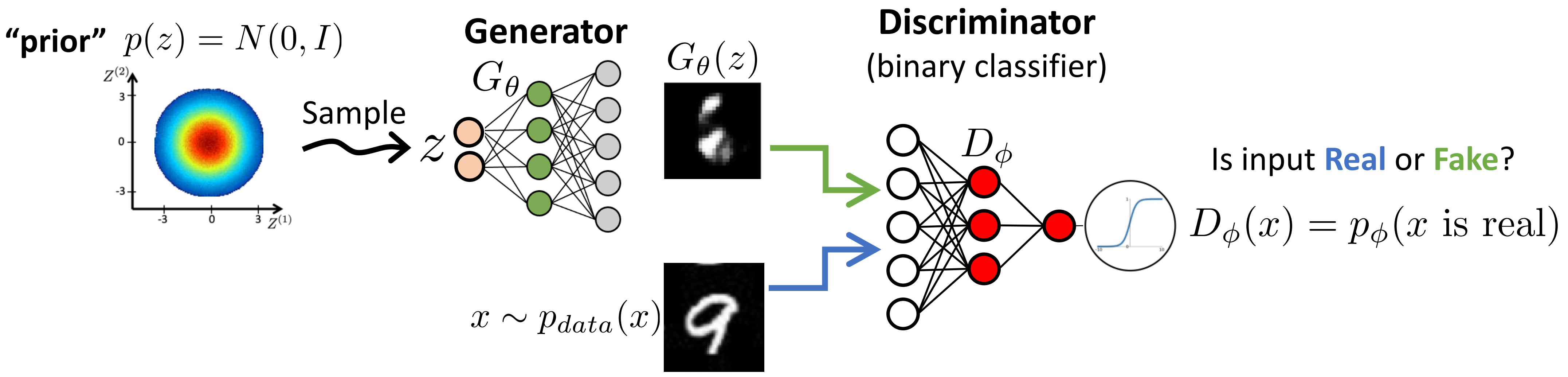
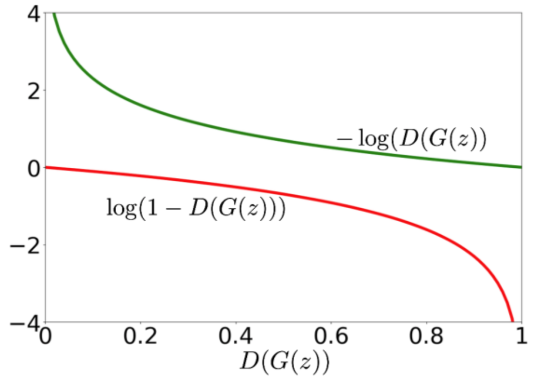
Limitations of basic GAN
and
Advanced Models

Generative Adversarial Networks



$$\{\theta', \phi'\} = \arg\min_{\theta} \max_{\phi} \mathbb{E}_{x \sim p_{data}(x)} \left[\log D_\phi(x)\right] + \mathbb{E}_{z \sim p(z)} \left[\log(1 - D_\phi(G_\theta(z)))\right]$$

Theoretically motivated
vs practical loss for G



Local Minima and Mode Collapse

Research on solutions:
-Better optimizers
-Better network architectures
-Better losses (Wasserstein, GAN-GP,…)
-…

Two potential losses for G
from previous video lecture,
one offering better gradients.

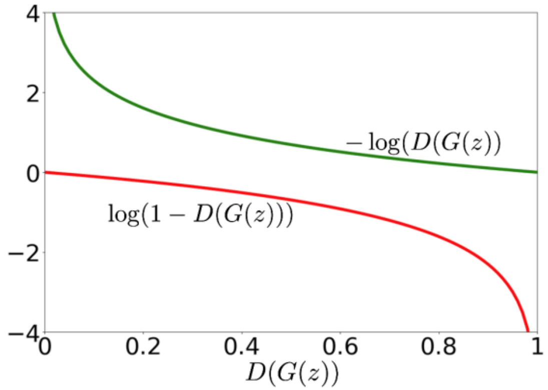
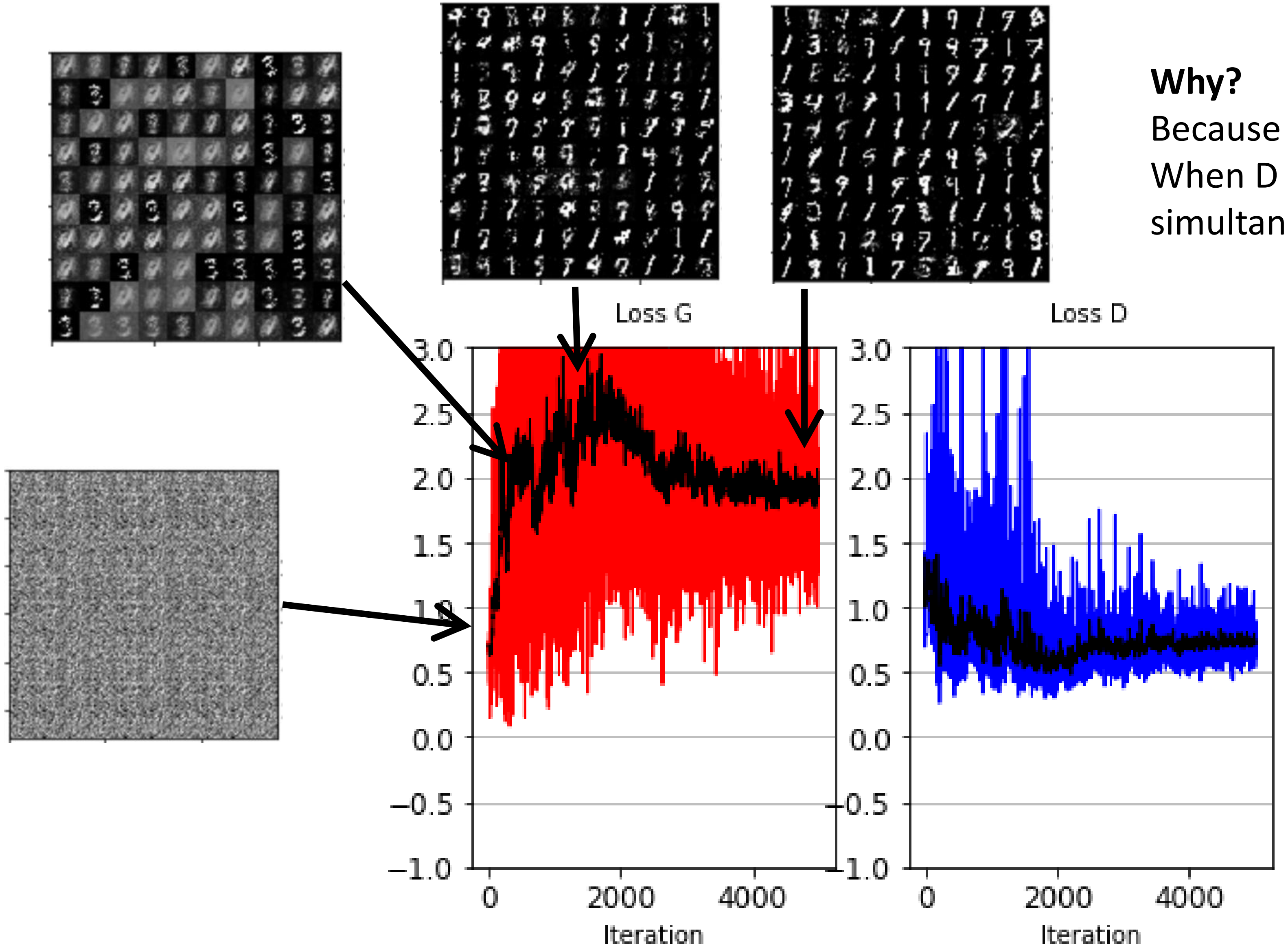


Image: Fedus et al, Many paths to equilibrium: GANS do not need to decrease a divergence at every step, 2018

Problem: Generator's Loss is not decreasing



Why?
Because D also improves during training.
When D changes, if D reduces its loss, it simultaneously increases G's loss.

