EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

Image Analysis II

Generative Adversarial Networks (GANs)

https://www.whichfaceisreal.com/

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

Doprava zdarma

Udržitelný materiál

A.I.Chair

PHILIPPE STARCK

Jídelní židle A.I. Chair od designéra Philippe Starck je prvním počinem Kartellu a umělé inteligence. Více informací

Barva : Vyberte

● 2 varianty skladem

Může být u vás doma do **3 dnů**.

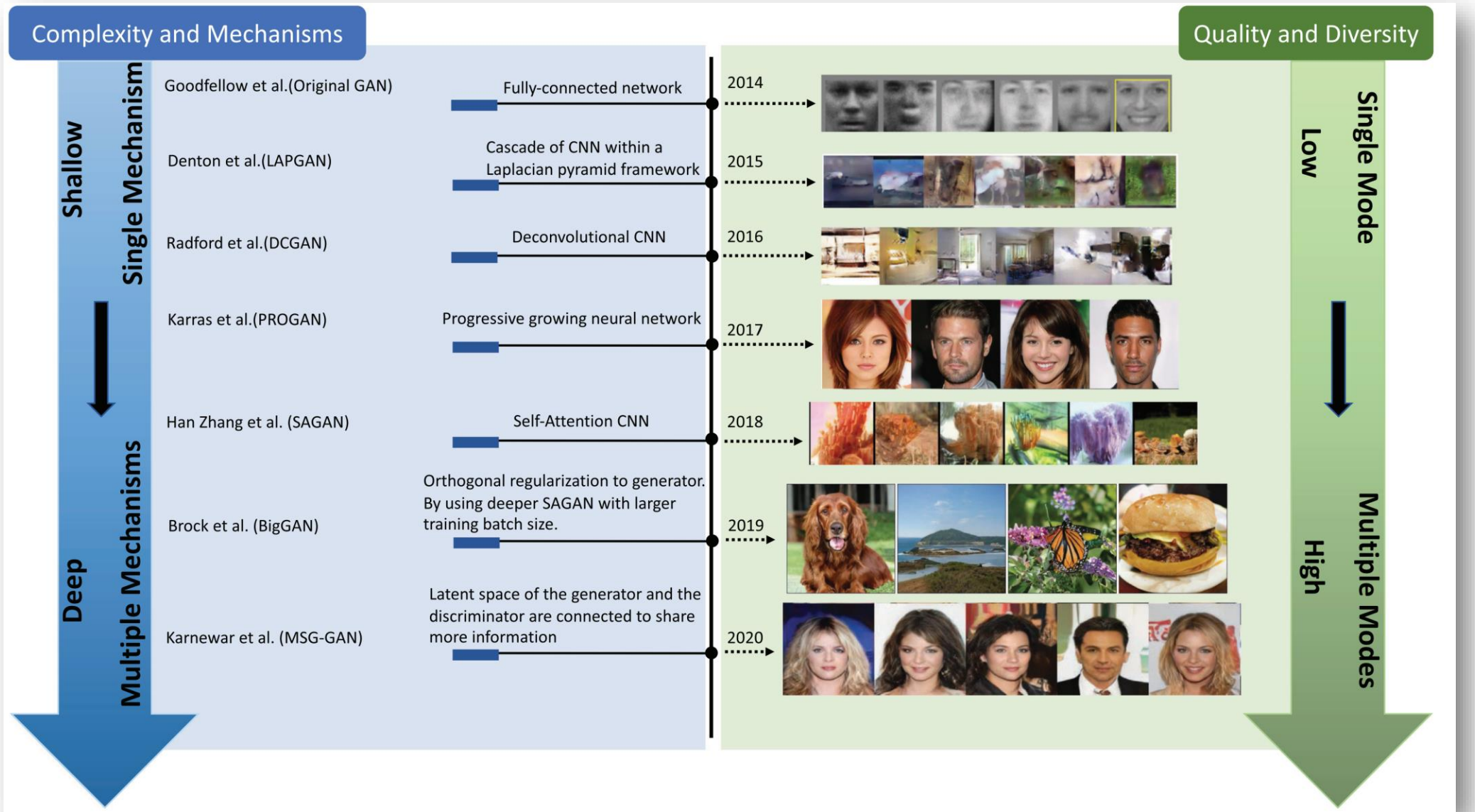
6 230 Kč

5 149 Kč bez DPH

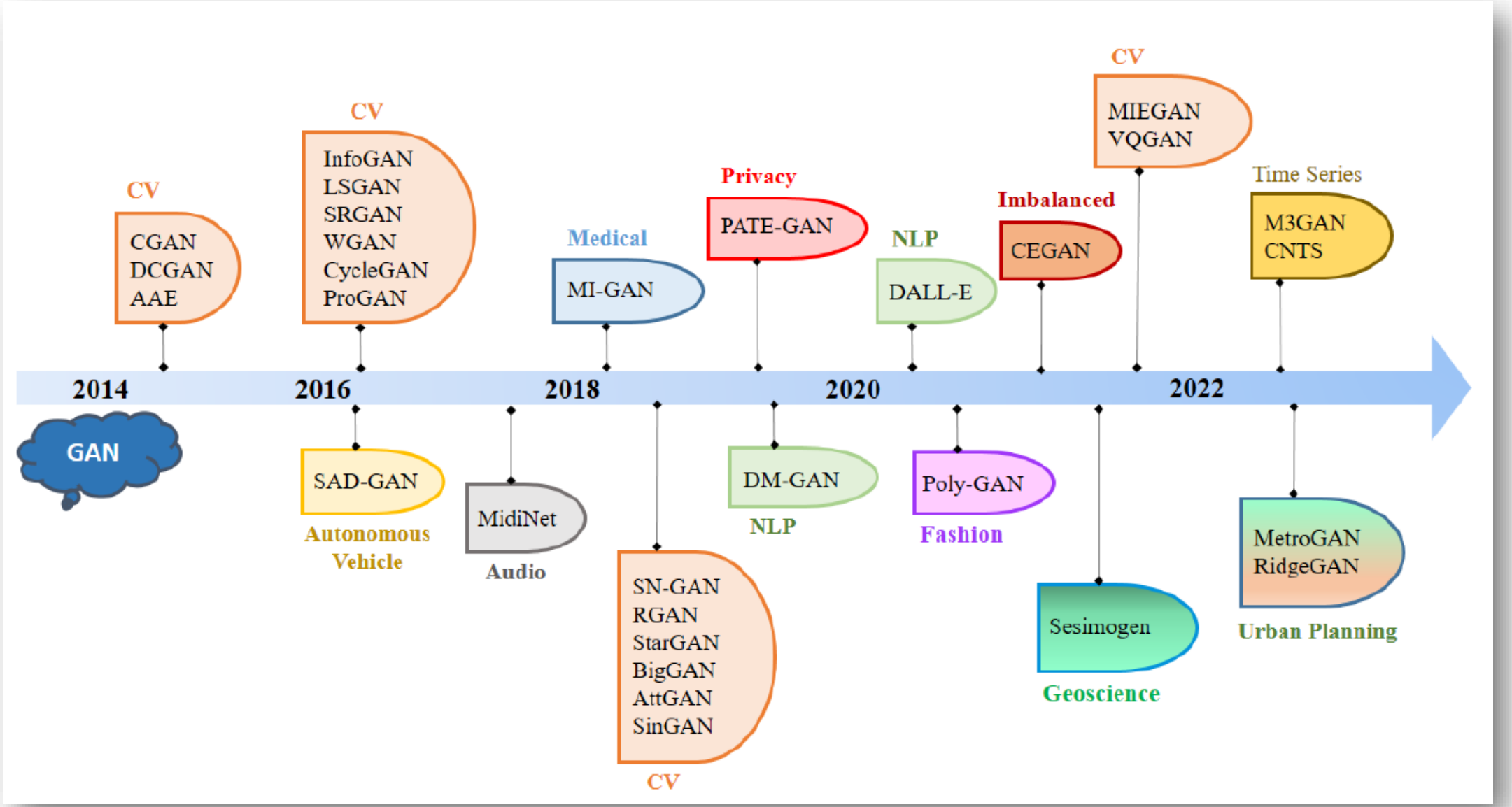
1 ks

DO KOŠÍKU

Naši designéři vám poradí s výběrem: **Kontaktujte nás!**



https://www.researchgate.net/publication/349189619_Generative_Adversarial_Networks_in_Computer_Vision_A_Survey_and_Taxonomy



https://www.researchgate.net/publication/373551906_Ten_Years_of_Generative_Adversarial_Nets_GANs_A_survey_of_the_state-of-the-art

https://www.researchgate.net/publication/349189619_Generative_Adversarial_Networks_in_Computer_Vision_A_Survey_and_Taxonomy

https://github.com/dongb5/GAN-Timeline

https://github.com/hindupuravinash/the-gan-zoo

LoGAN: Generating Logos with a Generative Adversarial Neural Network Conditioned on color

