Neural Network Tools and Principles, part 2

Generative models

Intelligent Computational Media, Spring 2021
Prof. Perttu Hämäläinen
Aalto University

Generative models

- Learn a probability distribution $p(\mathbf{x})$ – e.g., facial images – and a way to draw samples from it
- More common in practice: learn a conditional distribution $p(\mathbf{y} \mid \mathbf{x})$
- Same as approximating some function $\mathbf{y}=f(\mathbf{x})$, but with multiple possible $\mathbf{y}$ for each $\mathbf{x}$
- Example: image colorization. Many possible interpretations.

input

U-Net (discriminative)

GAN, WAE... (generative)

Ground truth

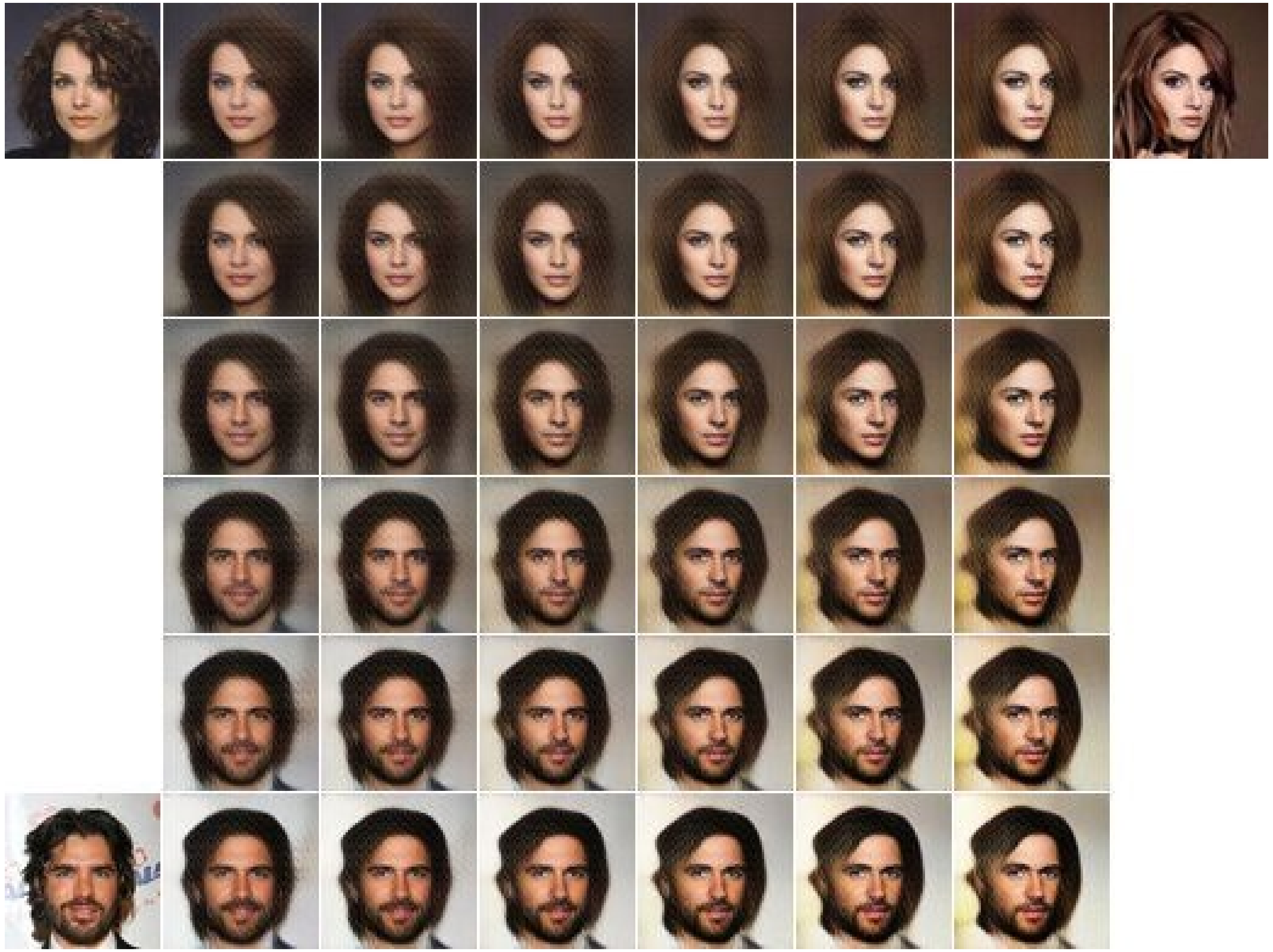
Basic principle

- Sample some random numbers and feed them through a network that converts them into images, sound, text...
- Three main types of models: Autoencoders (VAE, WAE), Generative Adversarial Networks (GAN), Flow-based models

Latent space, representation learning

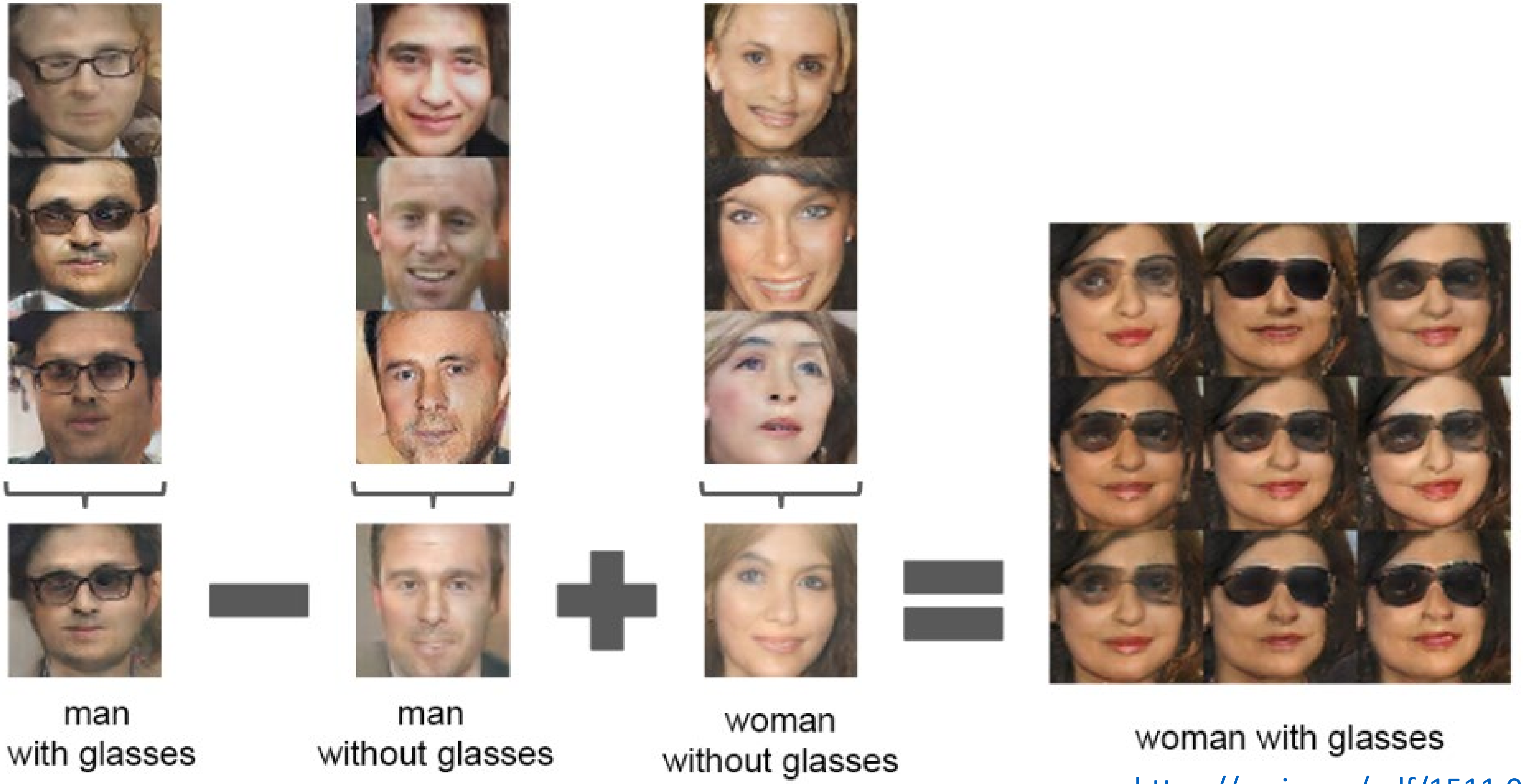
- If one samples image pixels or audio values, one gets noise.
- The models try to learn an *N*-dimensional ”latent space” where all positions (vectors of *N* numbers) in some region map to some meaningful image or sound, when passed through a generator or decoder network.
- If *N* is small, this enforces the networks to learn about relations. E.g., similar images should have similar latent encodings
- Directions often have semantic meaning: Along some axis, male faces might be to the left, and female to the right
- This allows interesting manipulations...

A 2D latent space of an autoencoder trained with faces



https://medium.com/@juliendespois/latent-space-visualization-deep-learning-bits-2-bd09a46920df