Training on MNIST
(from our Tutorial on GANs)

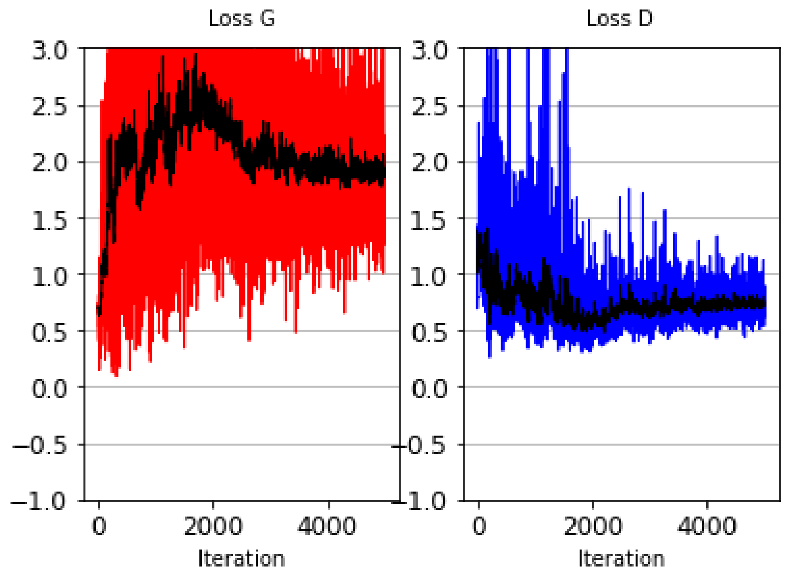
Discriminator's Accuracy as "Learned Loss function" for the Generator

Theoretical loss:

$$\mathcal{L}_G(z) = \log(1 - D_{\underline{\phi}}(G_{\underline{\theta}}(z)))$$

Practical loss:

$$\mathcal{L}_G(z) = -\log D_{\underline{\phi}}(G_{\underline{\theta}}(z))$$

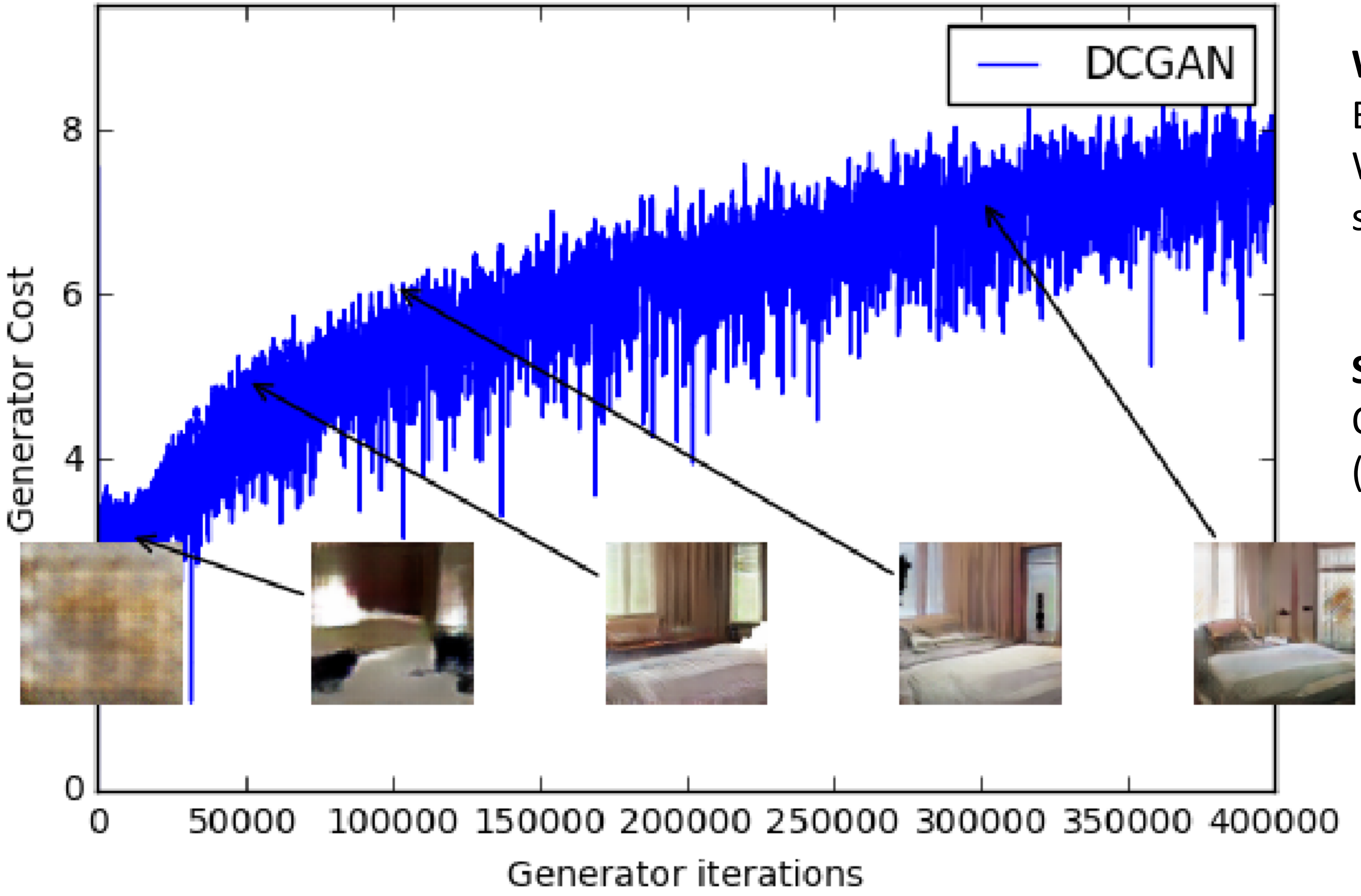


Generator's loss is **a function of the Discriminator** and its parameters.

G's loss decreases when G improves during training and pushes D(G(z)) to 1,
but also increases when D improves during training and pushes D(G(z)) to 0.

Therefore, even when results by G improve, we may see G's loss go up!

