Generative adversarial networks for computer vision

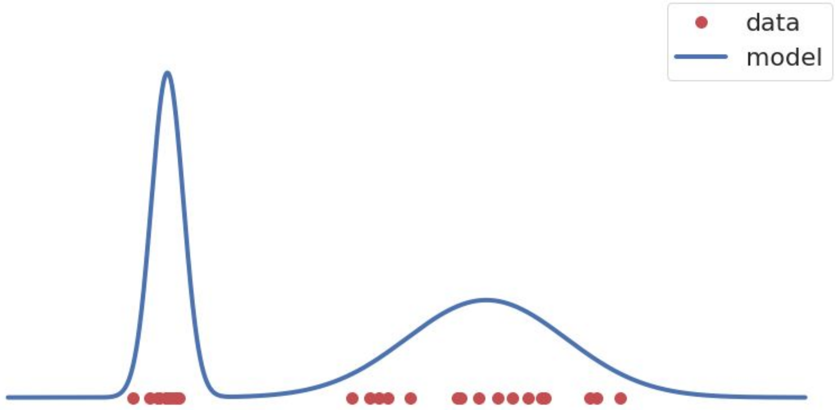
Mihaela Rosca
DeepMind & University College London

Disclaimer

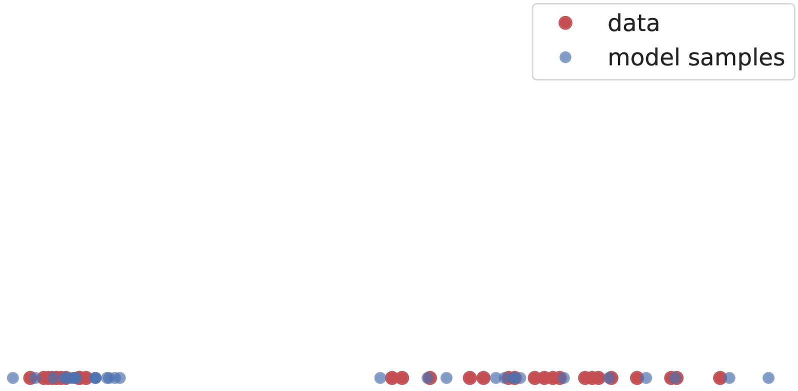
- This is one view of the literature in the field, there are many others.
- Each citation is meant to be used as a representative example. Further related work can be found using connectedpapers.com.

Goal: distribution learning from examples

Goal: distribution learning from examples



Goal: distribution learning from examples

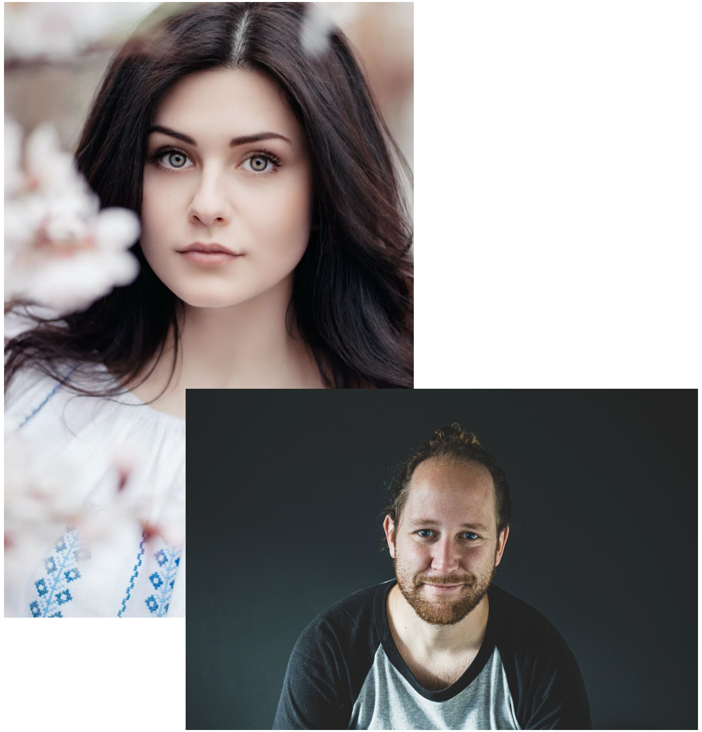


Generative adversarial networks

The data

Natural images

Faces

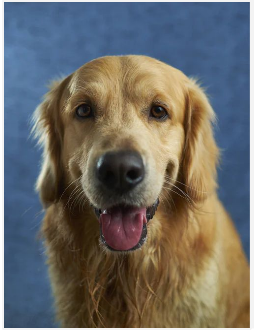


Other

Distribution learning with two player games

Discriminator

Learns to distinguish between real and generated data.