Latent space math

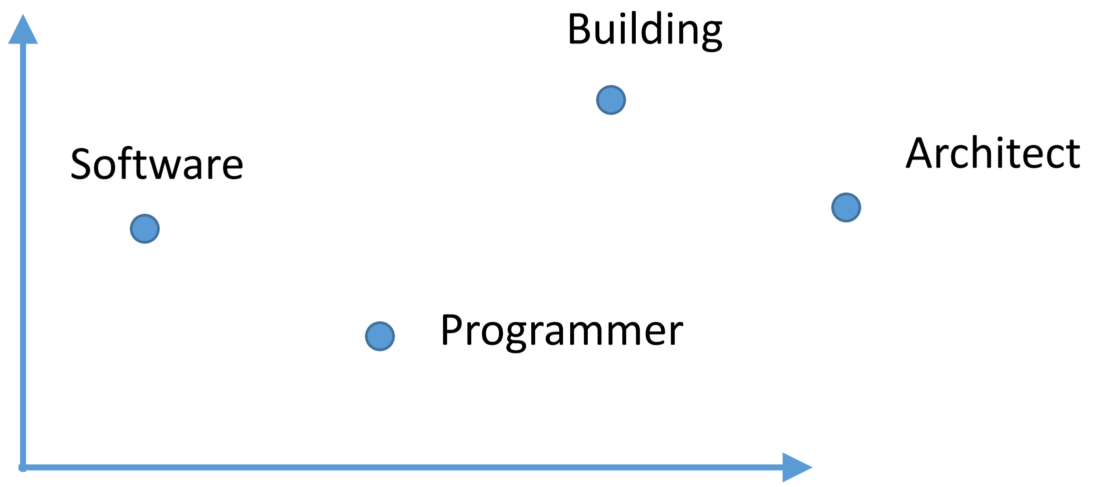


https://arxiv.org/pdf/1511.06434.pdf

Pixel-space math

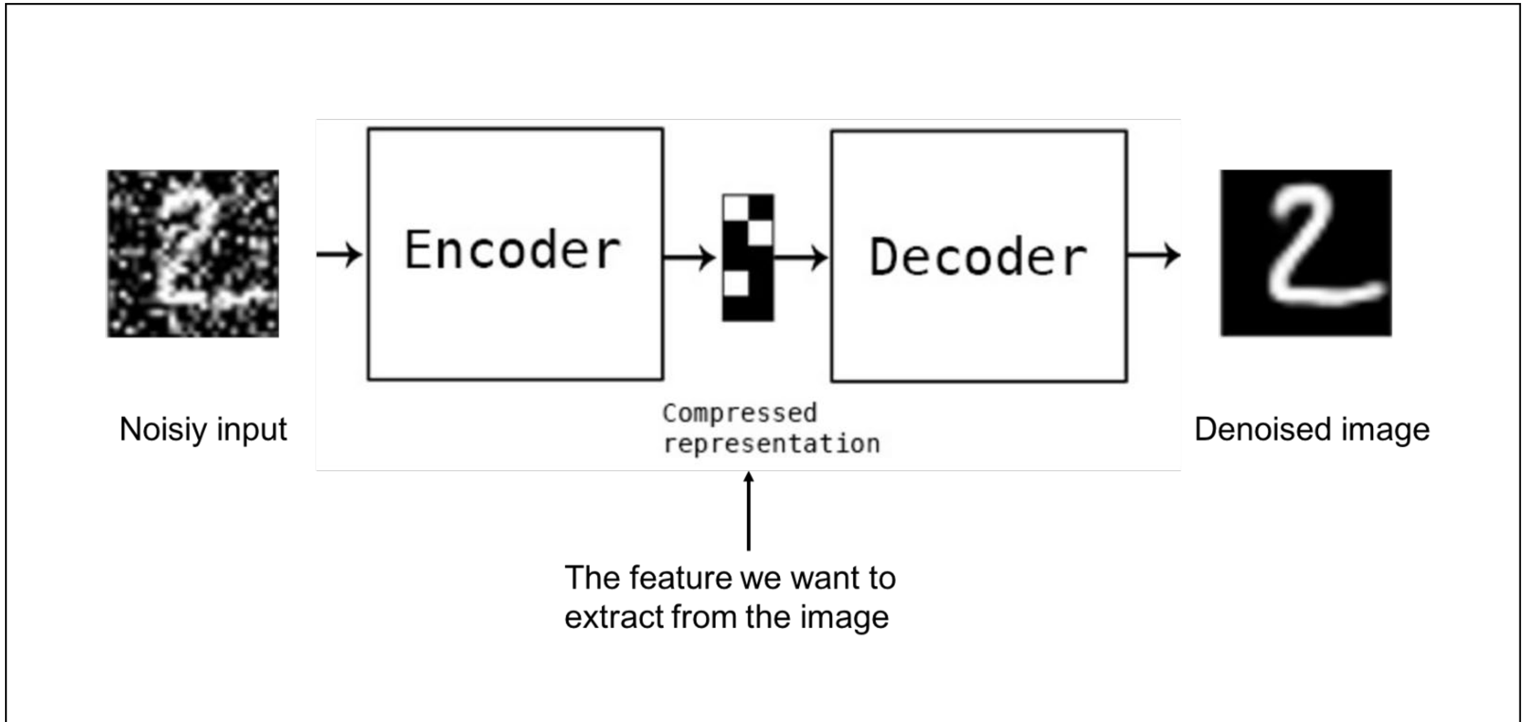
Same with words

- King - Man + Woman = Queen
- Note that this is equivalent to King - Man = Queen - Woman
- Software - Building + Architect = Programmer
- Precomputed dictionaries of such encodings are available, e.g., Fasttext (https://fasttext.cc/)



Variational autoencoder (VAE, 2014)

- Random samples in the latent space (the encoder outputs) of an autoencoder sometimes generate valid decoded output, sometimes not
- A basic autoencoder does not guarantee what happens when the encoding lies between training examples.
- VAE is an autoencoder where the compressed representation is forced to be normally distributed => easy to sample



Wasserstein Autoencoder (WAE, ICLR 2018)

- VAE is old and usually doesn't give great results (details beyond the scope of this lecture)
- WAE is the modern version, but a bit more costly to train

Wasserstein Auto-Encoders

Ilya Tolstikhin[1], Olivier Bousquet[2], Sylvain Gelly[2], and Bernhard Schölkopf[1]

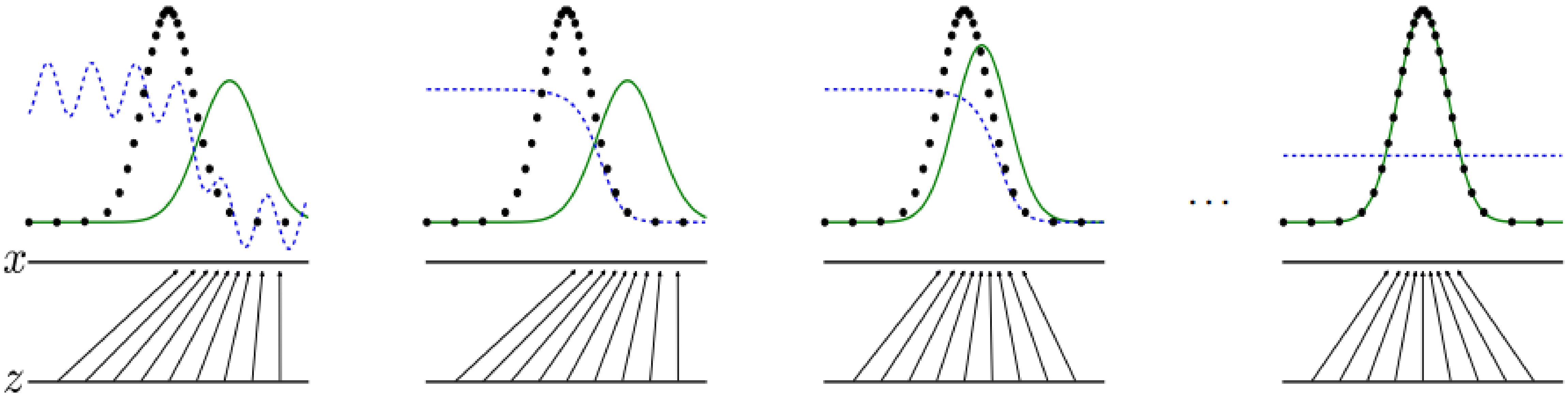
[1]Max Planck Institute for Intelligent Systems
[2]Google Brain

Abstract

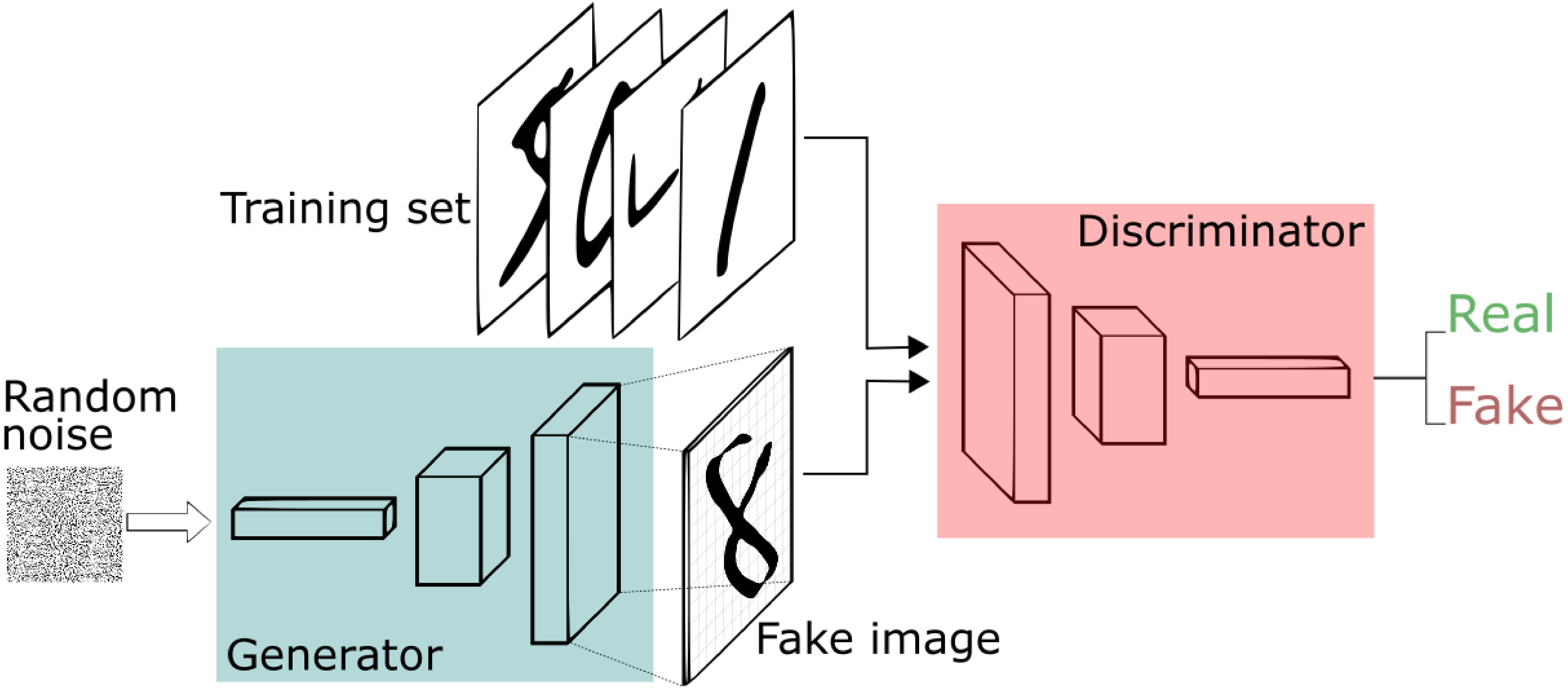
We propose the Wasserstein Auto-Encoder (WAE)—a new algorithm for building a generative model of the data distribution. WAE minimizes a penalized form of the Wasserstein distance between the model distribution and the target distribution, which leads to a different regularizer than the one used by the Variational Auto-Encoder (VAE) [1]. This regularizer encourages the encoded training distribution to match the prior. We compare our algorithm with several other techniques and show that it is a generalization of adversarial auto-encoders (AAE) [2]. Our experiments show that WAE shares many of the properties of VAEs (stable training, encoder-decoder architecture, nice latent manifold structure) while generating samples of better quality, as measured by the FID score.