Problem: Generator's Loss is not decreasing



From: Arjovsky et al, Wasserstein GAN, 2017

Why?
Because D also improves during training.
When D changes, if D reduces its loss, it simultaneously increases G's loss.

Solution?
Open research on alternative GAN losses (e.g. Wasserstein GAN)

Limited uses of basic GAN:

How to use the basic GAN for inferring latent variables (code) of a given sample?

How to use the basic GAN for altering a feature of a specific input sample?

Can we use basic GAN for compression?

How to use the basic GAN for pre-training a supervised classifier?

We cannot, because basic GAN has no encoder x -> z.   Only generator z -> x

…but there are 1000s of extensions for solving these!

Advanced models for unsupervised learning

From: Makhzani et al, Adversarial Auto-Encoders, 2016

Adversarial Auto-Encoder

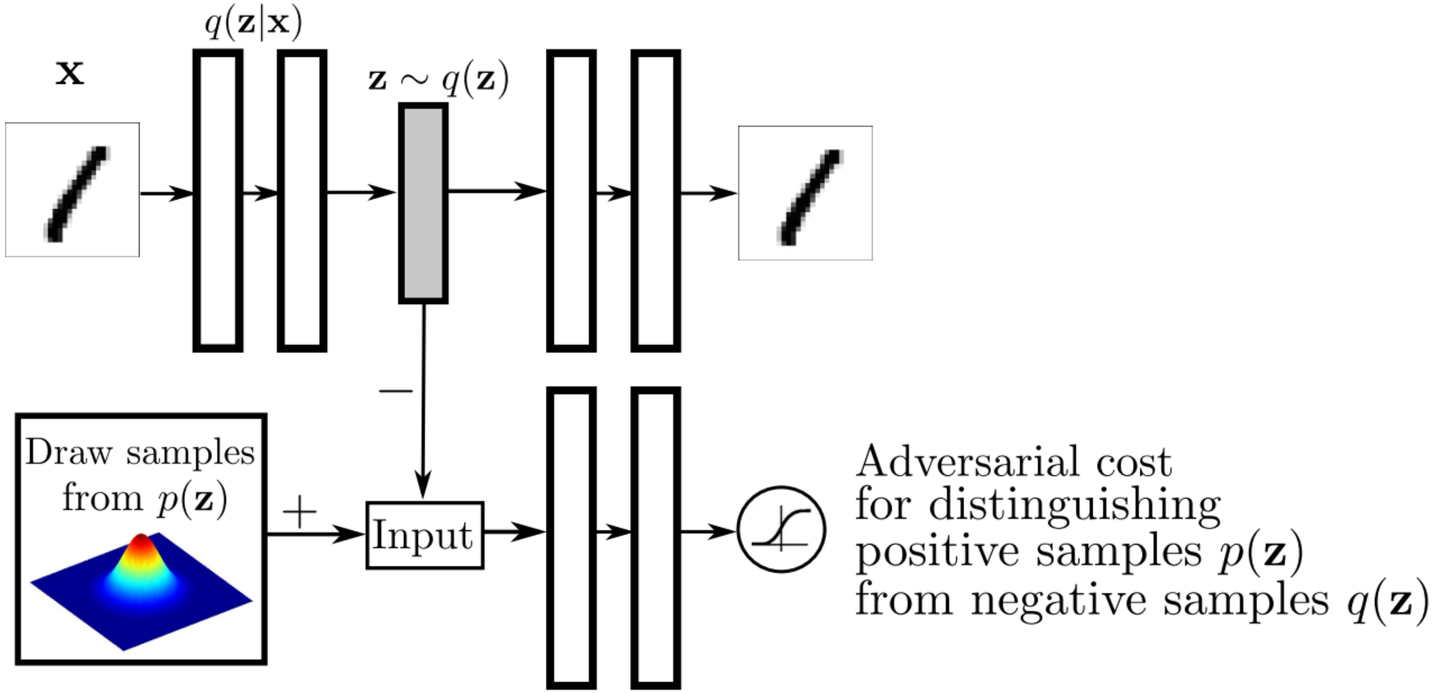


Figure 1: Architecture of an adversarial autoencoder. The top row is a standard autoencoder that reconstructs an image $\mathbf{x}$ from a latent code $\mathbf{z}$. The bottom row diagrams a second network trained to discriminatively predict whether a sample arises from the hidden code of the autoencoder or from a sampled distribution specified by the user.

From: Makhzani et al, Adversarial Auto-Encoders, 2016

VAE-GAN

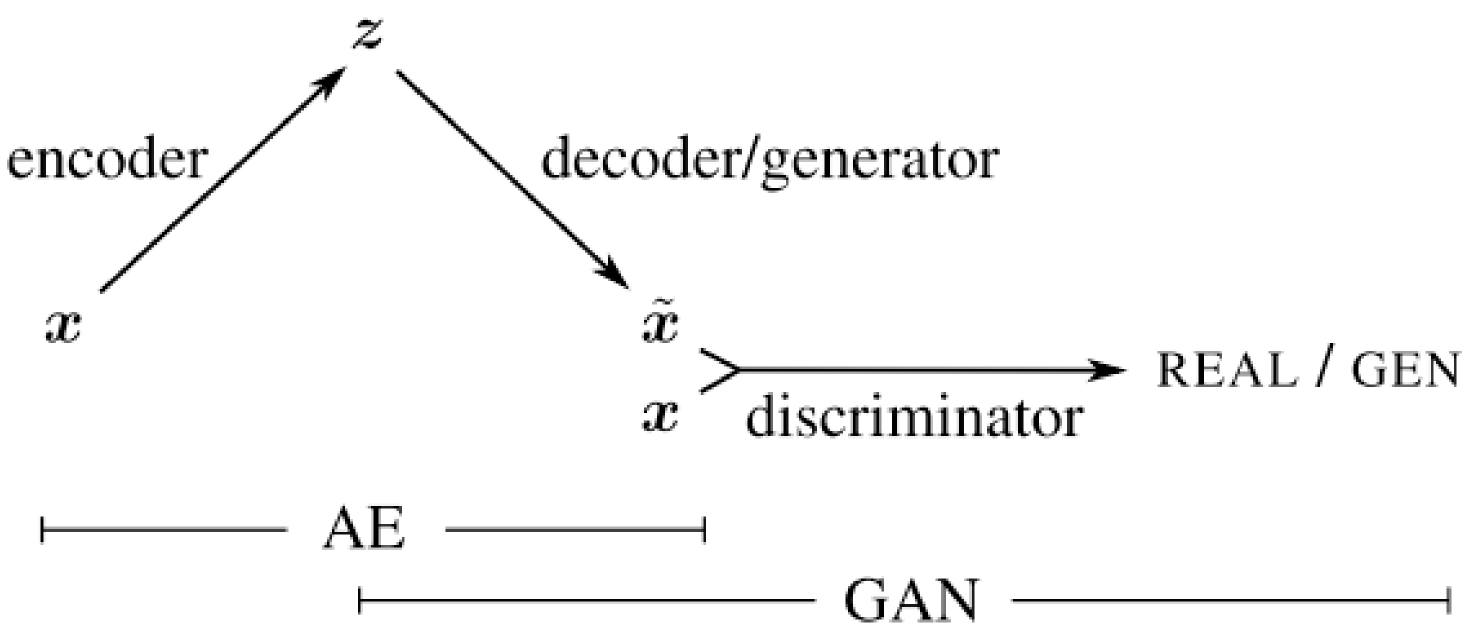


Figure 1. Overview of our network. We combine a VAE with a GAN by collapsing the decoder and the generator into one.