vs

Created by Minus icons
from Noun Project

Generator

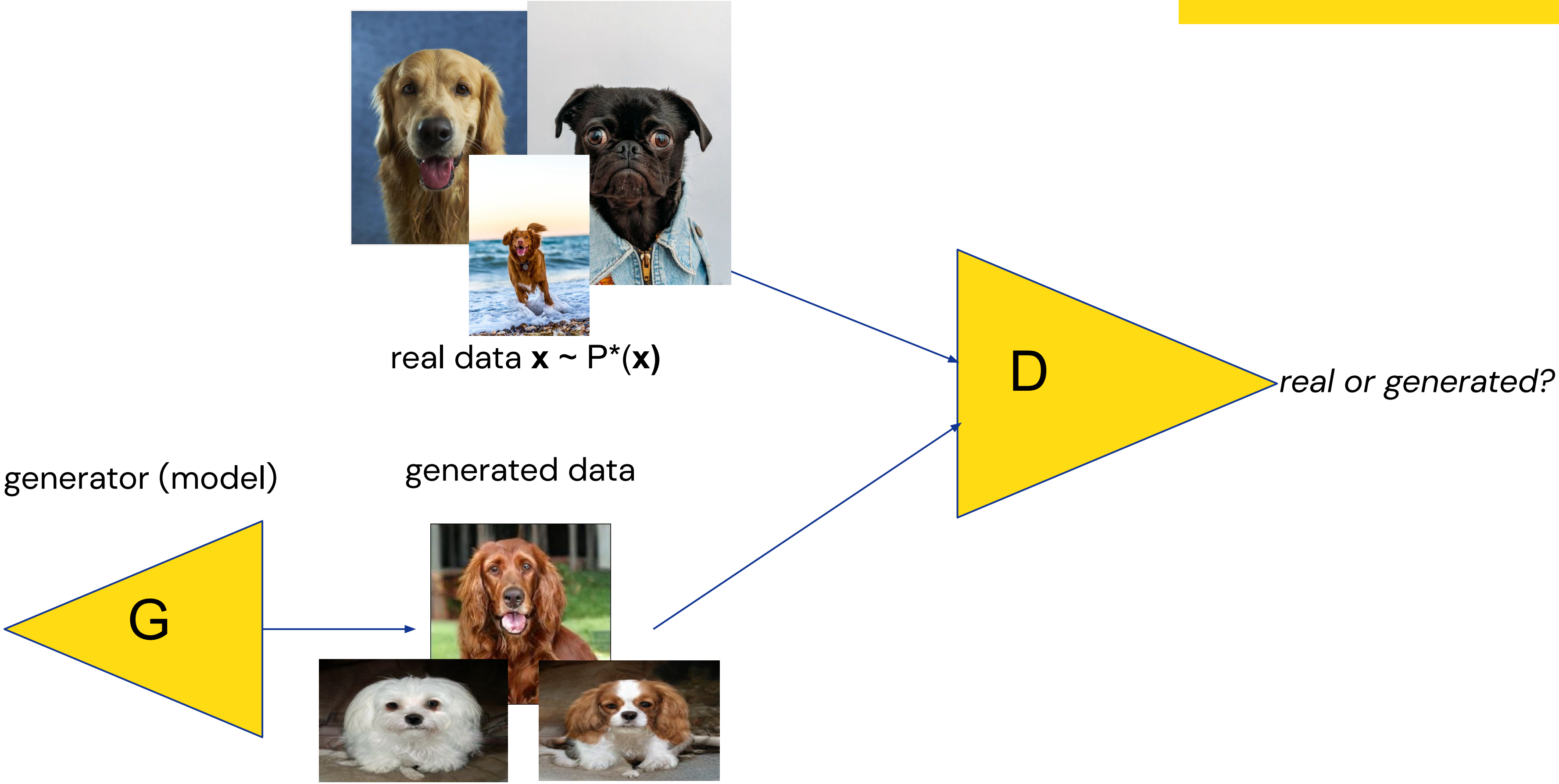
Learns to generate data to “fool” the discriminator.

Created by Minus icons
from Noun Project

Generative adversarial networks

Want to learn more?

Goodfellow, et al. **Generative adversarial networks..** Neural Information Processing Systems **(2014)**



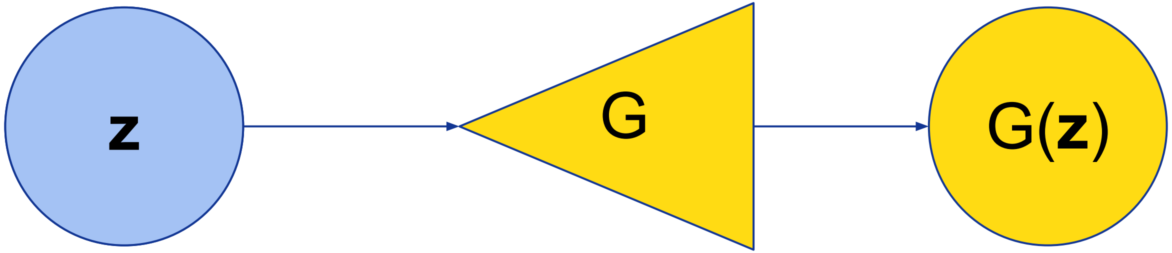
Slide thanks to Jeff Donahue.

The generator

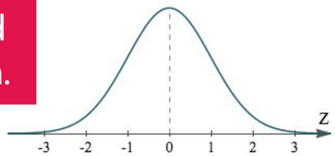
latent ("noise") vector
$\mathbf{z} \sim P(\mathbf{z})$

generator G:
a deep neural network

generated data
$G(\mathbf{z})$



Generator input is random noise to account for spread of data distribution.



Generative adversarial networks

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} \underbrace{\mathbb{E}_{p^*(\mathbf{x})}\left[\log \mathcal{D}_{\boldsymbol{\phi}}(\mathbf{x})\right]}_{\text{log-probability that D correctly predicts real data } \mathbf{x} \text{ are real}} + \underbrace{\mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})}\left[\log(1 - \mathcal{D}_{\boldsymbol{\phi}}(\mathbf{x}))\right]}_{\text{log-probability that D correctly predicts generated data are generated}}$$

Generative adversarial networks

Want to learn more?

Goodfellow, et al. **Generative adversarial networks..** Neural Information Processing Systems (**2014**)

$$\min_G \max_D V(D,G) = \underbrace{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]}_{\text{log-probability that D correctly predicts real data } \mathbf{x} \text{ are real}} + \underbrace{\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]}_{\text{log-probability that D correctly predicts generated data G}(\mathbf{z}) \text{ are generated}}$$

discriminator's (D) goal: **maximize** prediction accuracy

generator's (G) goal: **minimize** D's prediction accuracy,
by **fooling** D into believing its outputs G(**z**) are real as often as possible

Are GANs learning the true distribution?

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} \mathbb{E}_{p^*(\mathbf{x})} \left[\log \mathcal{D}_{\boldsymbol{\phi}}(\mathbf{x}) \right] + \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \left[\log(1 - \mathcal{D}_{\boldsymbol{\phi}}(\mathbf{x})) \right]$$

If the discriminator (D) is optimal:
the generator is minimizing the Jensen Shannon divergence between the true and generated distributions.

Connection to optimality:

$$JSD(p^* || p_\theta) = 0 \implies p_\theta = p^*$$

Discriminators as learned loss functions

Want to learn more?

Arora, et al **Generalization and Equilibrium in Generative Adversarial Nets.** ICLR **(2017)**

$$\min_G \boxed{\max_D V(D, G)}$$

We can think of D (the discriminator) as learning a loss function between the data and model distribution that can provide useful gradients to the model.

Algorithms

$$\min_G \boxed{\max_D V(D, G)}$$

min max problems require full optimization between the two players but that is not computationally feasible.

Instead we perform alternating update schemes: update the discriminator a few times for each generator update.

Algorithms

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

---

for number of training iterations **do**

  for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

From Goodfellow et all, 2014.

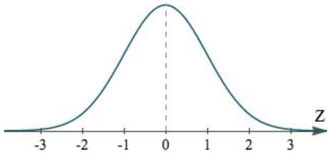
Slide thanks to Jeff Donahue.

