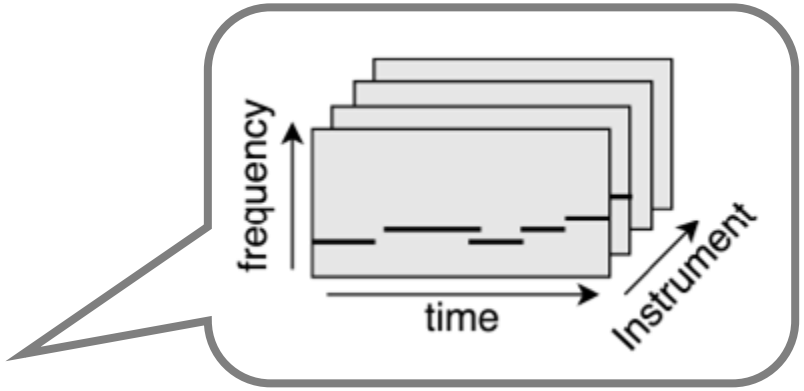
[1] "Large scale GAN training for high fidelity natural image synthesis," *ICLR* 2019
[2] "High fidelity speech synthesis with adversarial networks," *ICLR* 2020 submission
[3] "GANSynth: Adversarial neural audio synthesis," *ICLR* 2019
[4] https://tinyurl.com/y23ywv4s (on slideshare)
[5] "SVSGAN: Singing voice separation via generative adversarial network," *ICASSP* 2018
[6] "Cross-cultural music emotion recognition by adversarial discriminative domain adaptation," *ICMLA* 2018
[7] "Adversarial learning for improved onsets and frames music transcription," *ISMIR* 2019

Input/output representations

- **Symbolic output**
  - Piano rolls
  - MIDI events
  - Score

- **Audio output**
  - Sepctrogram
  - Waveform

I/O representations

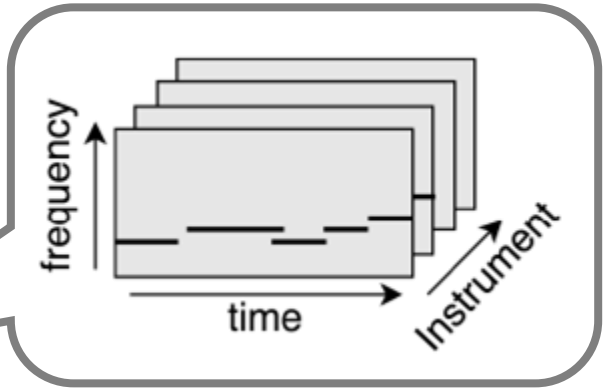


- **Symbolic output**
  - Piano rolls (image-like): easier for GANs to work with
  - MIDI events (text-like)
  - Score (hybrid)

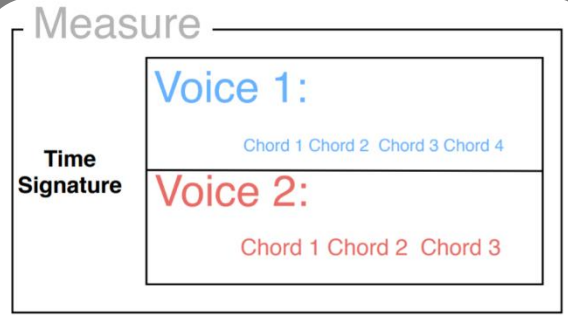
- **Audio output**
  - Sepctrogram (image-like)
  - Waveform

I/O representations

- **Symbolic output**
  - **Piano rolls (image-like):**
    MidiNet [1], MuseGAN [2]
  - **MIDI events (text-like):**
    Music Transformer [3], MuseNet [4]
  - **Scores (hybrid):**
    Thickstun [5], measure-by-measure [6]



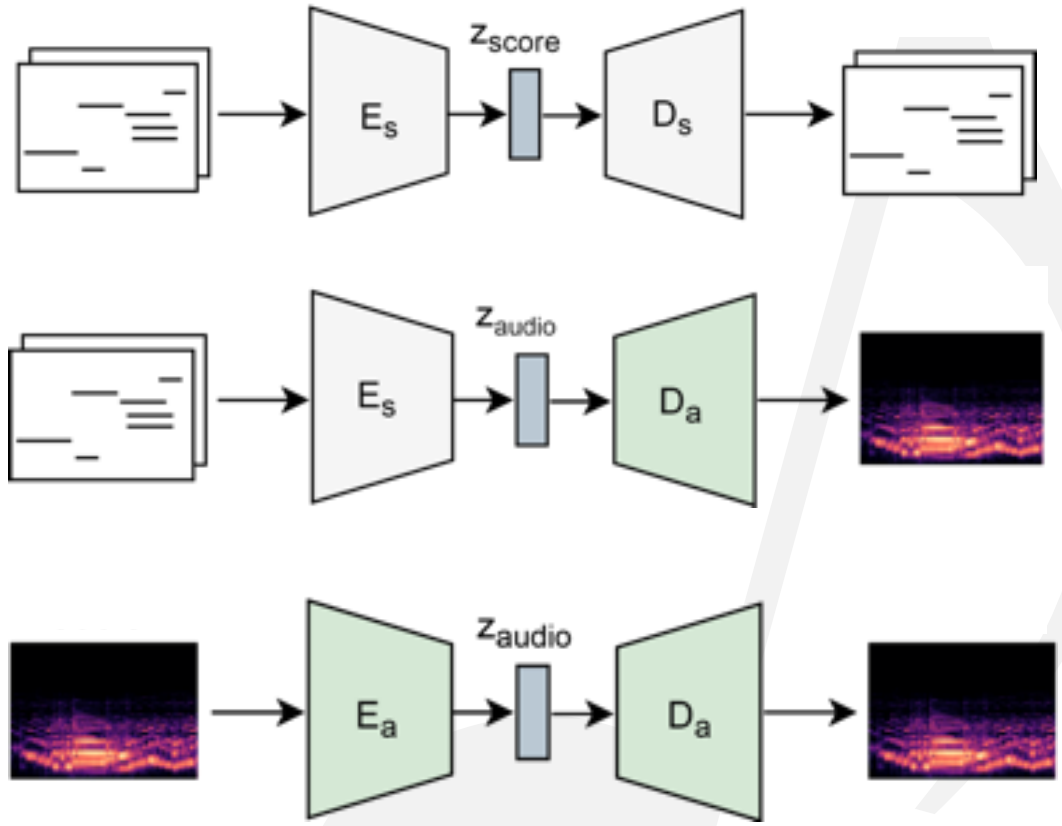

```
bach piano_strings start tempo90 piano:v72:G1
piano:v72:G2 piano:v72:B4 piano:v72:D4 violin:v80:G4
piano:v72:G4 piano:v72:B5 piano:v72:D5 wait:12
piano:v0:B5 wait:5 piano:v72:D5 wait:12 piano:v0:D5
wait:4 piano:v0:G1 piano:v0:G2 piano:v0:B4 piano:v0:D4
violin:v0:G4 piano:v0:G4 wait:1 piano:v72:G5 wait:12
piano:v0:G5 wait:5 piano:v72:D5 wait:12 piano:v0:D5
wait:5 piano:v72:B5 wait:12
```



[1] https://arxiv.org/abs/1703.10847, *ISMIR* 2017
[2] https://arxiv.org/abs/1709.06298, *AAAI* 2018
[3] https://openreview.net/pdf?id=rJe4ShAcF7, *ICLR* 2019
[4] https://openai.com/blog/musenet/
[5] https://arxiv.org/abs/1811.08045, *ISMIR* 2019
[6] https://openreview.net/forum?id=Hklk6xrYPB, *ICLR* 2020 submission

Scope of music generation

- **Generation from scratch**
  - X → melody
  - X → piano roll
  - X → audio
- **Conditional generation**
  - melody → piano roll (*accompaniment*)
  - piano roll → audio (*synthesis*)
  - piano roll → piano roll' (*rearrangement*)
  - audio → audio'
- **See Section 3: Case studies**
  - and, https://github.com/affige/genmusic_demo_list



We will talk about

1. **Symbolic melody generation**: MidiNet [1], SSMGAN [2]
2. **Arrangement generation**: MuseGAN [3], BinaryMuseGAN [4], LeadSheetGAN [5]
3. **Style transfer**: CycleGAN [6], TimbreTron [7], Play-as-you-like [8], CycleBEGAN [9]
4. **Audio generation**: WaveGAN [10], GANSynth [11]

---

[1] “MidiNet: A convolutional GAN for symbolic-domain music generation,” *ISMIR* 2017
[2] “Modeling self-repetition in music generation using structured adversaries,” *ML4MD* 2019
[3] “MuseGAN: Multi-track sequential GANs for symbolic music generation and accompaniment,” *AAAI* 2018
[4] “Convolutional GANs with binary neurons for polyphonic music generation,” *ISMIR* 2018
[5] “Lead sheet generation and arrangement by conditional GAN,” *ISMIR-LBD* 2018
[6] “Symbolic music genre transfer with CycleGAN,” *ICTAI* 2018
[7] “TimbreTron: A WaveNet(CycleGAN(CQT(Audio))) pipeline for musical timbre transfer,” *ICLR* 2019
[8] “Play as You Like: Timbre-enhanced multi-modal music style transfer,” *AAAI* 2019
[9] “Singing style transfer using cycle-consistent boundary equilibrium GANs,” *ICML workshop* 2018
[10] “Adversarial audio synthesis,” *ICLR* 2019
[11] “GANSynth: Adversarial neural audio synthesis,” *ICLR* 2019

Outline

Section 1

Overview of music generation research

Section 2

Introduction to GANs

Section 3

Case studies of GAN-based systems

Section 4

Current limitations

Future research directions

What is a GAN?

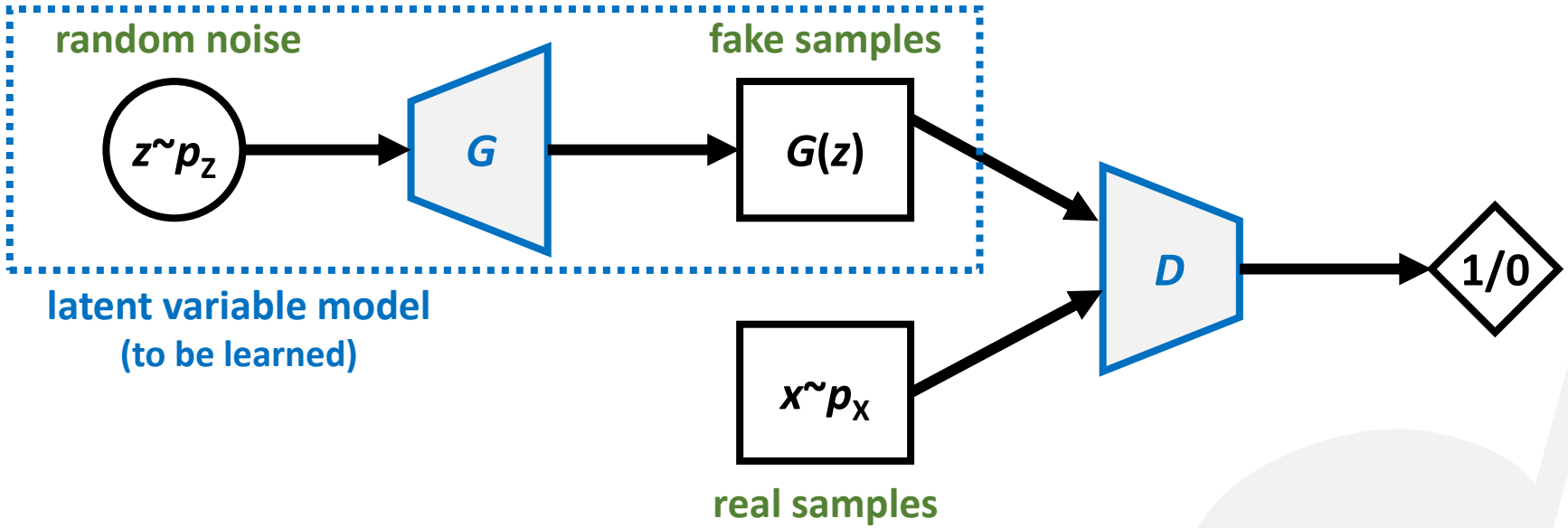
a ***generative*** model

a deep neural ***network***

Generative **A**dversarial **N**etwork

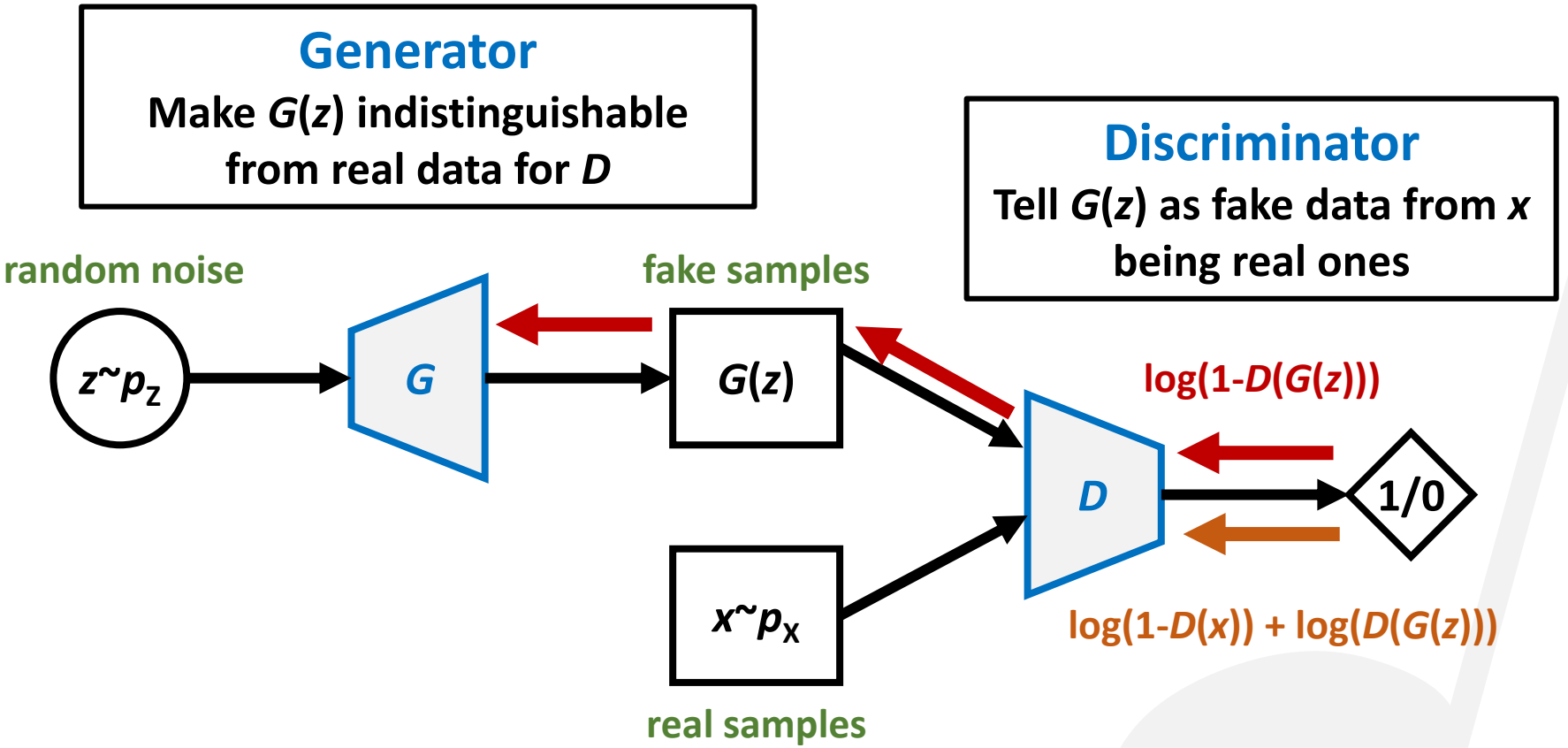
an ***adversarial*** game
between two competitors

A loss function for training generative models



Goodfellow et al., "Generative adversarial networks," *NeurIPS* 2014

A loss function for training generative models

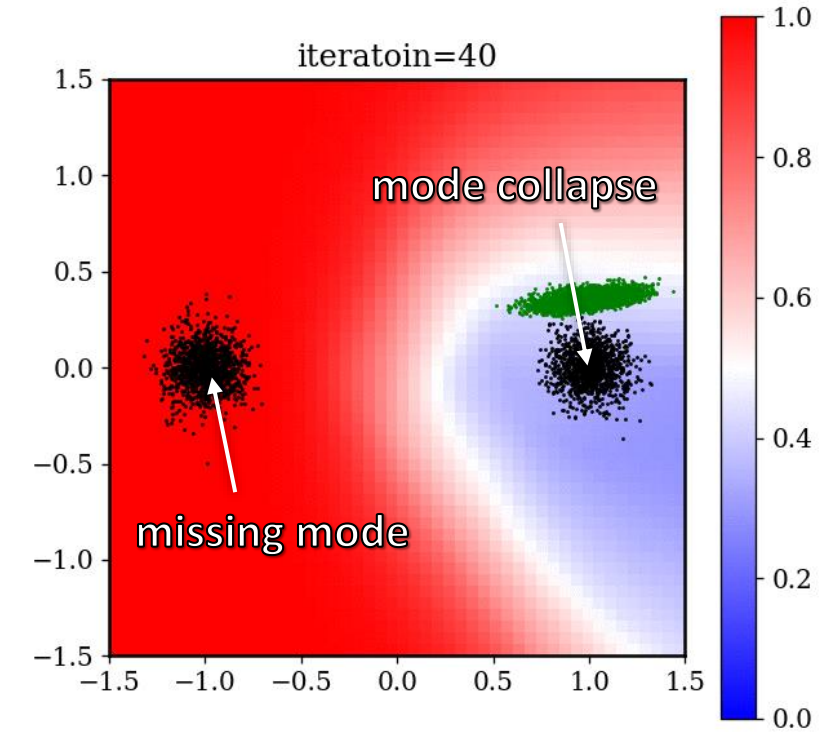


Goodfellow et al., "Generative adversarial networks," *NeurIPS* 2014

Problems of unregularized GANs

- **Key**—discriminator provides generator with gradients as a guidance for improvement
  - Discrimination is easier than generation
  - Discriminator tends to provide large gradients
  - Result in unstable training of the generator
- Common failure cases
  - Mode collapse
  - Missing modes

sharp color changes → large gradients



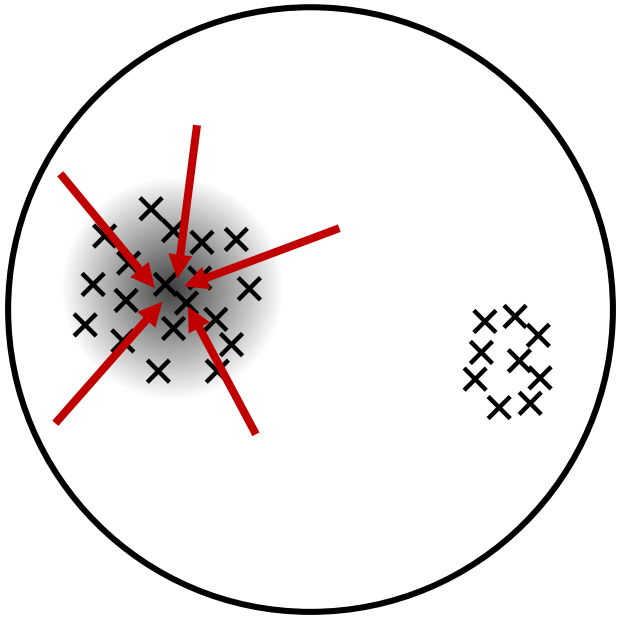
(Colors show the outputs of the discriminator)

Regularizing GANs

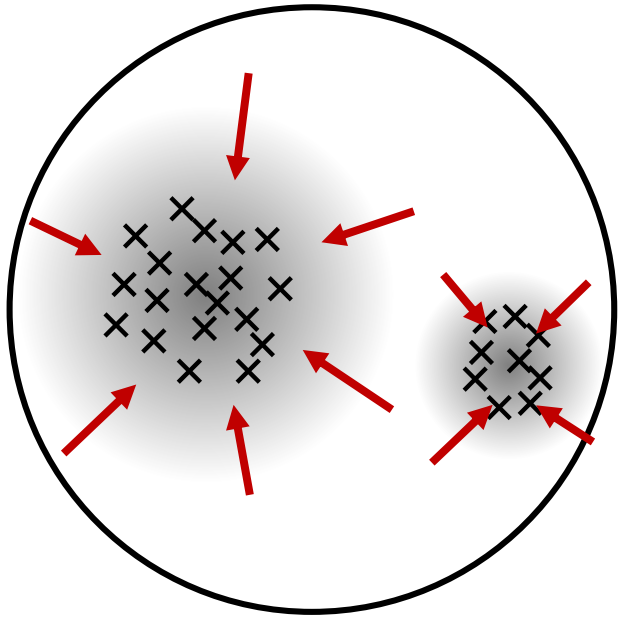
Advantages of *gradient regularization*

- **provide a smoother guidance to the generator**
- **alleviate mode collapse and missing modes issues**

Unregularized

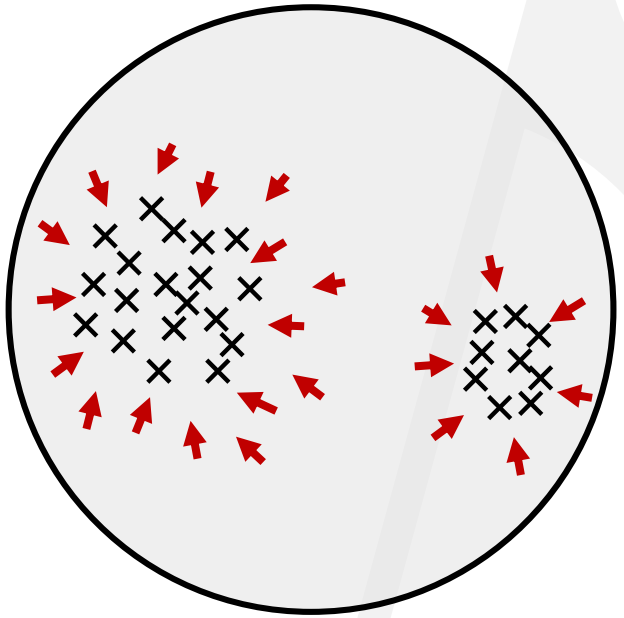


Locally regularized



gradient clipping [1]
gradient penalties [2,3]

Globally regularized



spectral normalization [4]

[1] Arjovsky et al., "Wasserstein generative adversarial networks," *ICML* 2017
[2] Gulrajani et al., "Improved training of Wasserstein GANs," *NeurIPS* 2017
[3] Kodali et al., "On convergence and stability of GANs," *arXiv* 2017
[4] Miyato et al., "Spectral normalization for generative adversarial networks," *ICLR* 2018

Coding session I—GAN for images

Google Colab notebook link

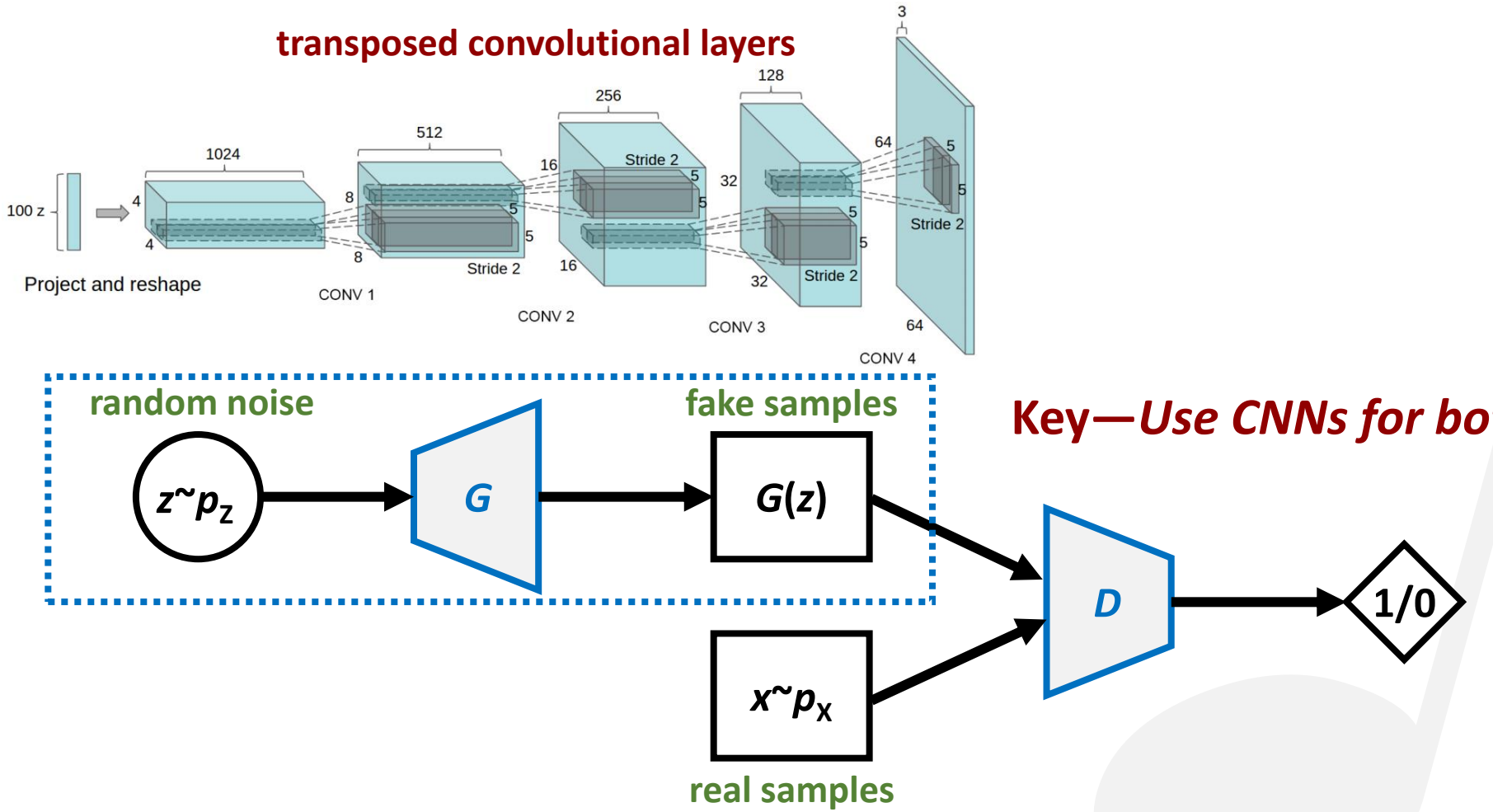
https://colab.research.google.com/drive/1Cnq9z3QvxIsVntlXKjPjbwttxeDH47Xl