From: Larsen et al, Autoencoding beyond pixels using a learned similarity metric, 2016

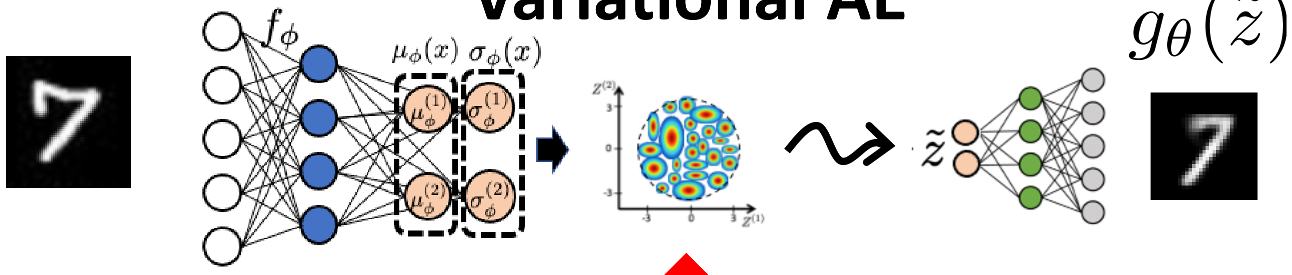
VAE-GAN

Reconstruction loss

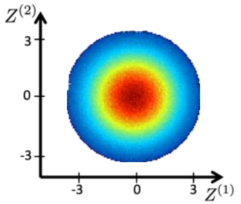
$$\cancel{\mathcal{L}_{rec}(x) = \frac{1}{d}\sum_{i}^{d}(x^{(i)} - g_{\theta}^{(i)}(\tilde{z}))^2}$$

Variational AE

$g_{\theta}(\tilde{z})$



$$\mathcal{L}_{reg}(x) = D_{KL}\left[p_{\phi}(z|x)||N(0,I)\right]$$

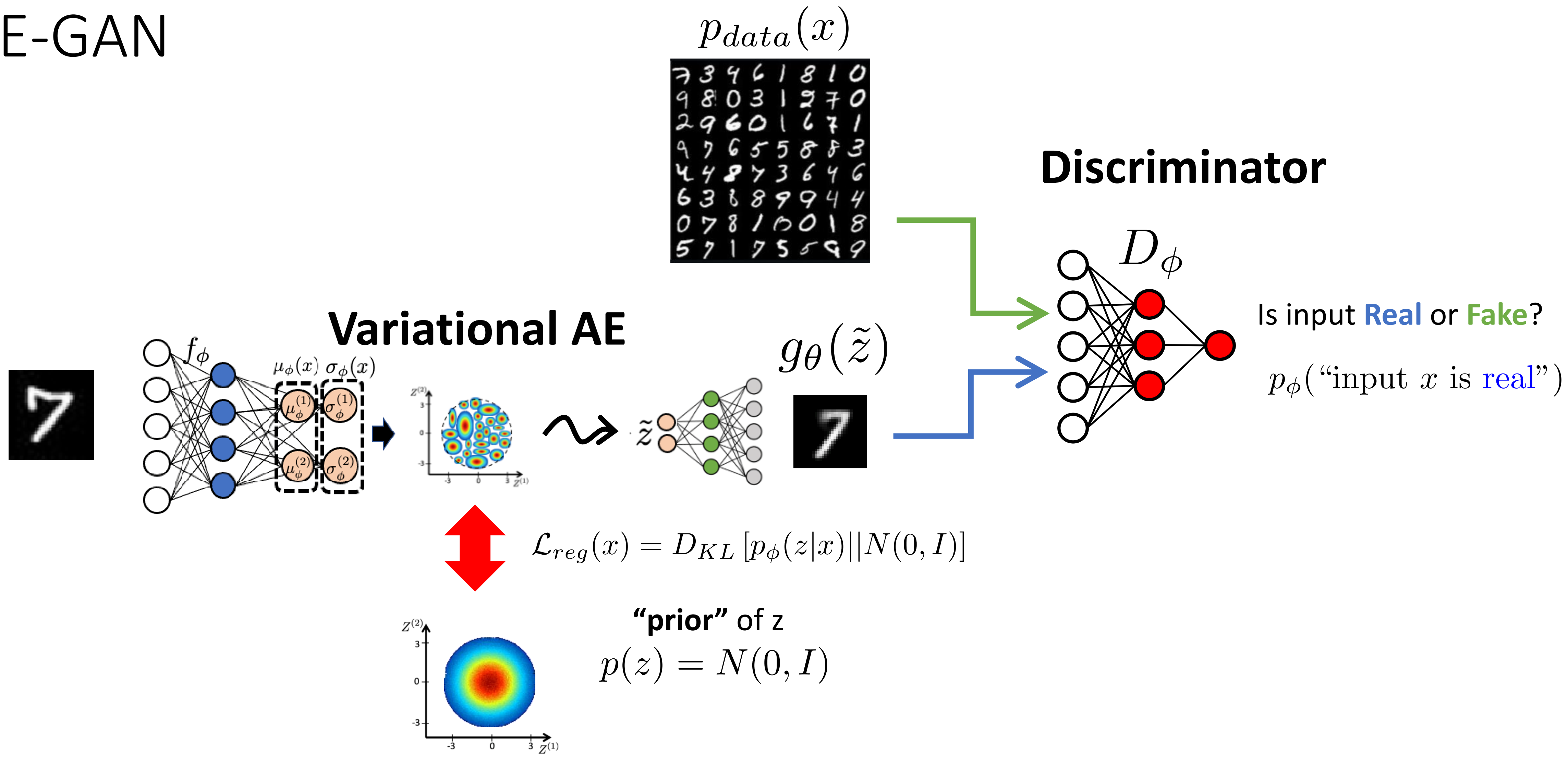


“prior” of z

$$p(z) = N(0, I)$$

From: Larsen et al, Autoencoding beyond pixels using a learned similarity metric, 2016