Generative Adversarial Networks (GANs)

- Produces highest-quality image and sound samples (so far)
- A *generator* network maps random vectors **z** to data vector **x**
- A *discriminator* network: 2-class classifier, trained to distinguish actual data from **x**

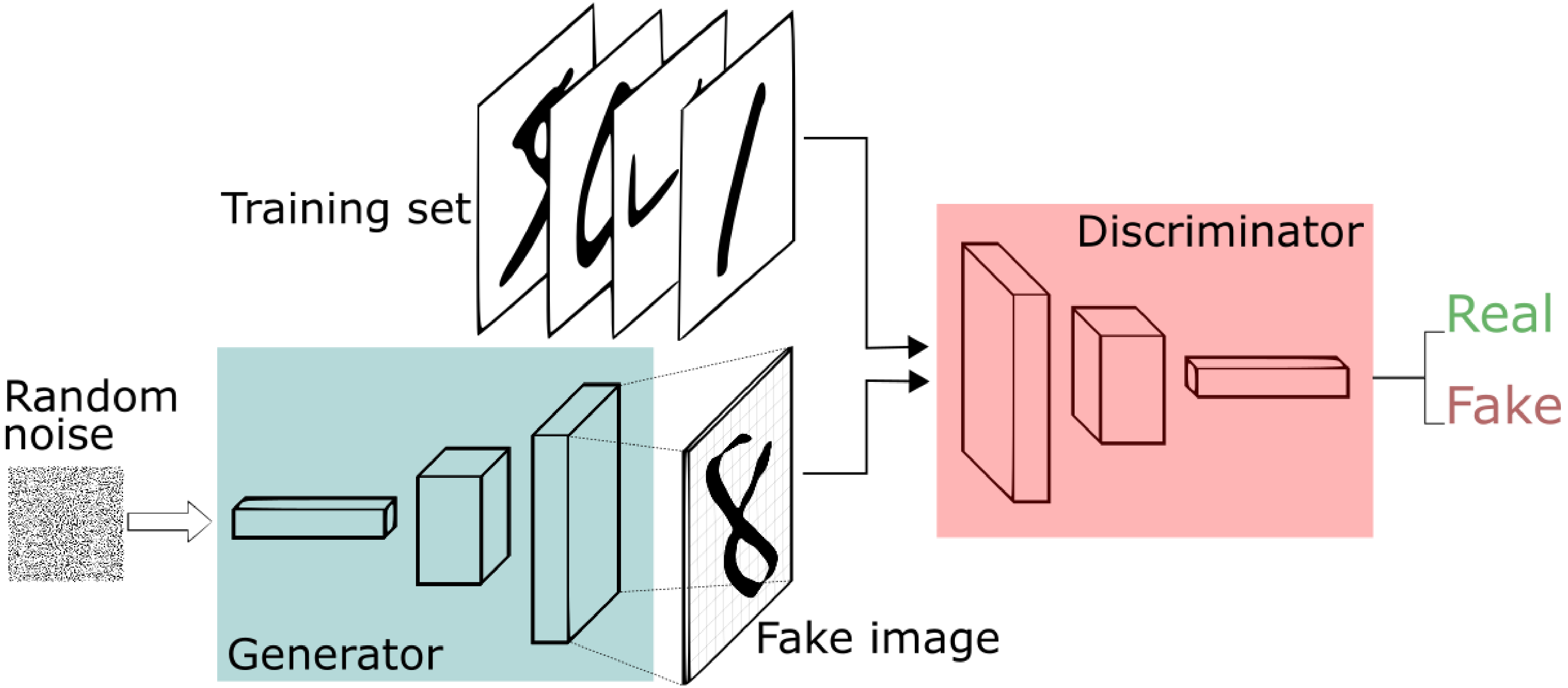


GAN: two networks in a single compute graph

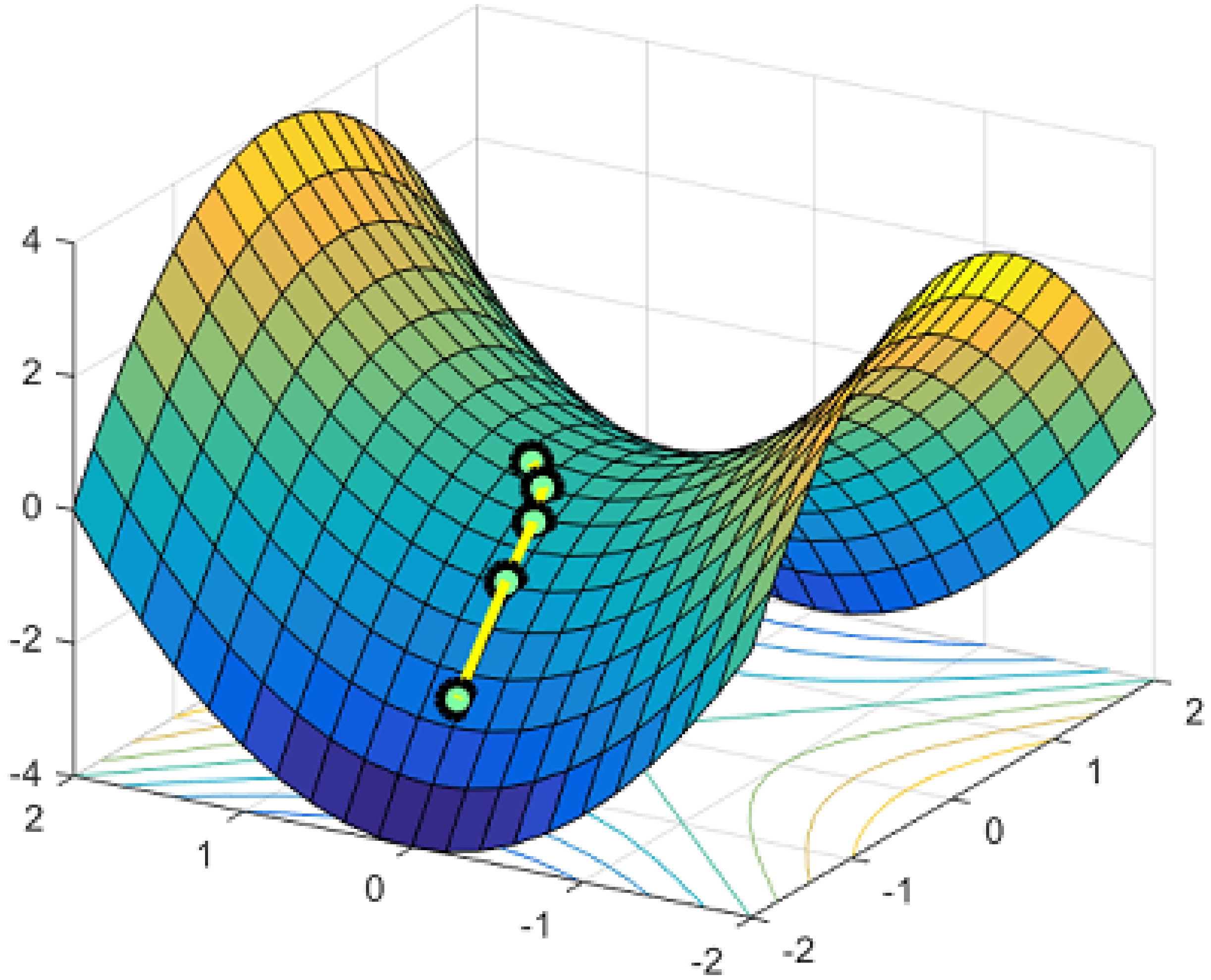


GANs are trained in interleaved manner

1. Keep generator fixed, optimize discriminator to maximize discriminator performance with both generated and real data
2. Keep discriminator fixed, optimize generator to minimize discriminator performance with generated data



Problem: saddle-point optimization is unstable



Improving GAN training stability

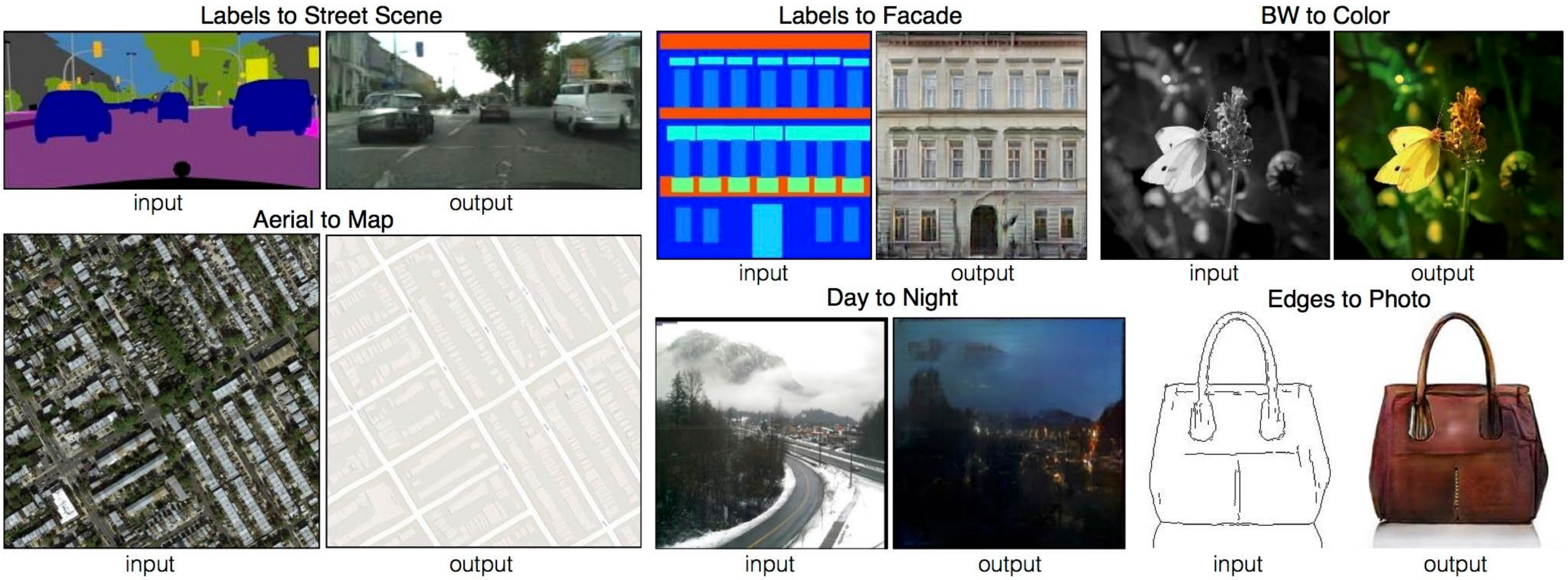
Yadav, A., Shah, S., Xu, Z., Jacobs, D., & Goldstein, T. (2017). Stabilizing Adversarial Nets With Prediction Methods. *arXiv preprint arXiv:1705.07364*. (Using saddle-point optimization theory)

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of Wasserstein gans. In *Advances in Neural Information Processing Systems* (pp. 5769-5779).

