

Deep Learning Techniques for Music Generation

Compound and GAN (6)

Jean-Pierre Briot

`Jean-Pierre.Briot@lip6.fr`

Laboratoire d'Informatique de Paris 6 (LIP6)
Sorbonne Université – CNRS



Programa de Pós-Graduação em Informática (PPGI)
UNIRIO



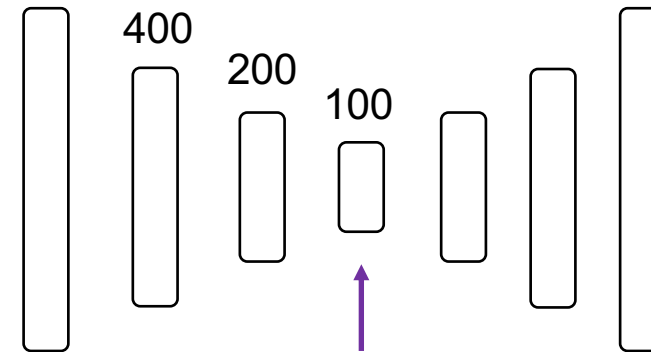
Architectures

Architectures

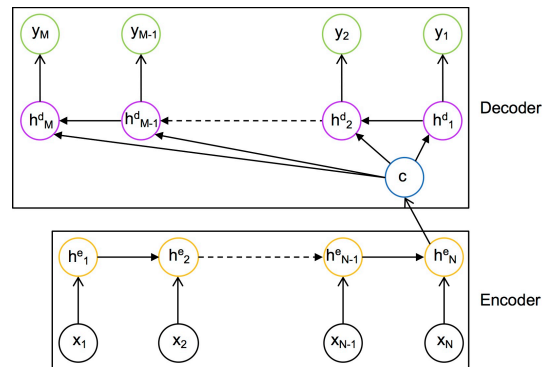
- *Feedforward* *mini-bach.py*
- *Autoencoder* *auto-bach.py*
 - *Variational Autoencoder (VAE)* *VRAE*
- *Recurrent (RNN)*
 - *LSTM* *lstm.py, Celtic*
- Generative Adversarial Networks (GAN)
- Restricted Boltzmann Machine (RBM)
- Reinforcement Learning (RL)

Compound Architectures

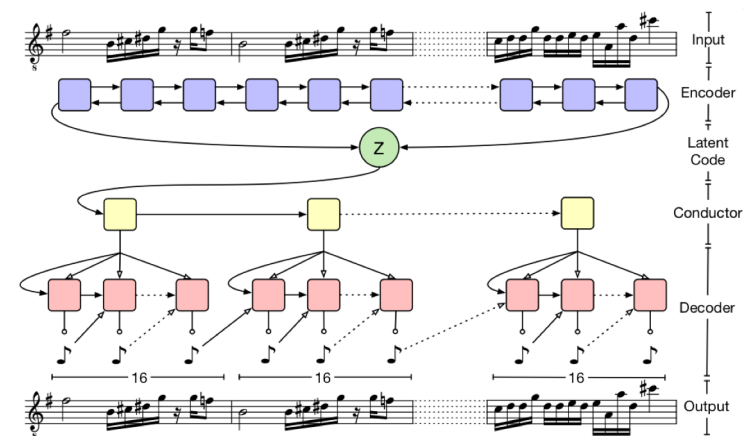
- Autoencoder Stack = Autoencoderⁿ
 - DeepHear, auto-bach.py



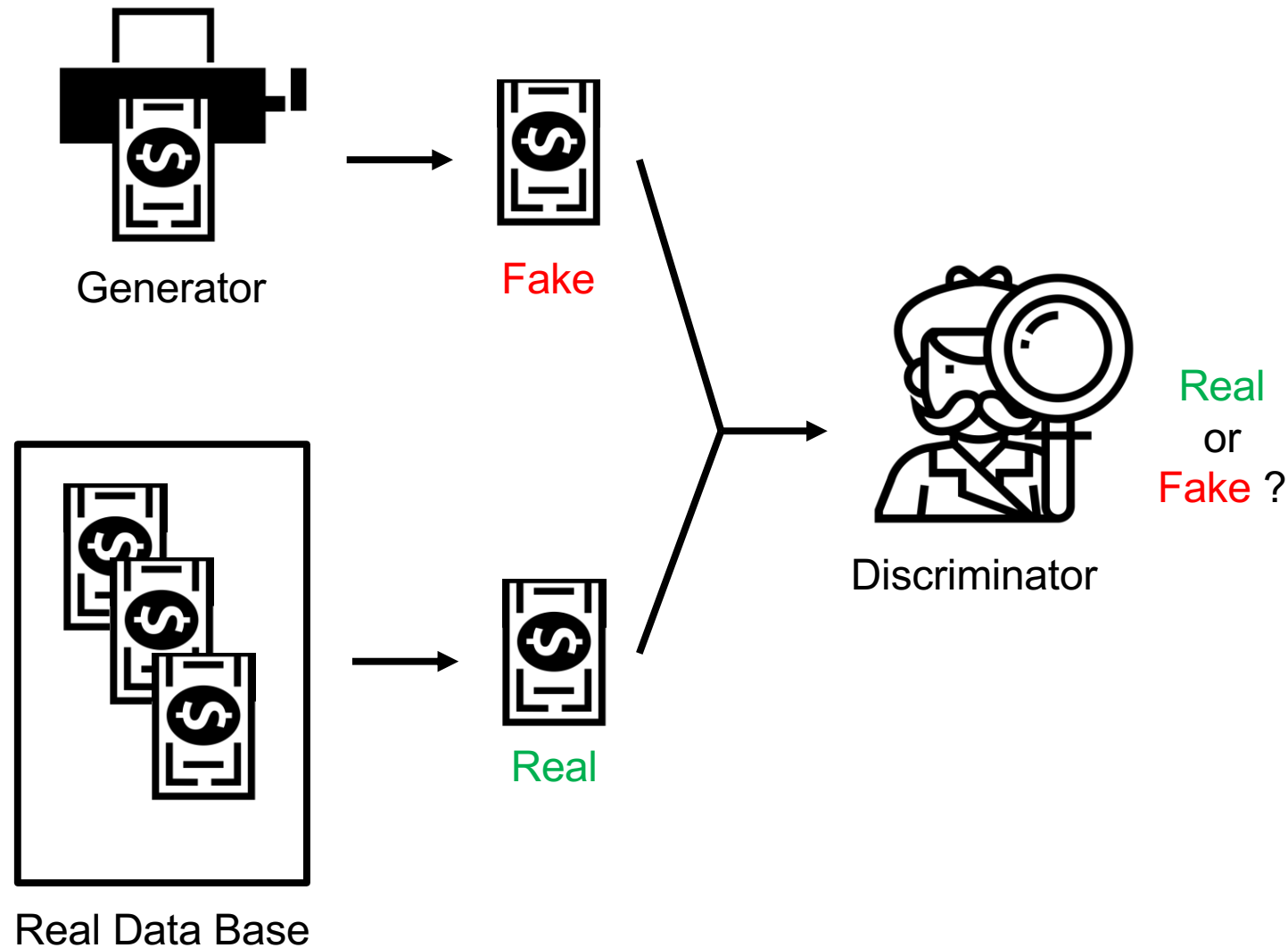
- Autoencoder(RNN, RNN) = RNN Encoder-Decoder
 - VRAE