Ajkel Mino
Department of Data Science and Knowledge Engineering
Maastricht University
Maastricht, Netherlands
ajkimino@gmail.com

Gerasimos Spanakis
Department of Data Science and Knowledge Engineering
Maastricht University
Maastricht, Netherlands
jerry.spanakis@maastrichtuniversity.nl

Fig. 6. Results from the generation of 64 logos per class after 400 epochs of training. Classes from left to right top to bottom: green, purple, white, brown, blue, cyan, yellow, gray, red, pink, orange, black.

https://www.researchgate.net/publication/373551906_Ten_Years_of_Generative_Adversarial_Nets_GANs_A_survey_of_the_state-of-the-art
https://arxiv.org/pdf/1810.10395.pdf
https://www.researchgate.net/publication/349189619_Generative_Adversarial_Networks_in_Computer_Vision_A_Survey_and_Taxonomy

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

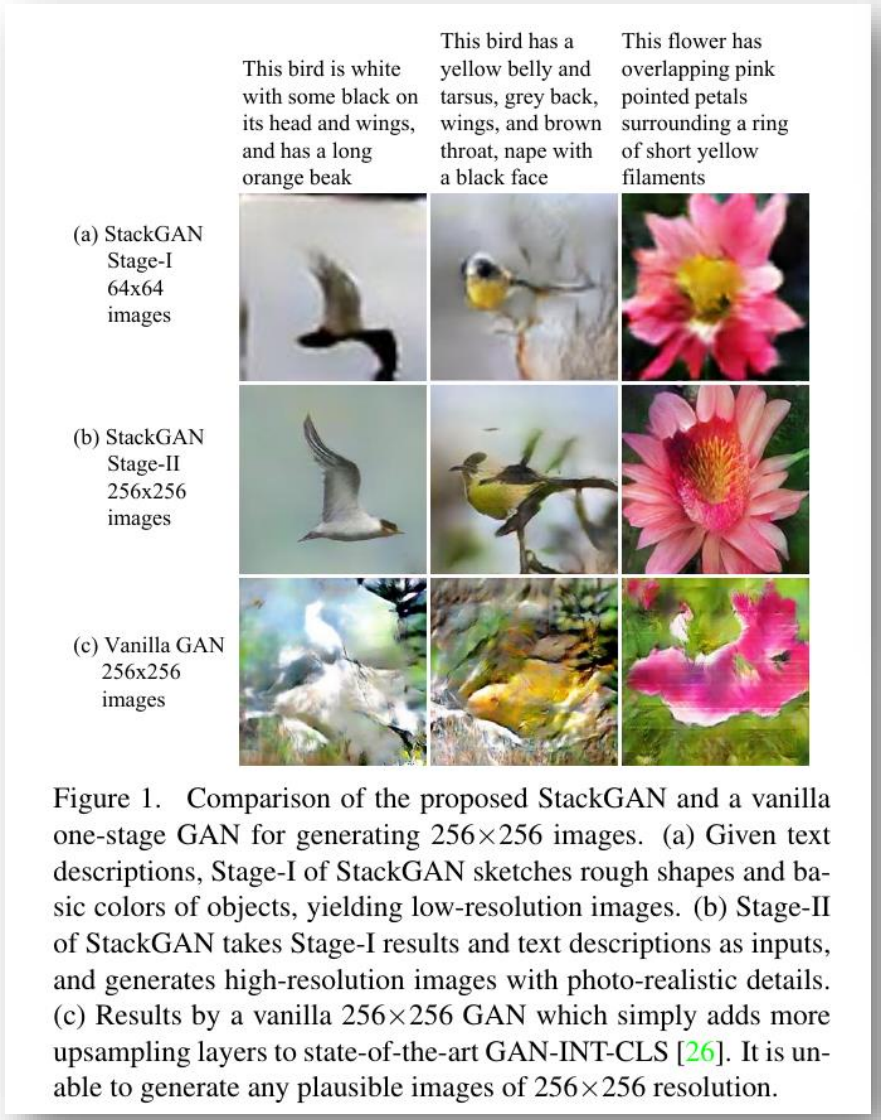


Figure 1. Comparison of the proposed StackGAN and a vanilla one-stage GAN for generating 256×256 images. (a) Given text descriptions, Stage-I of StackGAN sketches rough shapes and basic colors of objects, yielding low-resolution images. (b) Stage-II of StackGAN takes Stage-I results and text descriptions as inputs, and generates high-resolution images with photo-realistic details. (c) Results by a vanilla 256×256 GAN which simply adds more upsampling layers to state-of-the-art GAN-INT-CLS [26]. It is unable to generate any plausible images of 256×256 resolution.

StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks

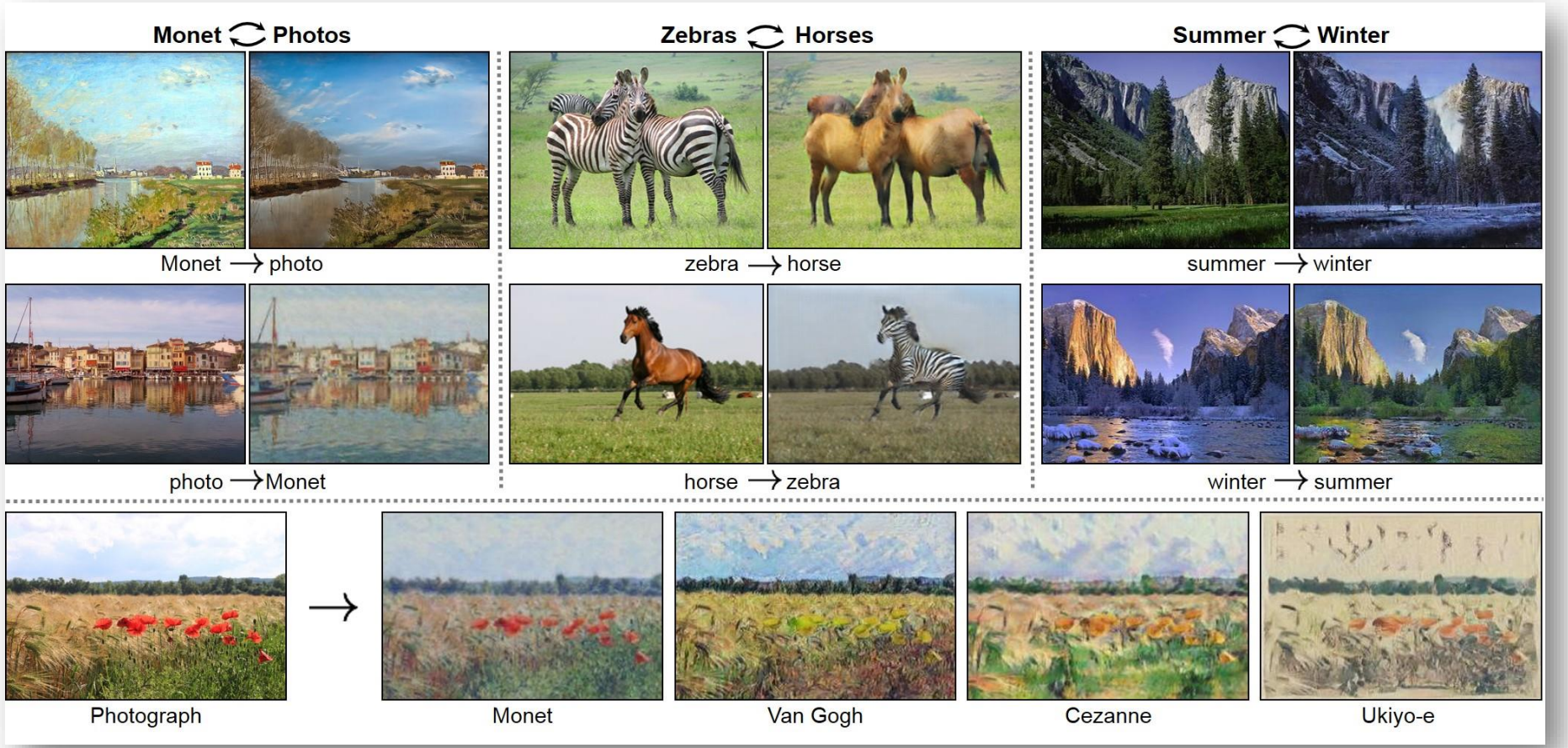
Han Zhang[1], Tao Xu[2], Hongsheng Li[3],
Shaoting Zhang[4], Xiaogang Wang[3], Xiaolei Huang[2], Dimitris Metaxas[1]