Demos and source code

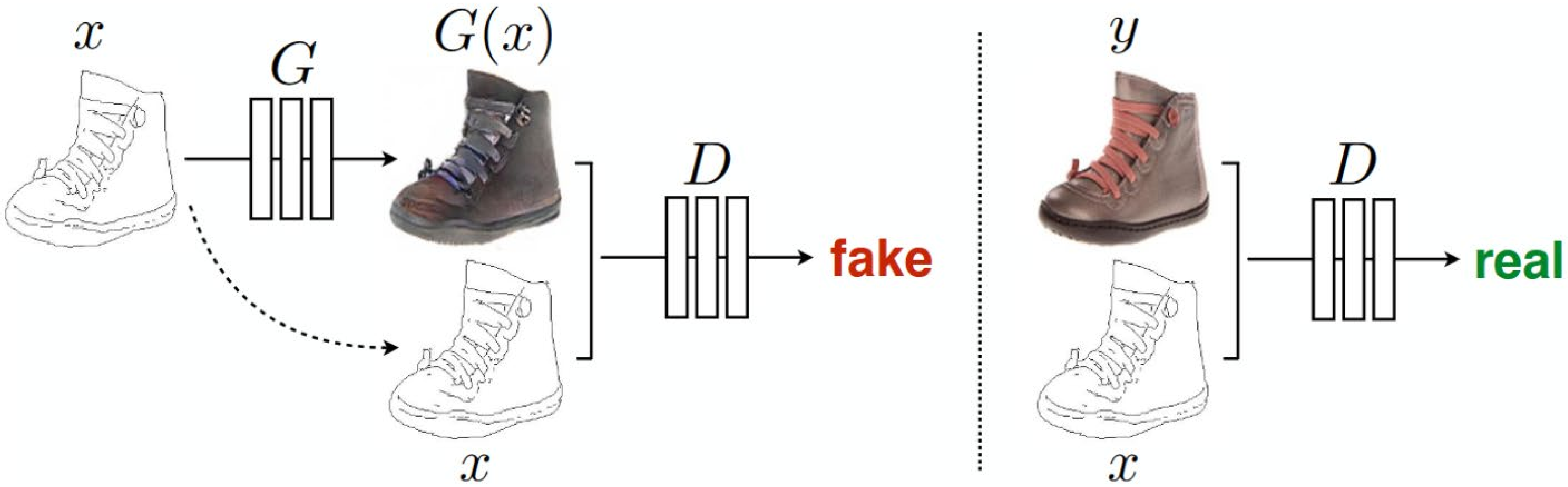
- Browser-based interactive visualization: https://cs.stanford.edu/people/karpathy/gan/
- Browser-based image-to-image (pix2pix) translation: https://affinelayer.com/pixsrv/
- An accessible tutorial: https://deeplearning4j.org/generative-adversarial-network
- BigGAN art creation tool: https://www.artbreeder.com/
- State of the art code from NVIDIA:
- https://github.com/NVlabs/stylegan2
- https://github.com/NVIDIA/pix2pixHD

pix2pix (Isola et al. 2017)



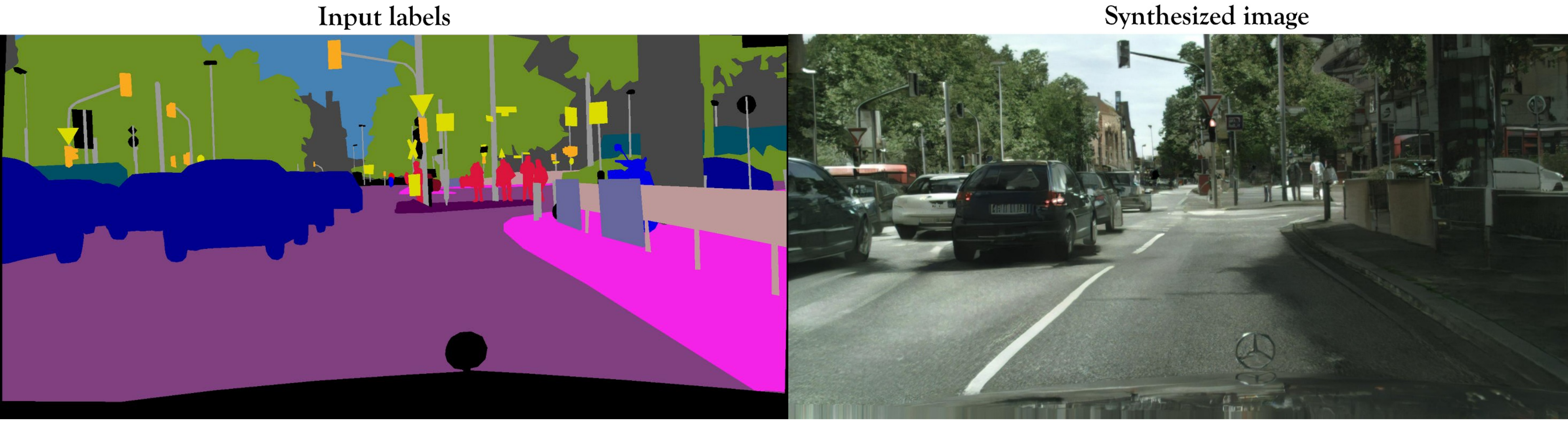
pix2pix (Isola et al. 2017)

- pix2pix is an example of conditional GAN
- Generator input: edge map, noise
- Discriminator input: edge map, real or generated images



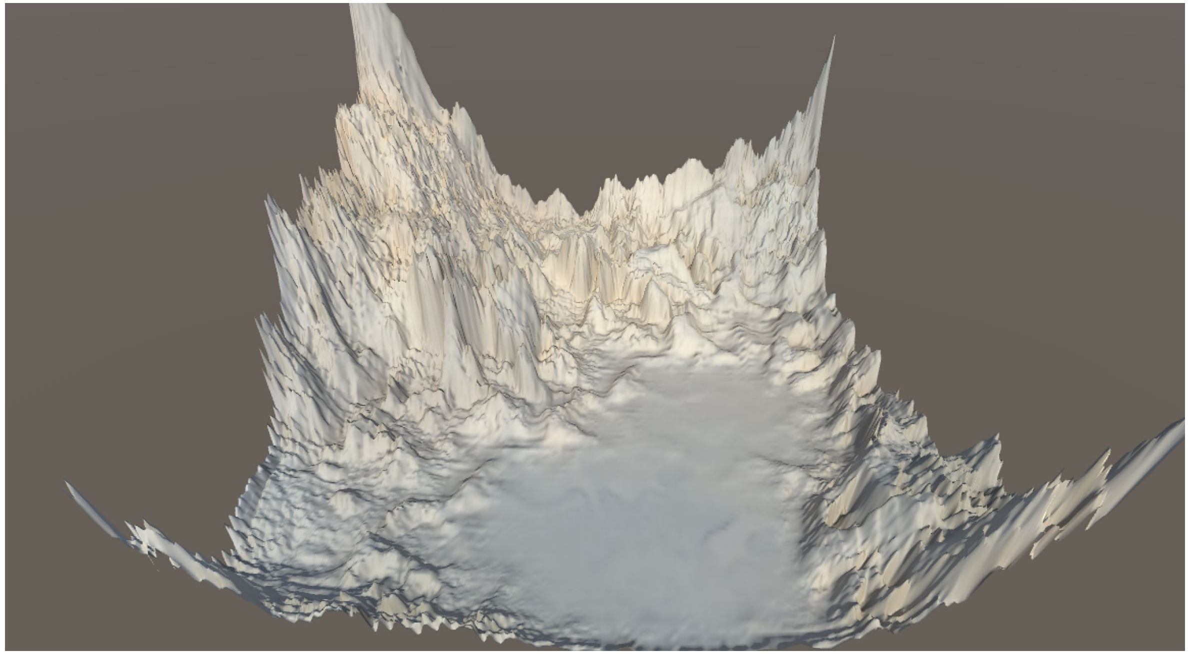
pix2pixHD

https://github.com/NVIDIA/pix2pixHD

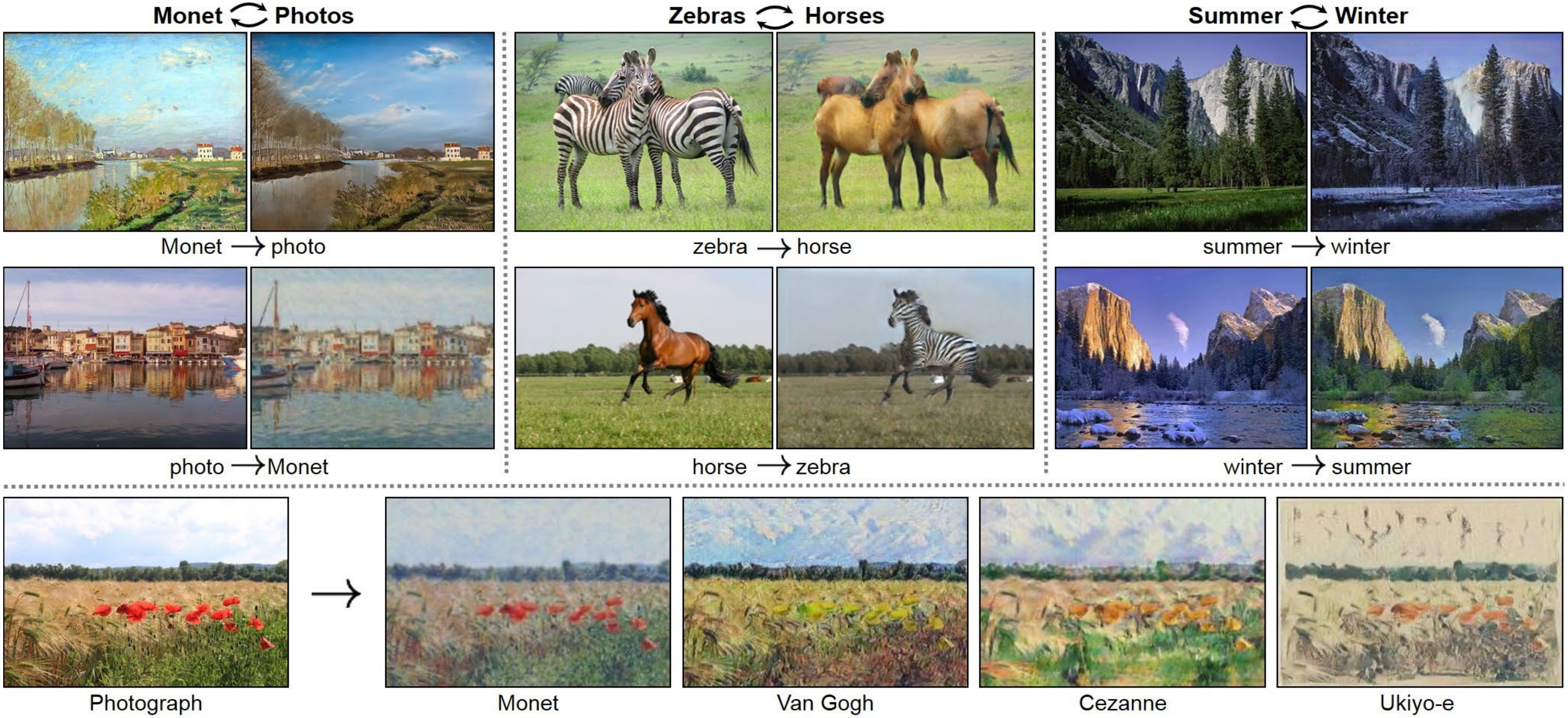


Generating terrain (Beckham 2017)