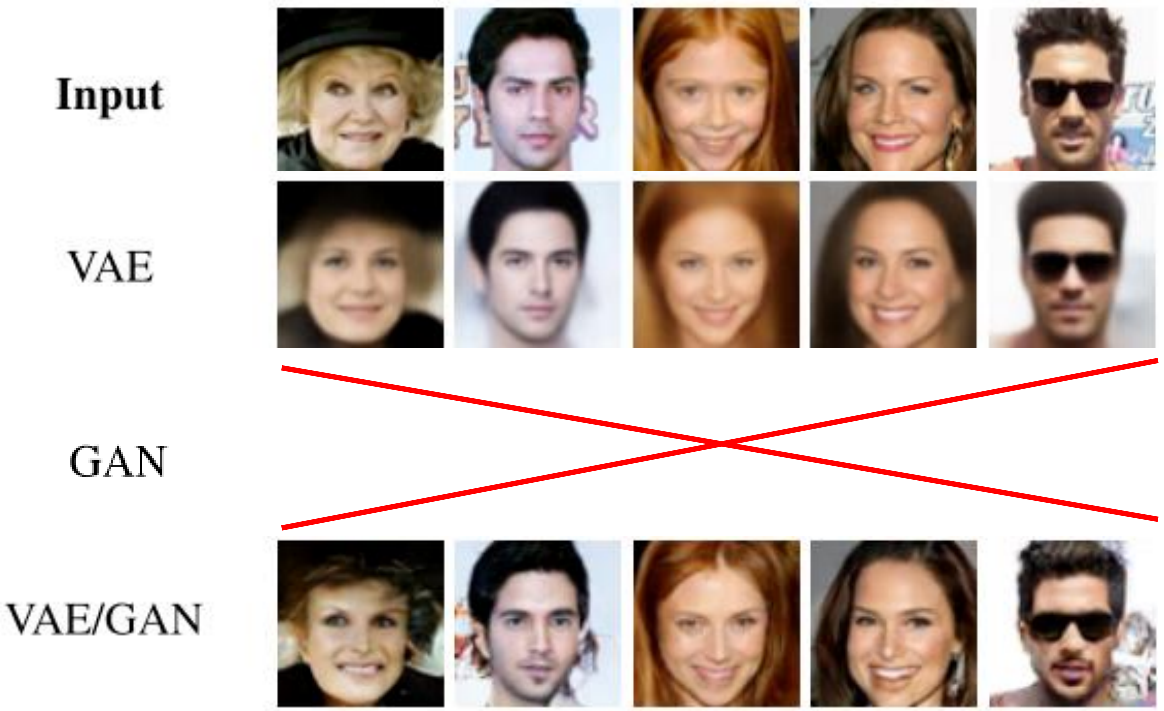
VAE-GAN



From: Larsen et al, Autoencoding beyond pixels using a learned similarity metric, 2016

VAE-GAN

Encoding => Reconstruction



Generation of new datapoints

From: Larsen et al, Autoencoding beyond pixels using a learned similarity metric, 2016

Discriminator's Accuracy as "**Learned Loss function**" for the Generator

$$\mathcal{L}_G(z) = \log(1 - D_{\phi}(G_{\theta}(z)))$$

Theoretical loss

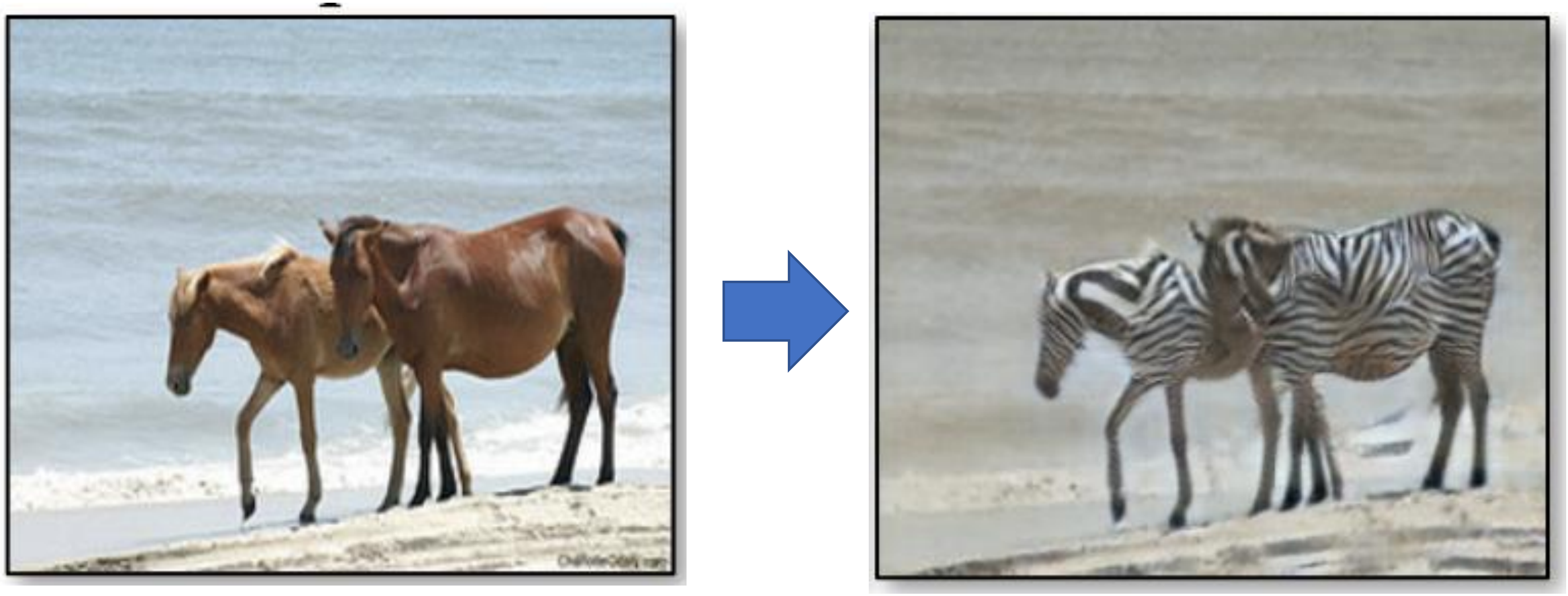
$$\mathcal{L}_G(z) = -\log D_{\phi}(G_{\theta}(z))$$

Practical loss

Other **"engineered" losses were designed** to measure a **specific type** of error. For example:
Reconstruction loss: intensity squared error
VAE regularizer: divergence from prior
Cross Entropy: entropy (difference) between predicted labels and real labels
Dice loss: Overlap of predicted segmentation with real segmentation