You can also find the link on the tutorial website

https://salu133445.github.io/ismir2019tutorial/

Click 

Deep convolutional GAN (DCGAN)



Radford et al., "Unsupervised representation learning with deep convolutional GANs," *ICLR* 2016

GANs vs VAEs

	GAN	VAE
Objective	(generator) fool the discriminator (discriminator) tell real data from fake ones	reconstruct real data using pixel-wise loss
Results	tend to be sharper 	tend to be more blurred 
Diversity	Higher	Lower
Stability	Lower	Higher

Larsen et al., "Autoencoding beyond pixels using a learned similarity metric," *ICML* 2016

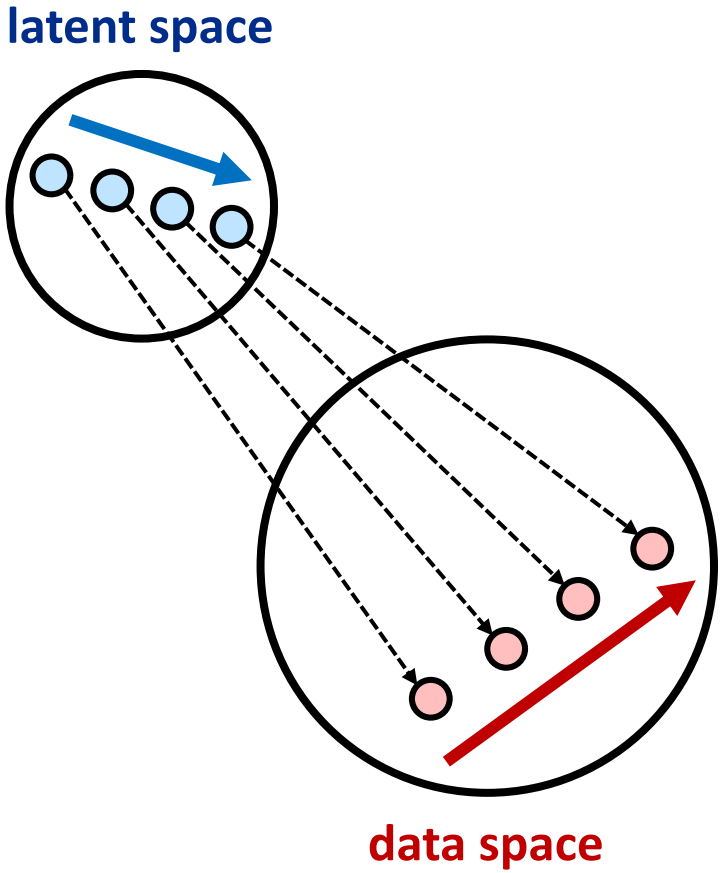
State of the arts—BigGANs

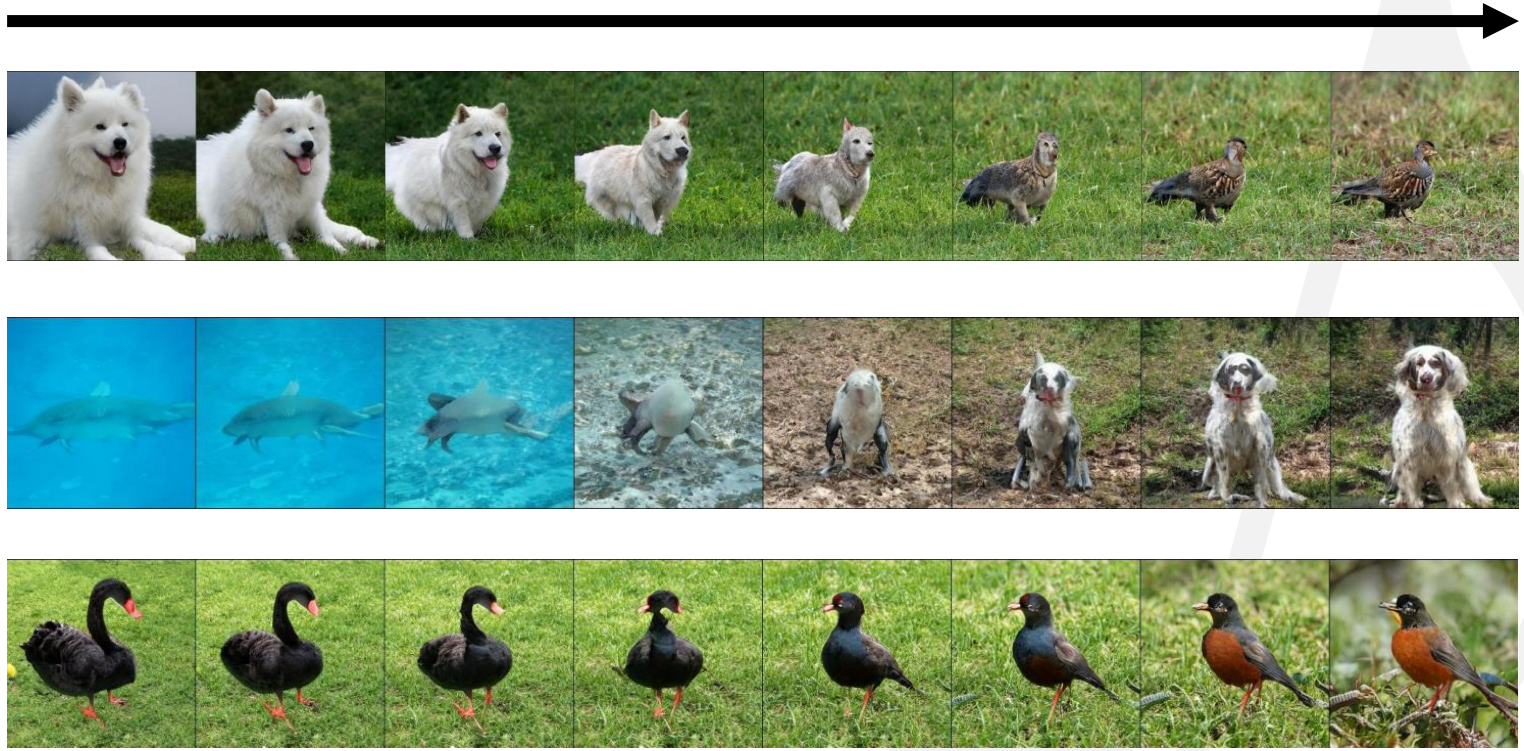
(hard classes)

[Colab notebook demo]
https://colab.research.google.com/github/tensorflow/hub/blob/master/examples/colab/biggan_generation_with_tf_hub.ipynb

Brock et al., "Large scale GAN training for high fidelity natural image synthesis," *ICLR* 2019

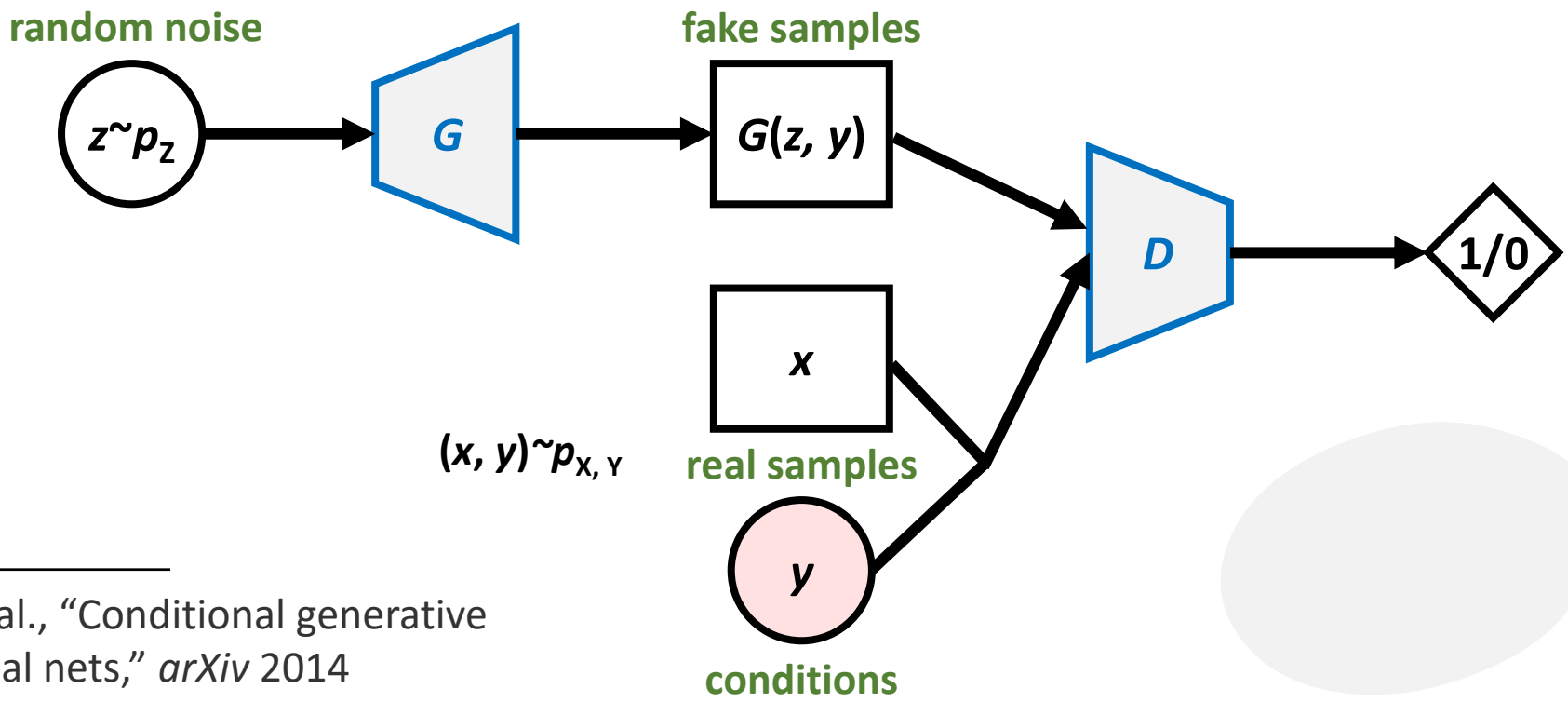
Interpolation on the latent space





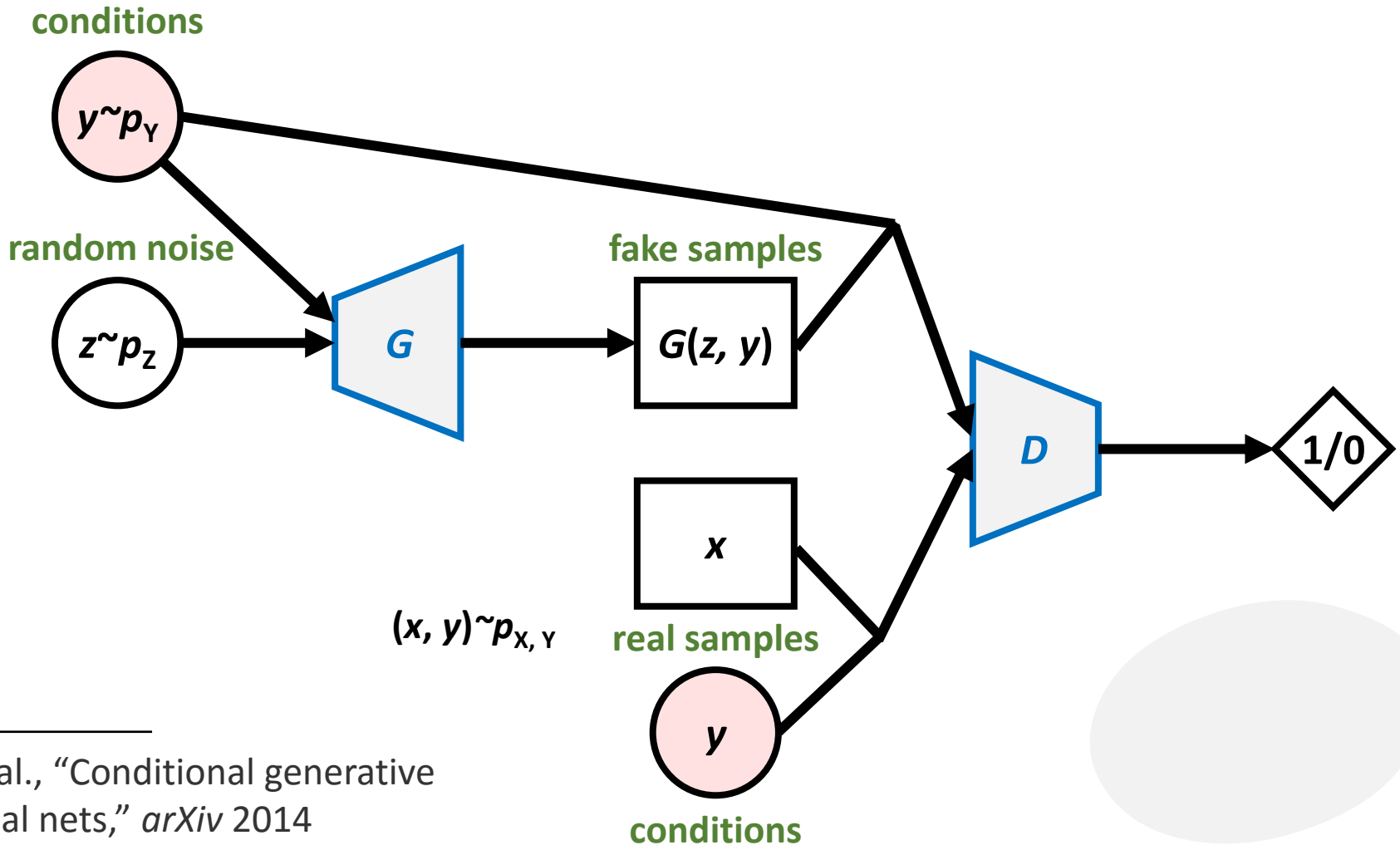
Brock et al., “Large scale GAN training for high fidelity natural image synthesis,” *ICLR* 2019

Conditional GAN (CGAN)



Mirza et al., "Conditional generative adversarial nets," *arXiv* 2014

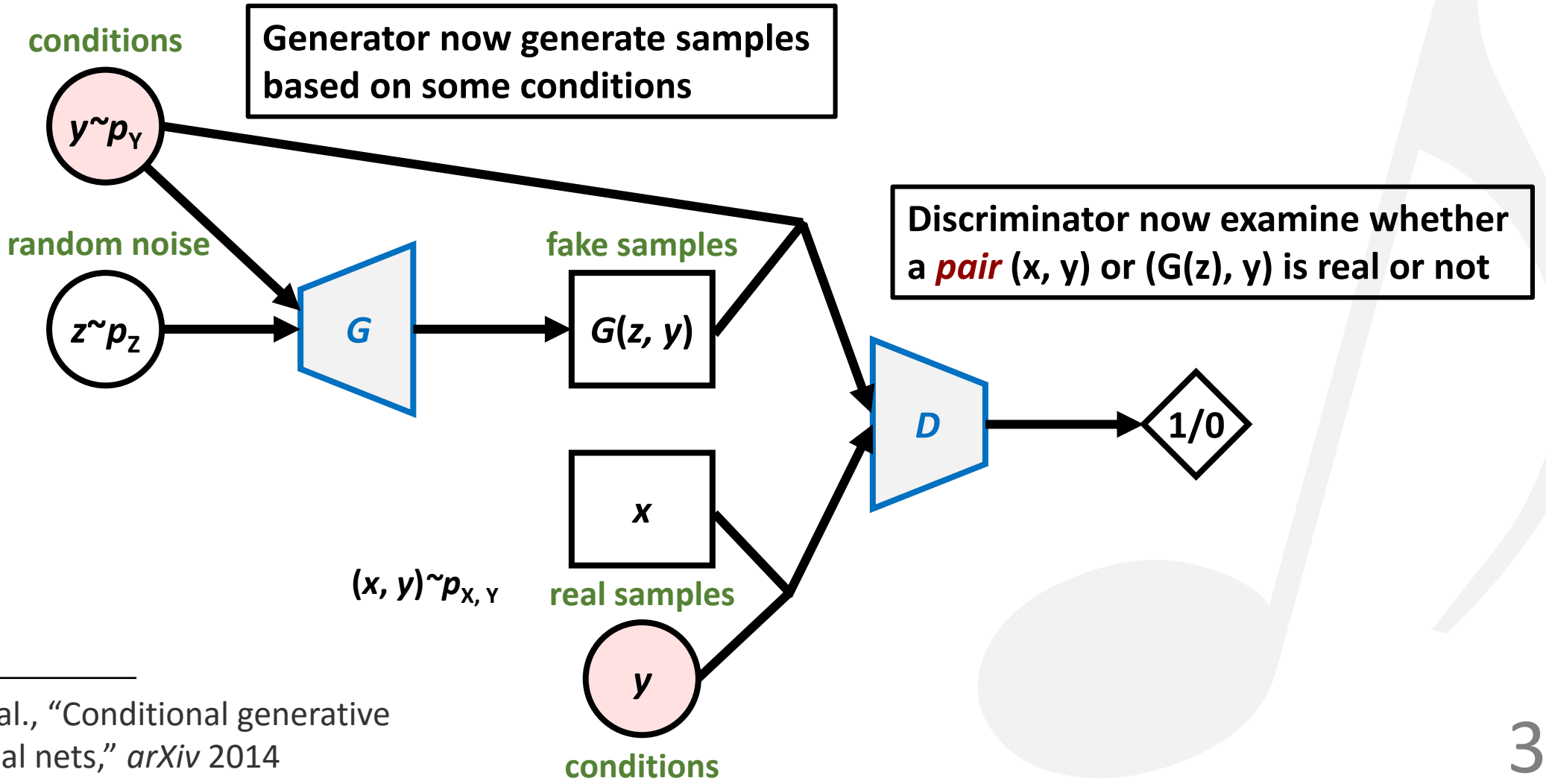
Conditional GAN (CGAN)



Mirza et al., "Conditional generative adversarial nets," *arXiv* 2014

Conditional GAN (CGAN)