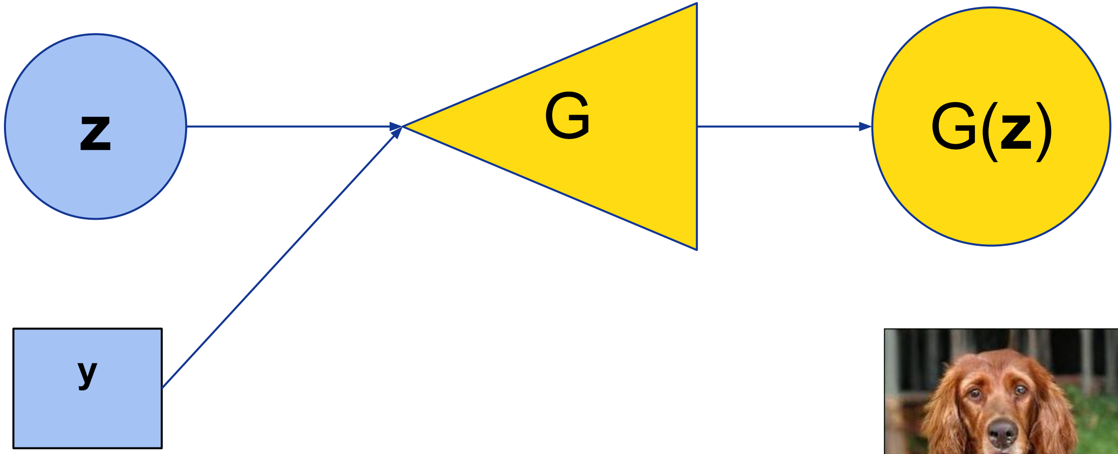
Conditional GANs

latent ("noise") vector
z ~ P(**z**)

generator G:
a deep neural network

generated data
G(**z**)

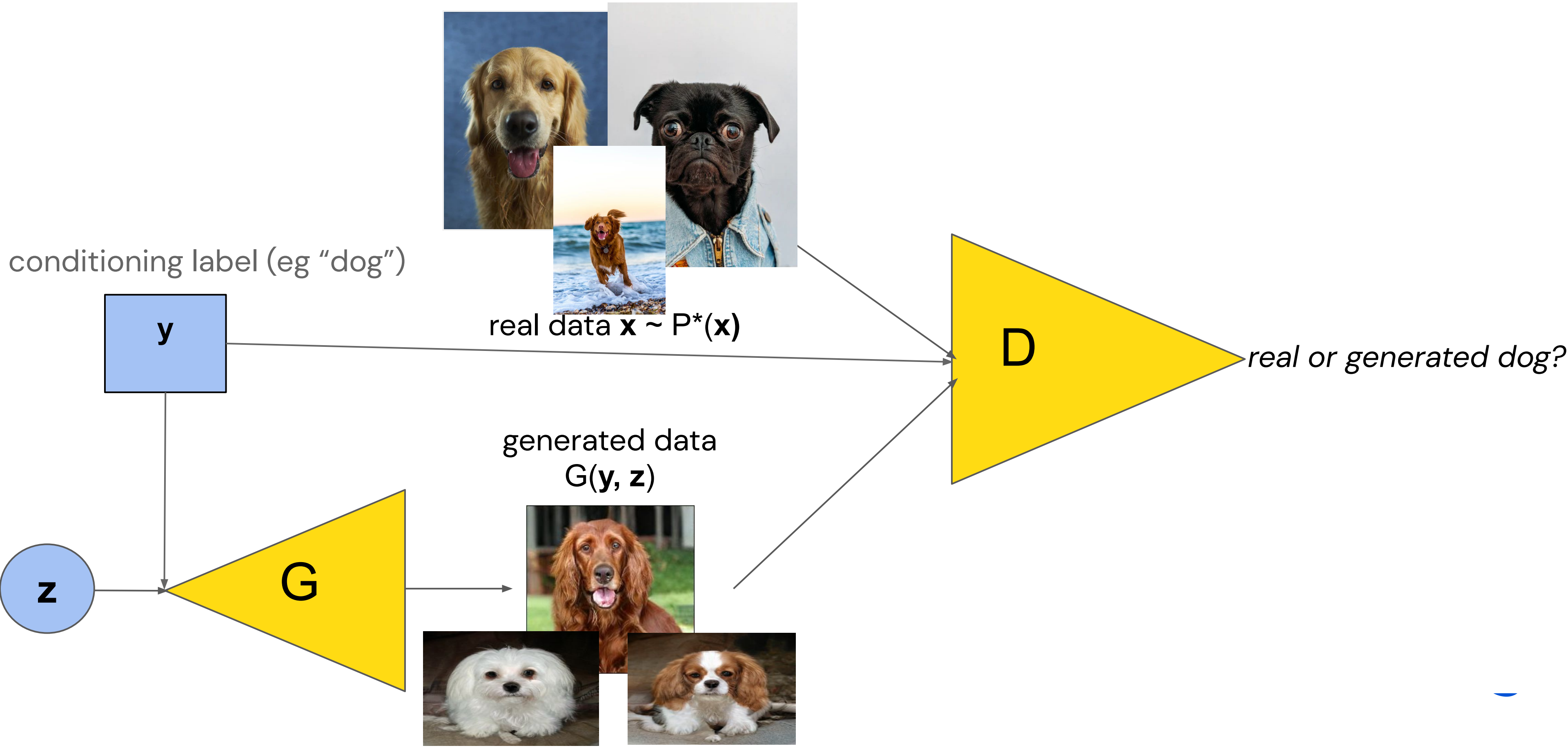




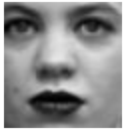
Add conditioning generation to specify information about generated sample.

Slide thanks to Jeff Donahue.