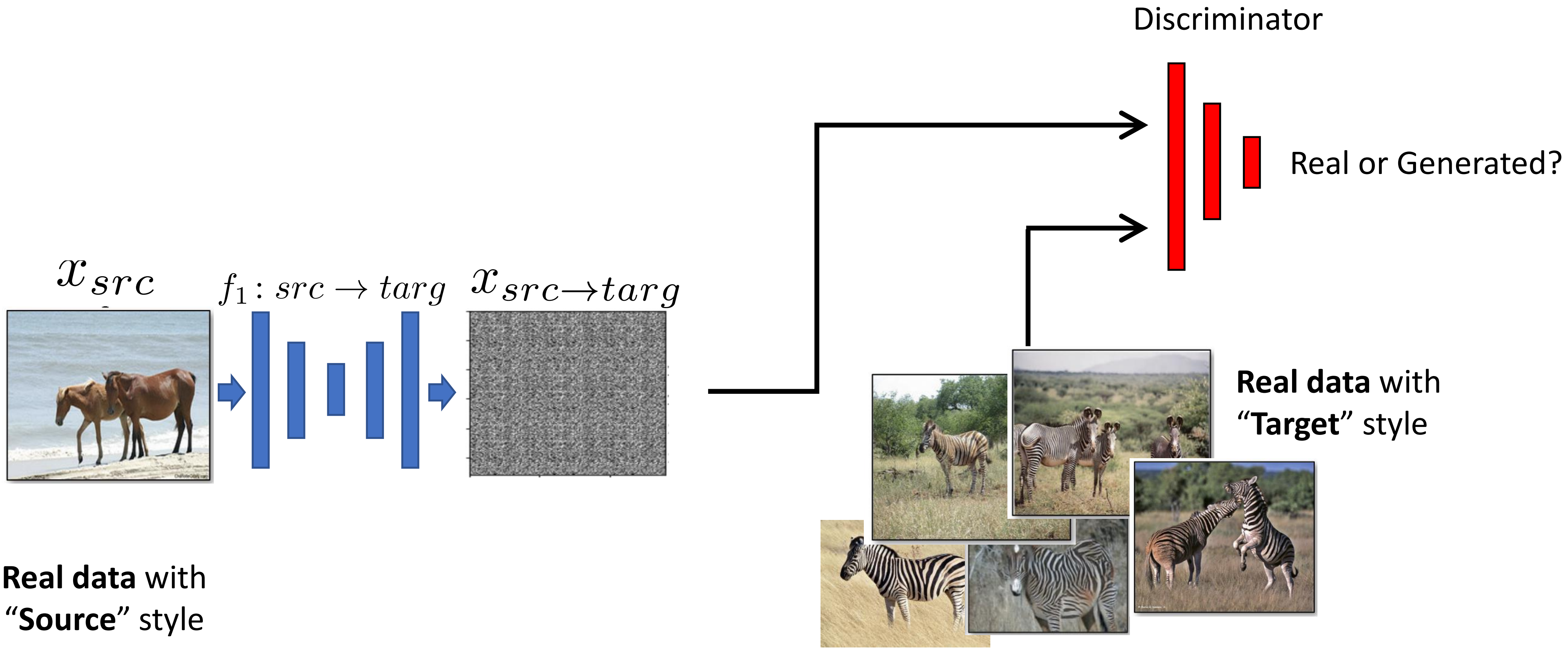
Discriminator's accuracy is a "Learned Loss" for the Generator:
D can learn to distinguish generated/reals based on any type of features!
Real or Fake?
Day or Night?
Summer or Winter?
Cartoon or Photo?

Style Transfer



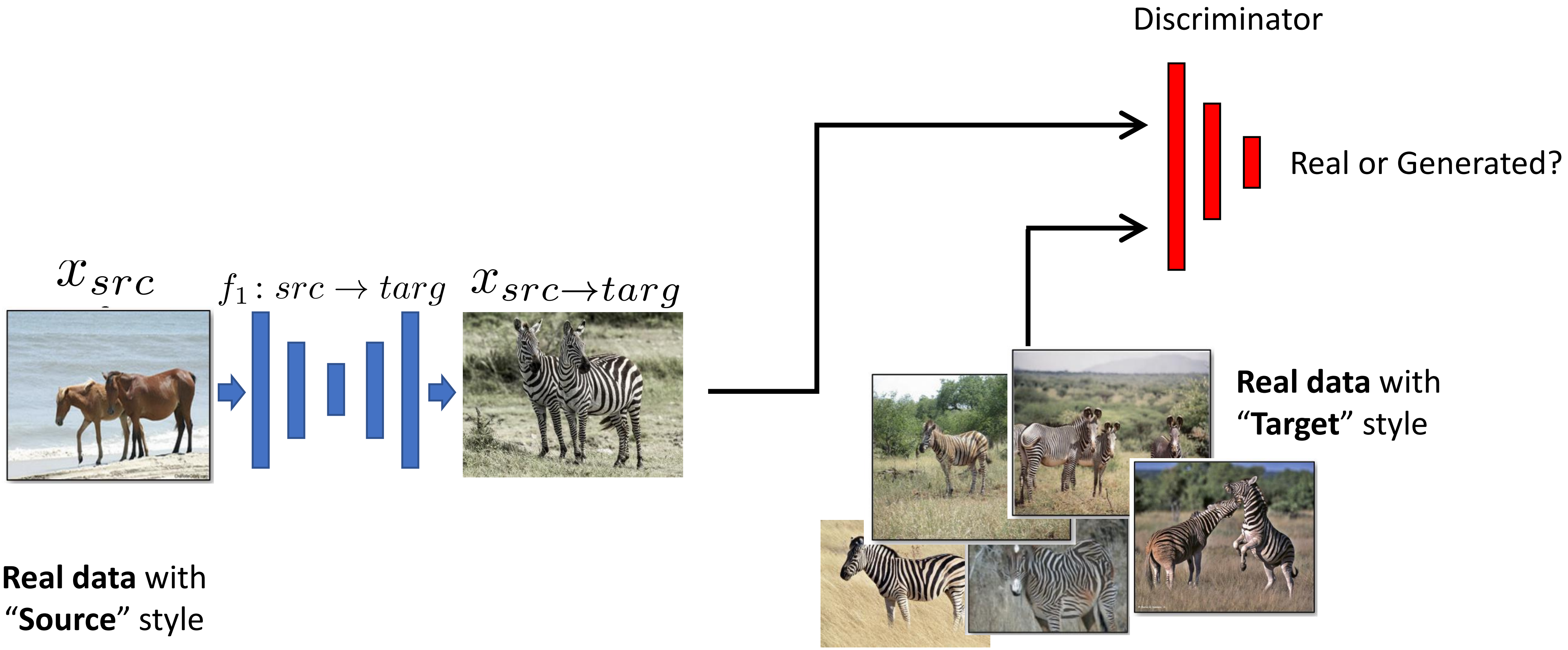
Horse => Zebra video:
https://www.youtube.com/watch?v=9reHvktowLY