Key—*Feed conditions to both G and D*



Mirza et al., "Conditional generative adversarial nets," *arXiv* 2014

Conditional GAN—Samples

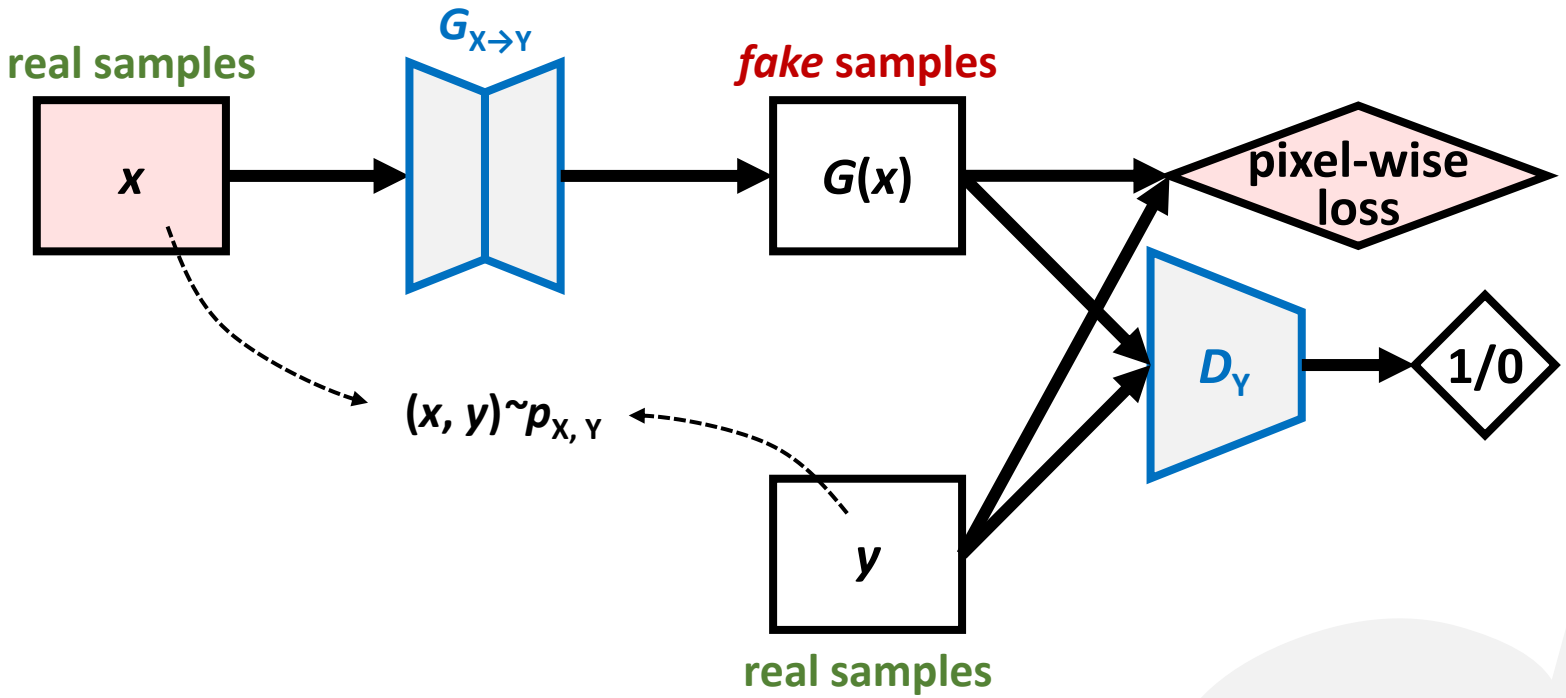
Dog

Cat

Tiger

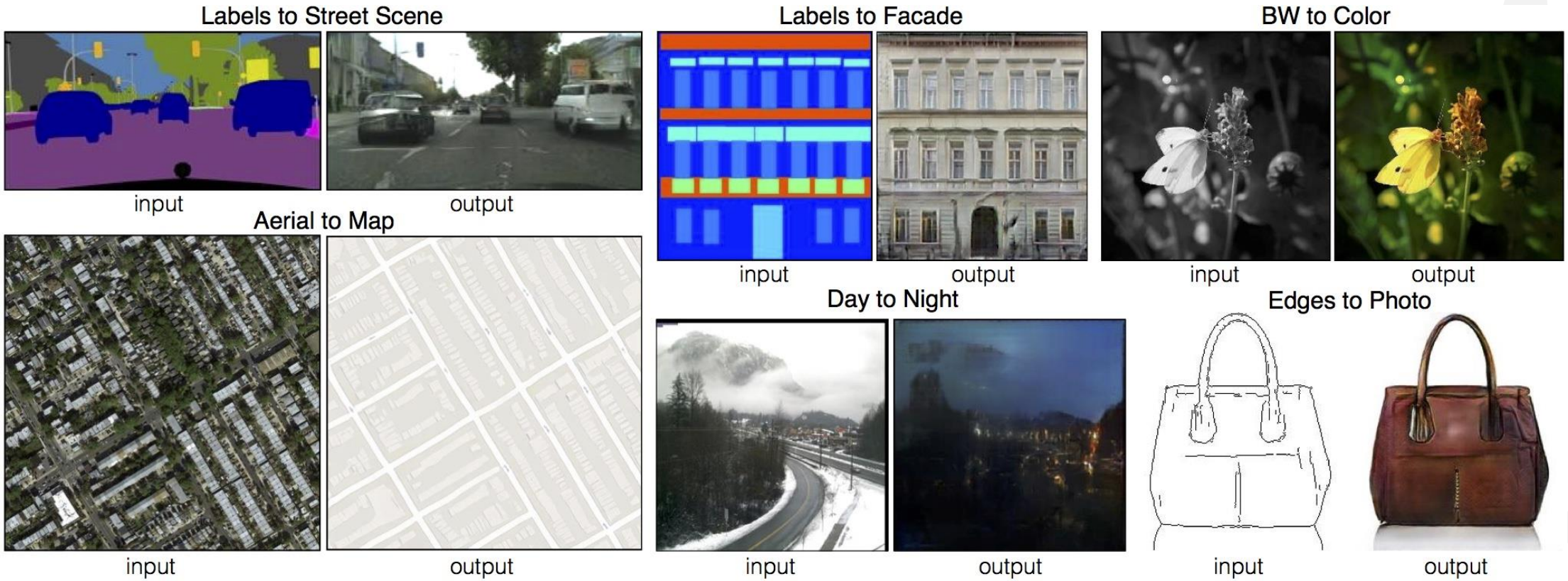
pix2pix

Key—Use pixel-wise loss for supervisions



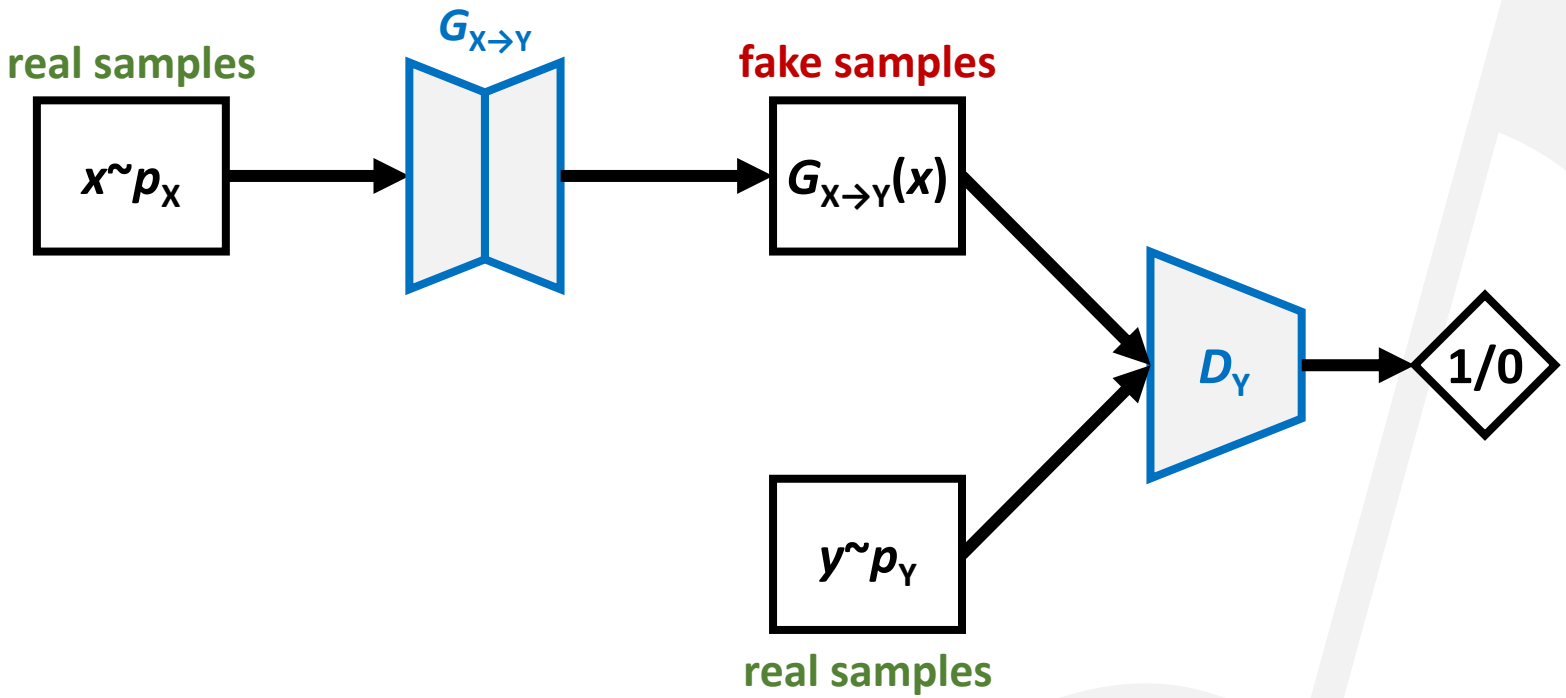
Isola et al., "Image-to-image translation with conditional adversarial nets," *CVPR* 2017

pix2pix—Samples



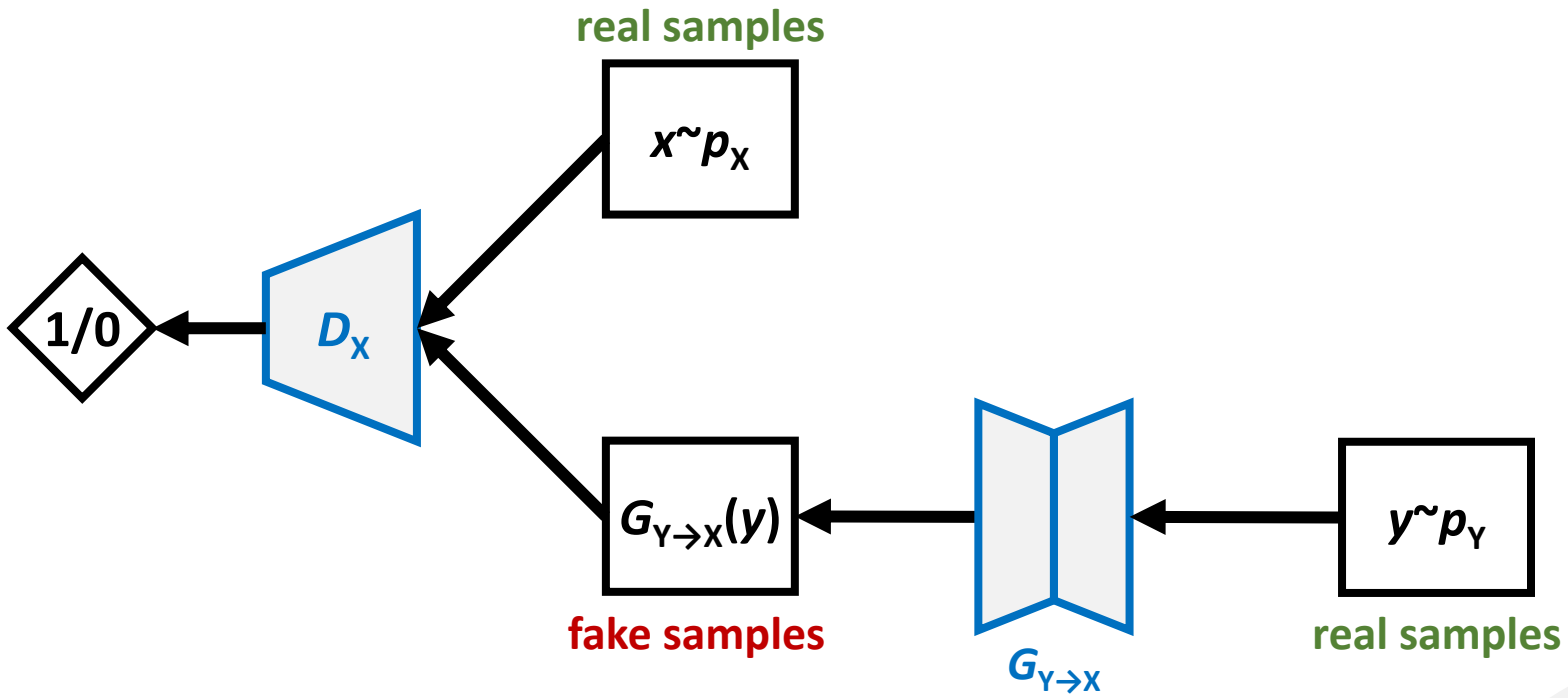
Isola et al., "Image-to-image translation with conditional adversarial nets," *CVPR* 2017

Cycle-consistent GAN (CycleGAN)



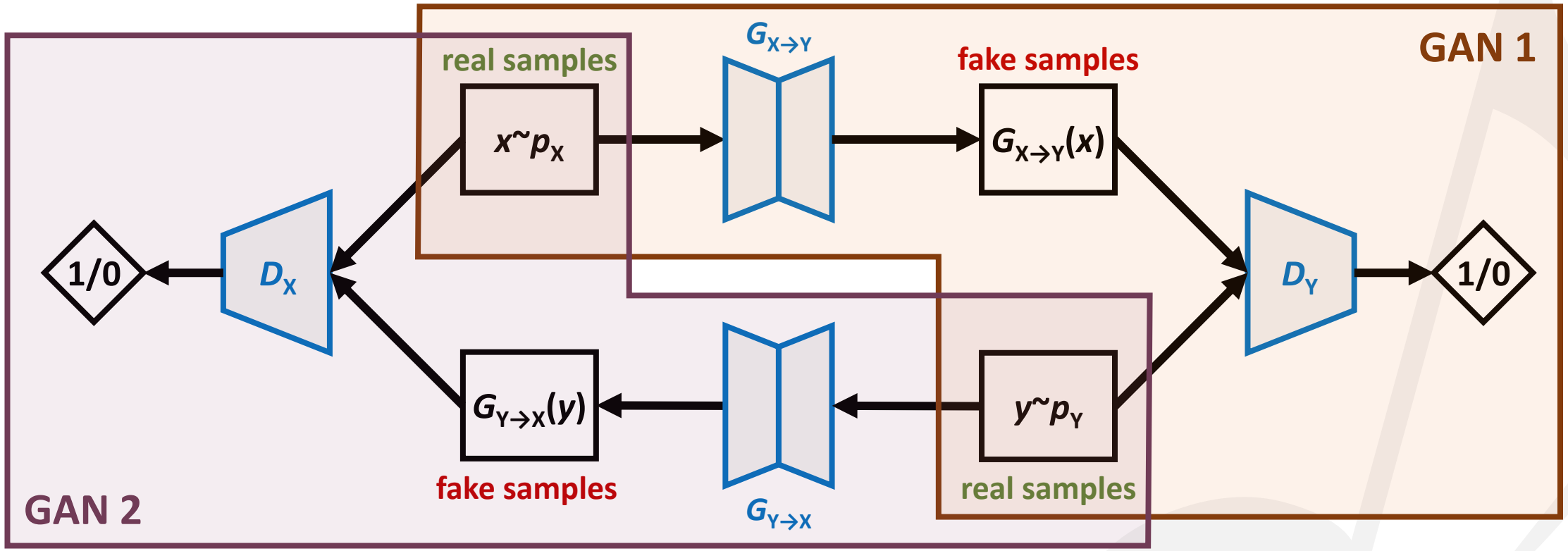
Zhu et al., "Unpaired image-to-image translation using cycle-consistent adversarial networks," *ICCV* 2017

Cycle-consistent GAN (CycleGAN)



Zhu et al., "Unpaired image-to-image translation using cycle-consistent adversarial networks," *ICCV* 2017

Cycle-consistent GAN (CycleGAN)

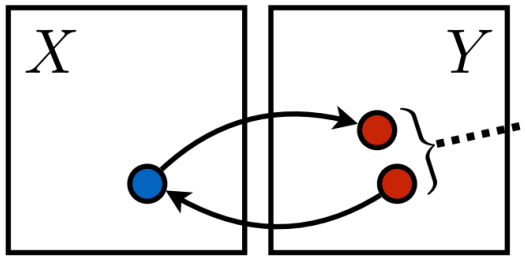


Zhu et al., "Unpaired image-to-image translation using cycle-consistent adversarial networks," *ICCV* 2017

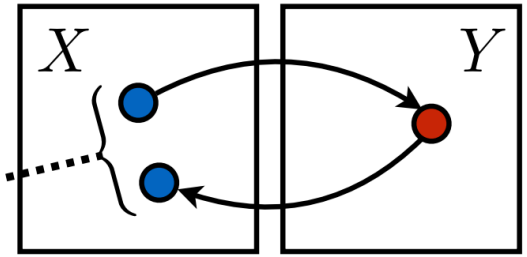
Cycle-consistent loss

Key—Use cycle-consistency loss for supervisions

CycleGAN



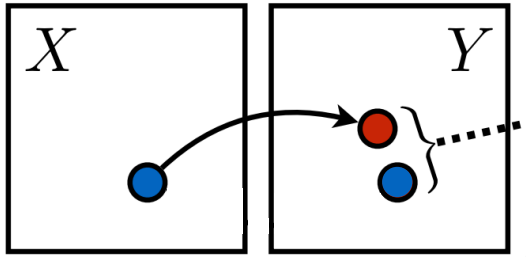
cycle-consistency loss



cycle-consistency loss

Unpaired samples in two domains

pix2pix



pixel-wise loss

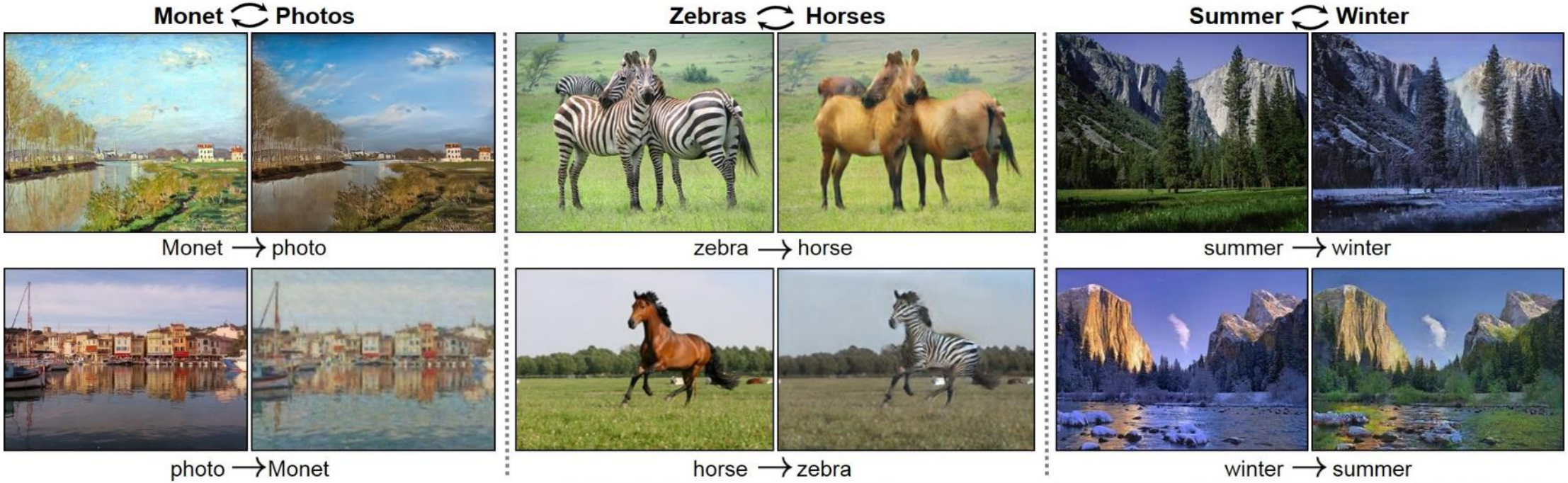
(each *x* ∈ X is mapped to a certain *y* ∈ Y)

Require paired (*x*, *y*) samples

Isola et al., "Image-to-image translation with conditional adversarial nets," *CVPR* 2017

Zhu et al., "Unpaired image-to-image translation using cycle-consistent adversarial networks," *ICCV* 2017

CycleGAN—Samples



(pix2pix) *We do not need a Monte's version for each photo → hard to acquire*

(CycleGAN) *We only need a collection of Monet's paintings and a collection of photos → easier to acquire*

Zhu et al., "Unpaired image-to-image translation using cycle-consistent adversarial networks," *ICCV* 2017

The many other GANs … and more!

Training criterions

- LSGAN
- WGAN
- EBGAN
- BEGAN
- GeometricGAN
- RaGAN

Optimization constraints

- WGAN
- WGANGP
- McGAN
- MMDGAN
- FisherGAN
- DRAGAN
- SNGAN

Training strategies

- UnrolledGAN
- LAPGAN
- StackedGAN
- StackGAN
- PGGAN
- StyleGAN
- BigGAN

Applications

- DCGAN
- InfoGAN
- CGAN
- ACGAN
- pix2pix
- CycleGAN
- CoGAN
- DAN

See more at https://github.com/hindupuravinash/the-gan-zoo

Open questions about GANs

- https://distill.pub/2019/gan-open-problems/

1. What are the trade-offs between GANs and other generative models?
2. What sorts of distributions can GANs model?
3. How can we Scale GANs beyond image synthesis?
4. What can we say about the global convergence of the training dynamics?
5. How should we evaluate GANs and when should we use them?
6. How does GAN training scale with batch size?
7. What is the relationship between GANs and adversarial examples?

Comparative studies on GANs

[1] Kunfeng Wang, Chao Gou, Yanjie Duan, Yilun Lin, Xinhu Zheng, and Fei-Yue Wang. Generative adversarial networks: Introduction and outlook. *IEEE/CAA Journal of Automatica Sinica*, 4(4):588-598, 2017.

[2] Mario Lučić, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are GANs created equal? A large-scale Study. *Proc. Advances in Neural Information Processing Systems 31*, pp. 700–709, 2018.

[3] Karol Kurach, Mario Lučić, Xiaohua Zhai, Marcin Michalski, and Sylvain Gelly. A large-scale study on regularization and normalization in GANs. *Proc. International Conference on Machine Learning (PMLR),* 97:3581-3590, 2019.

[4] Hao-Wen Dong and Yi-Hsuan Yang, “Towards a deeper understanding of adversarial losses,” *arXiv preprint arXiv:1901.08753*, 2019.

Tea Time!

Or try the next Google Colab notebook

https://colab.research.google.com/drive/1WrFtqo5LW8QfhiuhHmge9QLexWwS2BcM

Generating Music with GANs

Section 1

Overview of music generation research

Section 2

Introduction to GANs

Section 3

Case studies of GAN-based systems

Section 4

Current limitations &

Future research directions

Outline

Section 1

Overview of music generation research

Section 2

Introduction to GANs

Section 3

Case studies of GAN-based systems

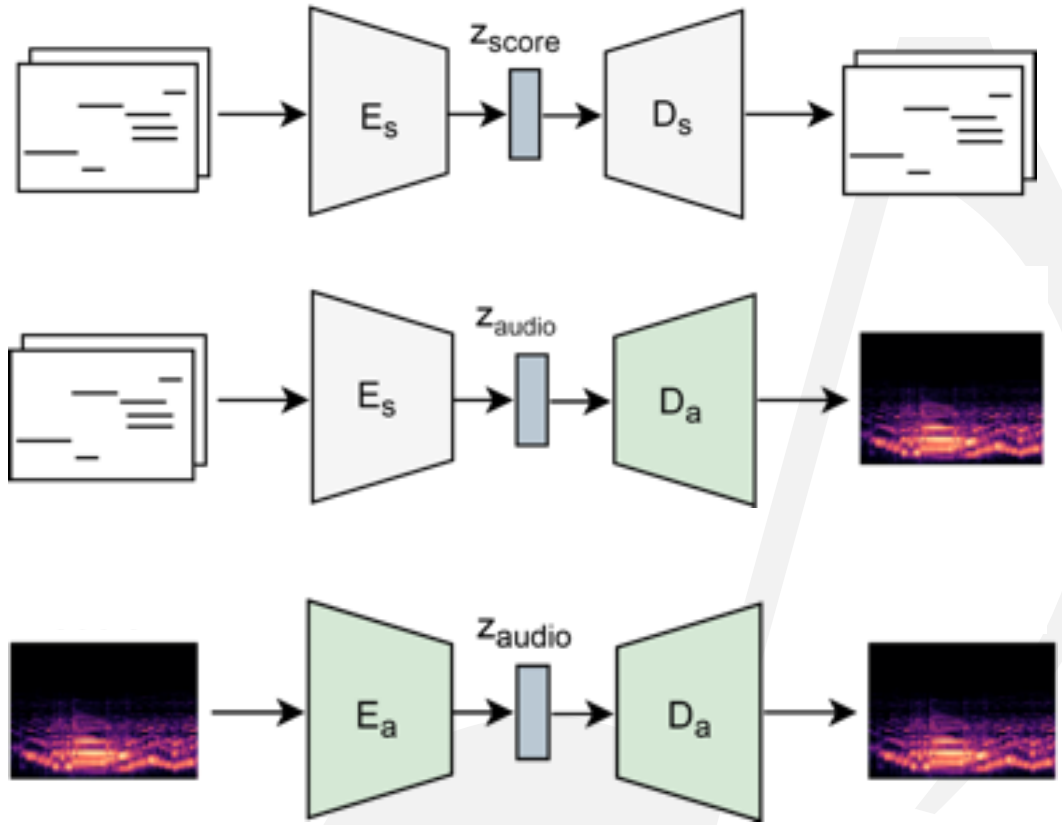
Section 4

Current limitations

Future research directions

Scope of music generation

1. **Symbolic melody generation**
   - X → melody
2. **Arrangement generation**
   - X → piano roll
   - melody → piano roll
3. **Style transfer**
   - piano roll → piano roll'
   - audio → audio'
4. **Audio generation**
   - X → audio
   - piano roll → audio



Some symbolic datasets to starts with

Dataset	Type	Format	Link
Wikifornia	lead sheet	XML	www.wikifonia.org
HookTheory	lead sheet	XML	www.hooktheory.com/theorytab
Lakh MIDI Dataset	multitrack	MIDI	https://colinraffel.com/projects/lmd
Lakh Pianoroll Dataset	multitrack	npz	salu133445.github.io/lakh-pianoroll-dataset
Groove MIDI Dataset	drum	MIDI	magenta.tensorflow.org/datasets/groove
Midi Man	drum	MIDI	www.reddit.com/r/WeAreTheMusicMakers/comments/3anwu8/the_drum_percussion_midi_archive_800k/

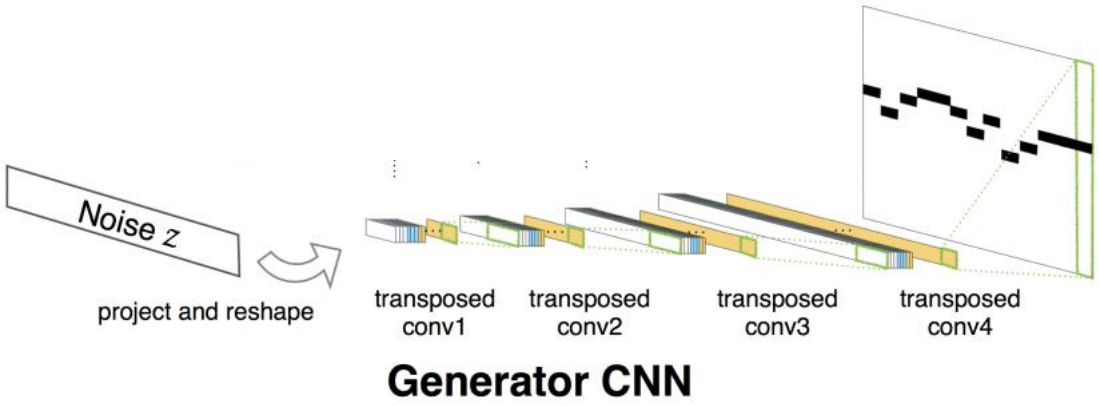
(Only datasets with *miscellaneous genres* are presented here)

See more at https://github.com/wayne391/symbolic-musical-datasets

GANs for Symbolic Melody Generation

X → melody

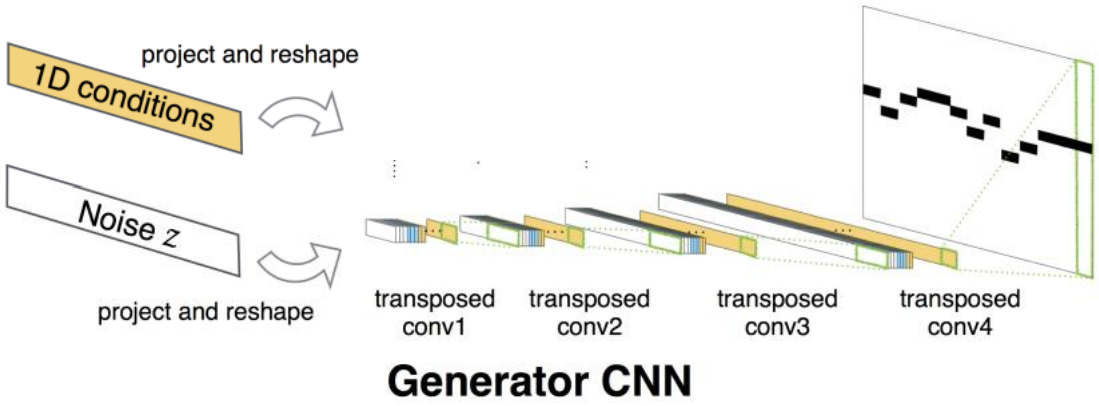
MidiNet



Yang et al., “MidiNet: A convolutional generative adversarial network for symbolic-domain music generation,” *ISMIR* 2017