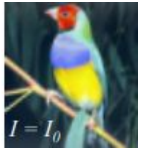
Conditional GANs



Computer vision applications



Goodfellow, et al. **Generative adversarial networks.** NIPS (**2014**)



Denton, et al. **Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks.** NIPS (**2015**)

Radford et al. **Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks** ICLR (**2015**)

Miyato et al. **Spectral normalization for Generative Adversarial Networks** ICLR (**2018**)

Karras et al. **Large Scale GAN Training for High Fidelity Natural Image Synthesis** ICLR (**2018**)

Brock et al. **Large Scale GAN Training for High Fidelity Natural Image Synthesis** ICLR (**2019**)

Karras et al. **A Style-Based Generator Architecture for Generative Adversarial Networks** CVPR (**2019**)

ela Rosca, 2021

Still image generation

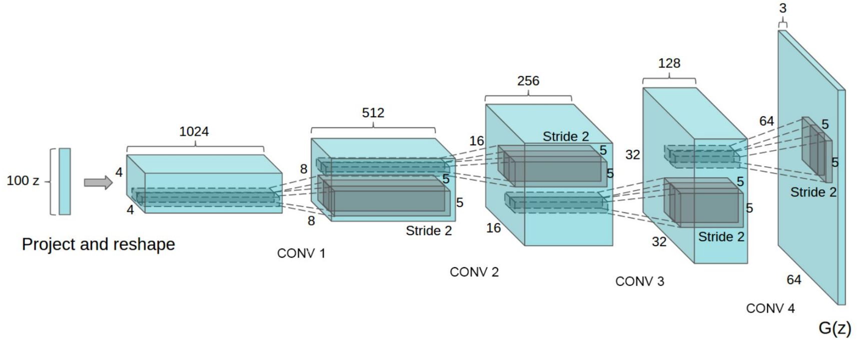
We will look at a few examples but highlight a few threads:

- architectures matter (a lot!)
- optimization matters
- conditioning matters
- progressively growing image resolution helps

Architectures matter

architectures: deep convolutional resnets

The importance of the architecture was first shown in DCGAN, but then a long tradition of papers showing the importance of architectures started.



Want to learn more?

Radford et all, **Unsupervised representation learning with deep convolutional generative adversarial networks** ICLR(2016)

Mihaela Rosca, 2021

Architectures matter

self attention: conv nets have local range - increase range using self attention.

Want to learn more?

Zhu et all, **Self-Attention Generative Adversarial Networks**, ICML(2019)

BigGAN - Class conditional generation

Uses deep resnets and self attention.

Want to learn more?

Brock et all **Large scale gan training for high fidelity natural image synthesis** Neurips (2018)

Tricks in BigGAN

large batch sizes

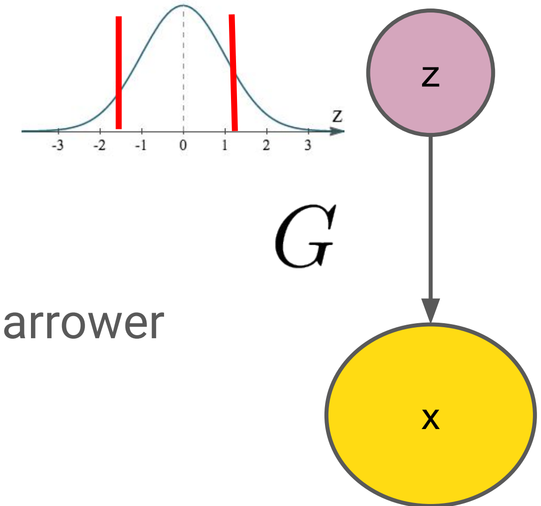
- optimization in two player games can be tricky
- large batch sizes increase stability and improve performance

truncation trick

- increase sample quality by sampling

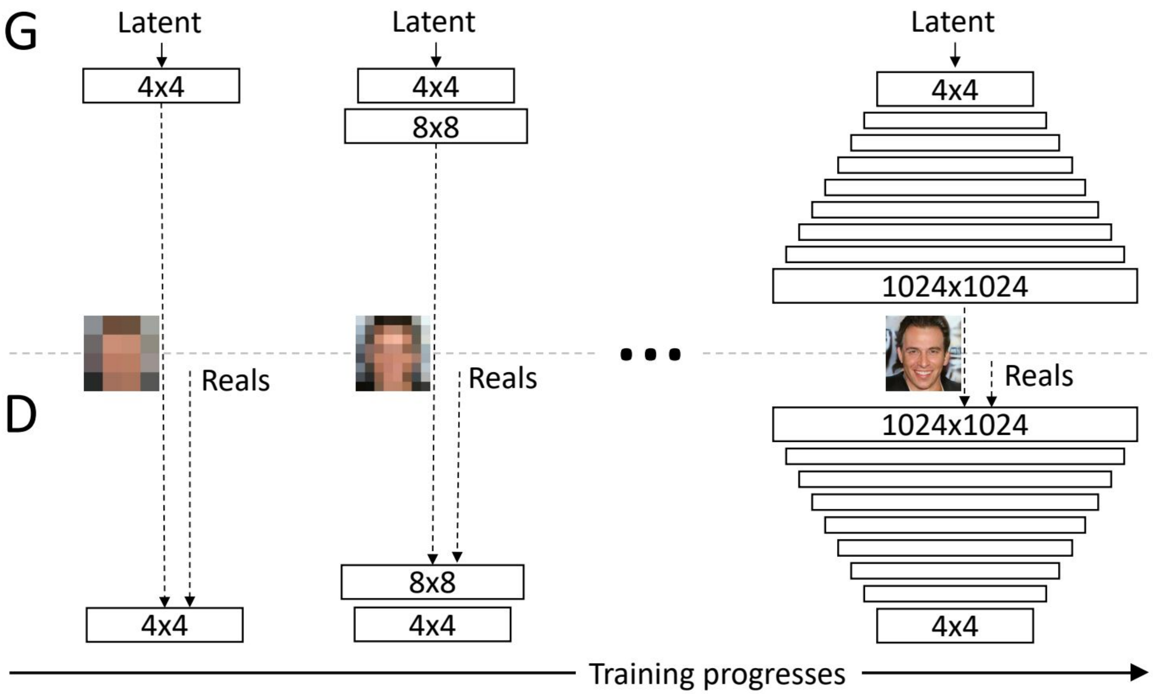
  the input of the generator from a narrower