- RNN Variational Encoder-Decoder
 - Music-VAE



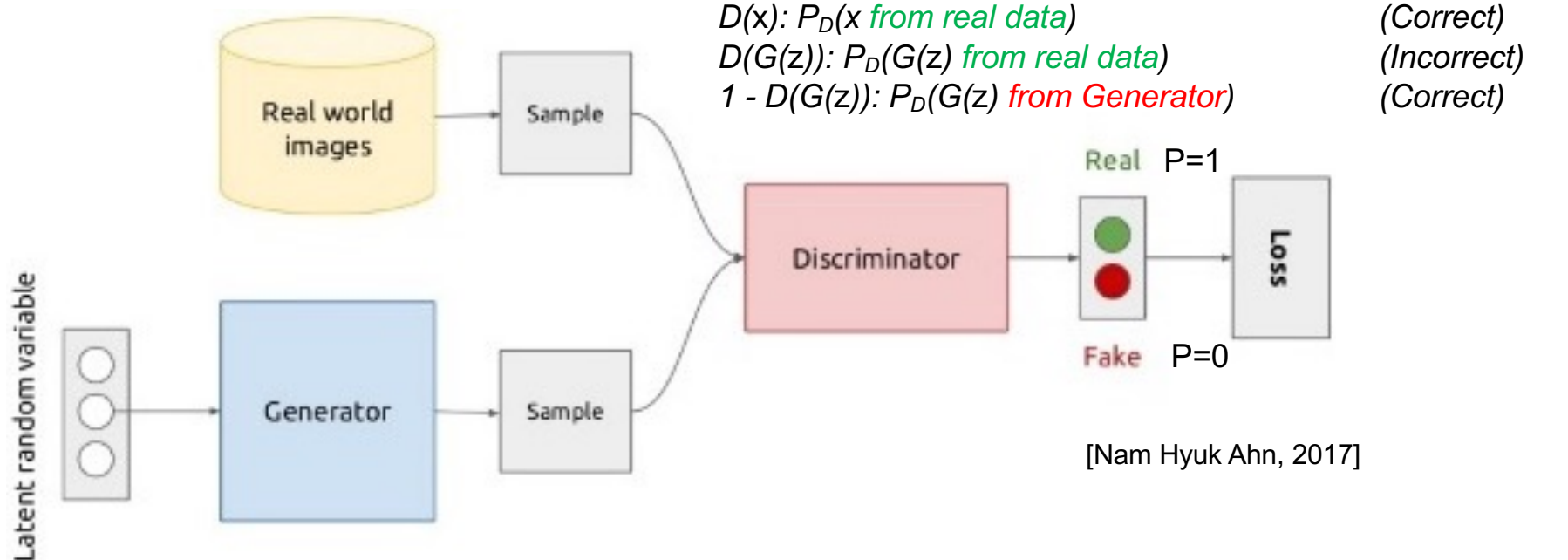
Generative Adversarial Networks (GAN) [Goodfellow et al., 2014]



Generative Adversarial Networks (GAN) [Goodfellow et al., 2014]

- Training Simultaneously 2 Neural Networks
 - Generator
 - » Transforms Random noise Vectors into *Faked* Samples
 - Discriminator
 - » Estimates probability that the Sample came from training data rather than from G
 - Minimax 2-player game

$$\min_G \max_D V(G, D) = \log(D(x)) + \log(1 - D(G(z)))$$

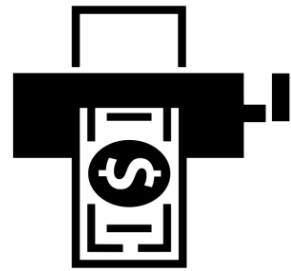


[Nam Hyuk Ahn, 2017]