MidiNet

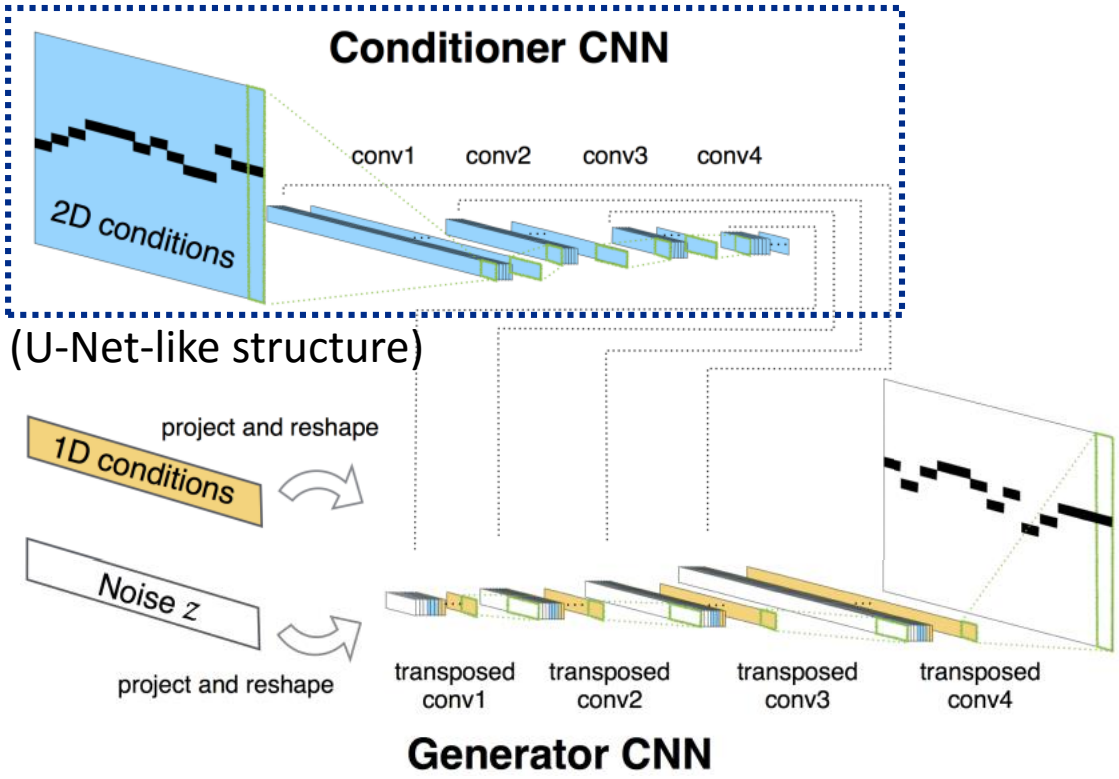
chords



Yang et al., "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation," *ISMIR* 2017

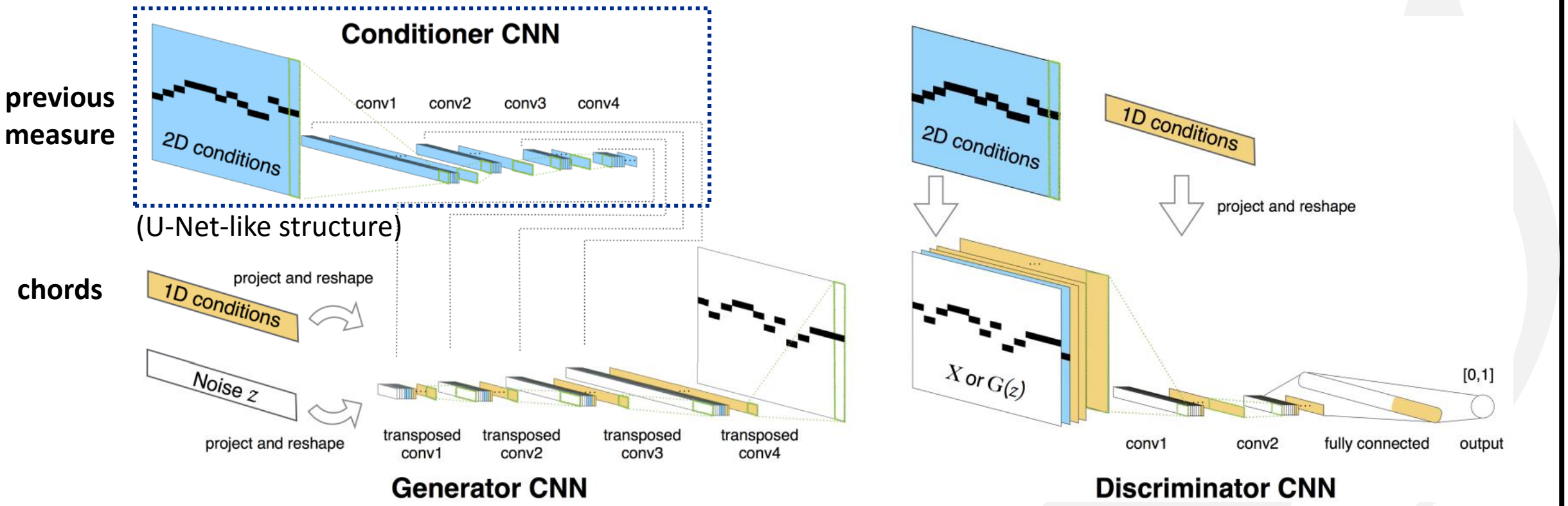
MidiNet

previous measure



Yang et al., "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation," *ISMIR* 2017

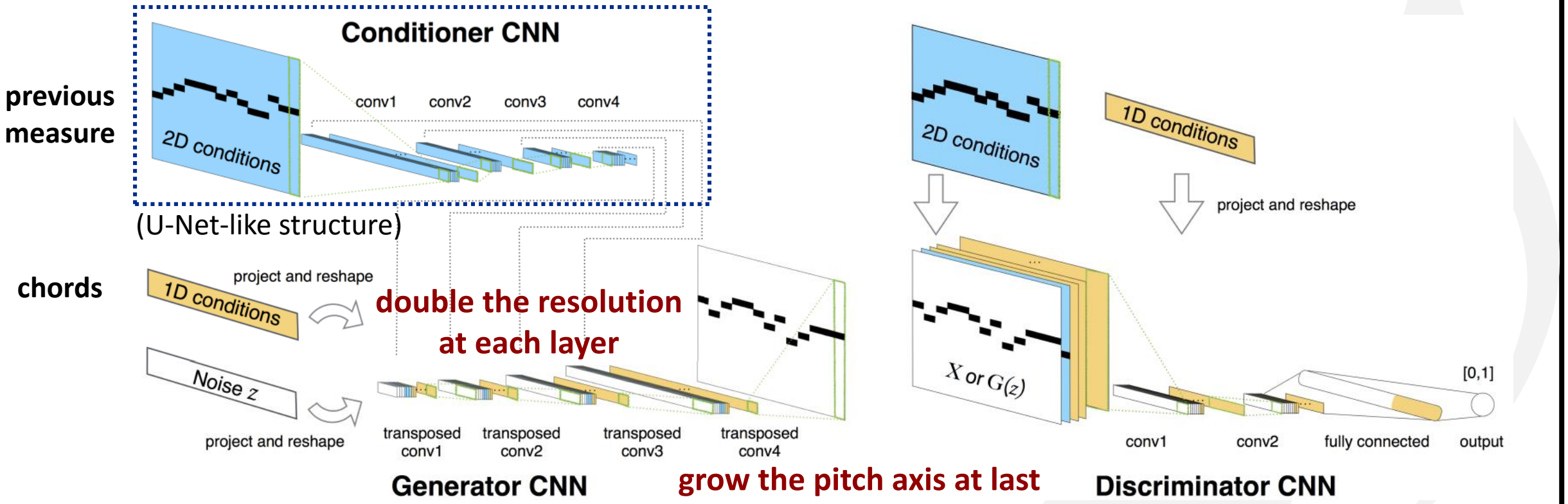
MidiNet



Yang et al., "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation," *ISMIR* 2017

MidiNet

Key—Design CNN kernel sizes to match our understanding of music

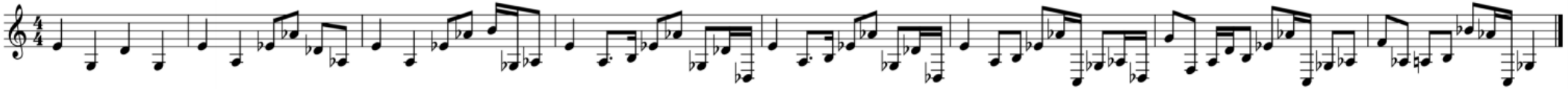


Yang et al., "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation," *ISMIR* 2017

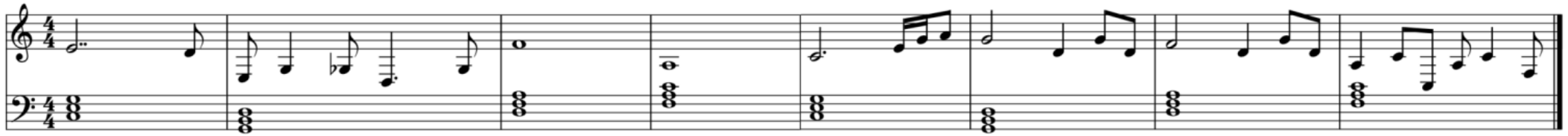
MidiNet—Samples

1D + 2D conditions
(*previous bar* and *chords*)

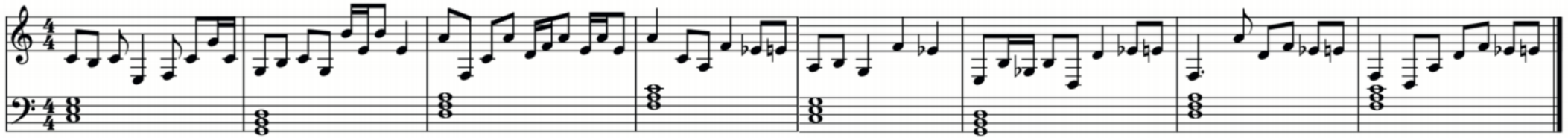
1D condition only
(*chords* only)



(a) MidiNet model 1



(b) MidiNet model 2

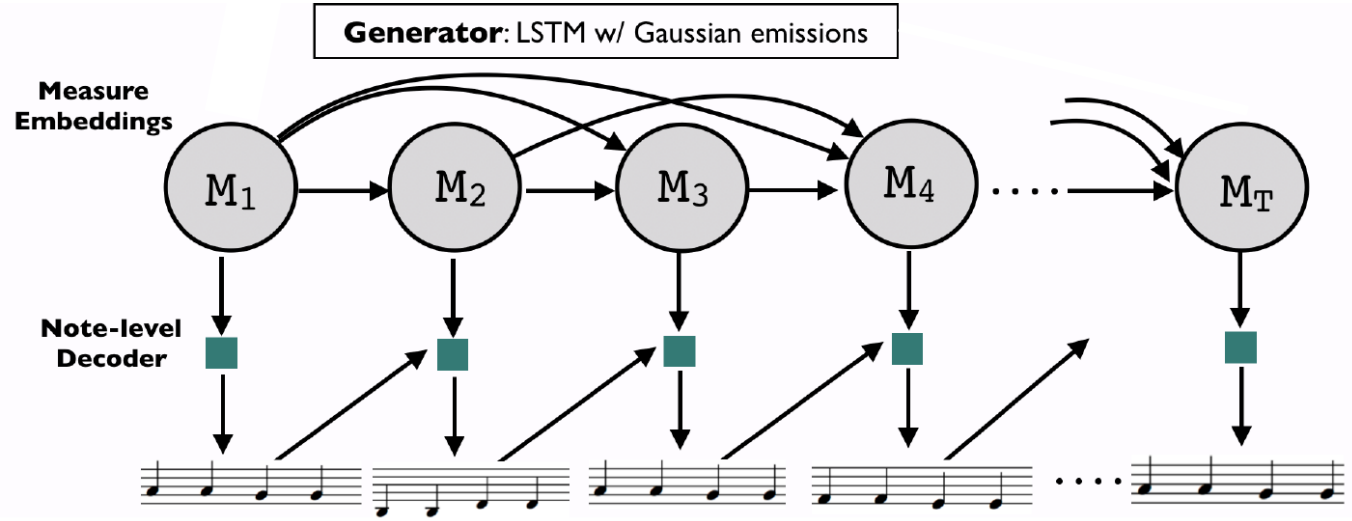


(c) MidiNet model 3

More samples can be found at https://richardyang40148.github.io/TheBlog/midinet_arxiv_demo.html

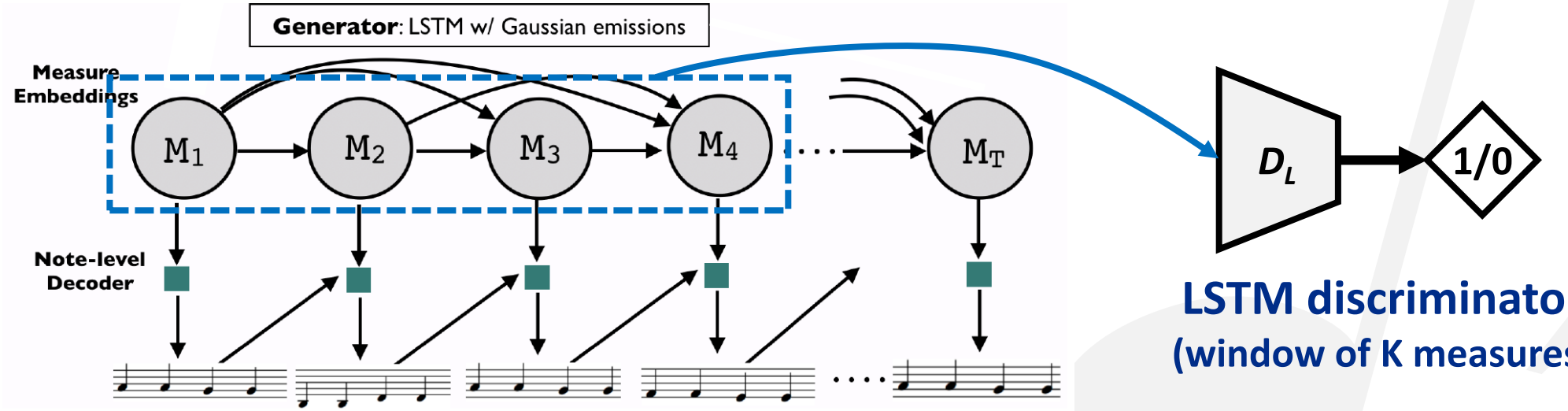
Yang et al., "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation," *ISMIR* 2017

SSMGAN



Jhamtani and Berg-Kirkpatrick, "Modeling self-repetition in music generation using structured adversaries," *ML4MD* 2019

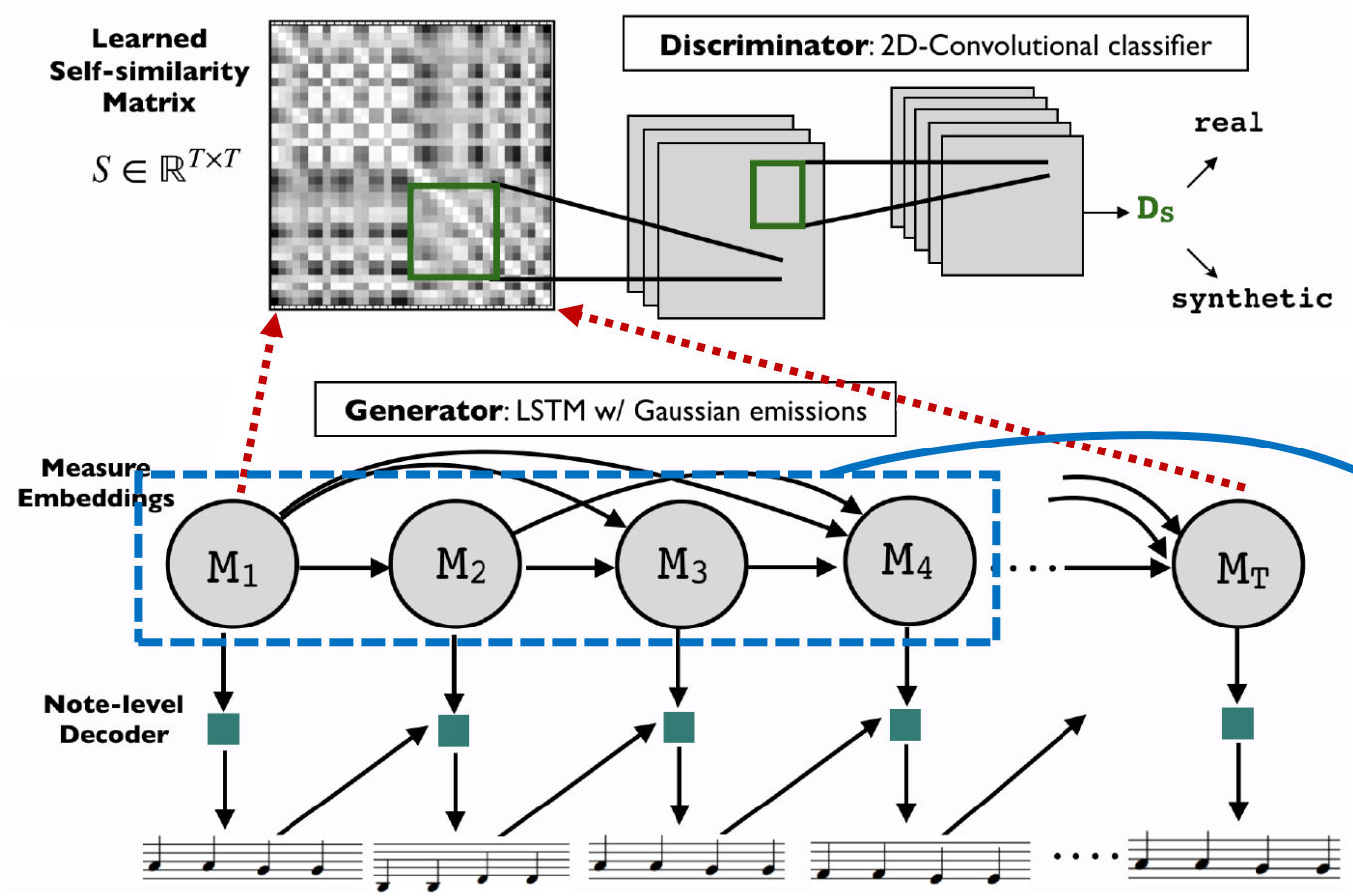
SSMGAN



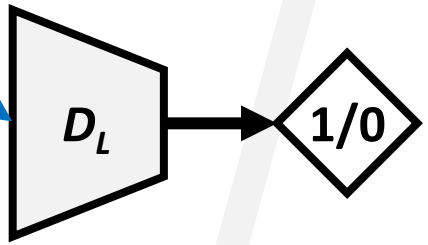
Jhamtani and Berg-Kirkpatrick, "Modeling self-repetition in music generation using structured adversaries," *ML4MD* 2019

SSMGAN

Key—Use the SSM discriminator to improve global musical structure



SSM discriminator
(all measures)



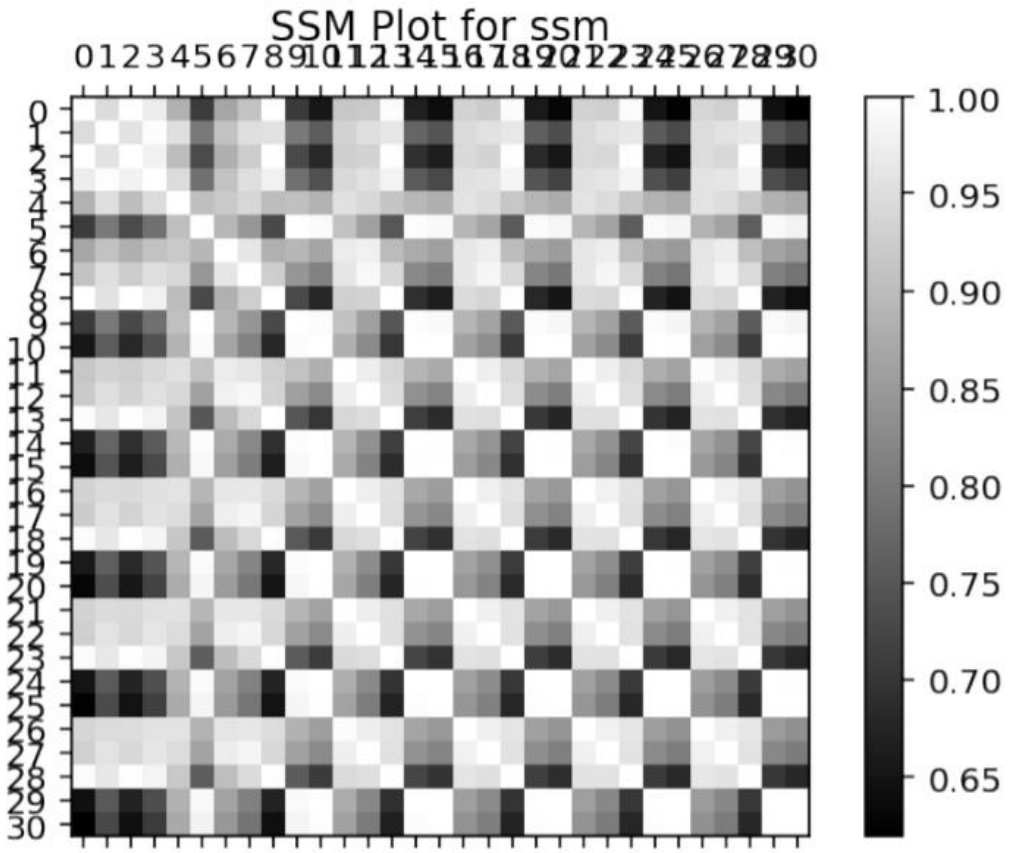
LSTM discriminator
(window of K measures)

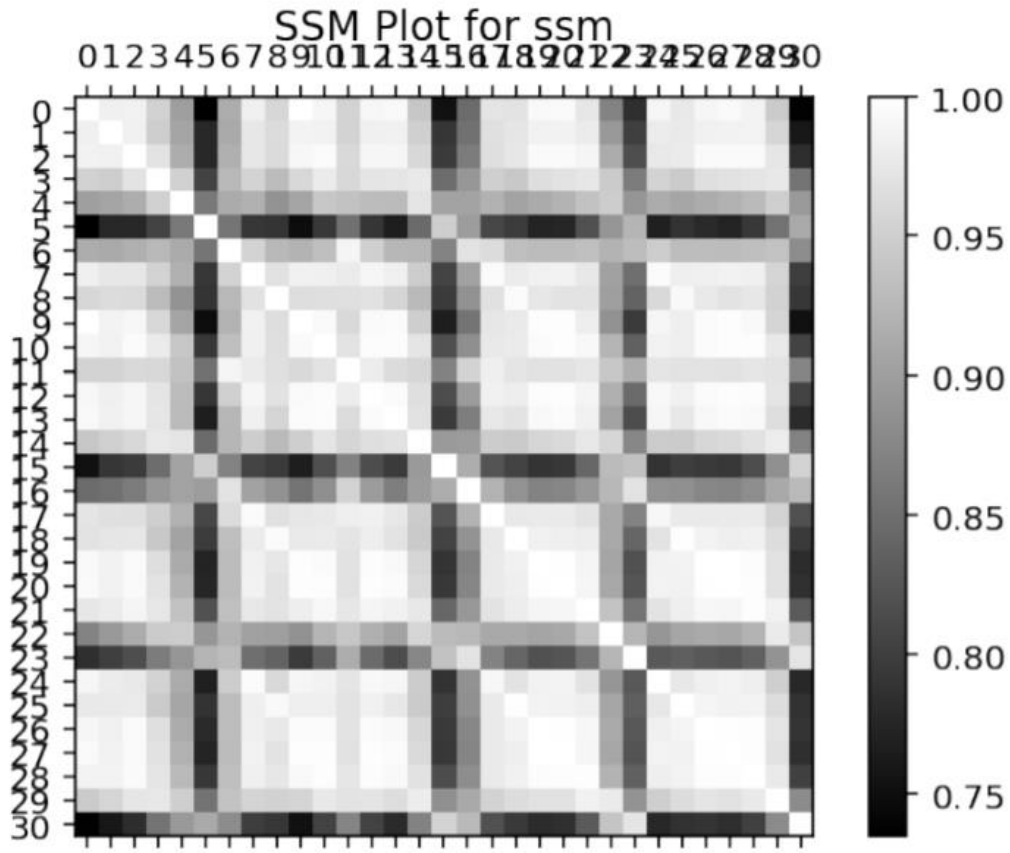
Jhamtani and Berg-Kirkpatrick, "Modeling self-repetition in music generation using structured adversaries," *ML4MD* 2019

SSMGAN—Samples

Sample 1

Sample 2





Jhamtani and Berg-Kirkpatrick, "Modeling self-repetition in music generation using structured adversaries," *ML4MD* 2019

Other GAN models for melody generation

- **C-RNN-GAN** [1]: use RNNs for the generator and discriminator
- **JazzGAN** [2]: given chords, compose melody; compare 3 representations
- **Conditional LSTM-GAN** [3]: given lyrics, compose melody
- **SSMGAN** [4]: use GAN to generate a self-similarity matrix (SSM) to represent self-repetition in music, and then use LSTM to generate melody given the SSM