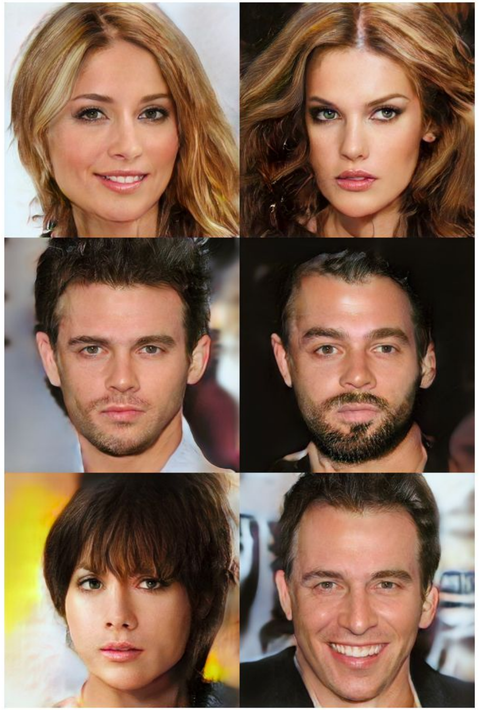
  distribution



ProgressiveGAN

Want to learn more?
Karras et all, **Progressive Growing of GANs for Improved Quality, Stability, and Variation**, ICLR (2019)





Mihaela Rosca, 2021

StyleGAN

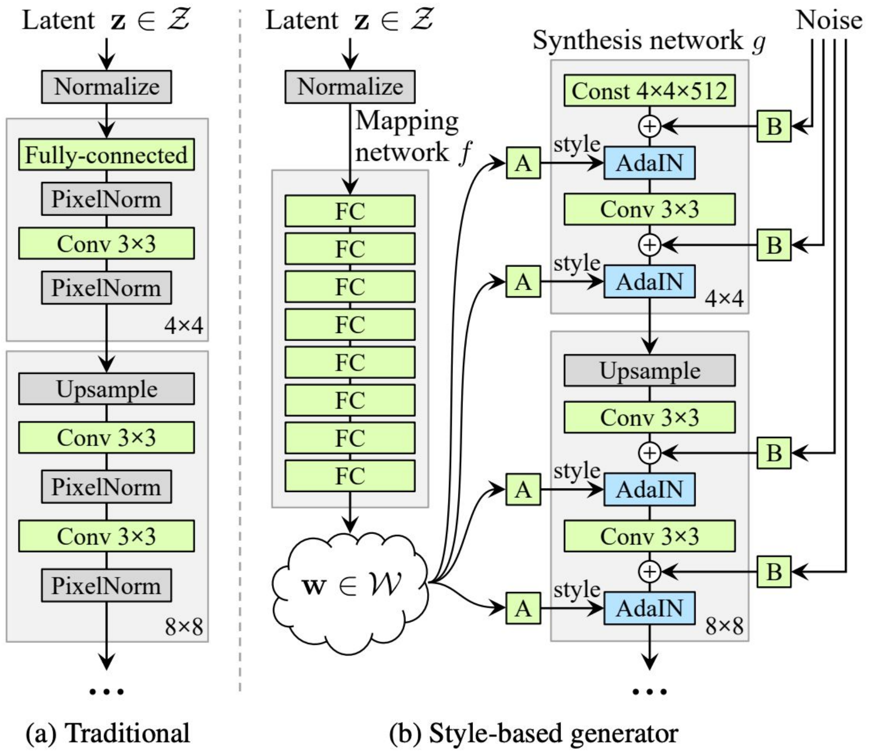
Photo realistic image equality at high resolution.

Want to learn more?

Karras et al,**A Style-Based Generator Architecture for Generative Adversarial Networks** ICLR (2019)

StyleGAN

- use progressive generation
- learn an embedding "style" vector that you inject at throughout the network