GAN Equation

- Binary Cross-Entropy:
 - $H_B(y, \hat{y}) = - (y \log \hat{y} + (1-y) \log (1-\hat{y}))$
 - $D(x) = 1$ $P_D(x \text{ from real data})$ *Correct*
 - $H_B(D(x), \hat{D}(x)) = - (D(x) \log \hat{D}(x) + (1-D(x)) \log (1-\hat{D}(x)))$
 - $H_B(D(x), \hat{D}(x)) = - \log D(x)$
 - $D(G(z)) = 0$ $P_D(G(z) \text{ from real data})$ *Incorrect*
 - $H_B(D(G(z)), \hat{D}(G(z))) = - (D(G(z)) \log \hat{D}(G(z)) + (1-D(G(z))) \log (1-\hat{D}(G(z))))$
 - $H_B(D(G(z)), \hat{D}(G(z))) = - \log (1-\hat{D}(G(z)))$
 - $H_B(D(x), \hat{D}(x)) + H_B(D(G(z)), \hat{D}(G(z))) = - (\log \hat{D}(x) + \log (1-\hat{D}(G(z))))$
- $$\min_G \max_D V(G, D) = \log(D(x)) + \log(1 - D(G(z)))$$

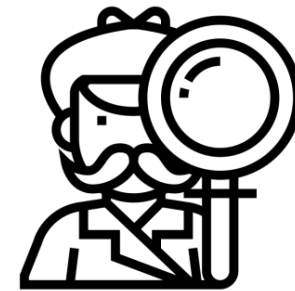
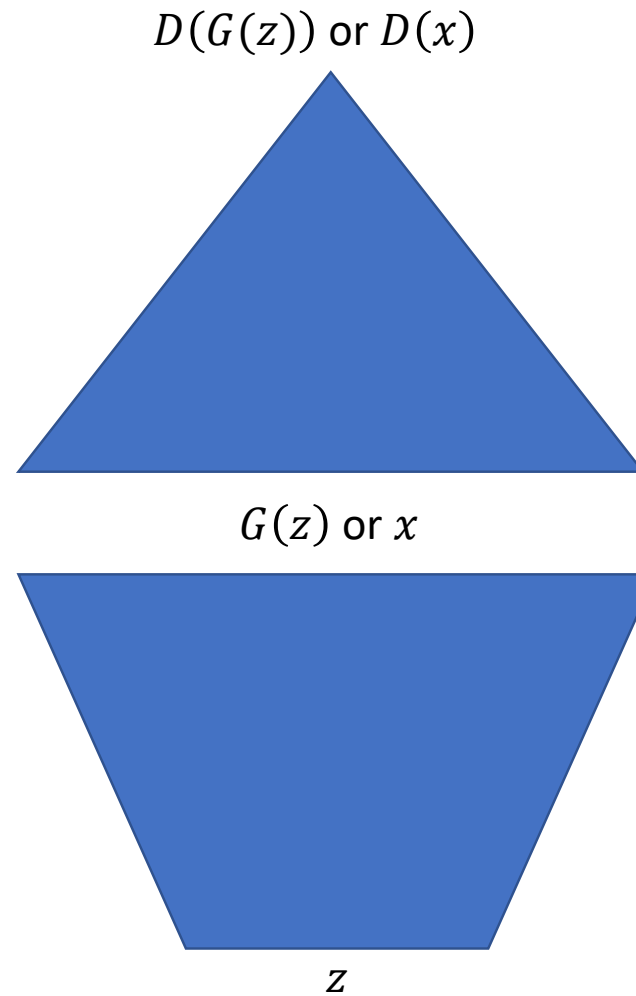
GAN and Turing Test



Generator



Discriminator



[Goodfellow, 2016]

GAN Basic Training Algorithm

- Initialize $\theta^{(G)}, \theta^{(D)}$
- For $t = 1:b:T$
 - Initialize $\Delta\theta^{(D)} = 0$
 - For $i = t:t + b - 1$
 - Sample $z_i \sim p(z_i)$
 - Compute $D(G(z_i)), D(x_i)$
 - $\Delta\theta_i^{(D)} \leftarrow$ Compute gradient of **Discriminator loss**, $J^{(D)}(\theta^{(G)}, \theta^{(D)})$
 - $\Delta\theta^{(D)} \leftarrow \Delta\theta^{(D)} + \Delta\theta_i^{(D)}$
 - Update $\theta^{(D)}$
 - Initialize $\Delta\theta^{(G)} = 0$
 - For $j = t:t + b - 1$
 - Sample $z_j \sim p(z_j)$
 - Compute $D(G(z_j)), D(x_j)$
 - $\Delta\theta_j^{(G)} \leftarrow$ Compute gradient of **Generator loss**, $J^{(G)}(\theta^{(G)}, \theta^{(D)})$
 - $\Delta\theta^{(G)} \leftarrow \Delta\theta^{(G)} + \Delta\theta_j^{(G)}$
 - Update $\theta^{(G)}$

Examples of GAN Generated Images



CelebFaces Attributes Dataset (CelebA)
> 200K celebrity images



[Brundage et al., 2018]

Synthetic (Generated) Celebrity images



[Karras et al., 2018]