[1]Rutgers University [2]Lehigh University [3]The Chinese University of Hong Kong [4]Baidu Research
{han.zhang, dnm}@cs.rutgers.edu, {tax313, xih206}@lehigh.edu
{hsli, xgwang}@ee.cuhk.edu.hk, zhangshaoting@baidu.com

https://github.com/hanzhanggit/StackGAN
https://www.researchgate.net/publication/373551906_Ten_Years_of_Generative_Adversarial_Nets_GANs_A_survey_of_the_state-of-the-art
https://www.researchgate.net/publication/349189619_Generative_Adversarial_Networks_in_Computer_Vision_A_Survey_and_Taxonomy

https://github.com/Sxela/ArcaneGAN/releases/tag/v0.2
https://www.researchgate.net/publication/373551906_Ten_Years_of_Generative_Adversarial_Nets_GANs_A_survey_of_the_state-of-the-art
https://www.researchgate.net/publication/349189619_Generative_Adversarial_Networks_in_Computer_Vision_A_Survey_and_Taxonomy

https://www.youtube.com/watch?v=N7KbfWodXJE
https://github.com/junyanz/CycleGAN



https://github.com/Sxela/ArcaneGAN/releases/tag/v0.2
https://www.researchgate.net/publication/373551906_Ten_Years_of_Generative_Adversarial_Nets_GANs_A_survey_of_the_state-of-the-art
https://www.researchgate.net/publication/349189619_Generative_Adversarial_Networks_in_Computer_Vision_A_Survey_and_Taxonomy

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network

Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi
Twitter
{cledig,ltheis,fhuszar,jcaballero,aacostadiaz,aaitken,atejani,jtotz,zehanw,wshi}@twitter.com