---

[1] Mogren, "C-RNN-GAN: Continuous recurrent neural networks with adversarial training," *CML* 2016

[2] Trieu and Keller, "JazzGAN: Improvising with generative adversarial networks," *MUME* 2018

[3] Yu and Canales, "Conditional LSTM-GAN for melody generation from lyrics," *arXiv* 2019

[4] Jhamtani and Berg-Kirkpatrick, "Modeling self-repetition in music generation using structured adversaries," *ML4MD* 2019

GANs for Symbolic Arrangement Generation

X → piano roll
melody → piano roll

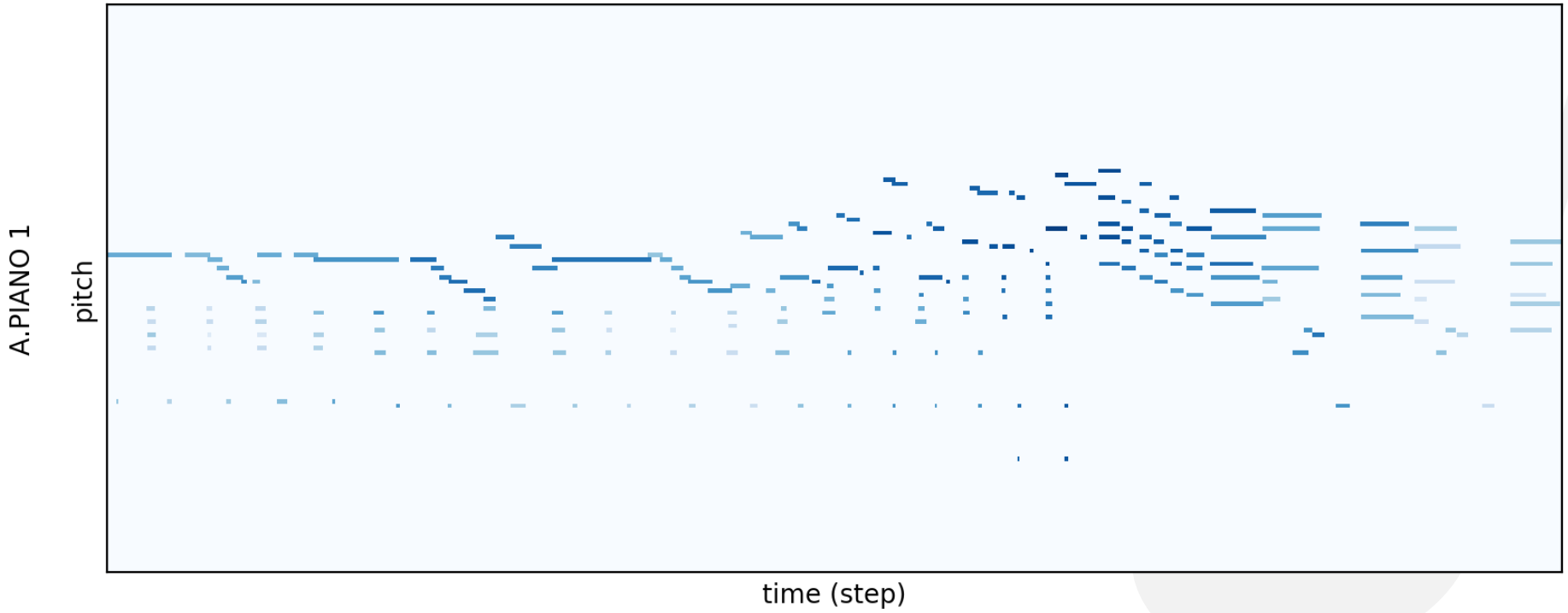
Challenges of arrangement generation

- **Temporal evolution**
  - Dynamics, emotions, tensions, etc.
- **Structure** (temporal)
  - Short-term structure → can somehow be generated with special models
  - Long-term structure → super hard
- **Instrumentation**
  - Multiple tracks
  - Functions of instruments

Couple with one another in a complex way in real world music

Why pianorolls?

- Deep learning friendly format → basically matrices
- Easier for GANs to work with

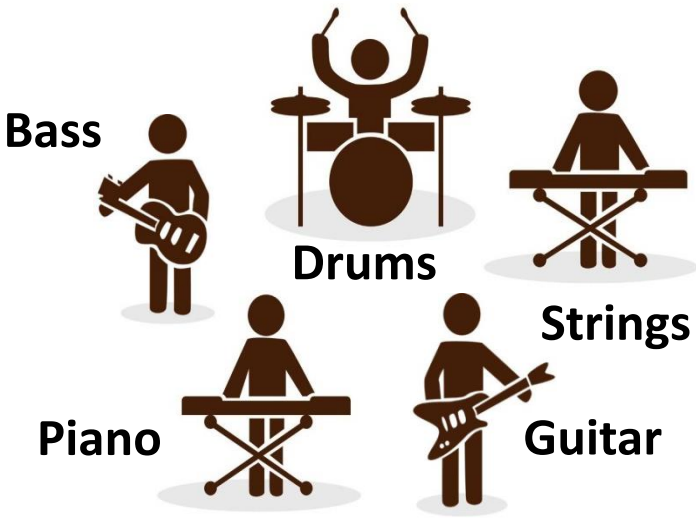


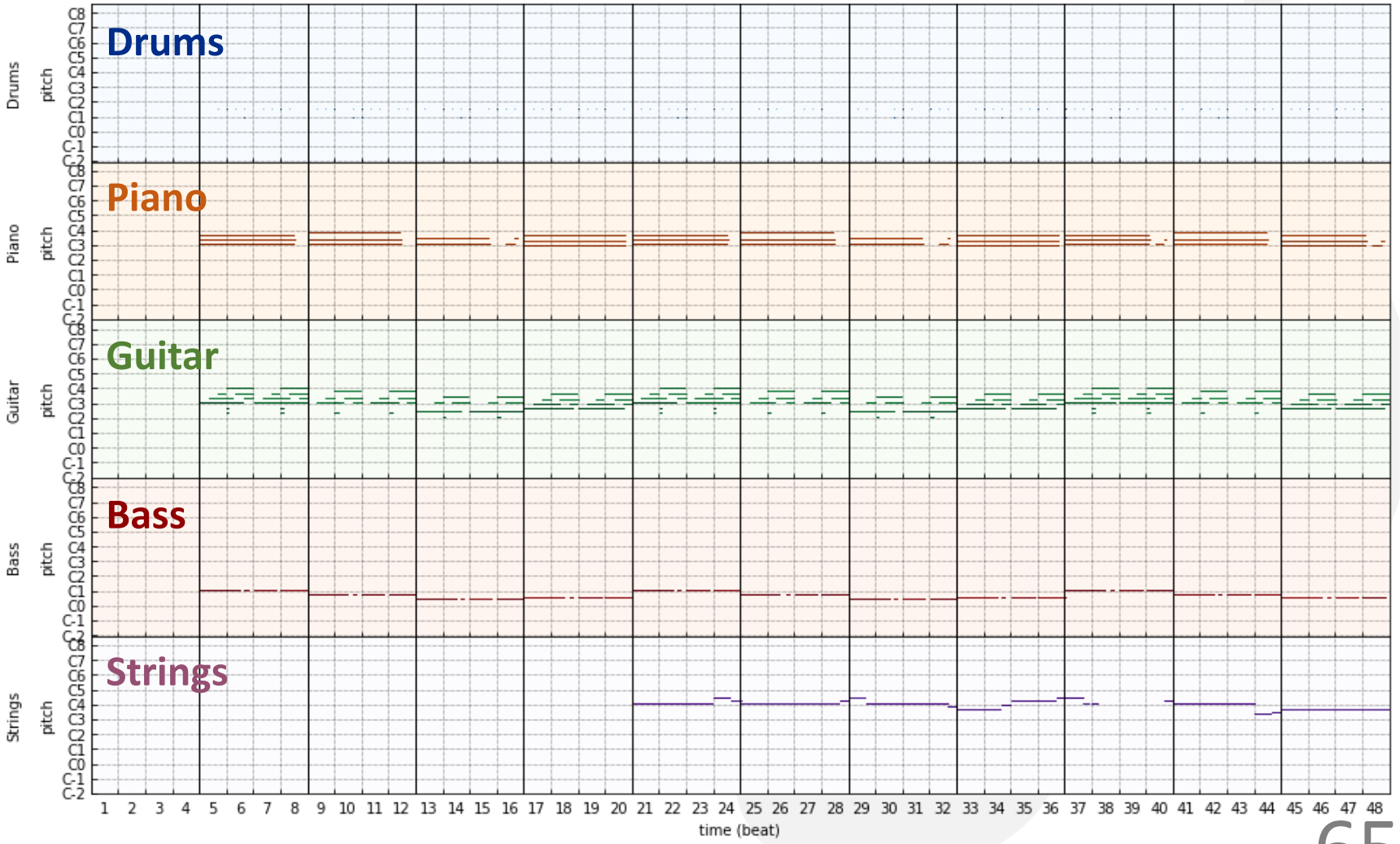
Pros and cons of pianorolls

- **Pros**
  - Can be purely symbolic → quantized by beats (or factors of beats)
  - Repetition and structure can be observed easily
  - No need to serialize polyphonic compositions

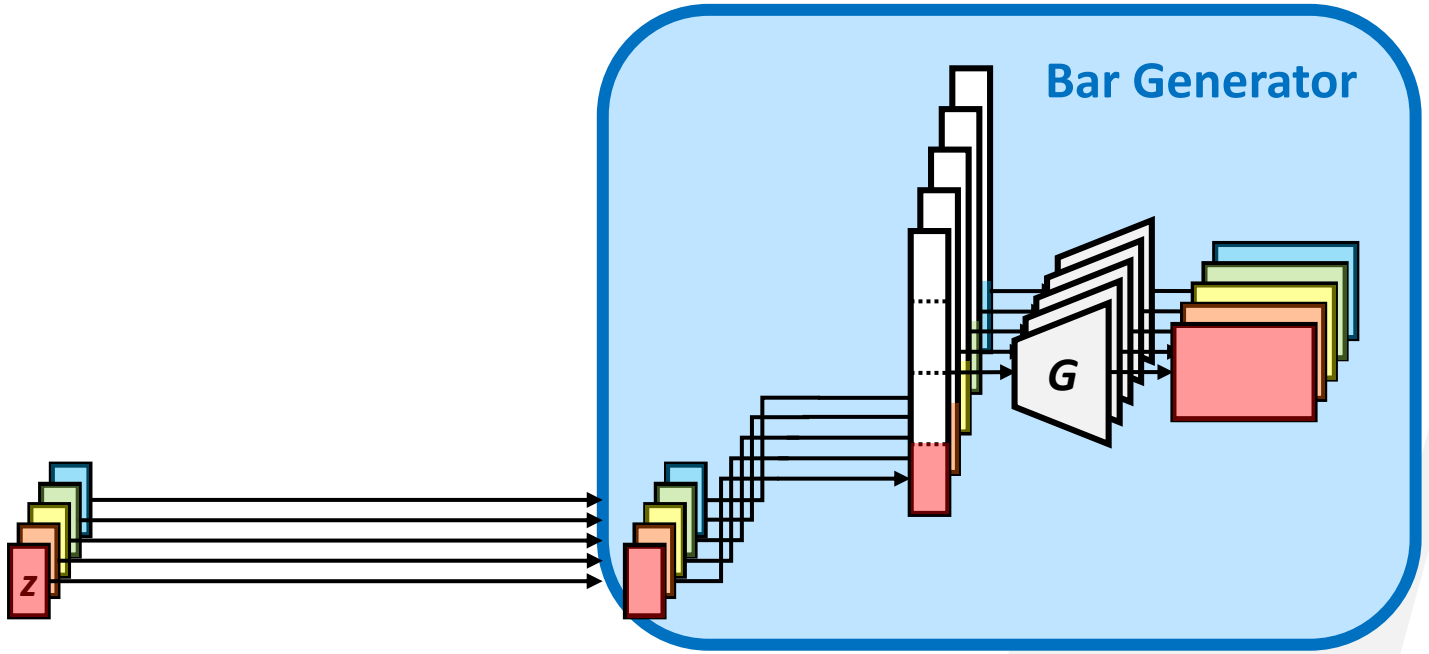
- **Cons**
  - Memory inefficient → mostly zero entries (i.e., sparse matrices)
  - Missing the concepts of “notes”
  - Hard to handle performance-level information unless using high resolution

Multitrack pianoroll



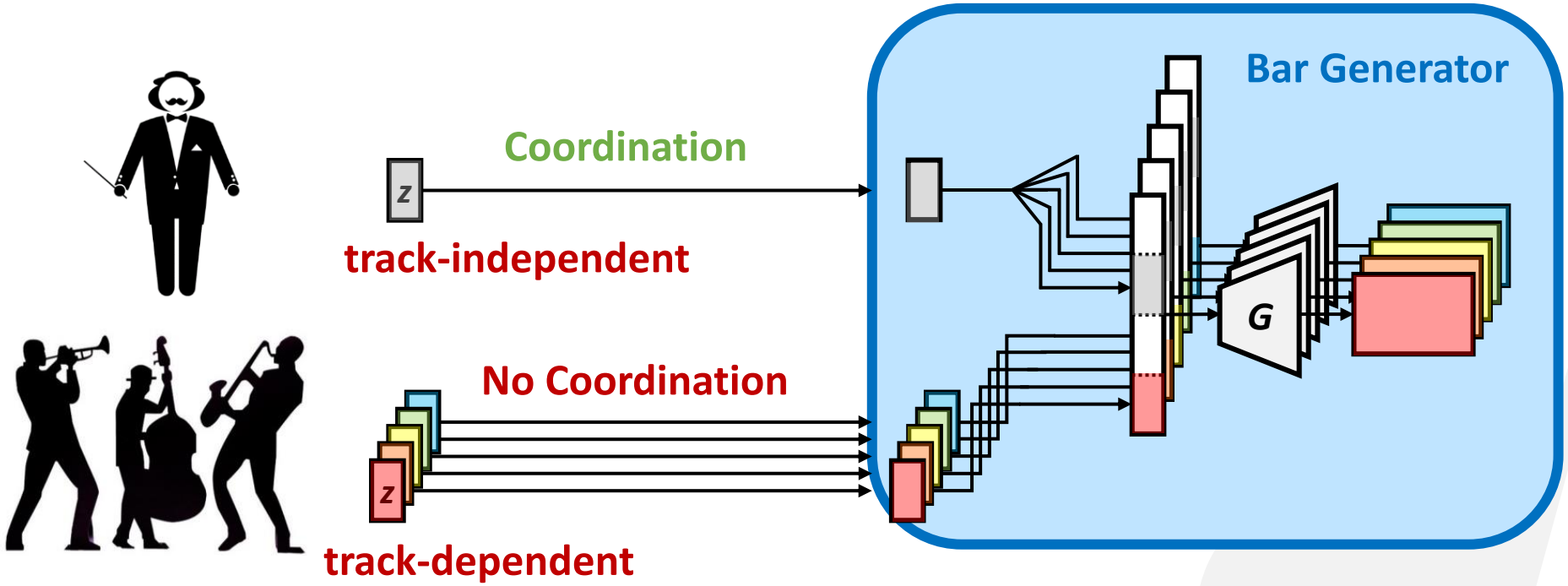


MuseGAN—Generator



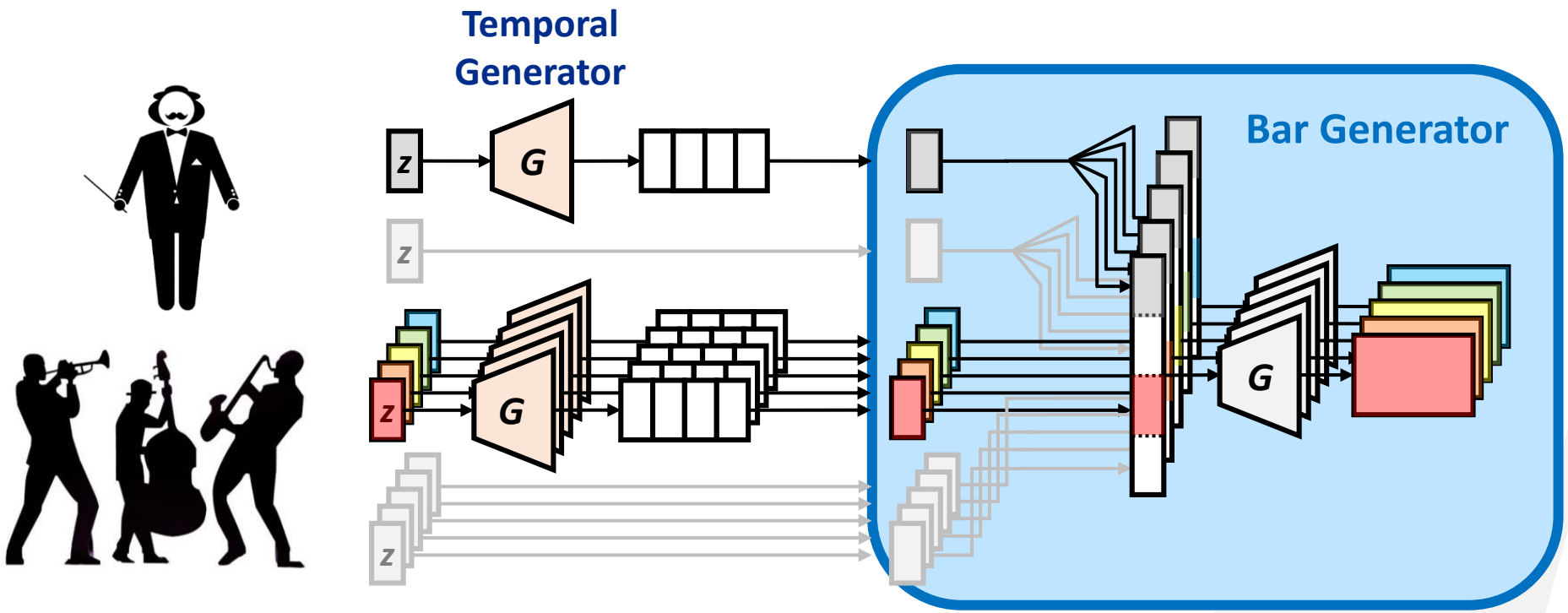
Dong et al., "MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," *AAAI* 2018

MuseGAN—Generator



Dong et al., "MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," *AAAI* 2018

MuseGAN—Generator

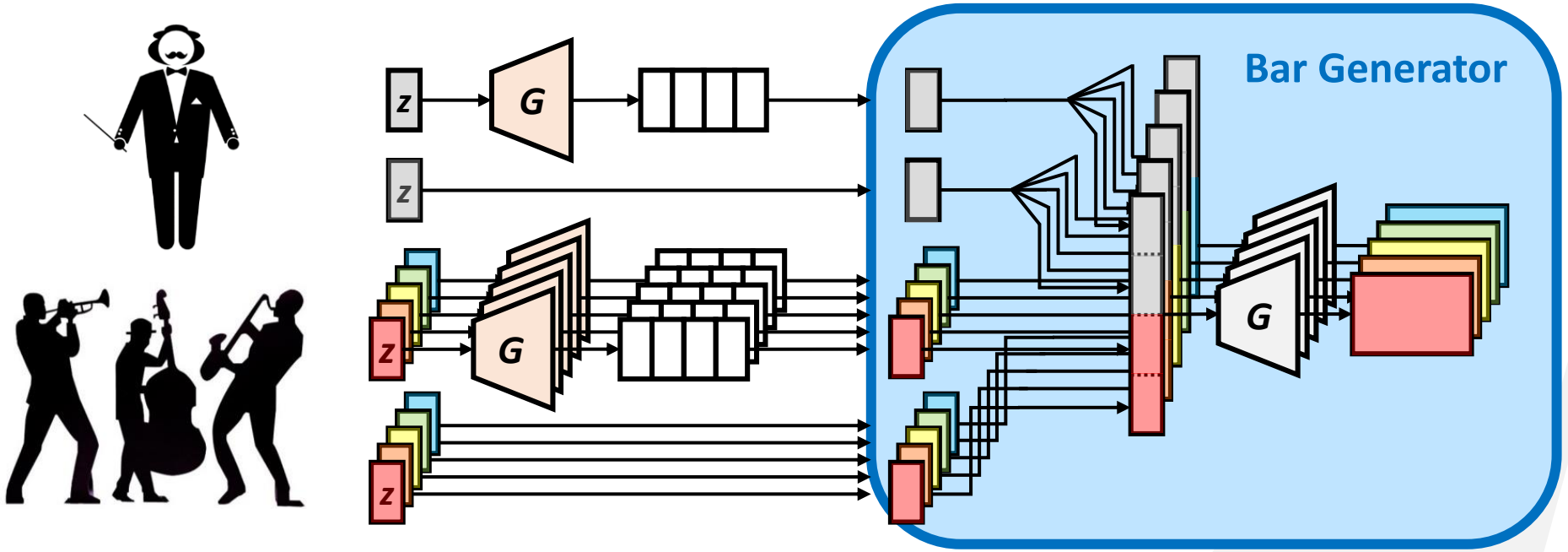


Dong et al., "MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," *AAAI* 2018

MuseGAN—Generator

Key—Use different types of latent variables to enhance controllability

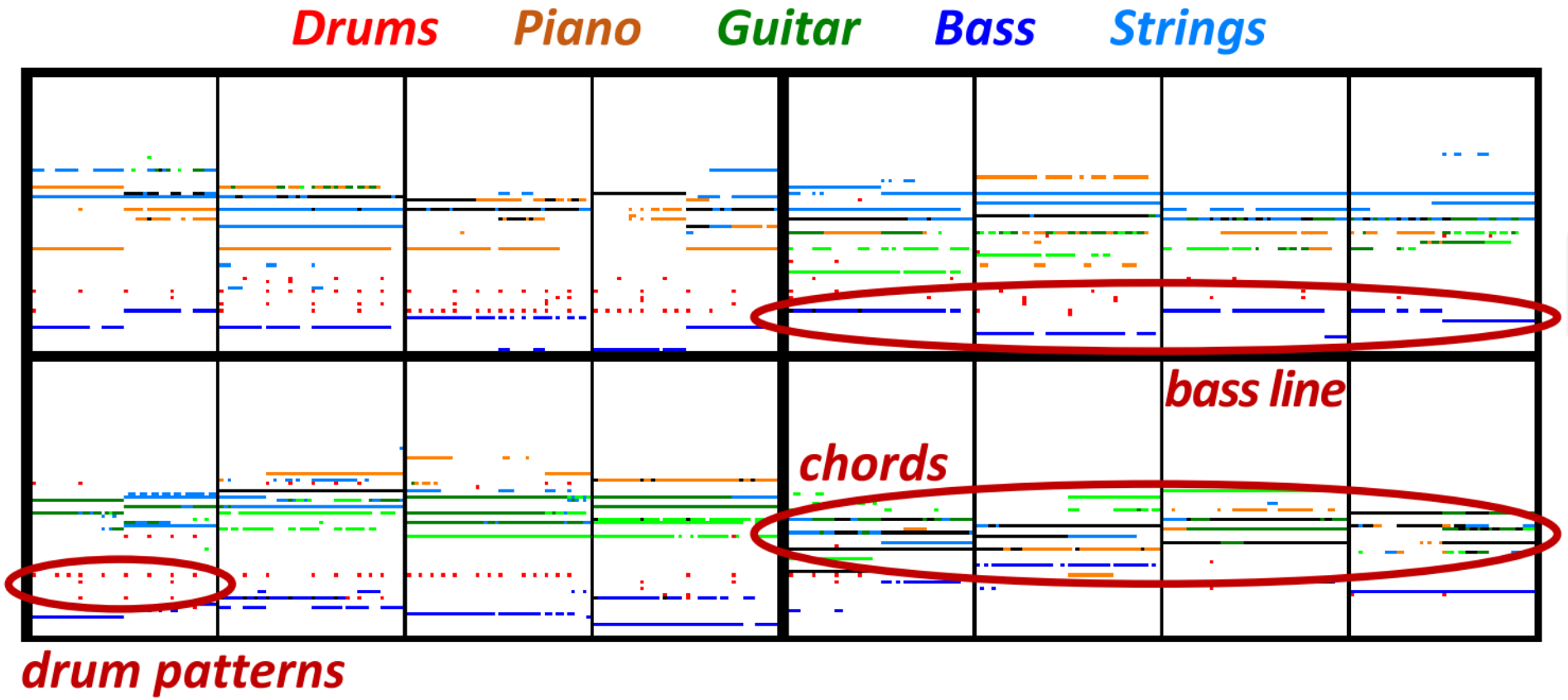
		Time	
		Dependent	Independent
Track	Dependent	**Melody**	**Groove**
	Independent	**Chords**	**Style**



Dong et al., "MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," *AAAI* 2018

MuseGAN—Samples

Sample 1 Sample 2



More samples can be found at https://salu133445.github.io/musegan/results

Dong et al., "MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," *AAAI* 2018

Before the coding session…

LPD (Lakh Pianoroll Dataset)

- 174,154 multi-track piano-rolls
- Derived from Lakh MIDI Dataset*
- Mainly pop songs
- Derived labels available

Pypianoroll (Python package)

- Manipulation & Visualization
- Efficient I/O
- Parse/Write MIDI files
- On PYPI (`pip install pypianoroll`)

We will use them in the next coding session!