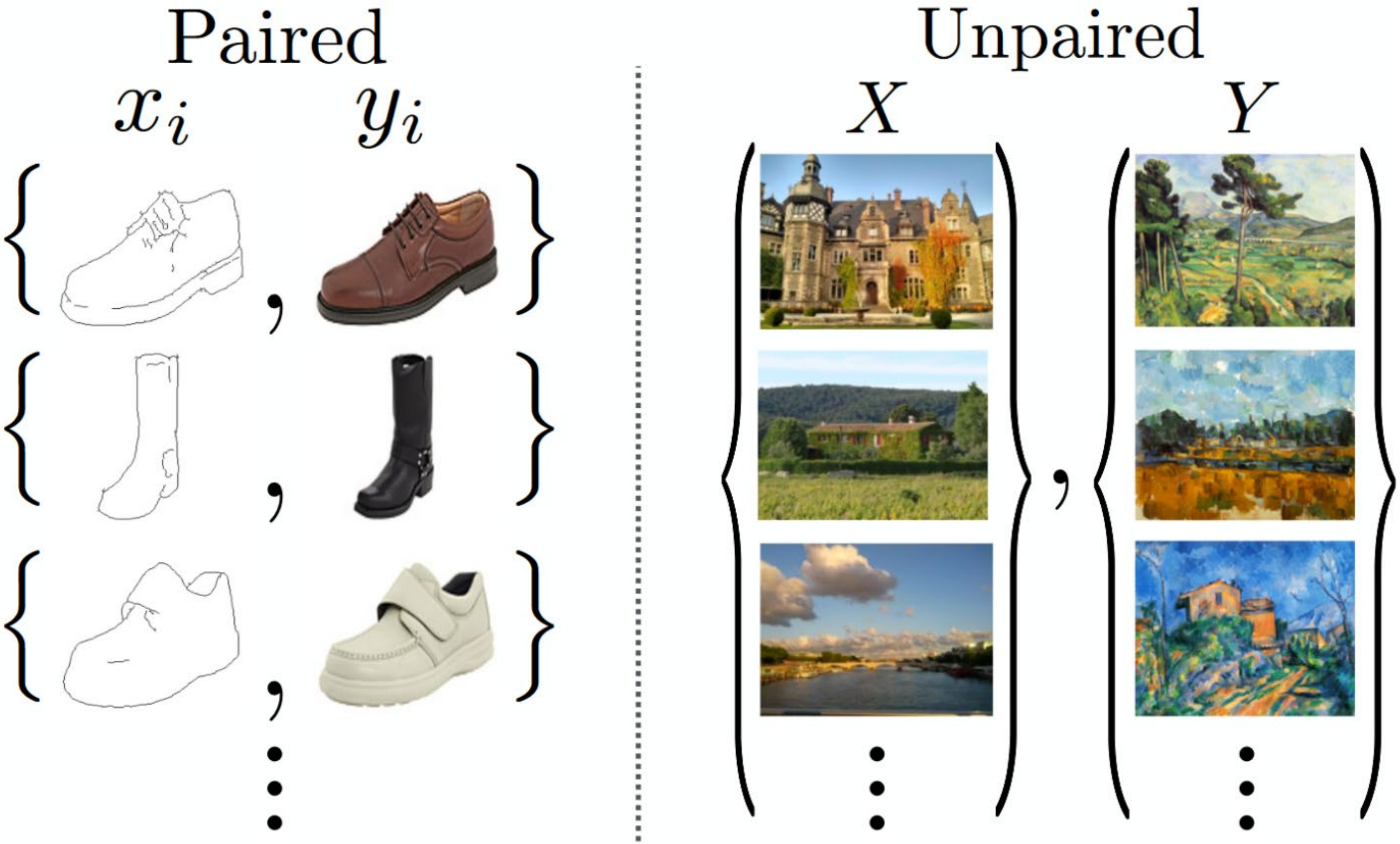
- Generate heightmaps with GAN
- Use pix2pix to generate textures from heightmaps
- Not yet super good results (dataset problem?)



CycleGAN (Jun-Yan Zhu et al. 2017)



CycleGAN: no training pairs needed!



CycleGAN: no training pairs needed!

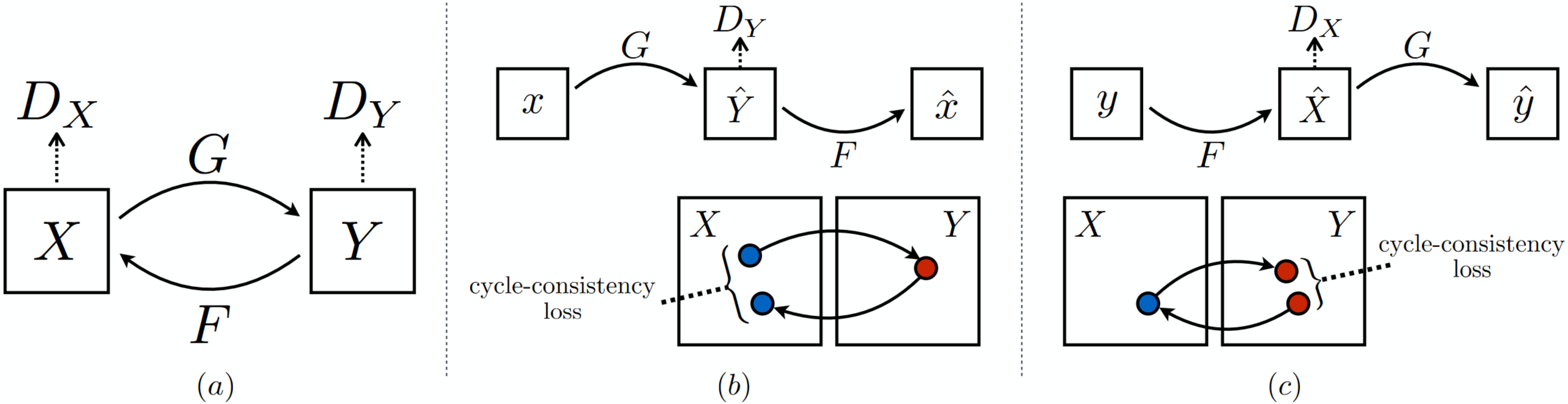
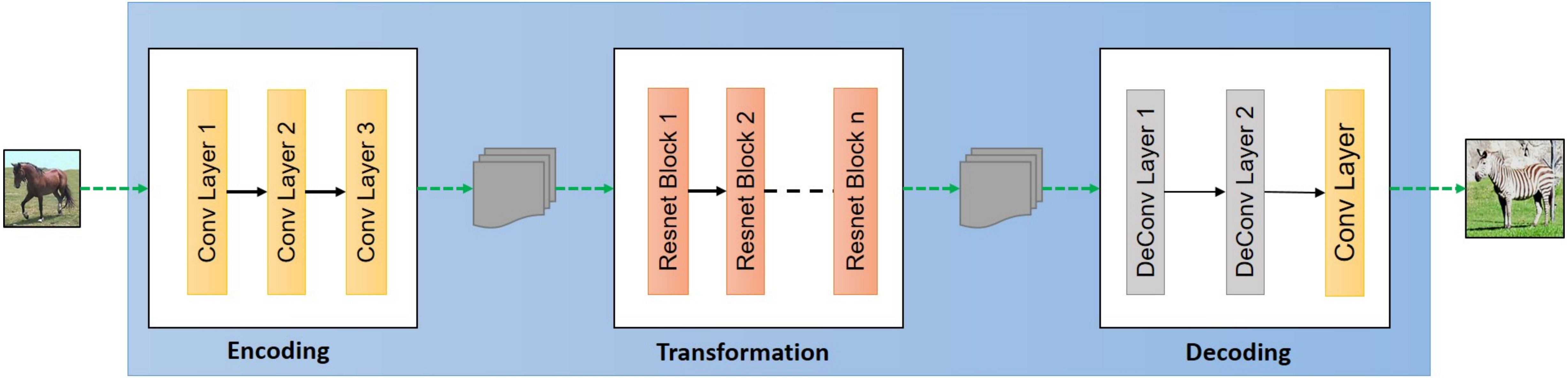


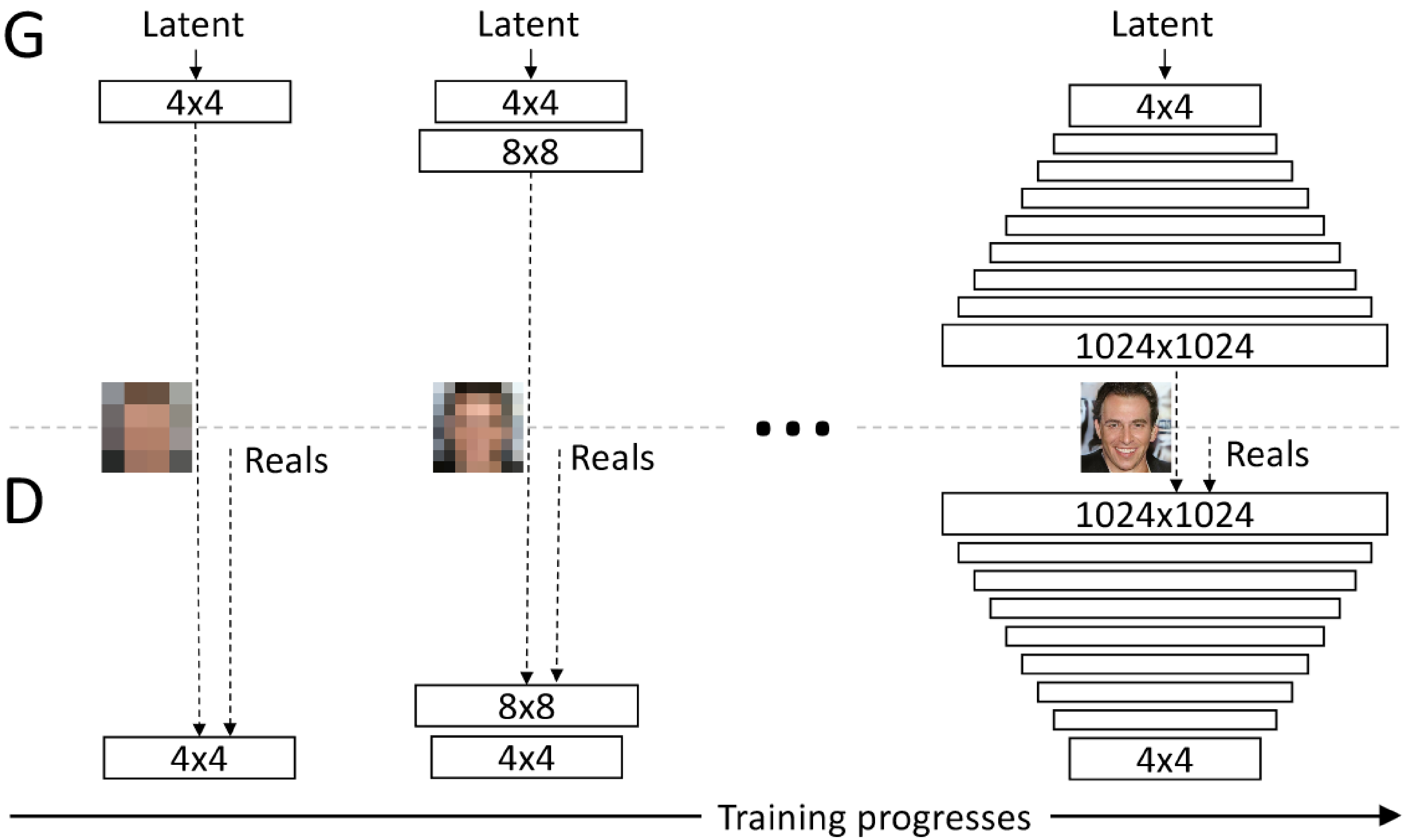
Figure 3: (a) Our model contains two mapping functions $G : X \to Y$ and $F : Y \to X$, and associated adversarial discriminators D_Y and D_X. D_Y encourages G to translate X into outputs indistinguishable from domain Y, and vice versa for D_X and F. To further regularize the mappings, we introduce two *cycle consistency losses* that capture the intuition that if we translate from one domain to the other and back again we should arrive at where we started: (b) forward cycle-consistency loss: $x \to G(x) \to F(G(x)) \approx x$, and (c) backward cycle-consistency loss: $y \to F(y) \to G(F(y)) \approx y$

CycleGAN: no training pairs needed!