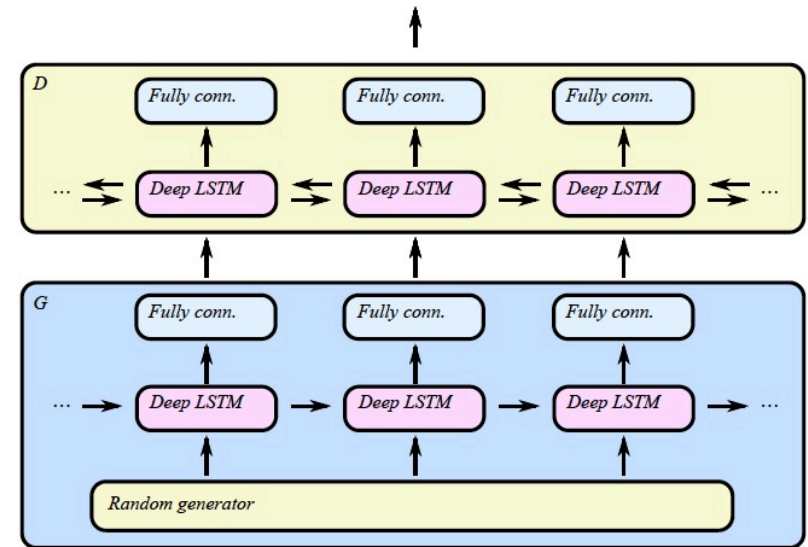
Using StyleGAN [Karras et al., 2018]



C-RNN-GAN [Mogren, 2016]

GAN(Bidirectional-LSTM², LSTM²)

- Discriminator considers the hidden layers (forward and backward) values to be (or not) representative of the Real data
 - Analog to RNN Encoder-Decoder which considers the hidden layer as the summary of a sequence
- Classical music Training Dataset



MidiNet [Yang et al., 2017]

- Conditioning information

- Previous measure
- Chord sequence

- Scope:

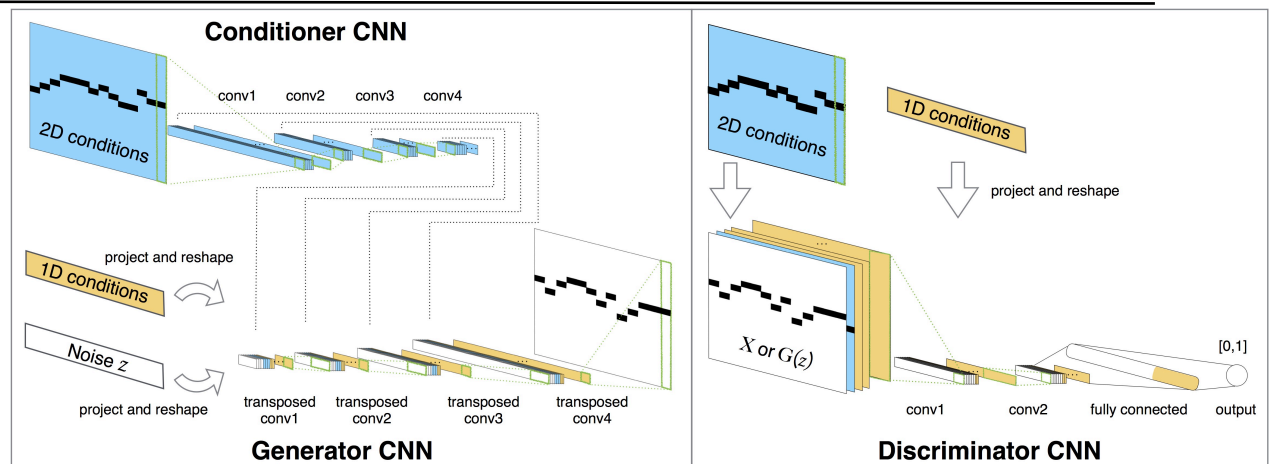
- Previous measure (1D conditions)
- Various previous measures (2D conditions)

- Fine control:

- Conditioning on previous measure 1D/2D and on chord sequence 1D/2D for one/all convolutional layers
- Ex: previous measure 1D and on chord sequence 2D for all convolutional layers
 - » Follows more chord sequence



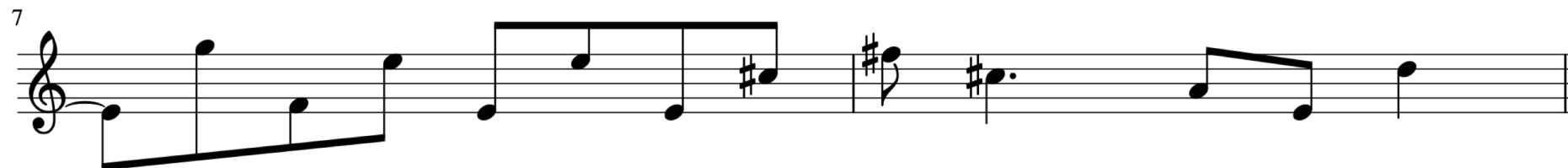
- Pop music dataset <https://soundcloud.com/vgtsv6jf5fwq/model3>



GAN Examples – Celtic Melodies (500 Epochs)



GAN Examples – Celtic Melodies (5000 Epochs)



GAN Examples – Bach Chorales

Piano



The first system of the musical score consists of two staves, both labeled 'Piano'. The top staff is in treble clef with a key signature of one sharp (F#) and a 4/4 time signature. It contains a melodic line with eighth and sixteenth notes, including a trill. The bottom staff is in bass clef with the same key signature and time signature, providing a harmonic accompaniment with eighth and sixteenth notes.