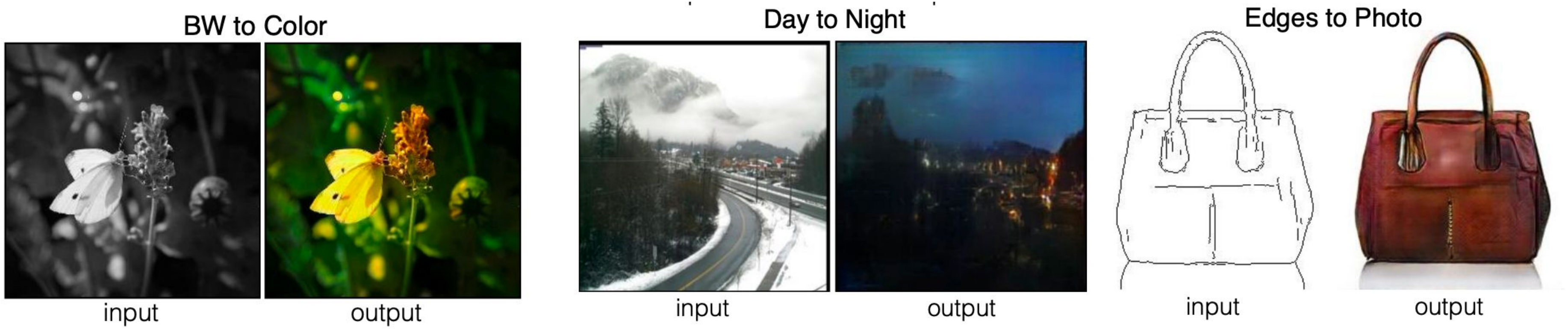
Mihaela Rosca, 2021

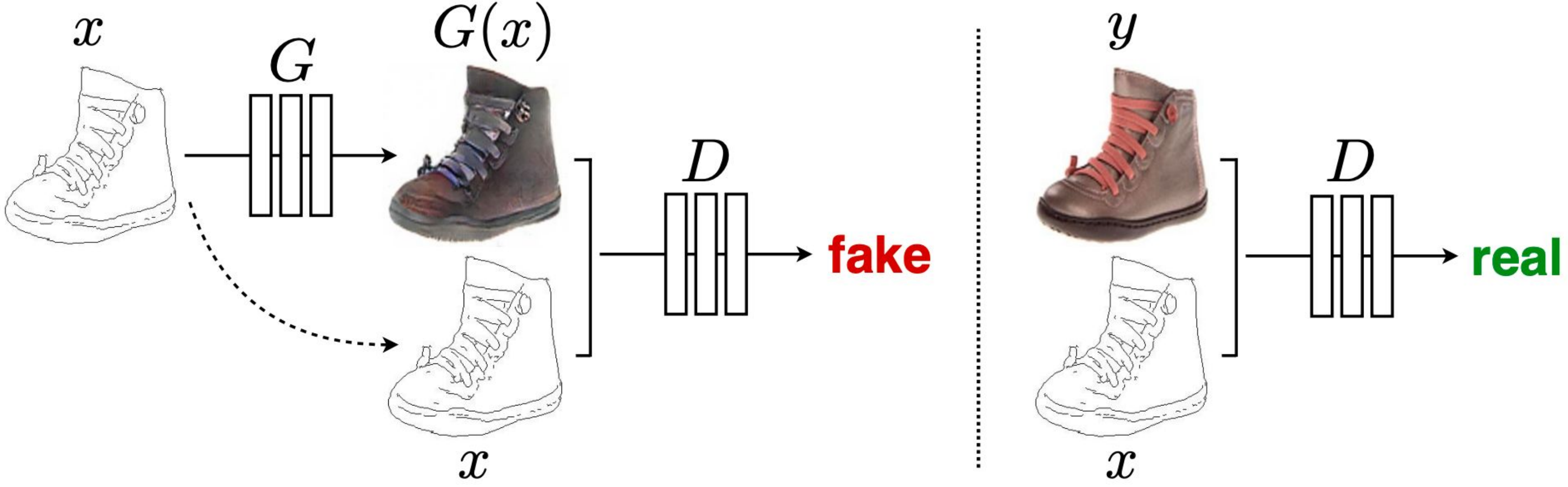
Interpolating between style embeddings

Mihaela Rosca, 2021

Image to image translation

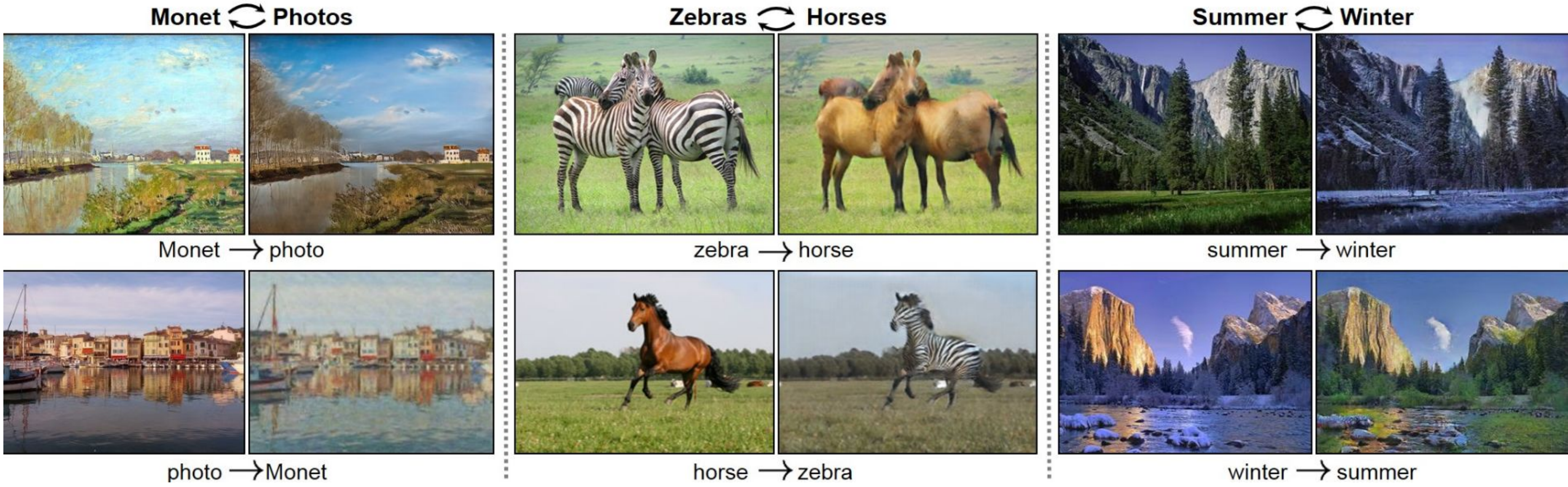
Supervised image to image translation - Pix2Pix





Requires paired examples.

Unsupervised image to image translation



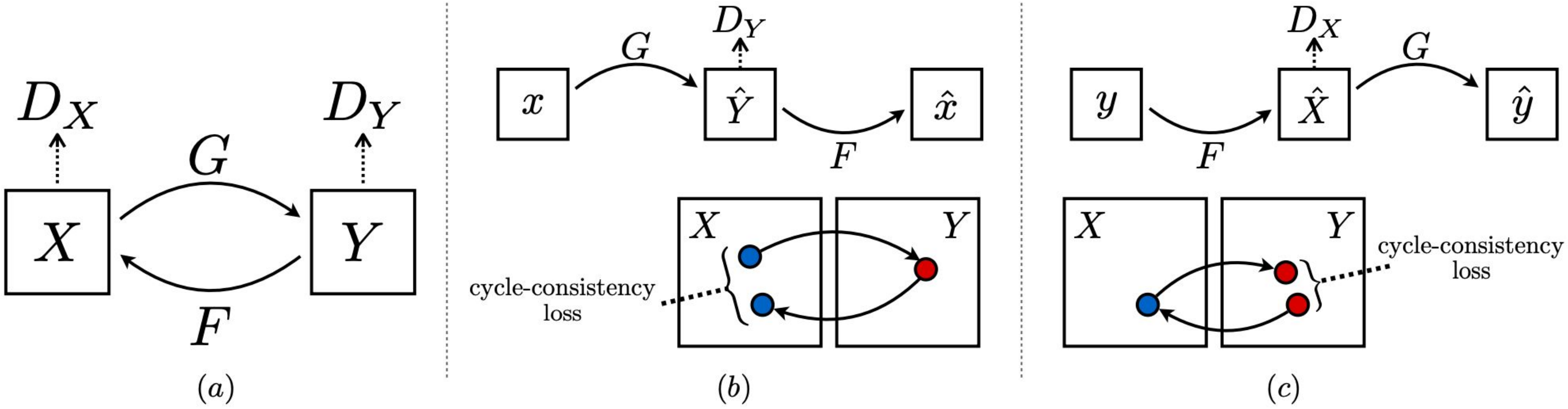
Want to learn more?