Progressive GANs (Karras et al. 2017)

Progressive GANs (Karras et al. 2017)



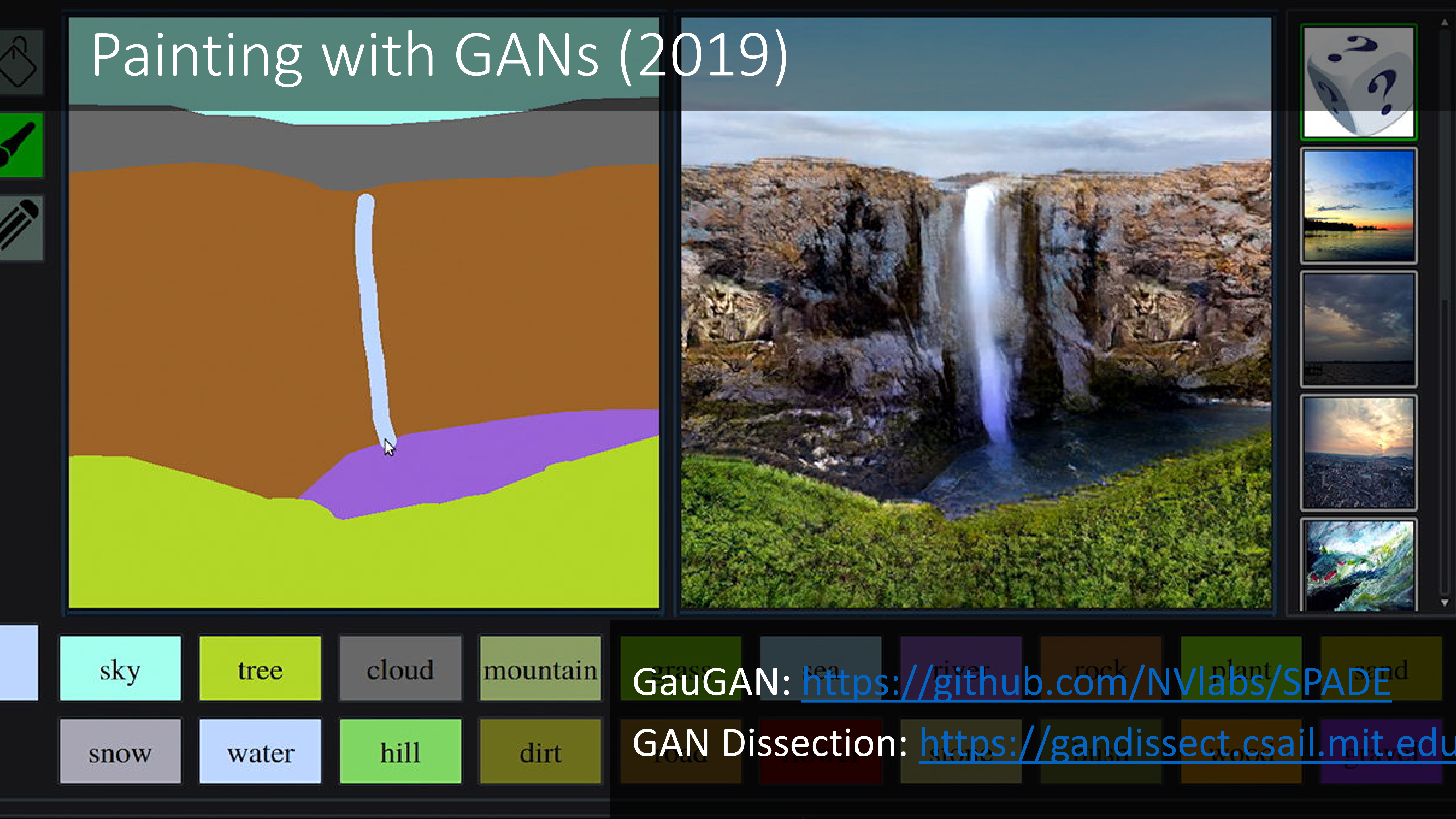
StyleGAN (Karras et al. 2019)

StyleGAN 2 (2020)




StyleGAN can provide similar results if one is lucky, but StyleGAN 2 produces high quality much more consistently and predictably

Painting with GANs (2019)



GauGAN: https://github.com/NVlabs/SPADE

GAN Dissection: https://gandissect.csail.mit.edu

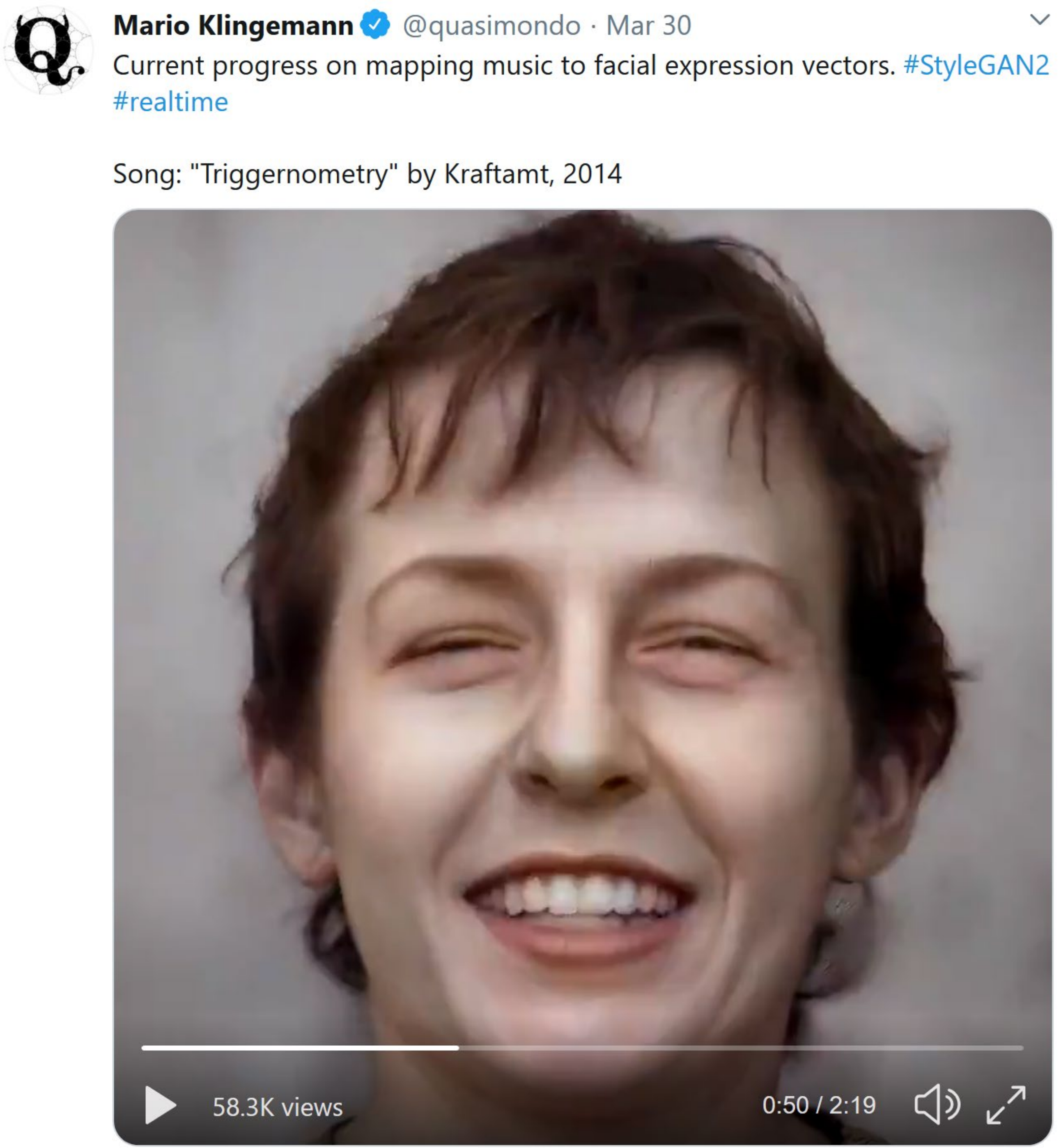
StyleGAN 2 ”circuit bending”