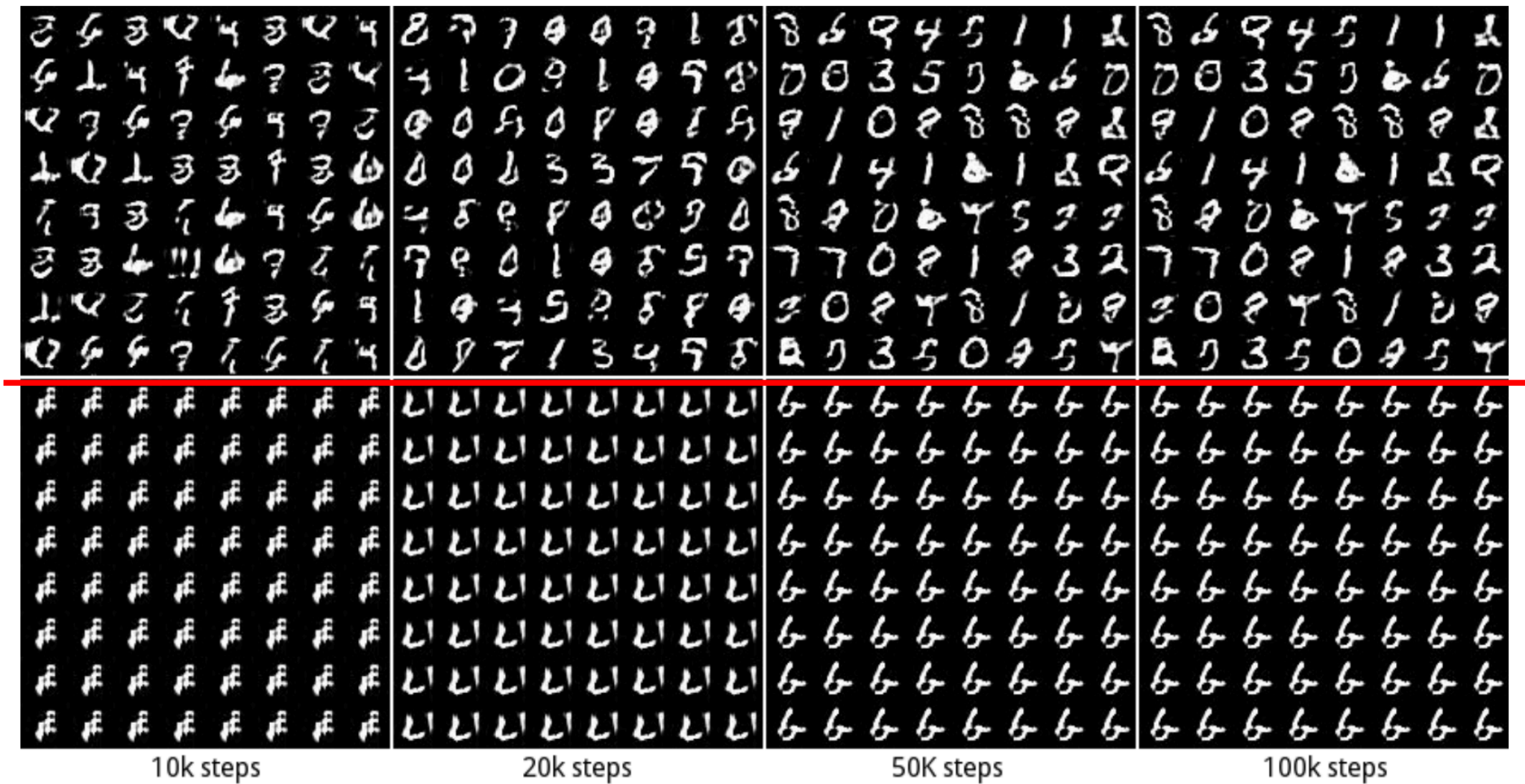
Piano



The second system of the musical score also consists of two staves, both labeled 'Piano'. The top staff is in treble clef with a key signature of one flat (Bb) and a 4/4 time signature. It continues the melodic line from the first system. The bottom staff is in bass clef with the same key signature and time signature, continuing the harmonic accompaniment.