[Lakh MIDI Dataset] https://colinraffel.com/projects/lmd/
[Pypianoroll] https://salu133445.github.io/pypianoroll
[Lakh Pianoroll Dataset] https://salu133445.github.io/lakh-pianoroll-dataset
Dong et al., "Pypianoroll: Open source Python package for handling multitrack pianorolls," *ISMIR-LBD* 2018.

Coding session II—GAN for pianorolls

Google Colab notebook link

https://colab.research.google.com/drive/1WrFtqo5LW8QfhiuhHmge9QLexWwS2BcM

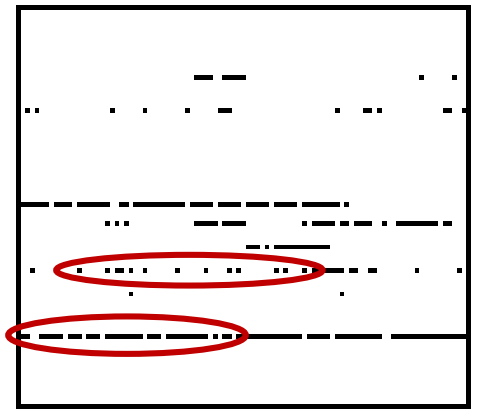
You can also find the link on the tutorial website

https://salu133445.github.io/ismir2019tutorial/

Click 

BinaryMuseGAN

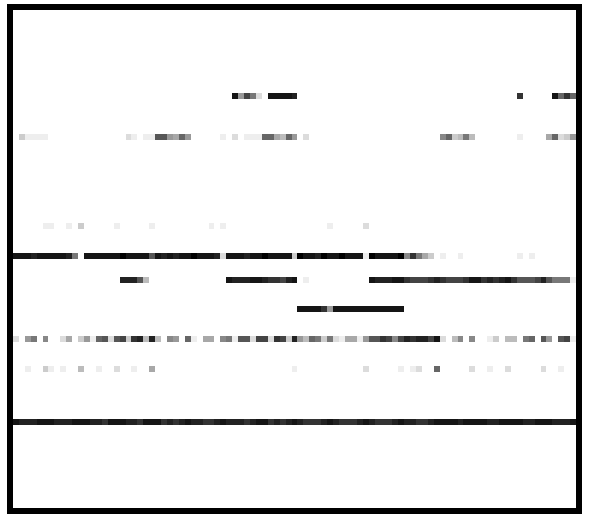
Key—Naïve binarization methods can easily lead to *overly-fragmented notes*



many overly-fragmented notes

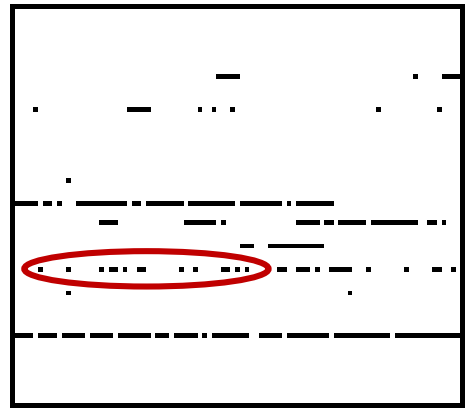
hard thresholding

MuseGAN's output

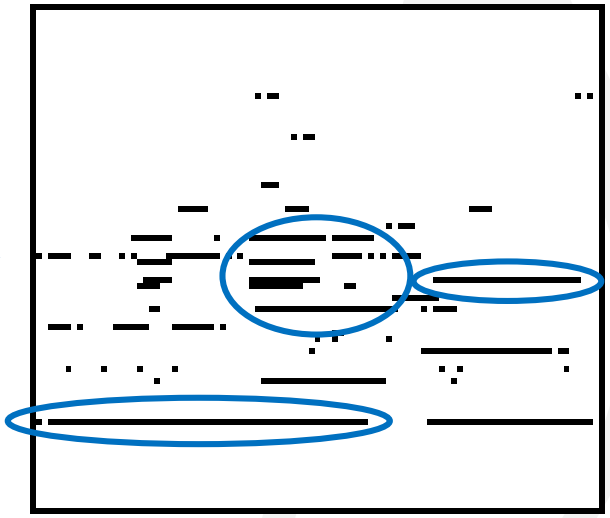


(real-valued)

Bernoulli sampling



BinaryMuseGAN (+DBNs)



less overly-fragmented notes

Dong and Yang, "Convolutional generative adversarial networks with binary neurons for polyphonic music generation," *ISMIR* 2018

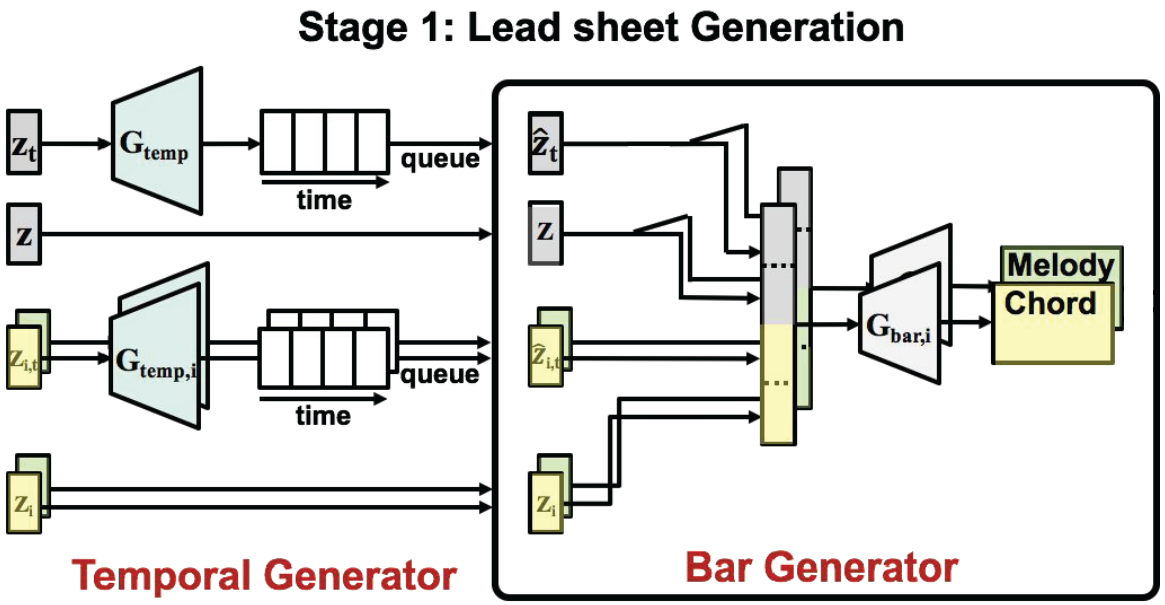
BinaryMuseGAN

- use *binary neurons* at the output layer of the generator
- use *straight-through estimator* to estimate the gradients for the binary neurons (which involves nondifferentiable operations)

	Generator's outputs	**Real data**
MuseGAN	***real-valued***	binary-valued
BinaryMuseGAN	binary-valued	binary-valued

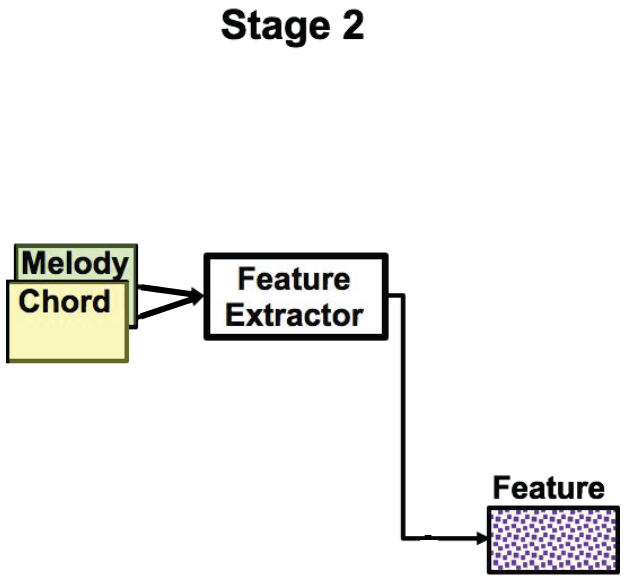
Dong and Yang, "Convolutional generative adversarial networks with binary neurons for polyphonic music generation," *ISMIR* 2018

LeadSheetGAN



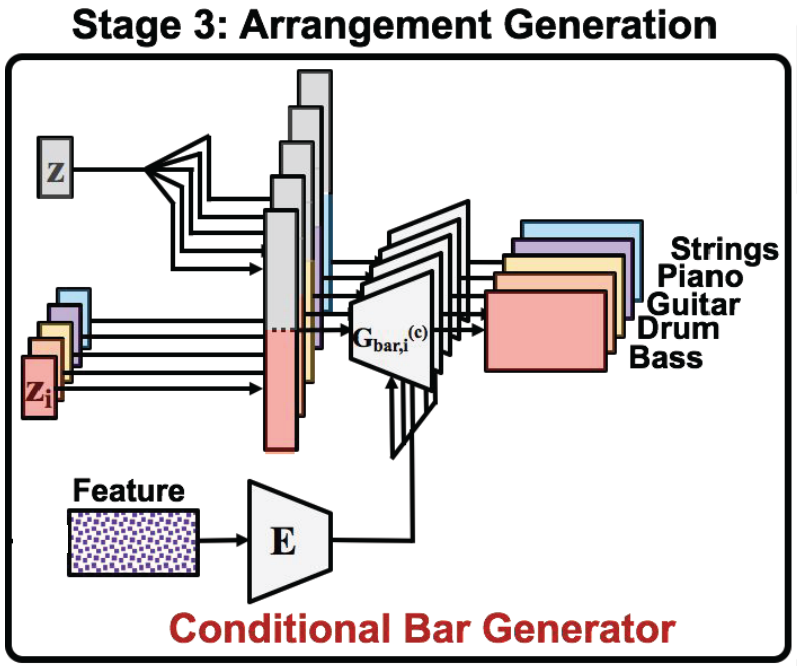
Liu and Yang, "Lead sheet generation and arrangement by conditional generative adversarial network," *ICMLA* 2018
Liu and Yang, "Lead sheet generation and arrangement via a hybrid generative model," *ISMIR-LBD* 2018

LeadSheetGAN



Liu and Yang, “Lead sheet generation and arrangement by conditional generative adversarial network,” *ICMLA* 2018
Liu and Yang, “Lead sheet generation and arrangement via a hybrid generative model,” *ISMIR-LBD* 2018

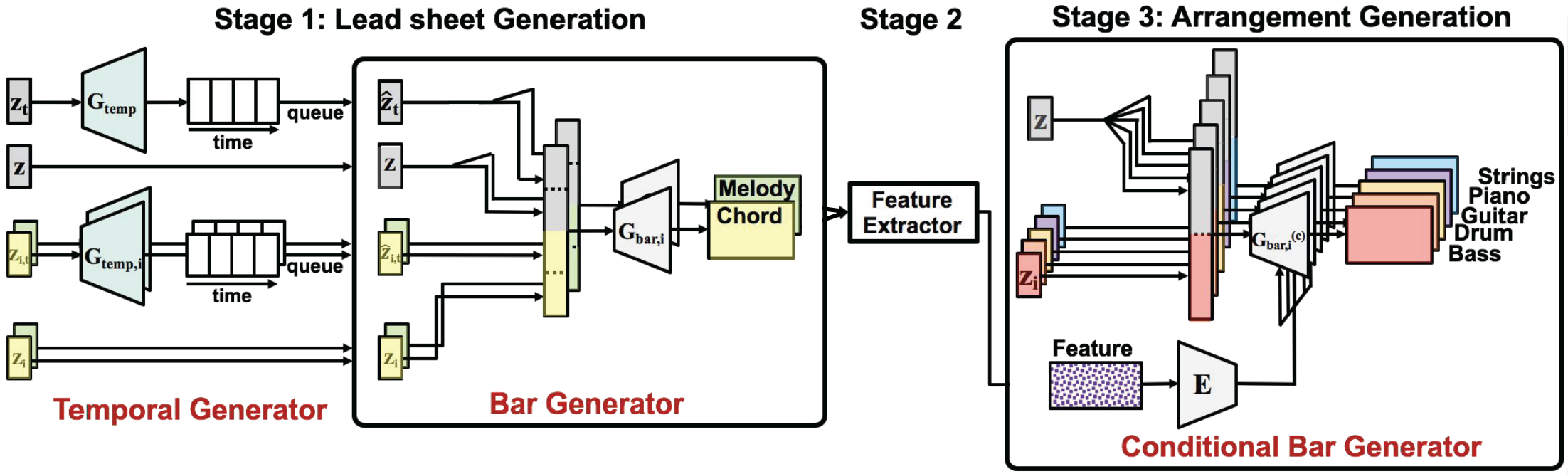
LeadSheetGAN



Liu and Yang, "Lead sheet generation and arrangement by conditional generative adversarial network," *ICMLA* 2018
Liu and Yang, "Lead sheet generation and arrangement via a hybrid generative model," *ISMIR-LBD* 2018

LeadSheetGAN

Key—First generate the lead sheet, then the arrangement



(Unconditional) GAN

Conditional GAN

Liu and Yang, "Lead sheet generation and arrangement by conditional generative adversarial network," *ICMLA* 2018
Liu and Yang, "Lead sheet generation and arrangement via a hybrid generative model," *ISMIR-LBD* 2018

LeadSheetGAN—Samples

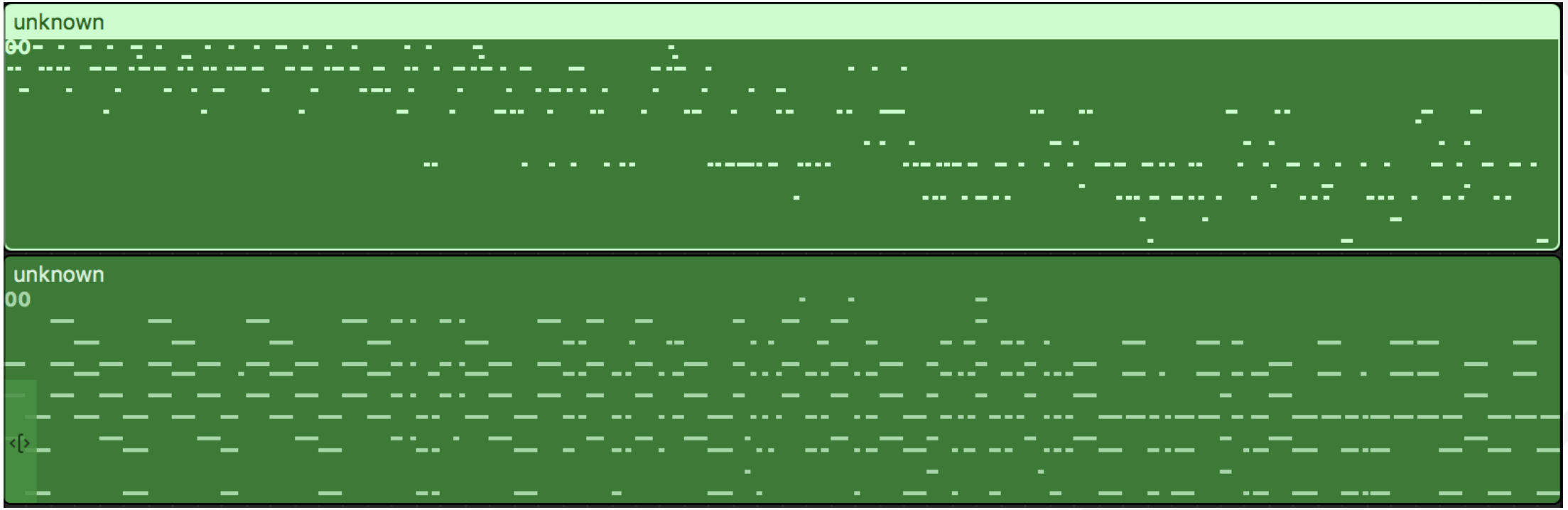
Marron 5

Payphone

latent space interpolation

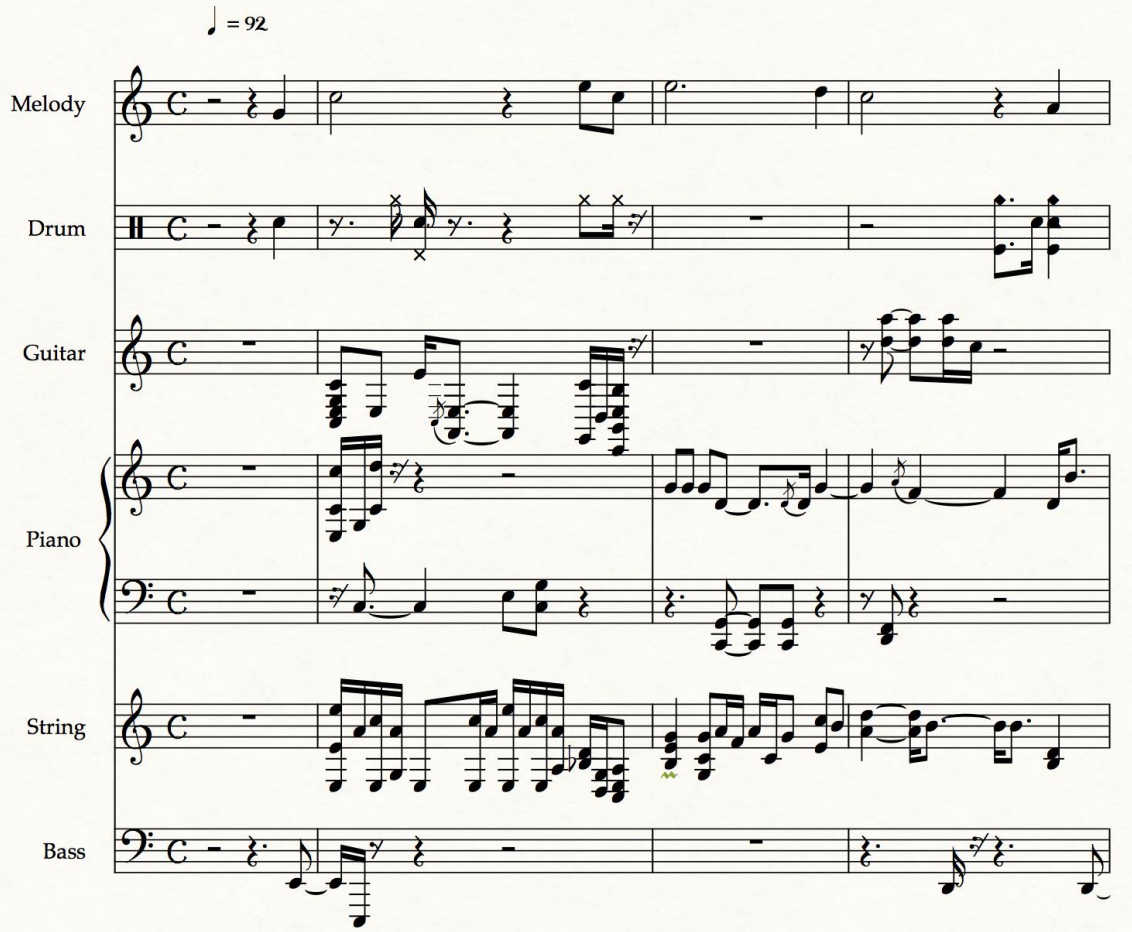
The Beatles

Hey Jude



Liu and Yang, “Lead sheet generation and arrangement by conditional generative adversarial network,” *ICMLA* 2018
Liu and Yang, “Lead sheet generation and arrangement via a hybrid generative model,” *ISMIR-LBD* 2018

LeadSheetGAN—Samples



Arrangement generation
given an "Amazing Graze" lead sheet

More samples can be found at https://liuhaumin.github.io/LeadsheetArrangement/results

Liu and Yang, "Lead sheet generation and arrangement by conditional generative adversarial network," *ICMLA* 2018

Tea Time!

Generating Music with GANs

Section 1

Overview of music generation research

Section 2

Introduction to GANs

Section 3

Case studies of GAN-based systems

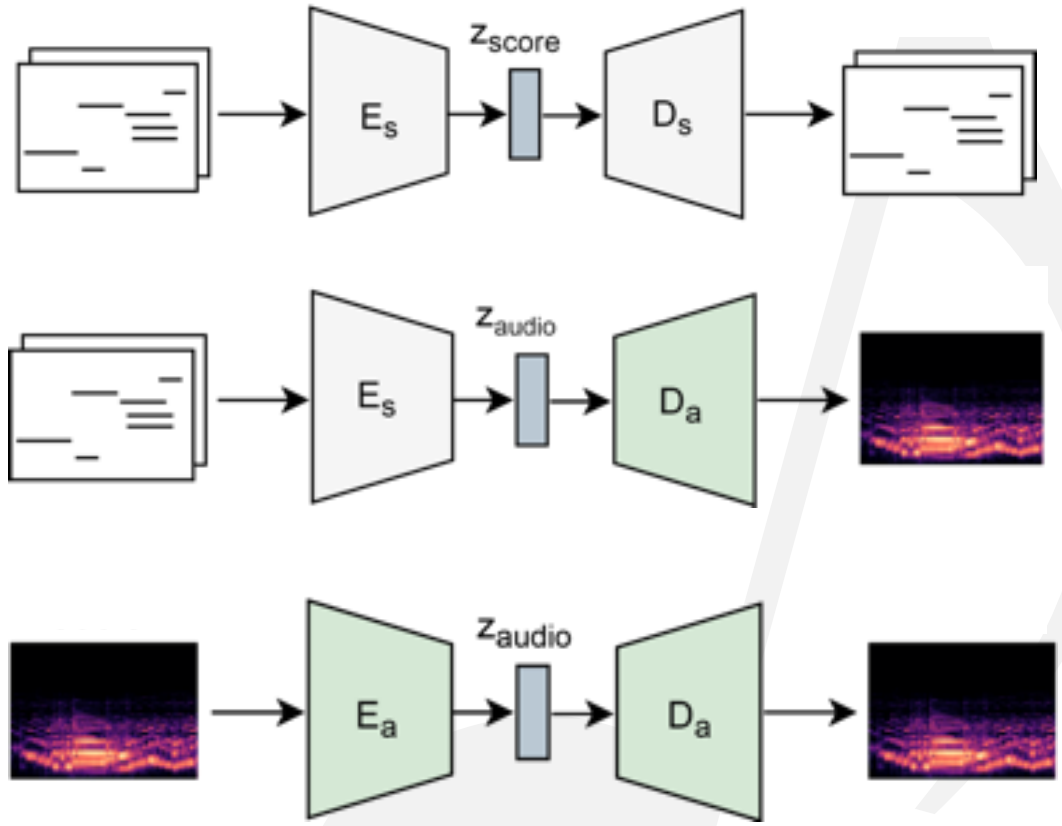
Section 4

Current limitations &

Future research directions

Scope of music generation

1. **Symbolic melody generation**
   - X → melody
2. **Arrangement generation**
   - X → piano roll
   - melody → piano roll
3. **Style transfer**
   - piano roll → piano roll'
   - audio → audio'
4. **Audio generation**
   - X → audio
   - piano roll → audio



CycleGANs for Music Style Transfer

piano roll → piano roll'
audio → audio'

Music style transfer

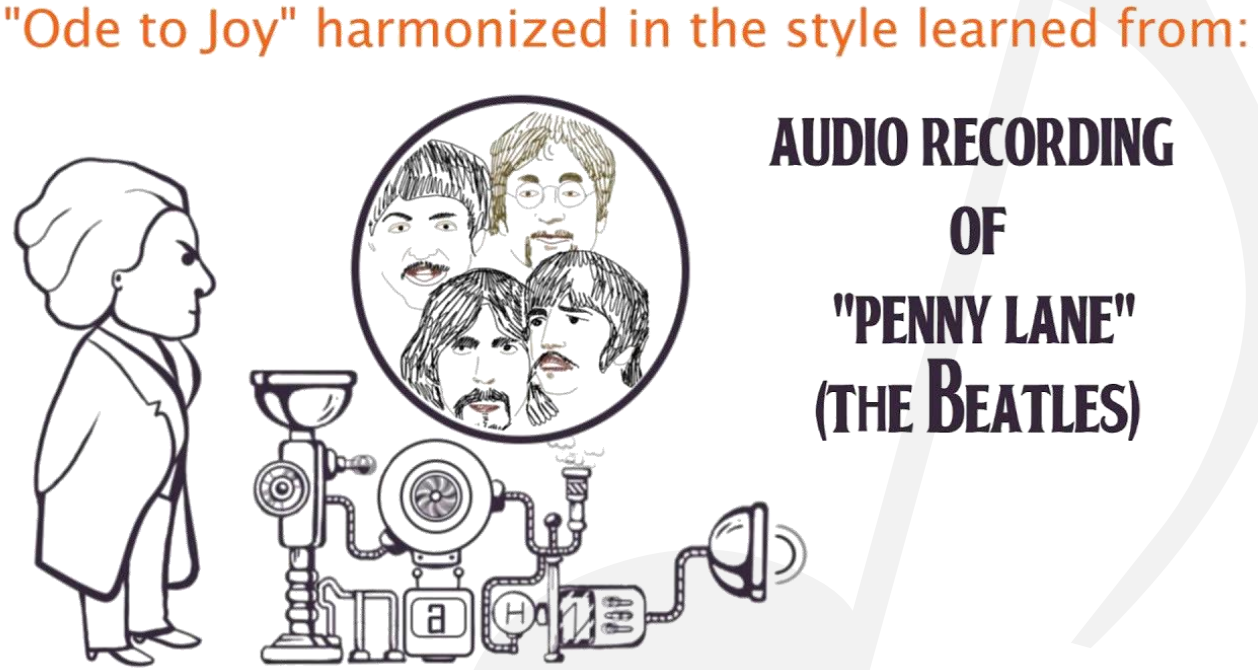
- Alter the "**style**," but keep the "**content**" fixed
- Three types of music style transfer [1]
  1. **composition style transfer** for score
  2. **performance style transfer** for performance control
  3. **timbre style transfer** for sound
- Little existing work on performance style transfer (e.g., [2]) uses deep learning