Finetuning a pretrained Nvidia StyleGAN 2 network with only 250 images (google image search with ”dragons”)

https://twitter.com/Norod78/status/1218282356391530496?s=20

Music visualization using StyleGAN 2

https://twitter.com/quasimondo/status/1244562140217905153?s=20



StyleGAN font generation

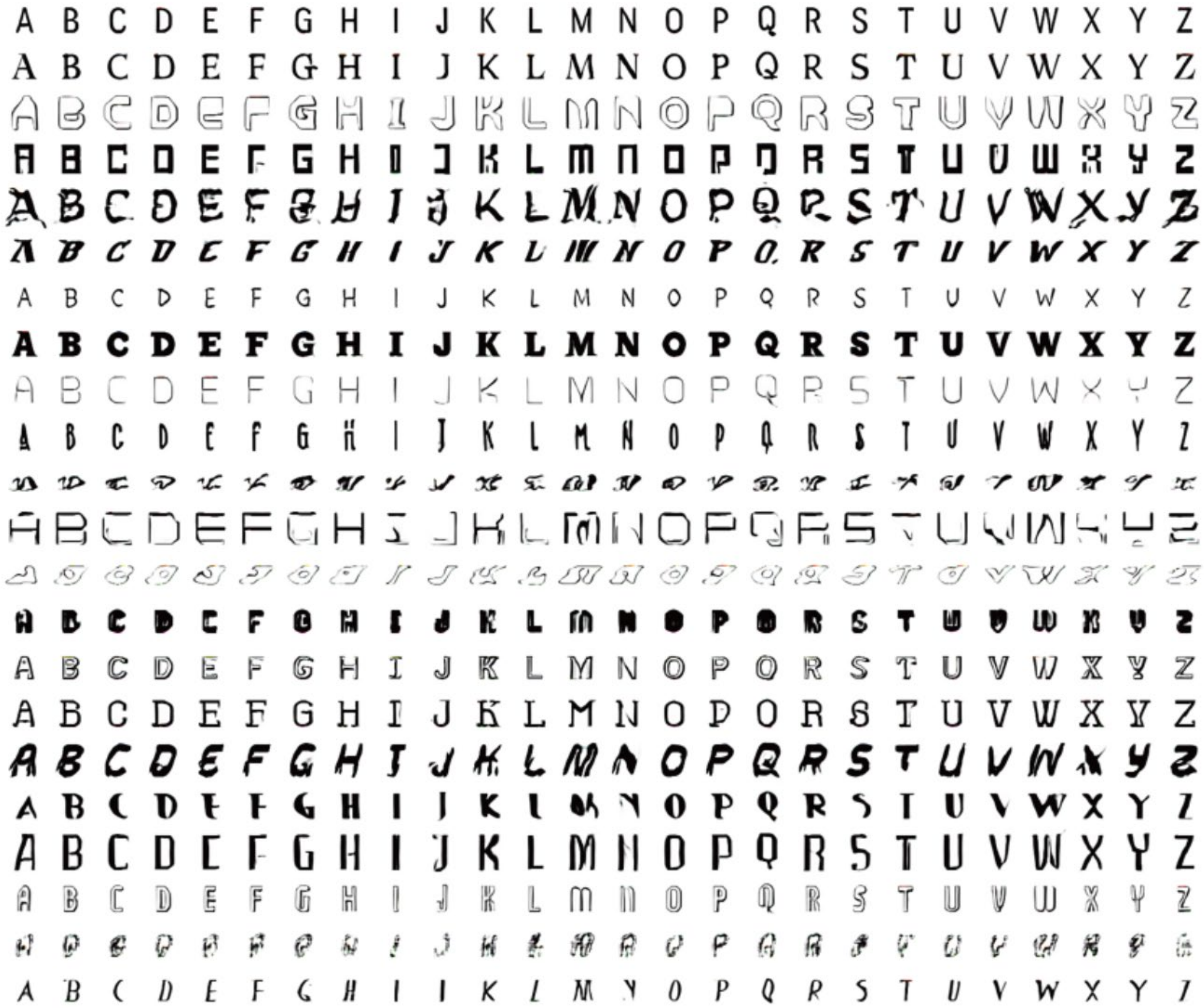
http://www.machinelearningfont.com/#mlfabout

[Submitted on 29 May 2019 (v1), last revised 30 May 2019 (this version, v2)]

GlyphGAN: Style-Consistent Font Generation Based on Generative Adversarial Networks

Hideaki Hayashi, Kohtaro Abe, Seiichi Uchida

In this paper, we propose GlyphGAN: style-consistent font generation based on generative adversarial networks (GANs). GANs are a framework for learning a generative model using a system of two neural networks competing with each other. One network generates synthetic images from random input vectors, and the other discriminates between synthetic and real images. The motivation of this study is to create new fonts using the GAN framework while maintaining style consistency over all characters. In GlyphGAN, the input vector for the generator network consists of two vectors: character class vector and style vector. The former is a one-hot vector and is associated with the character class of each sample image during training. The latter is a uniform random vector without supervised information. In this way, GlyphGAN can generate an infinite variety of fonts with the character and style independently controlled. Experimental results showed that fonts generated by GlyphGAN have style consistency and diversity different from the training images without losing their legibility.

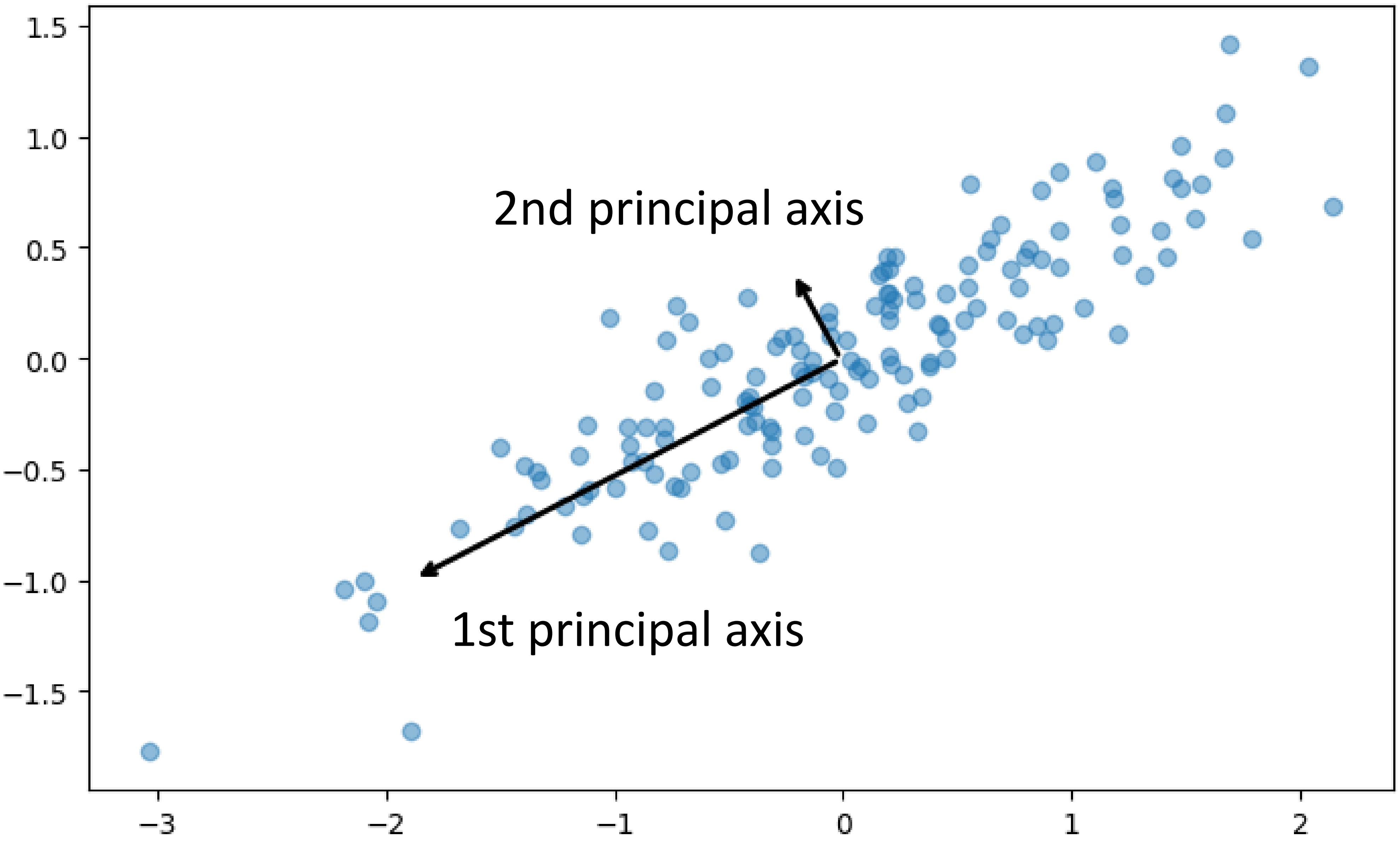


Other font generation projects

- https://github.com/erikbern/deep-fonts
- https://github.com/uchidalab/fontdesign_gan
- https://github.com/patrickgadd/feel-the-kern (with kerning)

PCA-based GAN control (Härkönen et al. 2020)

Principal Component Analysis (PCA)



BigGAN: Generating all 1000 ImageNet image classes

[Submitted on 28 Sep 2018 (v1), last revised 25 Feb 2019 (this version, v2)]

Large Scale GAN Training for High Fidelity Natural Image Synthesis

Andrew Brock, Jeff Donahue, Karen Simonyan

Despite recent progress in generative image modeling, successfully generating high-resolution, diverse samples from complex datasets such as ImageNet remains an elusive goal. To this end, we train Generative Adversarial Networks at the largest scale yet attempted, and study the instabilities specific to such scale. We find that applying orthogonal regularization to the generator renders it amenable to a simple "truncation trick," allowing fine control over the trade-off between sample fidelity and variety by reducing the variance of the Generator's input. Our modifications lead to models which set the new state of the art in class-conditional image synthesis. When trained on ImageNet at 128x128 resolution, our models (BigGANs) achieve an Inception Score (IS) of 166.5 and Frechet Inception Distance (FID) of 7.4, improving over the previous best IS of 52.52 and FID of 18.6.

ARTBREEDER

Extend your imagination

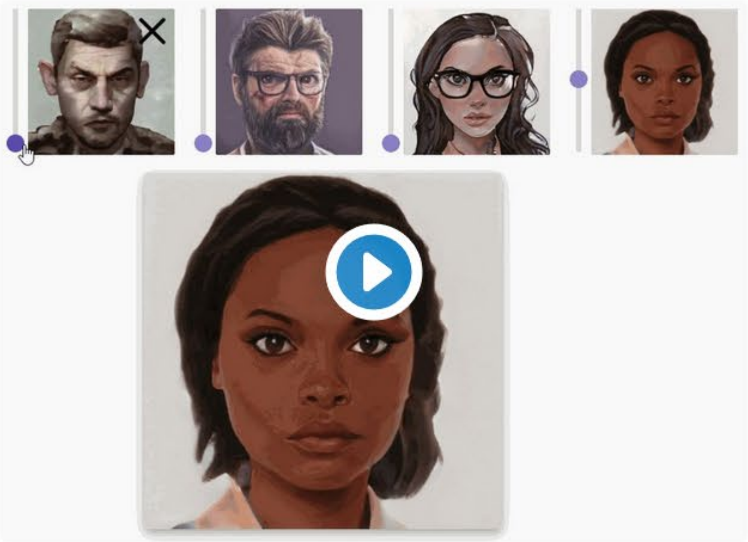
Thousands of users have collectively made 59946402 images

View Gallery | Start | Watch Intro

Bay Raitt
@bayraitt

Artbreeder is a nuclear powered pencil.

have a spin remixing some of mine here:artbreeder.com/bayraitt/starr…#artbreeder #ai #ganbreeder #conceptart #comics



♡ 8,991 2:29 AM - Sep 18, 2019

2,942 people are talking about this

Henry Lynch
@HenryLynch_Art

'New World Officer'

Character created with AI image breeding and paint-over. Fast and interesting. It is the future. @Artbreeder 👏#ConceptArt #CharacterDesign #SciFi #AI #ConceptArtist