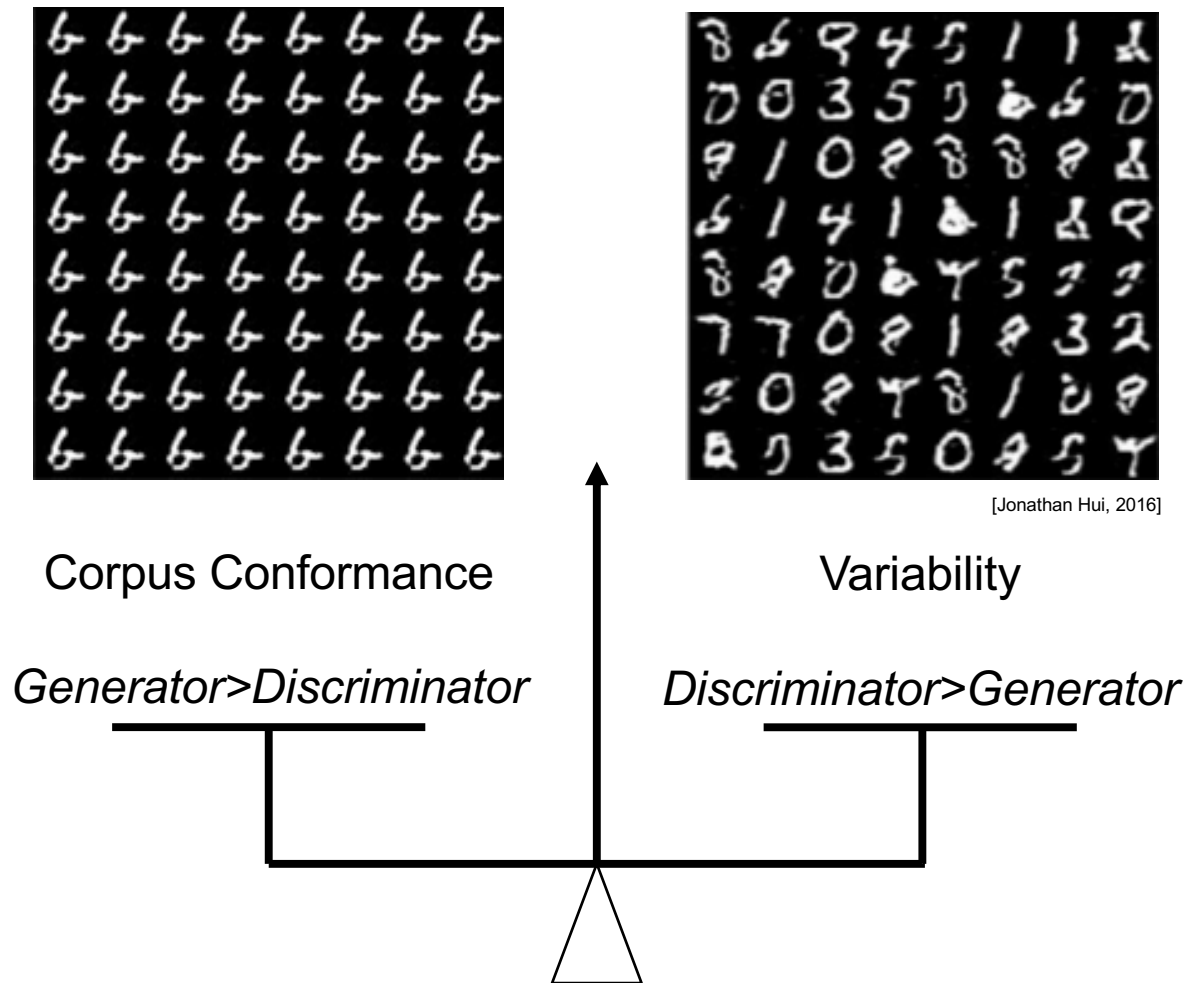


GAN Mode Collapse (1/3)

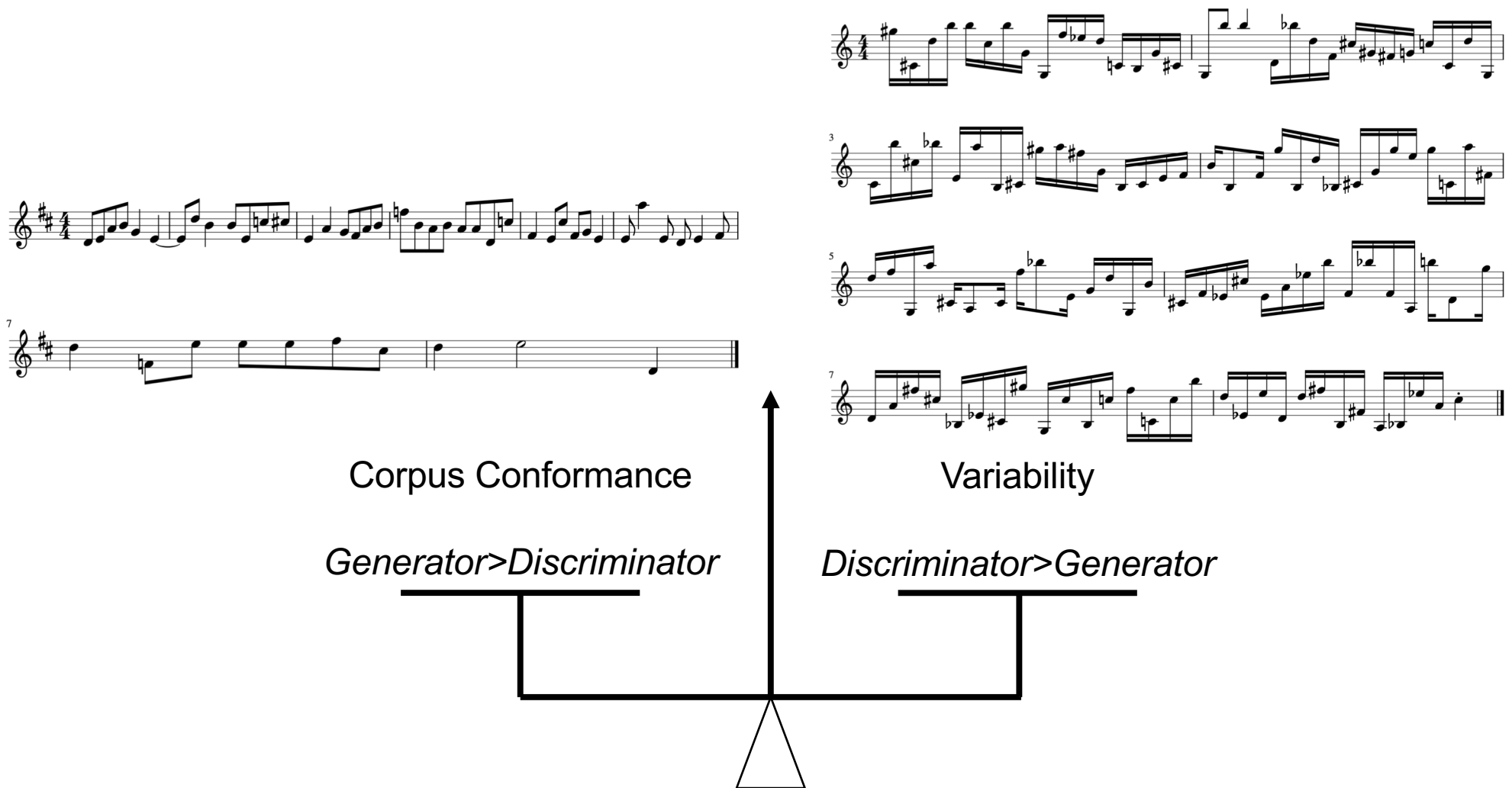


[Jonathan Hui, 2016]