Cycle GAN:
Converting “Style” of one data distribution to the “style” of another



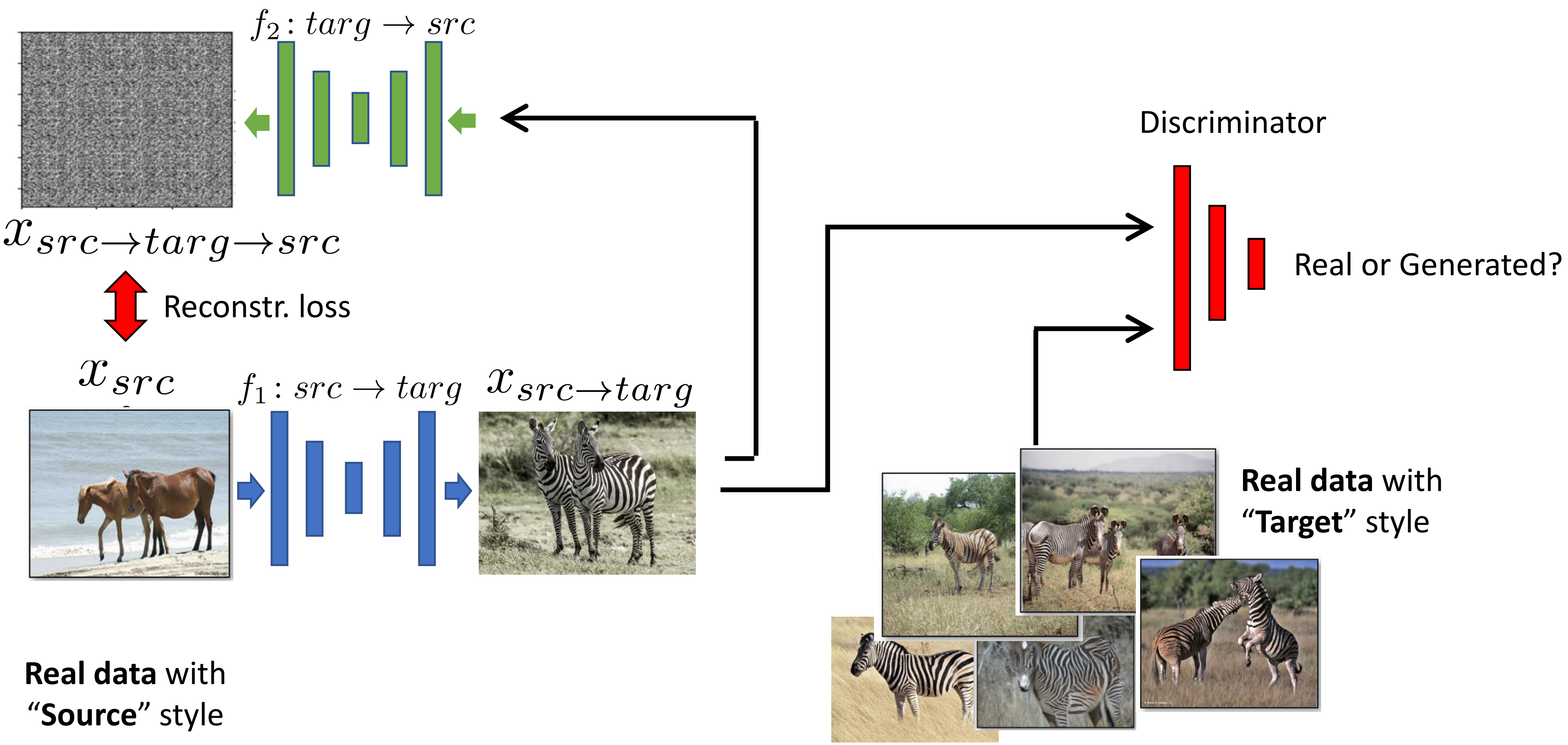
Zhu et al, Unpaired Image-to-Image Translation using Cycle-Conistent Adversarial Networks, 2017

Cycle GAN:
Converting “Style” of one data distribution to the “style” of another



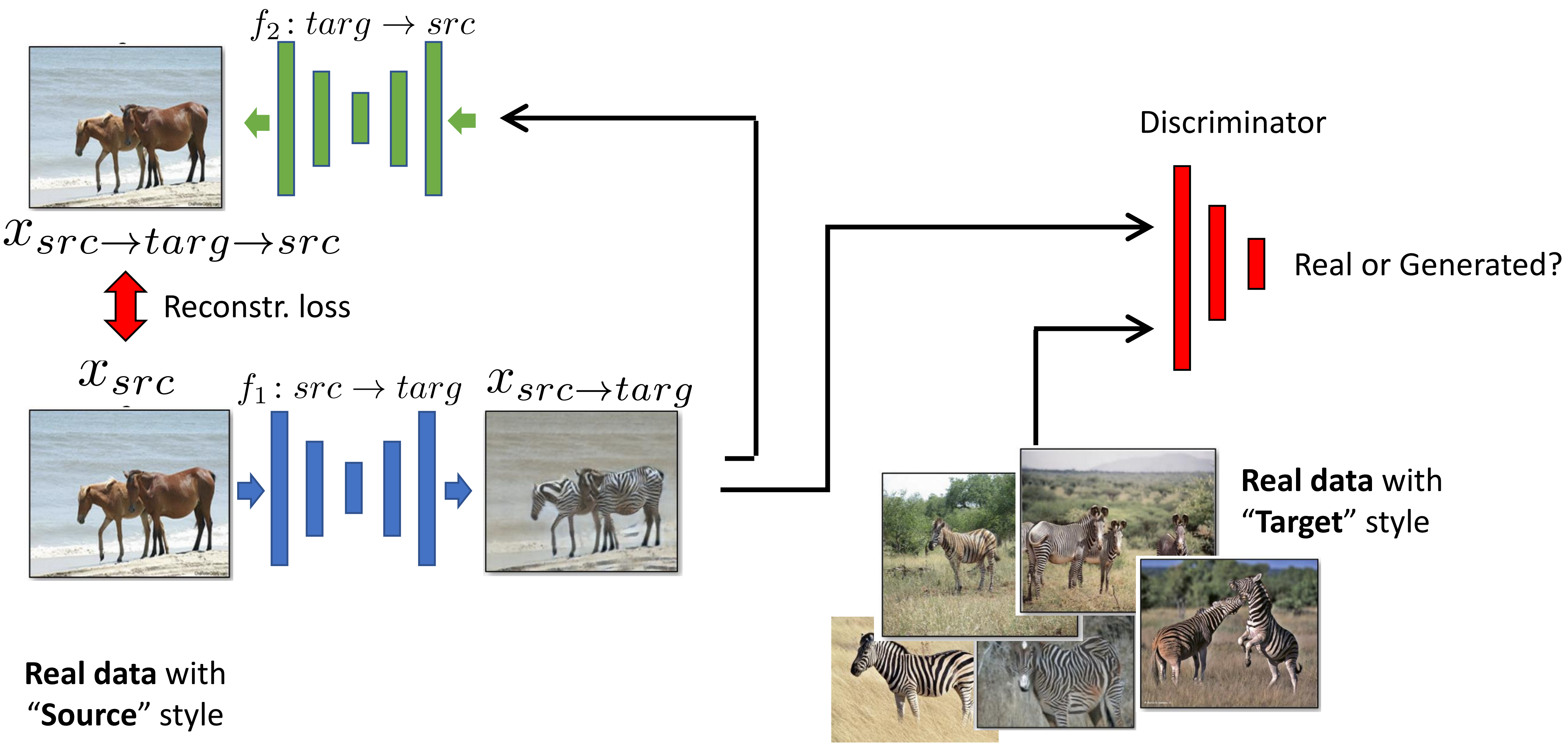
Zhu et al, Unpaired Image-to-Image Translation using Cycle-Conistent Adversarial Networks, 2017

Cycle GAN:
Converting “Style” of one data distribution to the “style” of another



Zhu et al, Unpaired Image-to-Image Translation using Cycle-Conistent Adversarial Networks, 2017