♡ 58 5:18 AM - Oct 9, 2019

See Henry Lynch's other Tweets

TELTHONA
@telthona

I generated bunch of concepts with new AI powered website - it's the best AI app that i tried so far! It's amaizng for creature exploration, mood thumbnailing and more ❤️ finally AI that i can truly use in the creative process! ganbreeder.app #Aimakesart #ganbreeder

♡ 69 8:41 PM - Jun 29, 2019

See TELTHONA's other Tweets

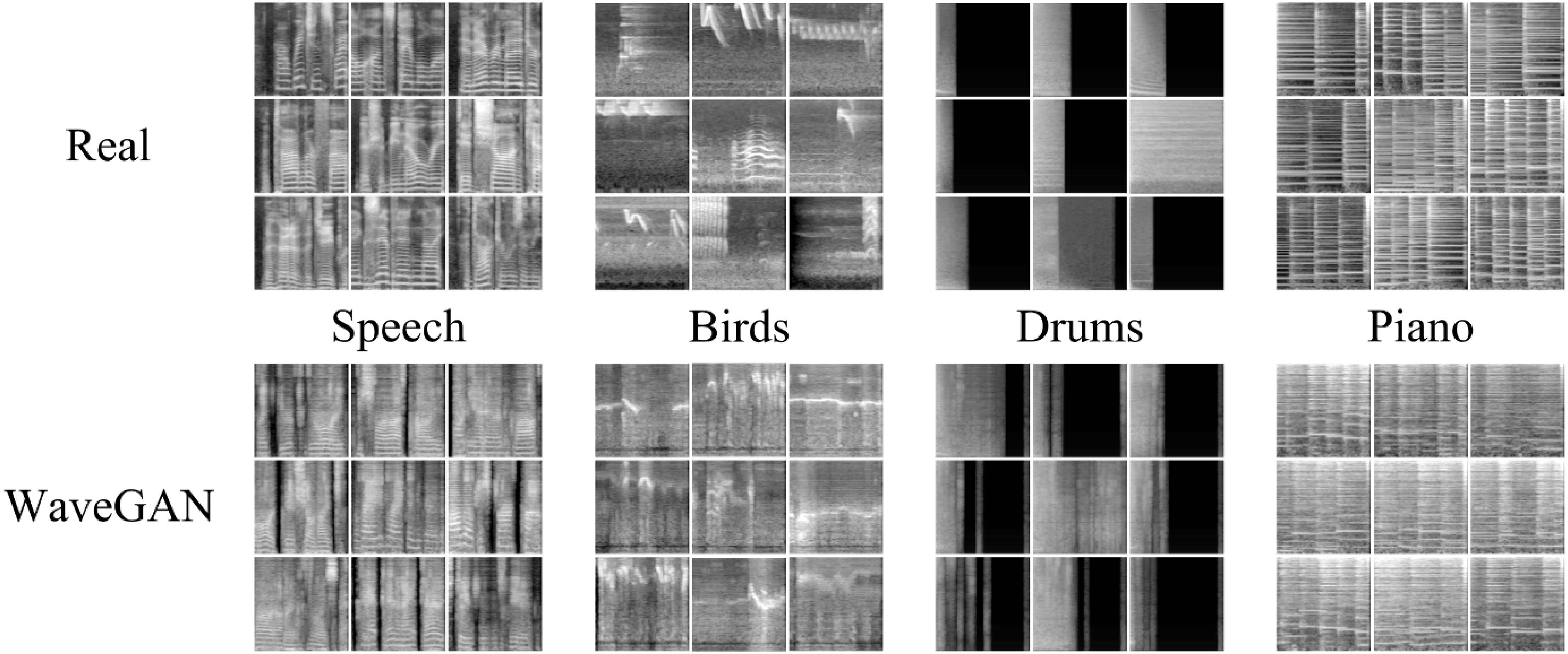
Facebook Removes Accounts With AI-Generated Profile Photos

Researchers said it appears to be the first use of artificial intelligence to support an inauthentic social media campaign.

ILLUSTRATION: WIRED STAFF; DIMITRI OTIS/GETTY IMAGES

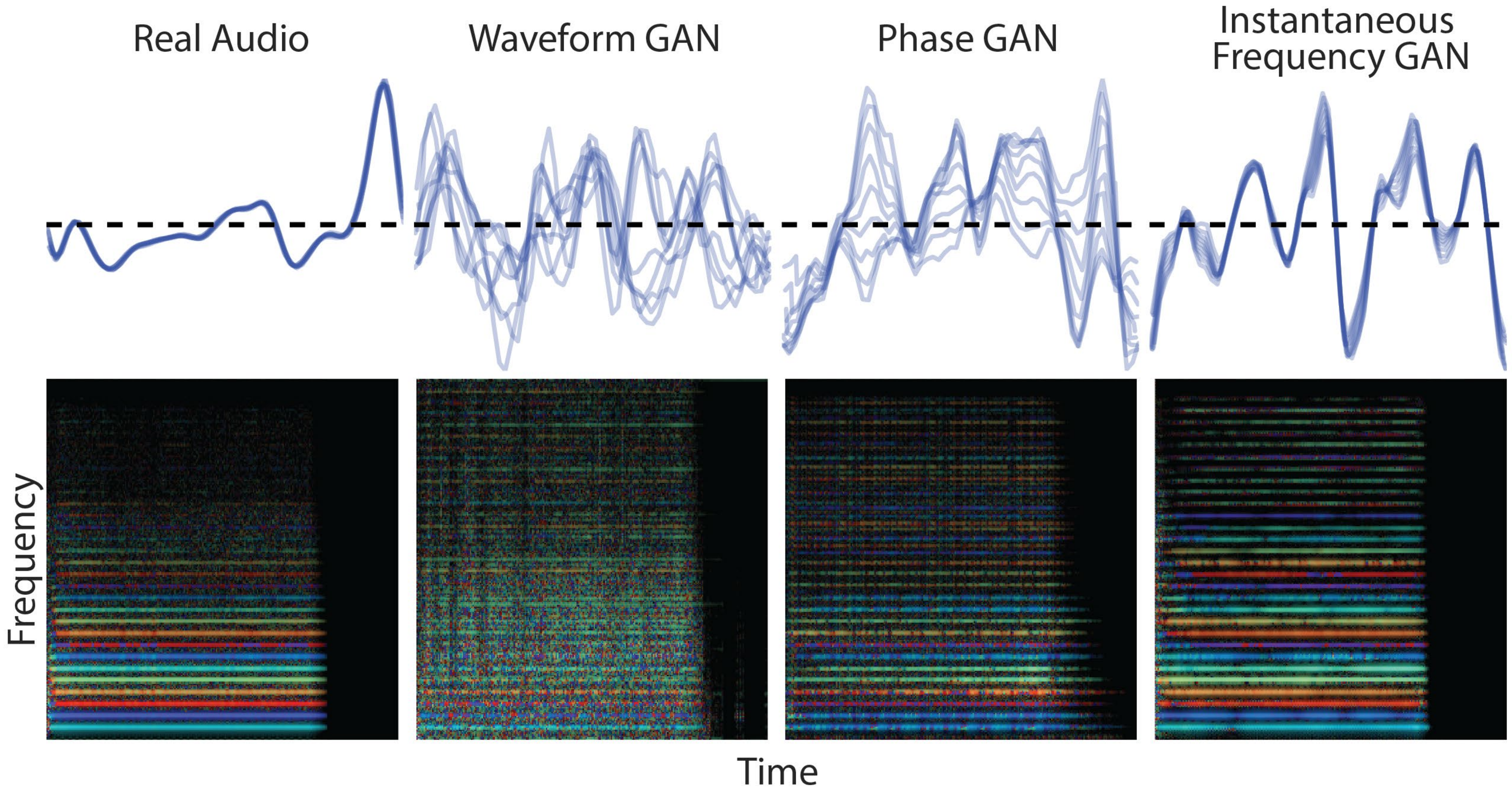
https://www.wired.com/story/facebook-removes-accounts-ai-generated-photos/

Audio and GANs: WaveGAN



https://github.com/chrisdonahue/wavegan

GANSynth (2019)



https://magenta.tensorflow.org/gansynth

More on audio generation in the next lecture

[Submitted on 16 Nov 2018]

Generating Albums with SampleRNN to Imitate Metal, Rock, and Punk Bands

CJ Carr, Zack Zukowski

This early example of neural synthesis is a proof-of-concept for how machine learning can drive new types of music software. Creating music can be as simple as specifying a set of music influences on which a model trains. We demonstrate a method for generating albums that imitate bands in experimental music genres previously unrealized by traditional synthesis techniques (e.g. additive, subtractive, FM, granular, concatenative). Raw audio is generated autoregressively in the time-domain using an unconditional SampleRNN. We create six albums this way. Artwork and song titles are also generated using materials from the original artists' back catalog as training data. We try a fully-automated method and a human-curated method. We discuss its potential for machine-assisted production.

Comments:	3 pages
Subjects:	**Sound (cs.SD)**; Audio and Speech Processing (eess.AS)
Journal reference:	Proceedings of the 6th International Workshop on Musical Metacreation (MUME 2018)
Cite as:	arXiv:1811.06633 **[cs.SD]**
	(or arXiv:1811.06633v1 **[cs.SD]** for this version)

http://dadabots.com

- Applications using GANs
  - Font generation
  - Anime character generation
  - Interactive Image generation
  - Text2Image (text to image)
  - 3D Object generation
  - Image Editing
  - Face Aging
  - Human Pose Estimation
  - Domain-transfer (e.g. style-transfer, pix2pix, sketch2image)
  - Image Inpainting (hole filling)

https://github.com/nashory/gans-awesome-applications

Glow: Generative Flow with Invertible 1×1 Convolutions

Diederik P. Kingma[*†], **Prafulla Dhariwal**[*]
[*]OpenAI
[†]Google AI

Abstract

Flow-based generative models (Dinh et al., 2014) are conceptually attractive due to tractability of the exact log-likelihood, tractability of exact latent-variable inference, and parallelizability of both training and synthesis. In this paper we propose *Glow*, a simple type of generative flow using an invertible 1×1 convolution. Using our method we demonstrate a significant improvement in log-likelihood on standard benchmarks. Perhaps most strikingly, we demonstrate that a flow-based generative model optimized towards the plain log-likelihood objective is capable of efficient realistic-looking synthesis and manipulation of large images. The code for our model is available at `https://github.com/openai/glow`.

1 Introduction

Two major unsolved problems in the field of machine learning are (1) data-efficiency: the ability to learn from few datapoints, like humans; and (2) generalization: robustness to changes of the task or its context. AI systems, for example, often do not work at all when given inputs that are different

[*]Equal contribution.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

Figure 1: Synthetic celebrities sampled from our model; see Section 3 for architecture and method, and Section 5 for more results.

Summary

- A *generative model* is needed when there are multiple possible outputs for a single input, and the outputs are not discrete-valued
- GANs are the dominant type of generative model
- Training GANs may be unstable, but things are improving rapidly
- StyleGAN 2 and BigGAN have been quickly adopted by artists and designers

A key takeaway: infinite ways to combine basic compute graph blocks