GAN Mode Collapse (2/4)



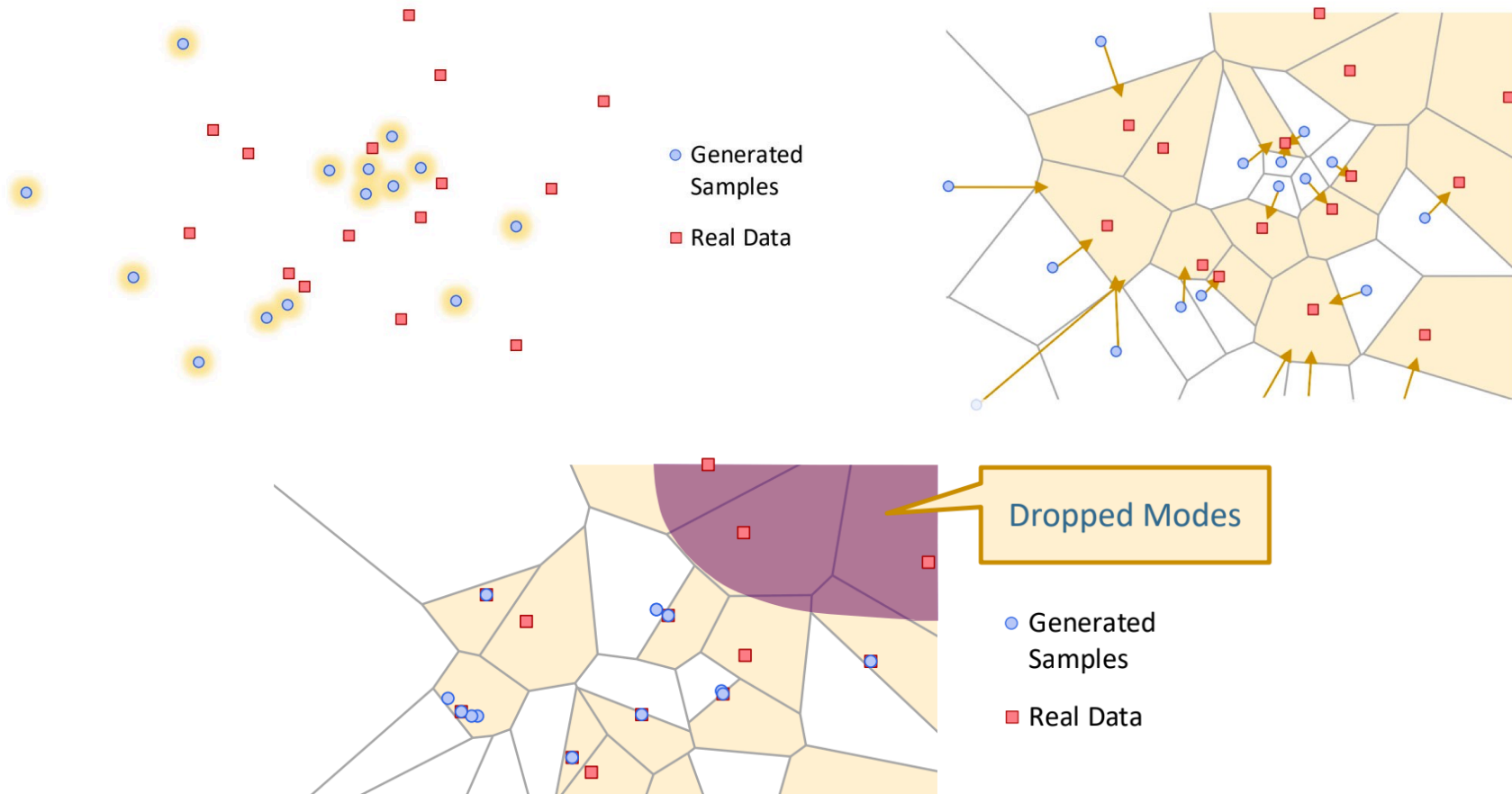
GAN Mode Collapse (2/4)



GAN Mode Collapse (3/3)

- G is trained extensively without sufficient updates to D
- The generated samples will converge to find the optimal content x^* that fools D the most, the most realistic sample from the discriminator perspective
- In this extreme case (**single point mode collapse**), x^* will be independent of z [Hui, 2018]
- Approach: Constantly update D
- Heuristic/Empirical approach
- High hyperparameters sensitivity