Cycle GAN:
Converting “Style” of one data distribution to the “style” of another

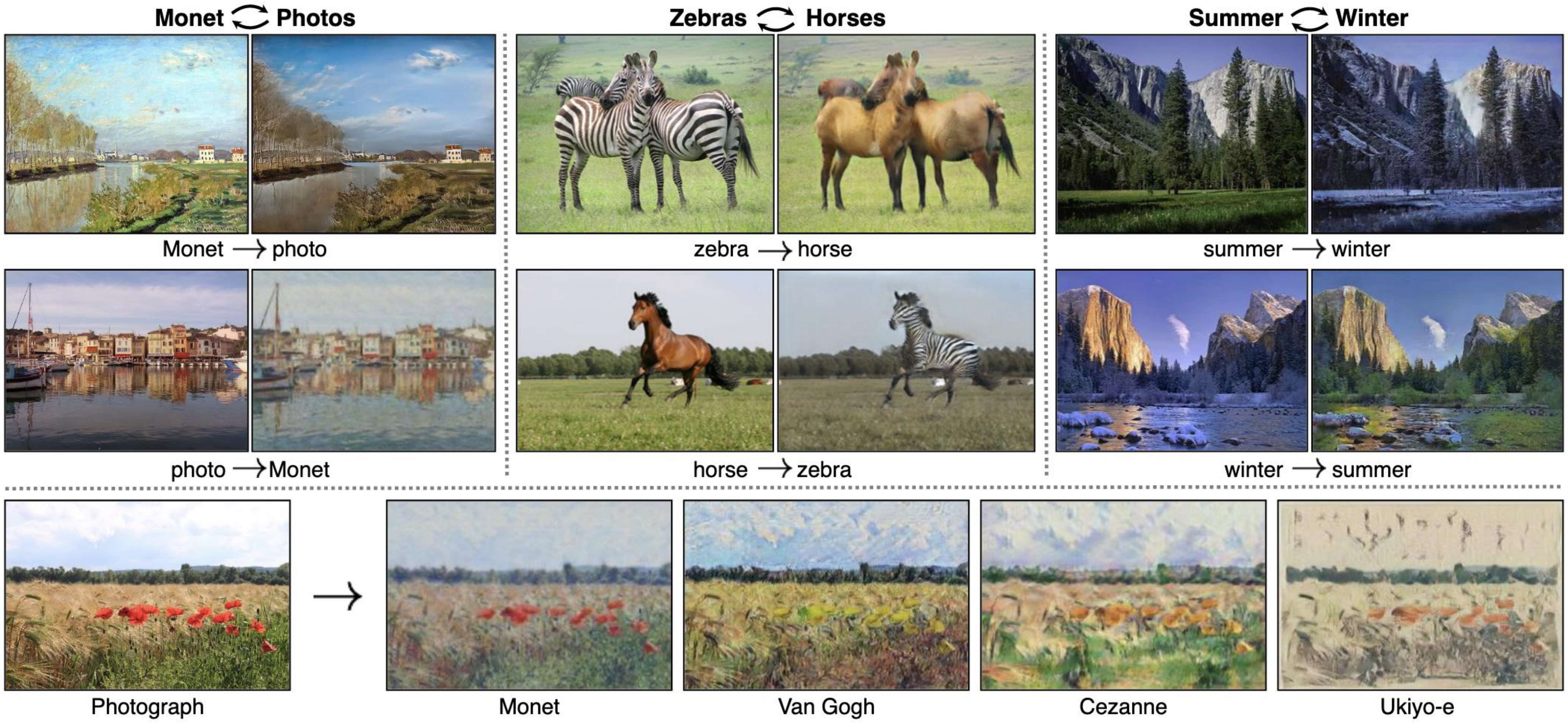


Zhu et al, Unpaired Image-to-Image Translation using Cycle-Conistent Adversarial Networks, 2017

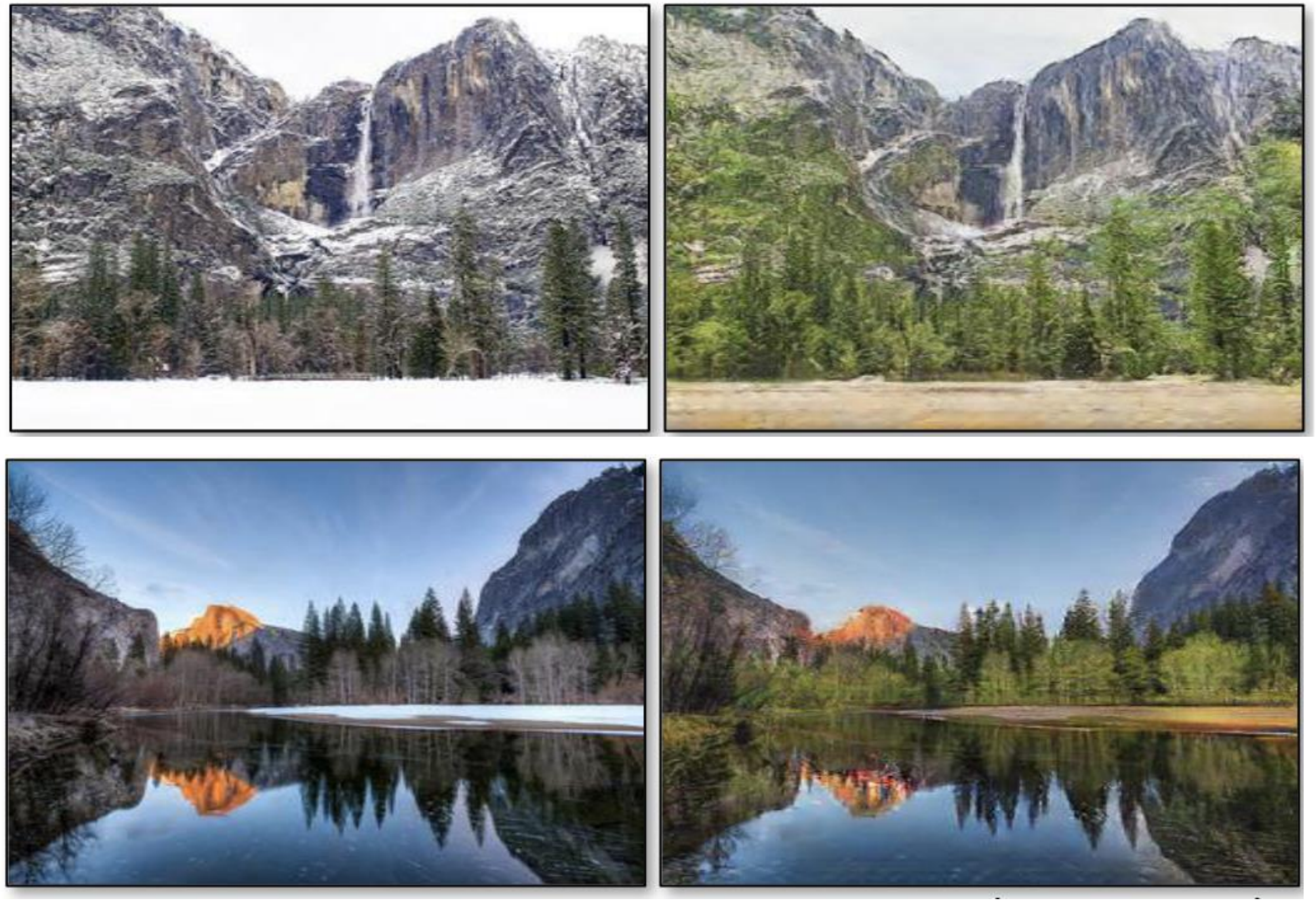
Cycle-GAN: Results

Horse => Zebra video:
https://www.youtube.com/watch?v=9reHvktowLY

Cycle GAN - Results:



Cycle GAN - Results:



Thank you very much