---

[1] Dai et al., "Music style transfer: A position paper," *MUME* 2018
[2] Shih et al., "Analysis and synthesis of the violin playing styles of Heifetz and Oistrakh," *DAFx* 2017

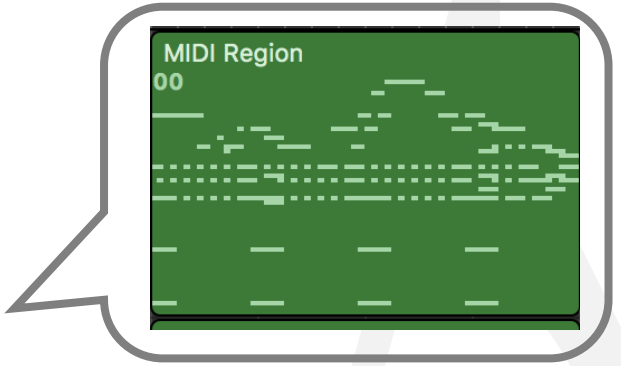
Composition style transfer (musical genre)

- Example: https://www.youtube.com/watch?v=buXqNqBFd6E
- Re-orchestrations of Beethoven's *Ode to Joy* by a collaboration between human and AI (Sony CSL Flow Machines)



Pachet, "A Joyful Ode to automatic orchestration," *ACM TIST* 2016

Composition style transfer (musical genre)



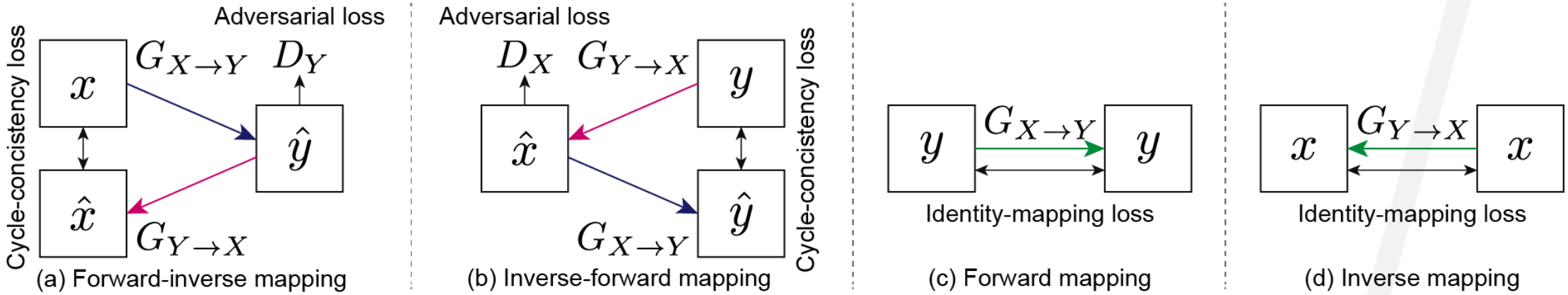
- Transfer among **Classic, Jazz, and Pop** [1,2]
- Model: standard convolutional **CycleGAN**
- I/O representation: **single-track piano roll** (64 x 84)
  - **merge all notes of all tracks (except for drums) into a single track**
  - discard drums
  - 7 octaves (C1-C8; hence 84 notes)
  - 4/4 time signature, 16 time steps per bar, 4 bars as a unit (hence 64 steps)
  - 10k+ four-bar phrases for each genre (**no paired data**)

---

[1] Brunner et al., "Symbolic music genre transfer with CycleGAN," *ICTAI* 2018
[2] Brunner et al., "Neural symbolic music genre transfer insights," *MML* 2019

Recap—CycleGAN

- Adversarial loss + identity mapping loss ("cycle consistency")



Zhu et al., "Unpaired image-to-image translation using cycle-consistent adversarial networks," *ICCV* 2017

Composition style transfer (musical genre)

- https://www.youtube.com/channel/UCs-bI_NP7PrQaMV1AJ4A3HQ

- Possible extensions:
  - Consider different voices (including drums) separately (instead of merging them)
  - Add recurrent layers to better model sequential information
  - Identify the melody line [1] and take better care of it

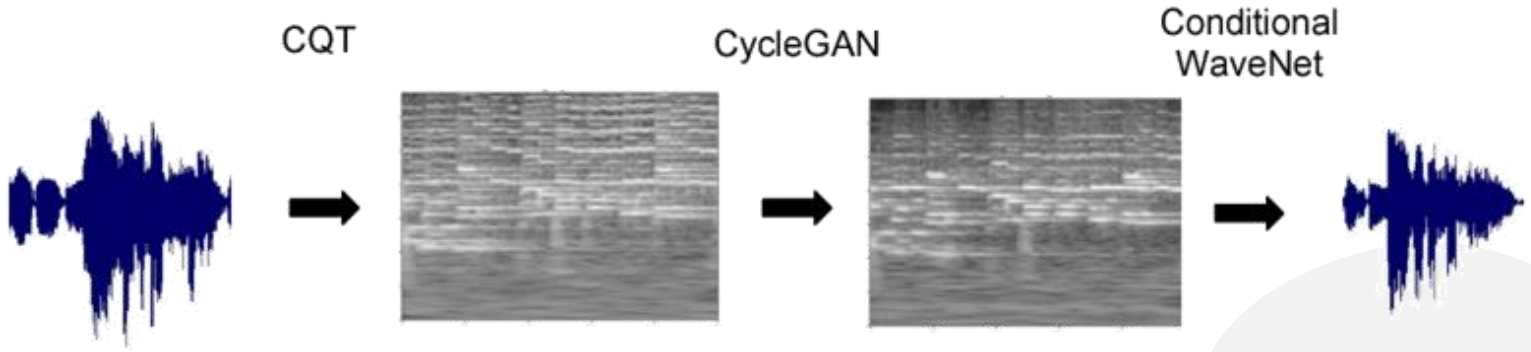
- Related:
  - Supervised genre style transfer (not using GANs) using synthesized data [2]

---

[1] Simonetta et al., "A Convolutional approach to melody line identification in symbolic scores," *ISMIR* 2019
[2] Cífka et al., "Supervised symbolic music style translation using synthetic data," *ISMIR* 2019

Timbre style transfer (instrumental sounds): TimbreTron

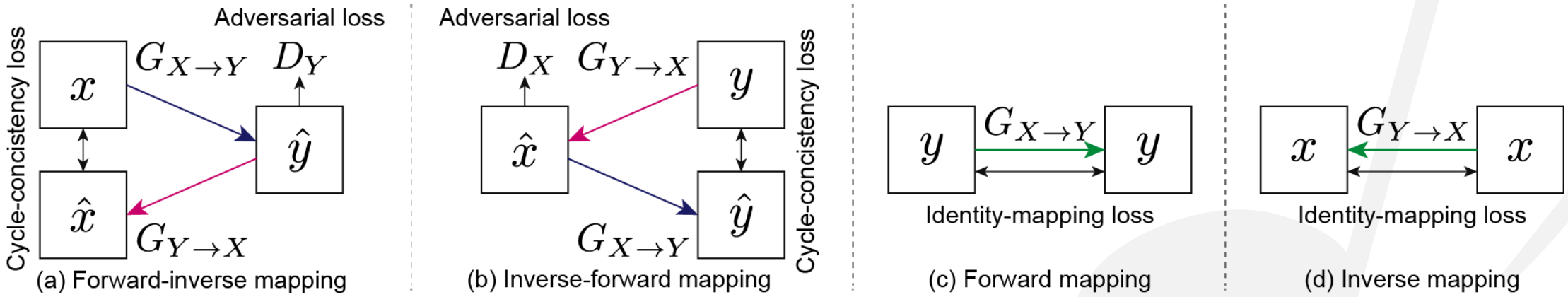
- Transfer among **piano**, **flute**, **violin**, **harpsichord** solos
  https://www.cs.toronto.edu/~huang/TimbreTron/samples_page.html
- Model: modified version of **CycleGAN**
- I/O representation: 4-second **CQT** (257 x 251)



Huang et al., "TimbreTron: A WaveNet(CycleGAN(CQT(Audio))) pipeline for musical timbre transfer," *ICLR* 2019

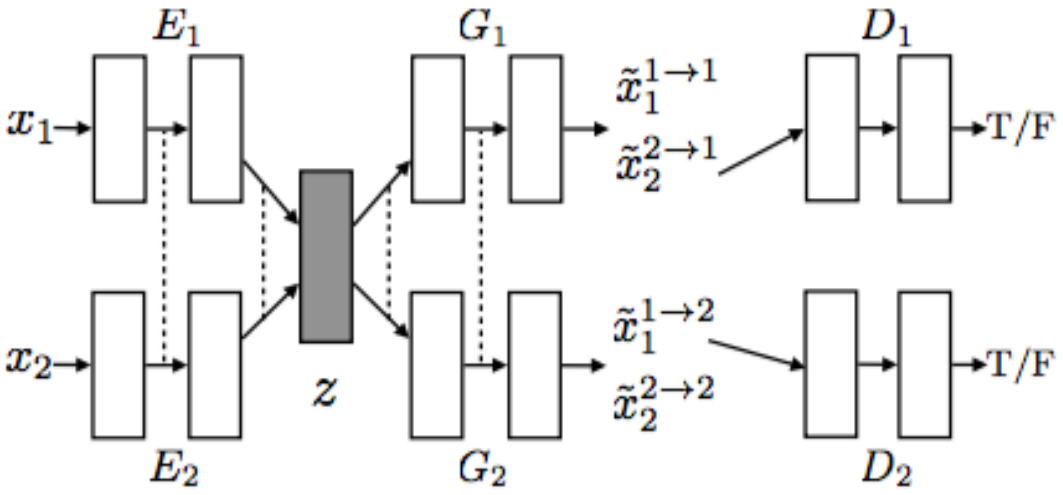
Drawback of cycleGAN

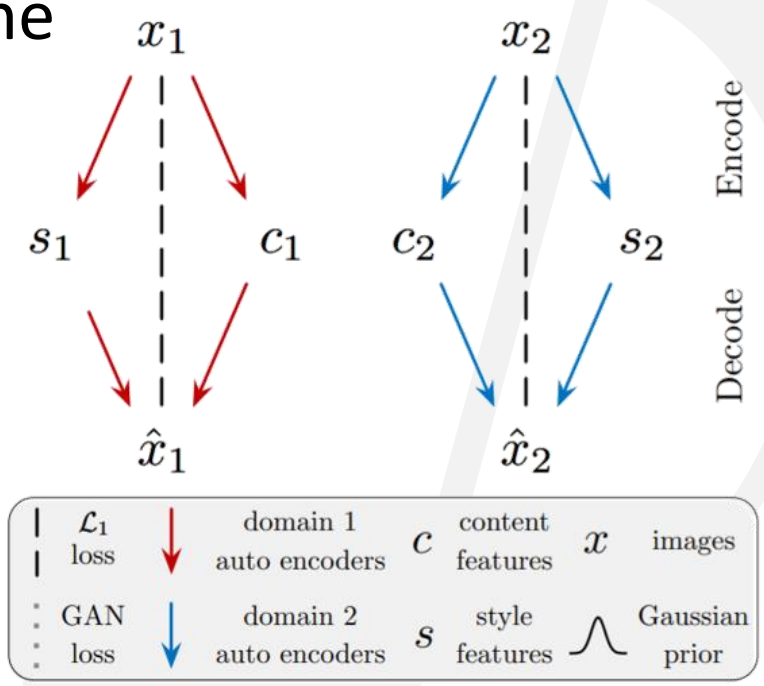
- Can work for only **two domains at a time**
  - For example, piano↔flute, paion↔violin, piano↔harpsichord, etc



MUNIT instead of cycleGAN

- **MUNIT** [1]: an advanced version of cycleGAN that incorporates encoders/decoders to get **disentangled "content" and "style" codes**
- Can work for **multiple domains** at the same time

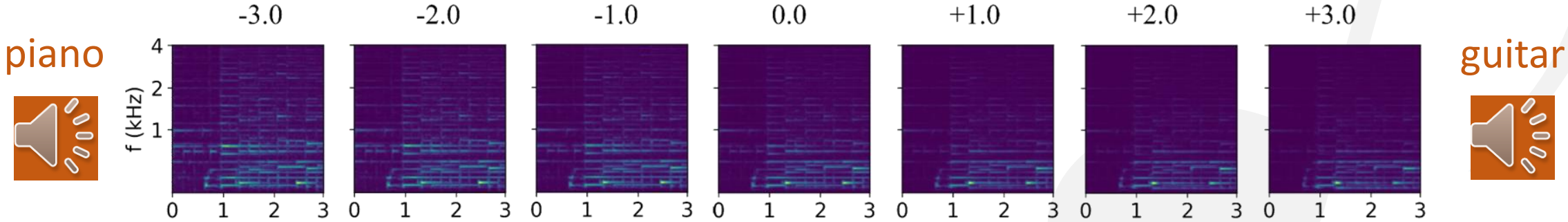




[1] Huang et al., "Multimodal unsupervised image-to-image translation," *ECCV* 2018

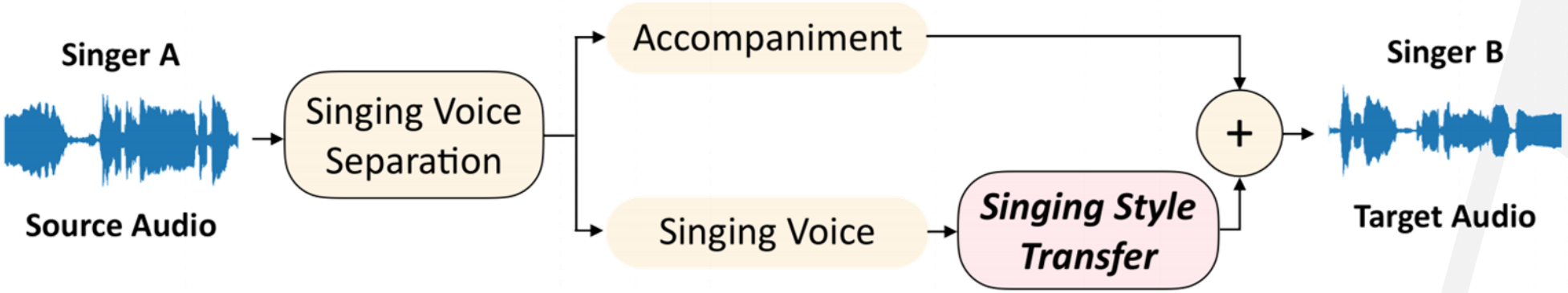
Timbre style transfer (instrumental sounds): Play-as-you-like

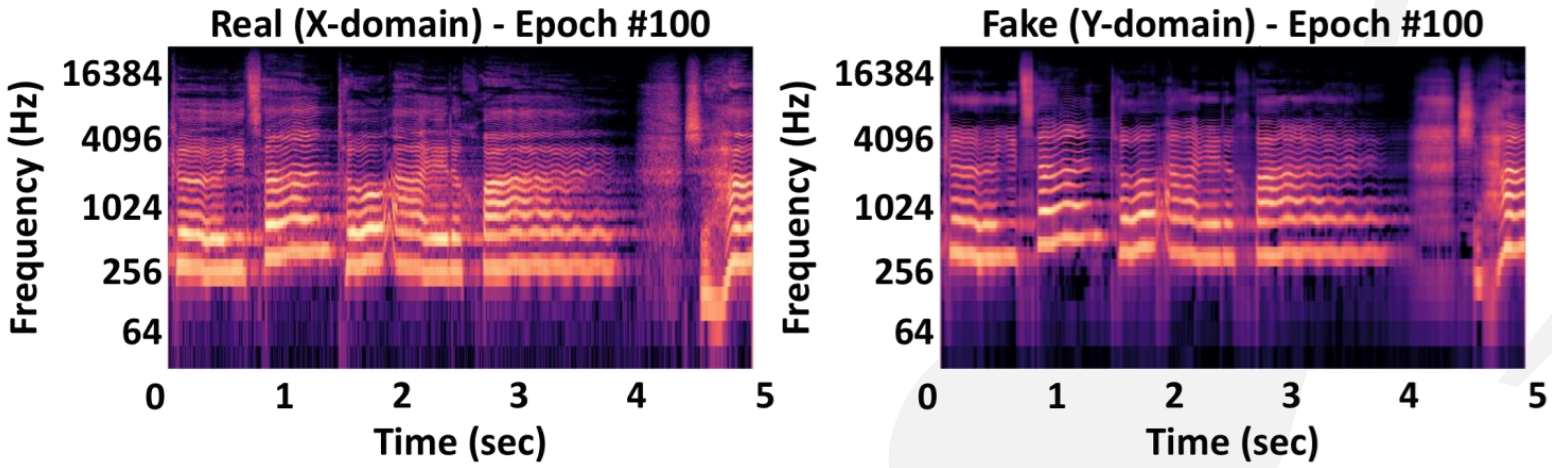
- Transfer among **piano**, **guitar** solos, and **string quartet** (https://tinyurl.com/y23tvhjx)
- Use MUNIT
- I/O representation: Mel-spectrogram + spectral difference + MFCC + spectral envelope
  - **Cycle consistency among the channel-wise features** (as regularizers)
- Style interpolation: https://soundcloud.com/affige/sets/ismir2019-gan-tutorial-supp-material



Lu et al., "Play as You Like: Timbre-enhanced multi-modal music style transfer," *AAAI* 2019

Timbre style transfer (singing): CycleBEGAN





Wu et al., "Singing style transfer using cycle-consistent boundary equilibrium generative adversarial networks," *ICML workshop* 2018

Timbre style transfer (singing): CycleBEGAN

- Transfer between female and male singing voices [1]
  http://mirlab.org/users/haley.wu/cybegan/ (check the outside test result)

- Model: **BEGAN** instead of GAN [2]
  - train the generator *G* such that the discriminator *D* (an encoder/decoder net) would reconstruct fake data as nicely as real data
  - have a mechanism to balance the power of *G* and *D*

  + **skip connections** (also used in [3]) + a **recurrent layer**

---

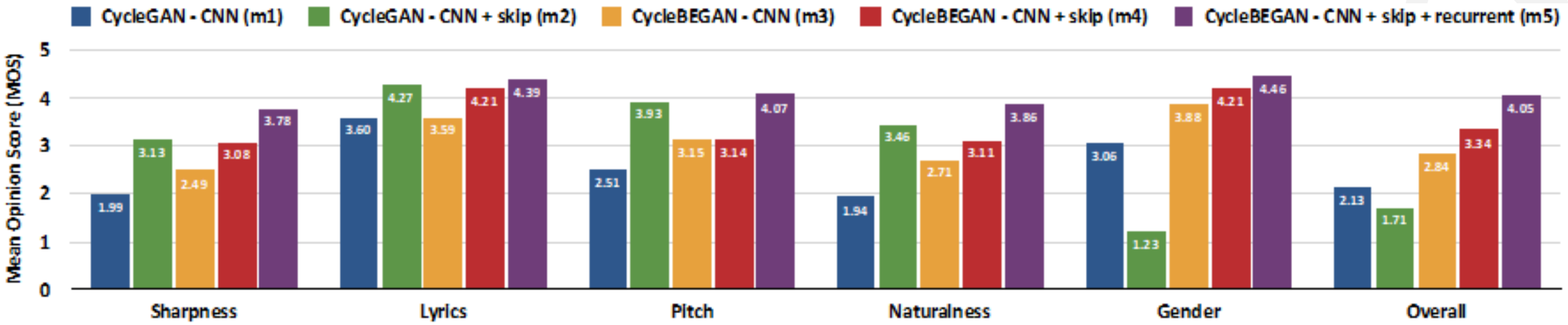
[1] Wu et al., "Singing style transfer using cycle-consistent boundary equilibrium generative adversarial networks," *ICML workshop* 2018

[2] Berthelot et al., "BEGAN: Boundary equilibrium generative adversarial networks," *arXiv* 2017

[3] Hung et al., "Musical composition style transfer via disentangled timbre representations," *IJCAI* 2019

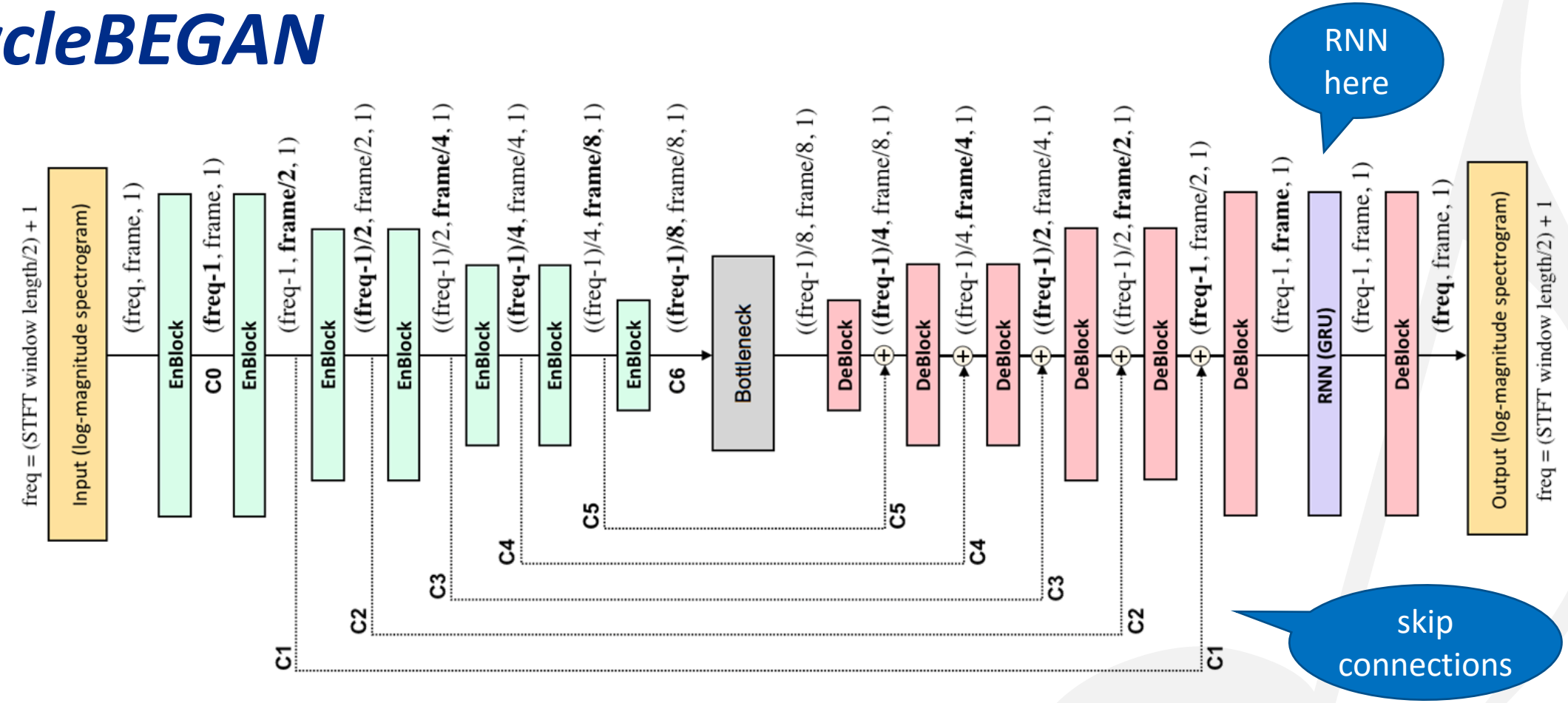
Timbre style transfer (singing): CycleBEGAN

- **Skip connections** contribute to sharpness, lyrics intelligibility, and naturalness
- **Recurrent layers** further improves everything, especially pitch accuracy



Wu et al., "Singing style transfer using cycle-consistent boundary equilibrium generative adversarial networks," *ICML workshop* 2018

Timbre style transfer (singing): CycleBEGAN



Also use an encoder/decoder architecture for the generator

"Singing style transfer using cycle-consistent boundary equilibrium generative adversarial networks," *ICML works.* '18

GANs for Music Audio Generation

X → audio

piano roll → audio

Generating instrument sounds using GANs

- Generate spectrograms
  - SpecGAN [1], TiFGAN [2], GANSynth [3]
- Generate waveforms
  - WaveGAN [1]
- There are also approaches that do not use GANs [4, 5, 6, 7]