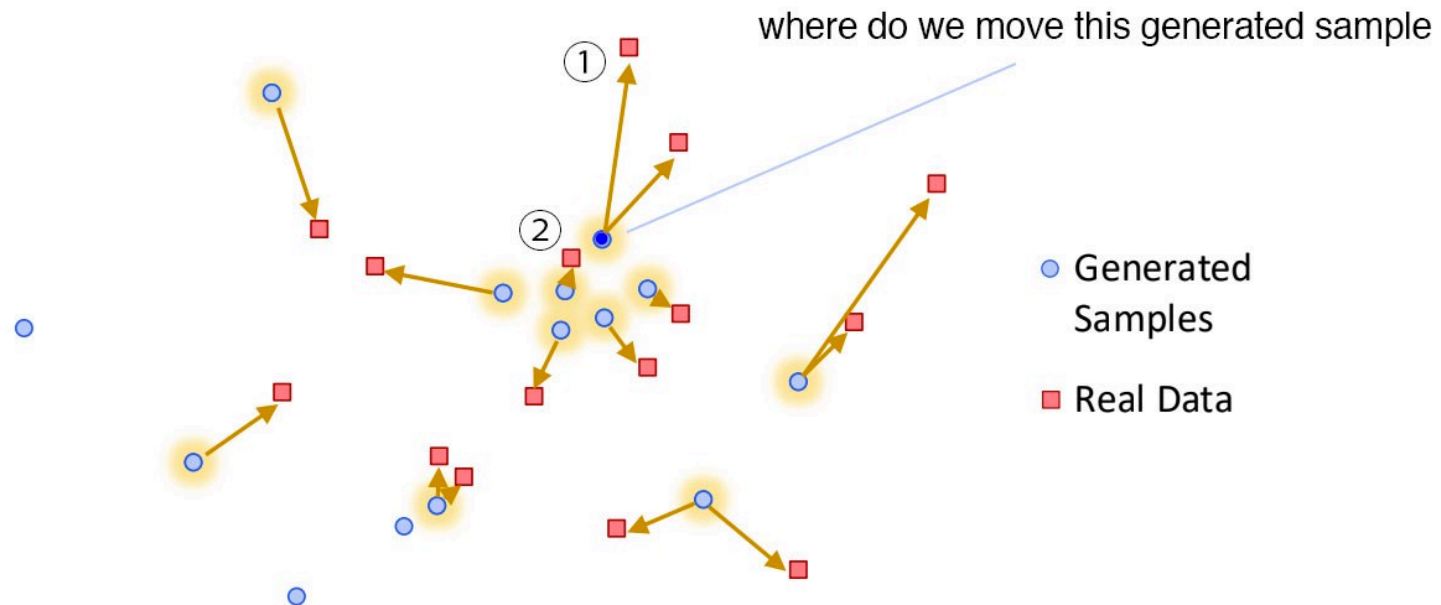
Explanation and Direction [Li, 2019]



Generated samples move toward the closest boundary
This ensures that each generated sample has a nearby data example

But it does not ensure that each real data has a nearby generated sample [Li, 2019]

Implicit Maximum Likelihood Estimation (IMLE) [Li, 2019]



- 1) For each real data, what is the closest generated sample?
- 2) The generated sample moves toward that real data

VAE vs GAN

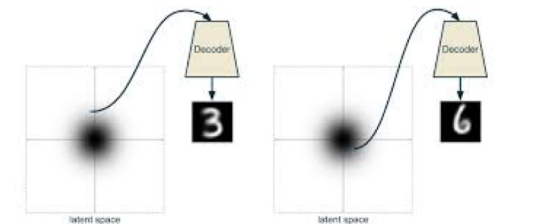
- VAE (Variational Autoencoder) and GAN (Generative Adversarial Networks)

Some Similarities:

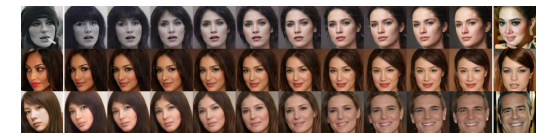
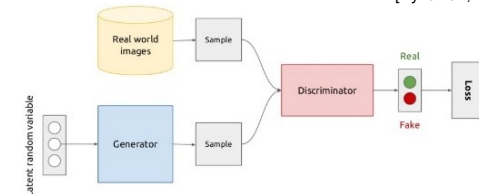
- Are both generative architectures
- Generate from random latent variables

Differences:

- VAE is representational of the whole training dataset
- GAN is not
- Smooth control interface for exploring latent data space
- GAN has (ex: interpolation) but not as for VAE
- GAN produces better quality content (ex: better resolution images)



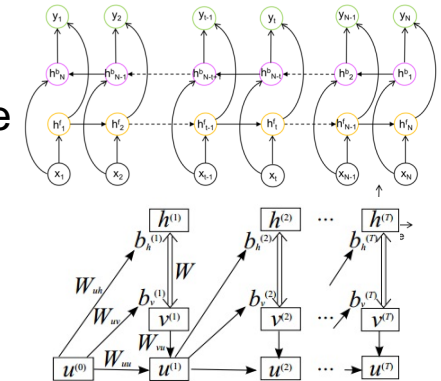
[Dykeman, 2016]



Compound Architectures

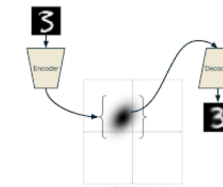
- Composition

- Bidirectional RNN, combining two RNNs, forward and backward in time
- RNN-RBM [Boulanger-Lewandowski et al., 2012], combining an RNN (horizontal/sequence) and an RBM (vertical/chords)



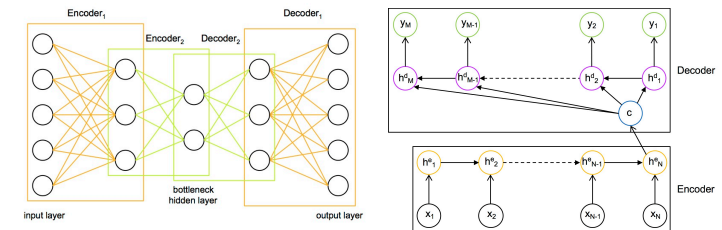
- Refinement

- Sparse autoencoder
- Variational autoencoder (VAE) = Variational(Autoencoder)



- Nested

- Stacked autoencoder = Autoencoderⁿ
- RNN Encoder-Decoder = Autoencoder(RNN, RNN)



- Pattern instantiation

- C-RBM [Lattner et al., 2016] = Convolutional(RBM)
- C-RNN-GAN [Mogren, 2016] = GAN(Bidirectional-LSTM², LSTM²)
- Anticipation-RNN [Hadjeres & Nielsen, 2017] = Conditioning(RNN, RNN)