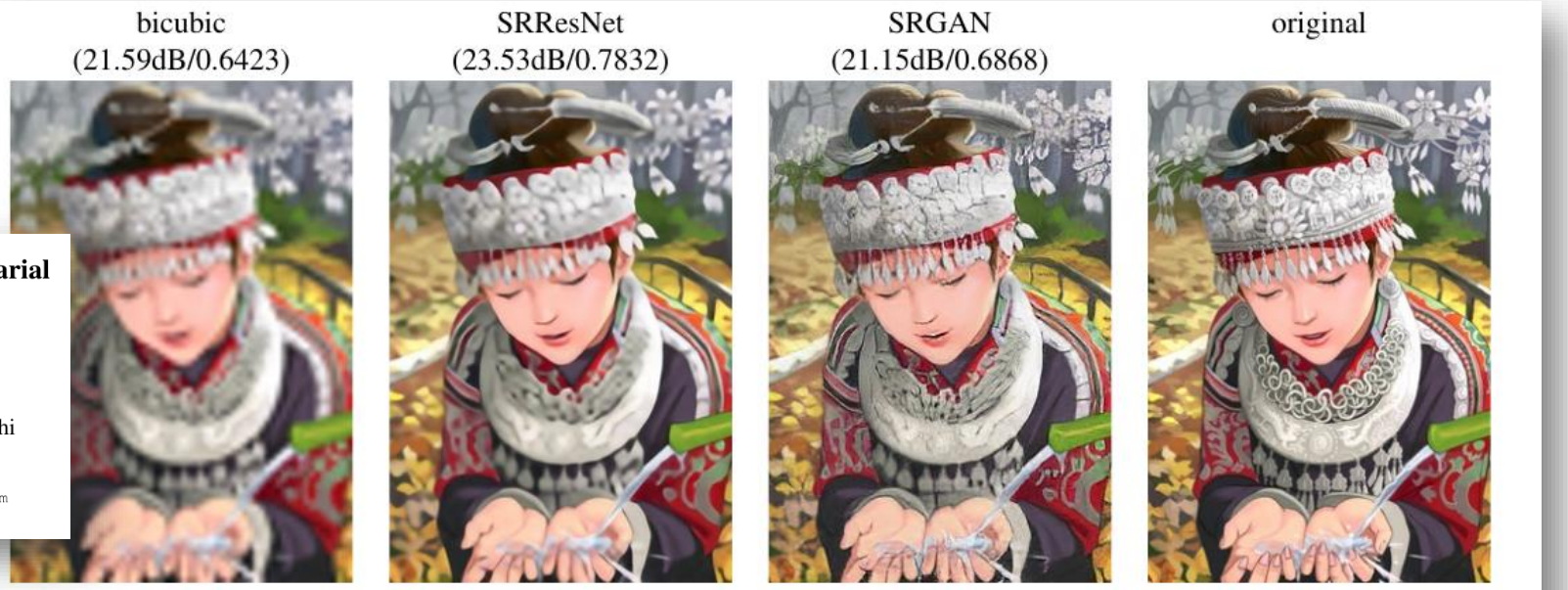


Figure 2: From left to right: bicubic interpolation, deep residual network optimized for MSE, deep residual generative mized for a loss more sensitive to human perception, original HR image. Corresponding PSNR and kets. [4× upscaling]

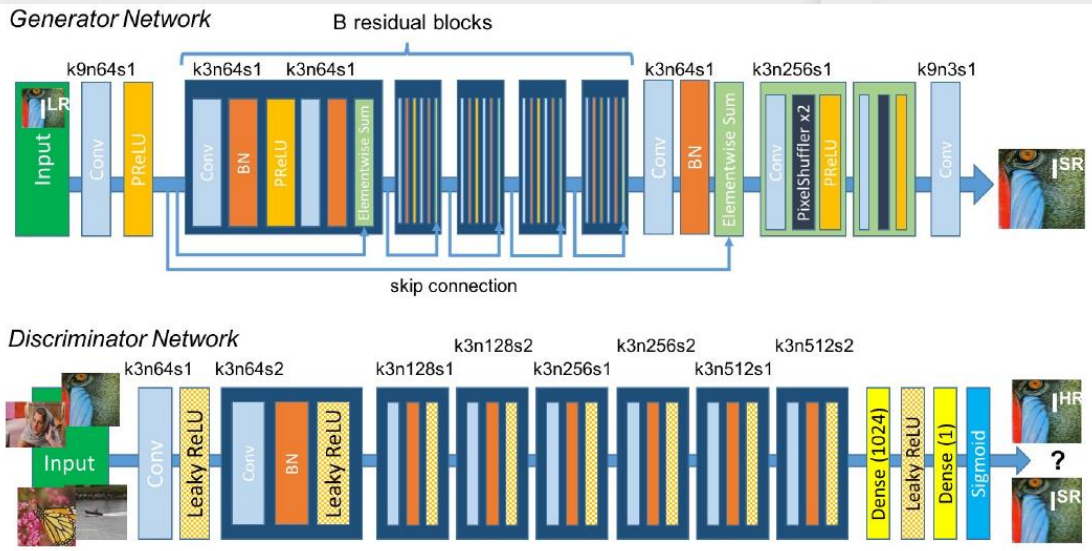


Figure 4: Architecture of Generator and Discriminator Network with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer.