Zhu et Il, **Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks,** ICCV (2017)

CycleGAN

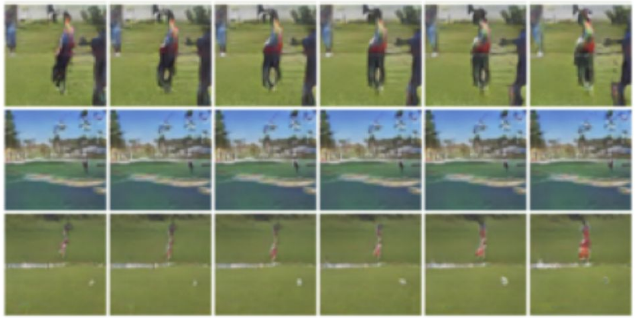
Does not require paired data!

zebra? **horse?**



Beyond still images

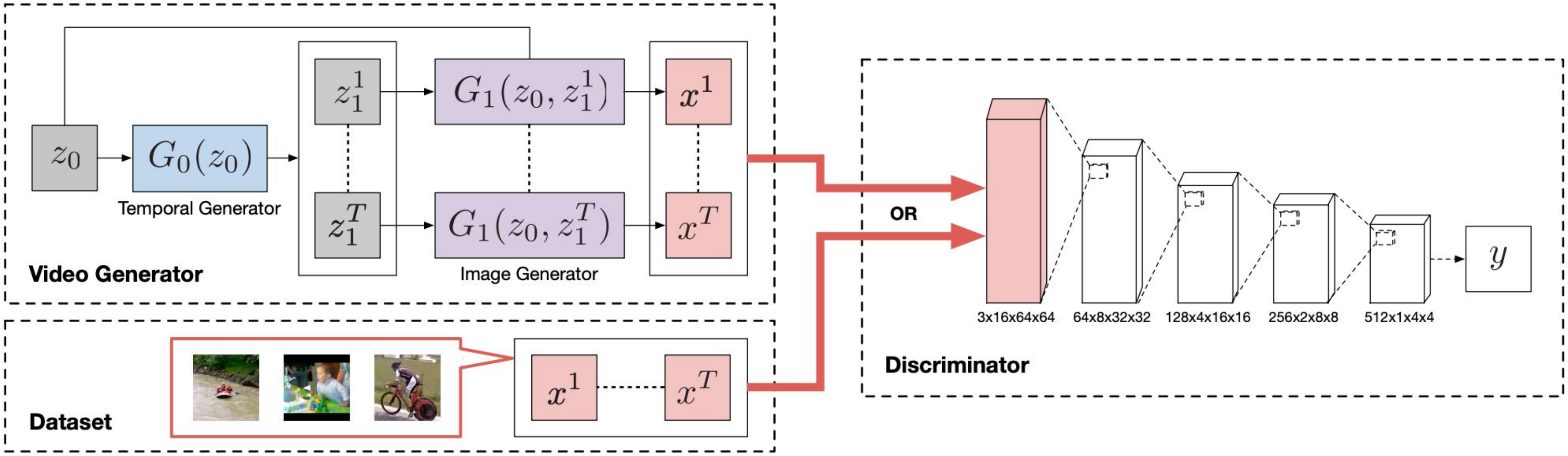
TGAN - video generation



Want to learn more?

Saito et al ,**Temporal Generative Adversarial Nets with Singular Value Clipping** ICCV (2017)

TGAN - video generation



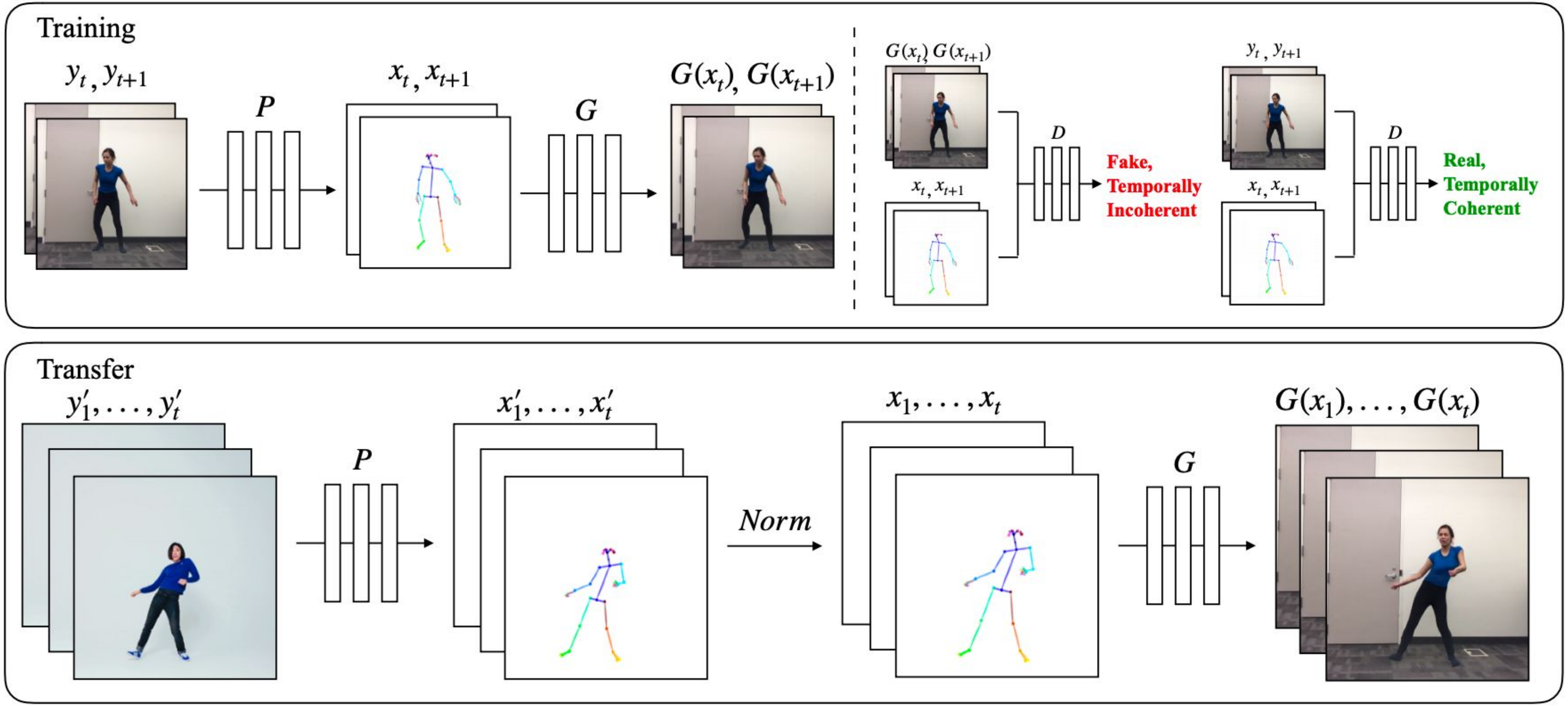
Source of noise matters: you have 1 for the entire sequence (determines sequence theme), and one per image!