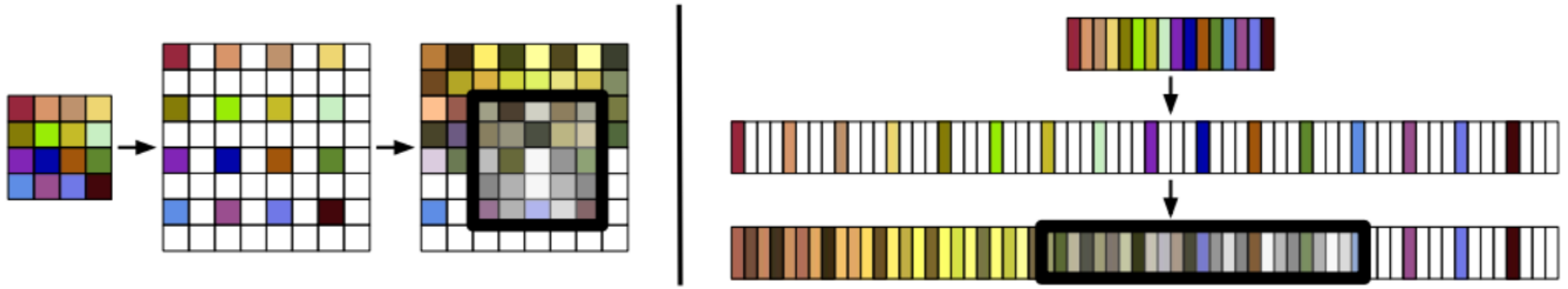
---

[1] Donahue et al., “Adversarial audio synthesis,” *ICLR* 2019
[2] Marafioti et al., “Adversarial generation of time-frequency features,” *ICML* 2019
[3] Engel et al., “GANSynth: Adversarial neural audio synthesis,” *ICLR* 2019
[4] Oord et al., “WaveNet: A generative model for raw audio,” *SSW* 2016
[5] Défossez et al., “SING: Symbol-to-instrument neural generator,” *NeurIPS* 2018
[6] Schimbinschi et al., “SynthNet: Learning to synthesize music end-to-end,” *IJCAI* 2019
[7] “PerformanceNet Score-to-audio music generation with multi-band convolutional residual network,” *AAAI* 2019

SpecGAN and WaveGAN

- Generating **1-second** audio at 16kHz (16,000 points)
  https://chrisdonahue.com/wavegan/

- Model: based on **DCGAN**
  - Flatten 2D convolutions into 1D (e.g., 5x5 2D convolution becomes length-25 1D)
  - Increase the stride factor for all convolutions (e.g., stride 2x2 becomes stride 4)
  - DCGAN outputs 64x64 images; add one more layer so that the output has 16,384 points

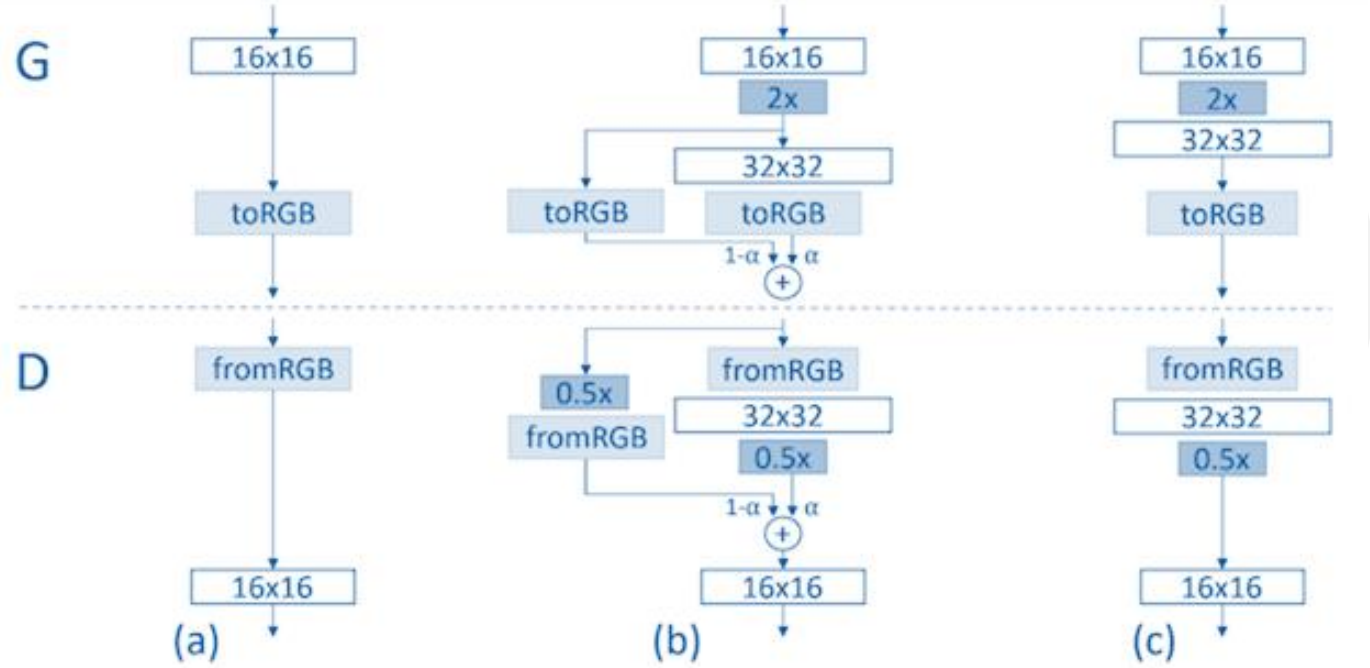


Donahue et al., "Adversarial audio synthesis," *ICLR* 2019

GANSynth

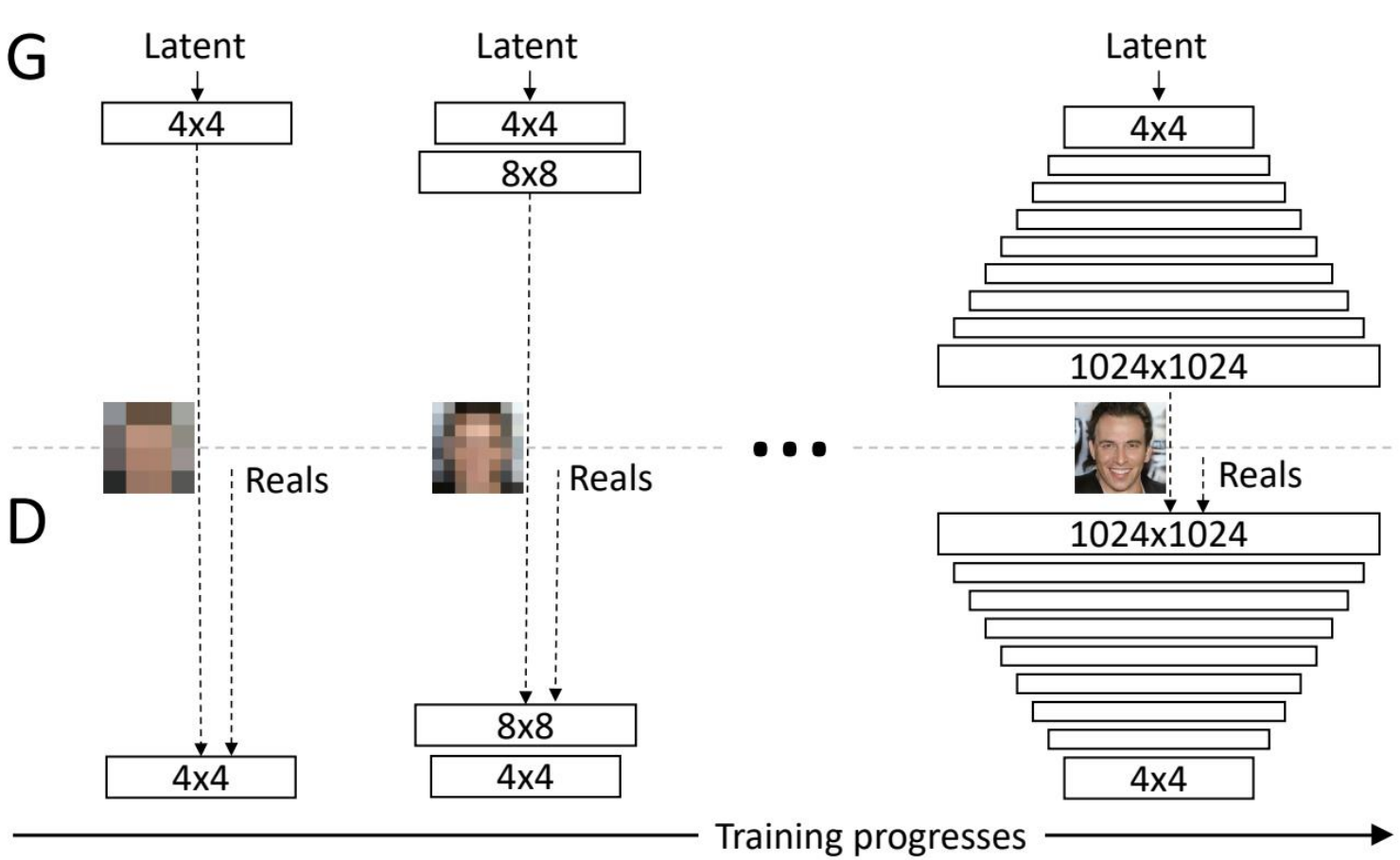
- Generating **4-second** audio at 16kHz [1]: large output dimension
  https://magenta.tensorflow.org/gansynth

- Model: based on **PGGAN** [2]
  - Progressively grow the GAN, **from low to high resolution**
  - During a resolution transition, interpolate between the output of two resolutions, with weight α linearly increasing from 0 to 1



[1] Engel et al., "GANSynth: Adversarial neural audio synthesis," *ICLR* 2019
[2] Karras et al., "Progressive growing of GANs for improved quality, stability, and variation," *ICLR* 2018

PGGAN: progressive growing of GANS



Karras et al., "Progressive growing of GANs for improved quality, stability, and variation," *ICLR* 2018

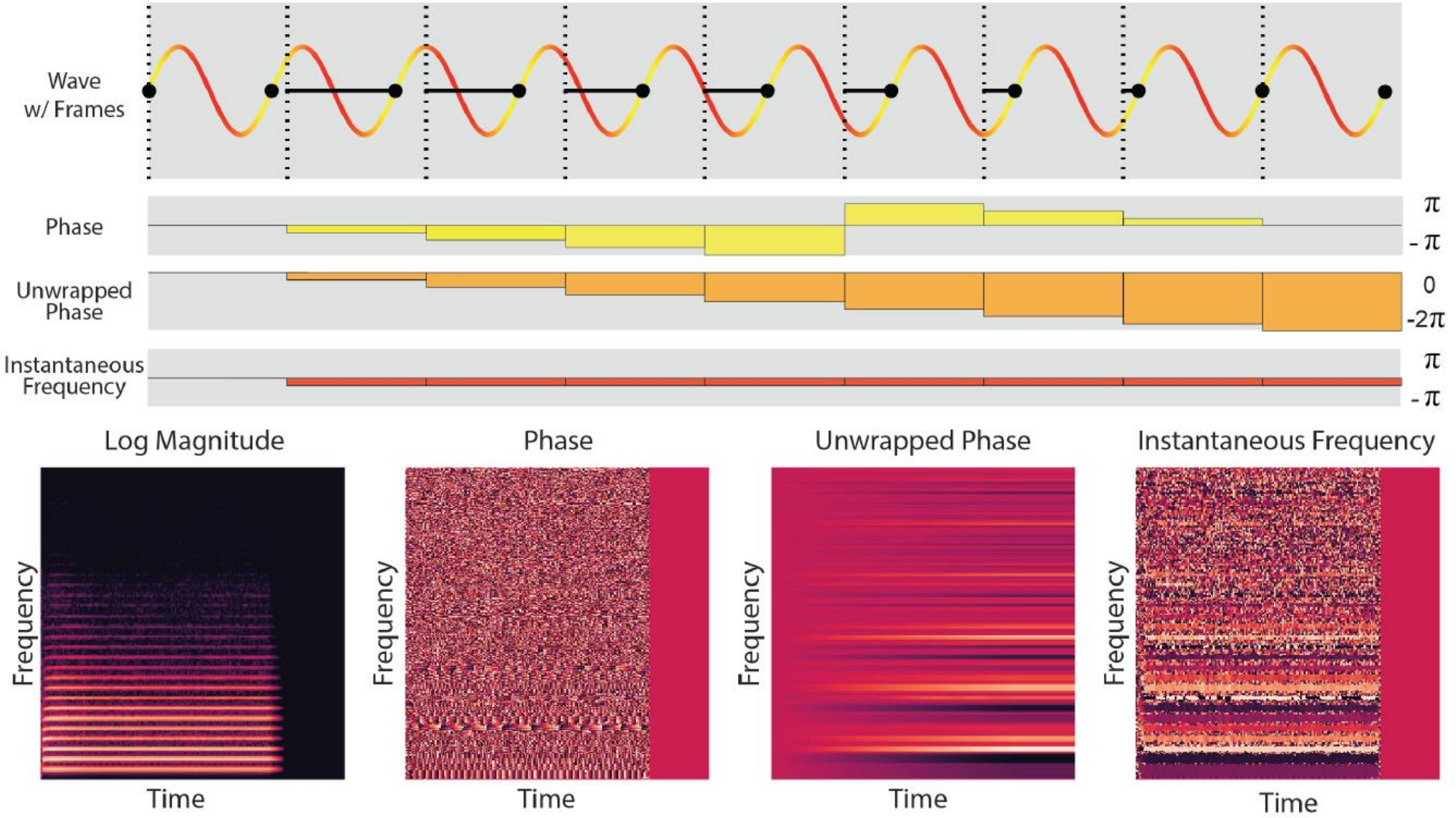
GANSynth

- Generating **4-second** audio at 16kHz
  https://magenta.tensorflow.org/gansynth
- Model: based on **PGGAN** (**conv2d**)
- Output: **mel-spectrogram** + **instantaneous freq (IF)**
  - Use IF to derive the phase, and then use inverse STFT to get the waveform
  - STFT window size 2048, stride 512 (so, about 128 frames)
  - 1024-bin mel-frequency scale
  - Target output tensor size: **128 x 1024 x 2**
  - (2x16 → 4x32 → … → 128x1024)

Generator	Output Size
concat(Z, Pitch)	(1, 1, 317)
conv2d	(2, 16, 256)
conv2d	(2, 16, 256)
upsample 2x2	(4, 32, 256)
conv2d	(4, 32, 256)
conv2d	(4, 32, 256)
upsample 2x2	(8, 64, 256)
conv2d	(8, 64, 256)
conv2d	(8, 64, 256)
upsample 2x2	(16, 128, 256)
conv2d	(16, 128, 256)
conv2d	(16, 128, 256)
upsample 2x2	(32, 256, 256)
conv2d	(32, 256, 128)
conv2d	(32, 256, 128)
upsample 2x2	(64, 512, 128)
conv2d	(64, 512, 64)
conv2d	(64, 512, 64)
upsample 2x2	(128, 1024, 64)
conv2d	(128, 1024, 32)
conv2d	(128, 1024, 32)
generator output	(128, 1024, 2)

Engel et al., "GANSynth: Adversarial neural audio synthesis," *ICLR* 2019

GANSynth: Why instantaneous frequency?

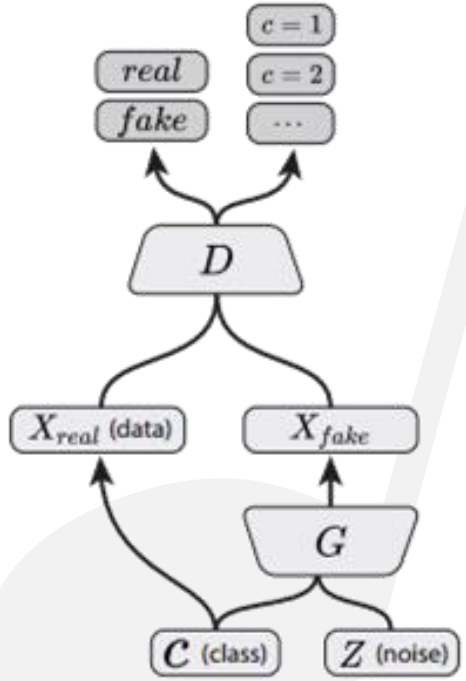


GANSynth: Pitch-conditioned generation

- Input to the generator
  - 256-dim random vector Z
  - 61-dim **one-hot vector** (MIDI 24-84) for **pitch conditioning**

- Auxiliary **pitch classification loss** for the discriminator (ACGAN [1])
  - In addition to the real/fake loss
  - Try to predict the pitch label

Generator	Output Size
concat(Z, Pitch)	(1, 1, 317)
conv2d	(2, 16, 256)



AC-GAN

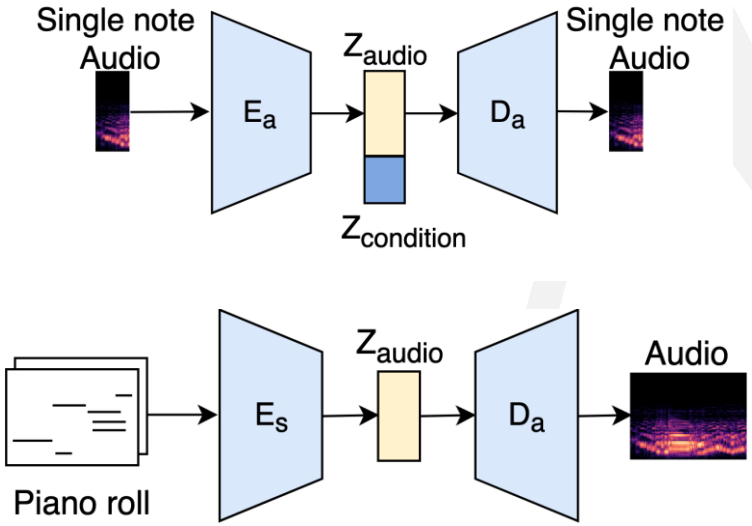
[1] Odena et al., "Conditional image synthesis with auxiliary classifier GANs," *ICML* 2017

GANSynth: Possible future extensions

- Modify the network to generate variable-length audio
- From **note-level synthesis**
  (i.e., condition on a **one-hot pitch vector**);

→ to **phrase-level syntehsis**
(i.e., condition on a **piano roll**) [1,2]

  - The IF might be noisy during note transitions
  - May need to deal with the use of playing techniques [3]



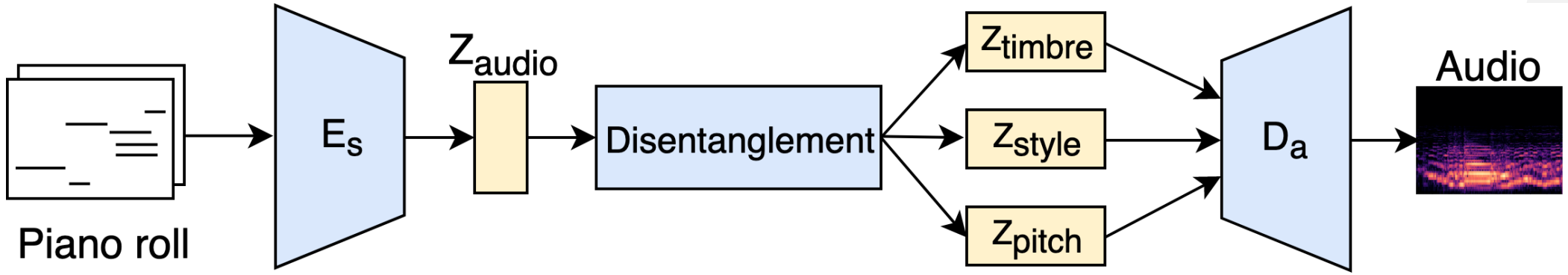
[1] Wang and Yang, "PerformanceNet: Score-to-audio music generation with multi-band convolutional residual network," *AAAI* 2019

[2] Chen et al., "Demonstration of PerformanceNet: A convolutional neural network model for score-to-audio music generation," *IJCAI demo paper,* 2019

[3] Su et al., "TENT: Technique-embedded note tracking for real-world guitar solo recordings," *TISMIR* 2019

PerformanceNet: Possible future extensions

- Building an " **AI Performer**" [1,2,3]



[1] Wang and Yang, "PerformanceNet: Score-to-audio music generation with multi-band convolutional residual network," *AAAI* 2019

[2] Chen et al., "Demonstration of PerformanceNet: A convolutional neural network model for score-to-audio music generation," *IJCAI demo paper,* 2019

[3] Oore et al., "This time with feeling: Learning expressive musical performance," *arXiv* 2018

Outline

Section 1

Overview of music generation research

Section 2

Introduction to GANs

Section 3

Case studies of GAN-based systems

Section 4

Current limitations

Future research directions

Limitations of GANs

- A bit difficult to train (it's helpful to know some tips and tricks [1])
- Only learn z→X mapping, not the inverse (X→z)
- Less explainable [2]
- Less clear how GANs can model text-like data or musical scores
- Unclear how GANs (and all other music composition models in general) can generate “new” music genres

[1] “How to Train a GAN? Tips and tricks to make GANs work,” https://github.com/soumith/ganhacks
[2] Kelz and Widmer, “Towards interpretable polyphonic transcription with invertible neural networks,” *ISMIR* 2019

The many other generative models

- Variational autoencoders (VAEs)
- Flow-based models
- Autoregressive models
- Attention mechanisms (transformers)
- Restricted Boltzmann machines (RBMs)
- Hidden Markov models (HMMs)
- *…and more!*

Future research directions

- Better network architectures for musical data, including piano rolls, MIDI events, musical scores, and audio
- Learning to generate music with better structure and diversity
- Better interpretability and human control (e.g., [1])
- Standardized test dataset and evaluation metrics [2]
- Cross-modal generation, e.g., “music + lyrics” or “music + video”
- Interactive music generation (e.g., [3])

---

[1] Lattner and Grachten, “High-level control of drum track generation using learned patterns of rhythmic interaction,” *WASPAA* 2019

[2] Katayose et al., “On evaluating systems for generating expressive music performance: the Rencon experience,” *Journal of New Music Research* 2012

[3] Hsiao et al., “Jamming with Yating: Interactive demonstration of a music composition AI,” *ISMIR-LBD* 2019

Future research directions

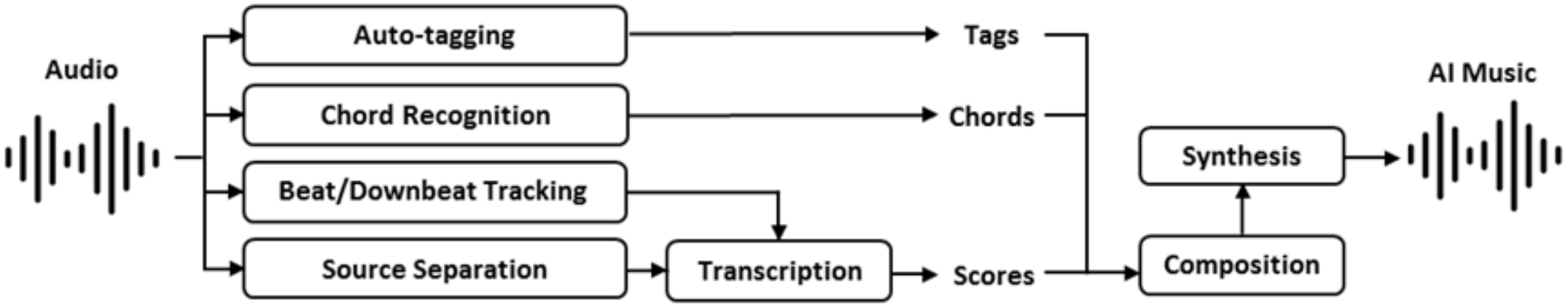
- Composition style transfer (musical genre) using the LPD dataset
- More work on performance style transfer
- Phrase-level audio generation instead of note-level synthesis [1]
- Multi-singer audio generation and style transfer
- Lyrics-free singing generation [2]
- EDM generation [3,4]

---

[1] “Demonstration of PerformanceNet: A convolutional neural network model for score-to-audio music generation,” *IJCAI* demo paper, 2019
[2] “Score and lyrics-free singing voice generation,” *ICLR* 2020 submission
[3] “DJnet: A dream for making an automatic DJ,” *ISMIR-LBD* 2017
[4] “Unmixer: An interface for extracting and remixing loops,” *ISMIR* 2019

Future research directions

- The “MIR4generation” pipeline
  - Learning to generate music (that is expressive) from machine-transcribed data (i.e., learning to compose and perform at the same time)



Yeh et al., “Learning to generate Jazz and Pop piano music from audio via MIR techniques,” *ISMIR-LBD* 2019

Thank you!
Any questions?

Contact

Hao-Wen Dong (salu.hwdong@gmail.com)
Yi-Hsuan Yang (yang@citi.sinica.edu.tw)

Tutorial website

salu133445.github.io/ismir2019tutorial/

中央研究院
ACADEMIA SINICA

中央研究院
資訊科技創新研究中心
Research Center for Information Technology Innovation
Academia Sinica

音樂與人工智慧實驗室
Music and Artificial Intelligence Lab

UC San Diego
JACOBS SCHOOL OF ENGINEERING

AI Labs.tw
台灣人工智慧實驗室