Motion transfer - everybody dance now!

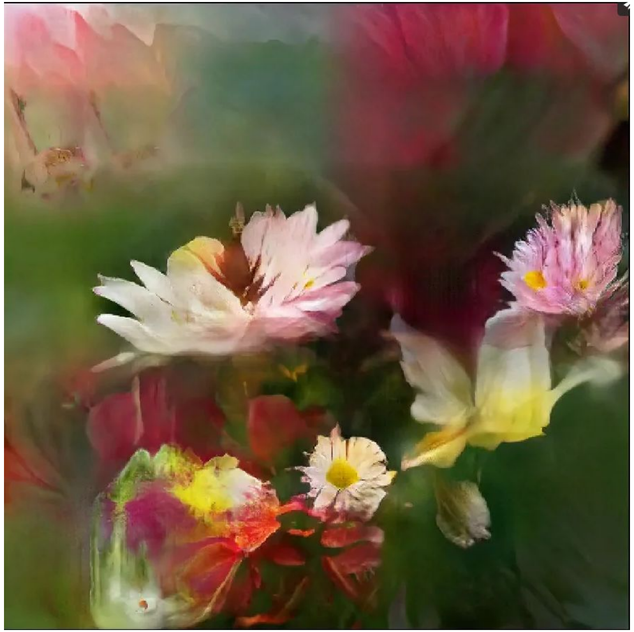
Want to learn more?

Chan et all,**Everybody dance now** ICCV (2019)

Motion transfer



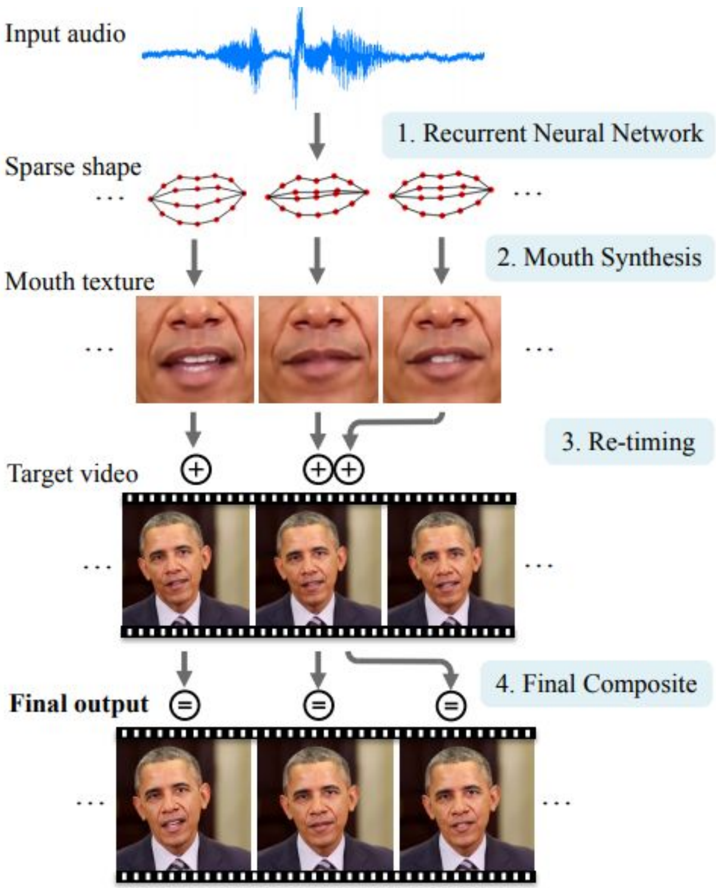
Art



Credit: Memo Akten (http://www.memo.tv)
http://www.memo.tv/portfolio/learning-to-see/

Ethical considerations

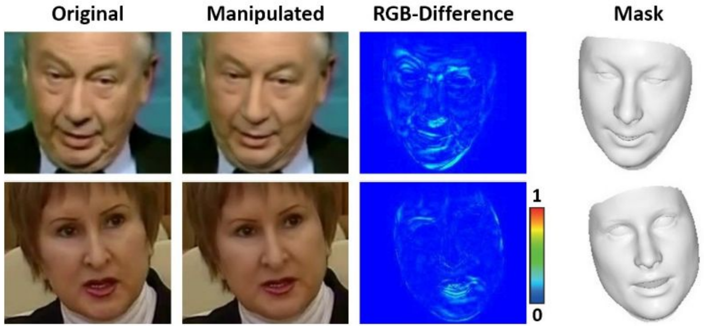
Examples - lip sync

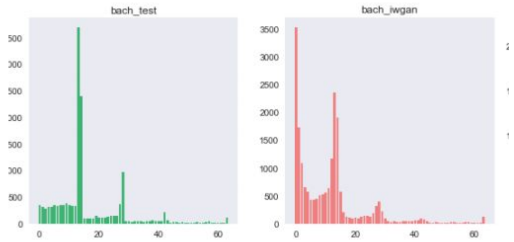


https://grail.cs.washington.edu/projects/AudioToObama/siggraph17_obama.pdf

Solutions

Data forensics

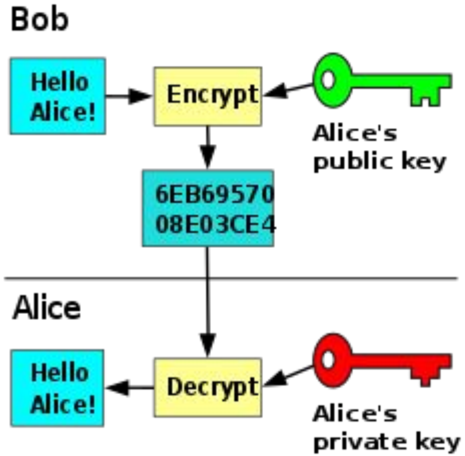




(a) Histogram of note durations

https://arxiv.org/pdf/1807.04919.pdf
https://arxiv.org/abs/1803.09179

Cryptography



Awareness

Public awareness

- critical thinking regarding media sources

Increase media awareness

- fact checking
- source verification

Our duty

Is to think about the ethical implications of the tools we build.

Conclusion

GANs have been incredibly successful in computer vision.

Conditioning information, architectures and progressive generation are key elements of the success of GANs.

The discriminator can be used to ensure temporal coherence.

Thank you!