https://arxiv.org/pdf/1609.04802.pdf

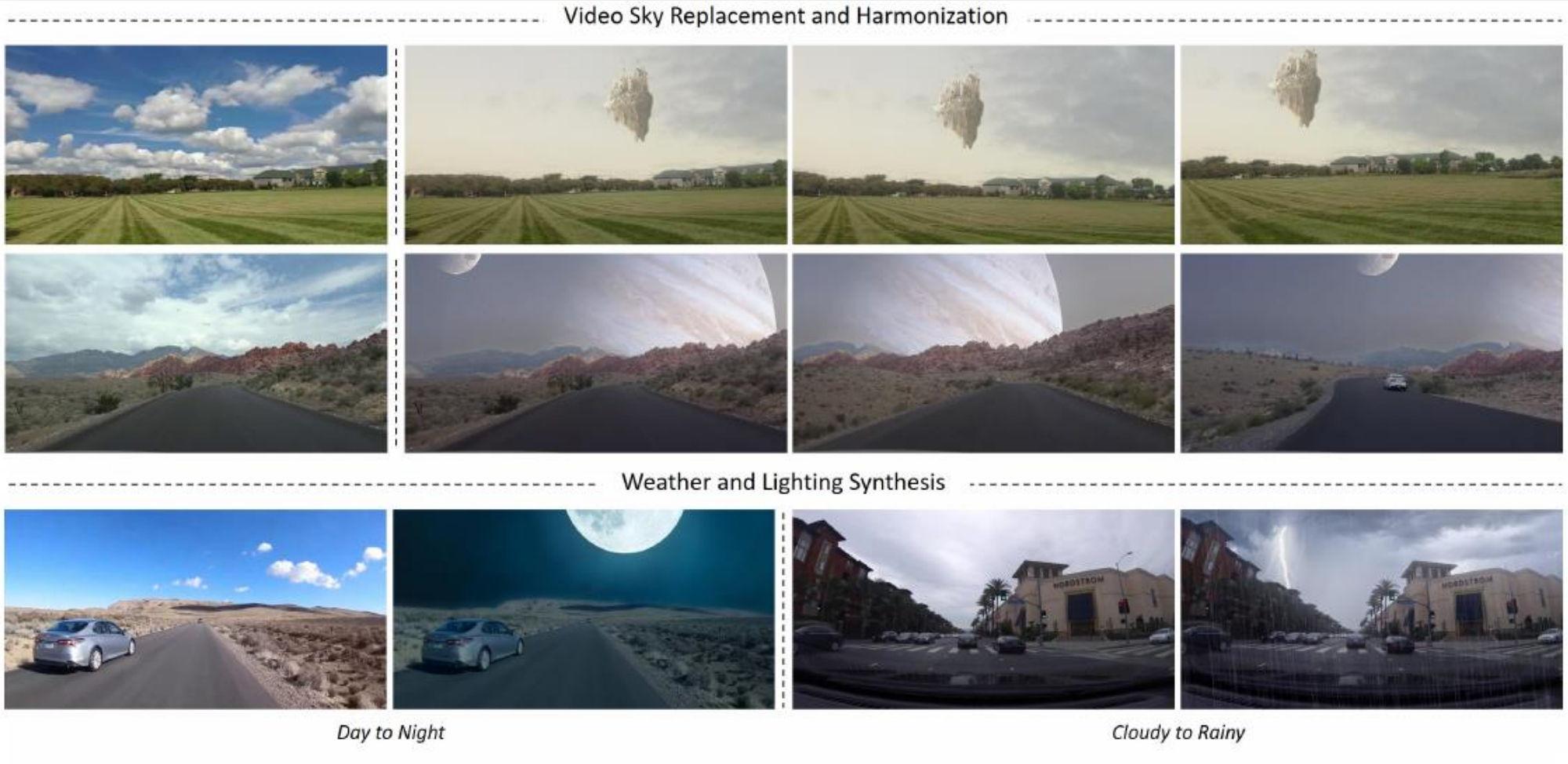


Figure 1: We propose a vision-based method for generating videos with controllable sky backgrounds and realistic weather/lighting conditions. First and second row: rendered sky videos - flying castle and Jupiter sky (leftmost shows a frame from input video and the rest are the output frames). Third row: weather and lighting synthesis (day to night, cloudy to rainy).

https://arxiv.org/pdf/2010.11800v1.pdf
https://www.youtube.com/watch?v=zal9Ues0aOQ

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

Generative Adversarial Nets

Ian J. Goodfellow[*], **Jean Pouget-Abadie**[†], **Mehdi Mirza**, **Bing Xu**, **David Warde-Farley**, **Sherjil Ozair**[‡], **Aaron Courville**, **Yoshua Bengio**[§]
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

Abstract

We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.

Figure 2: Visualization of samples from the model. Rightmost column shows the nearest training example of the neighboring sample, in order to demonstrate that the model has not memorized the training set. Samples are fair random draws, not cherry-picked. Unlike most other visualizations of deep generative models, these images show actual samples from the model distributions, not conditional means given samples of hidden units. Moreover, these samples are uncorrelated because the sampling process does not depend on Markov chain mixing. a) MNIST b) TFD c) CIFAR-10 (fully connected model) d) CIFAR-10 (convolutional discriminator and "deconvolutional" generator)

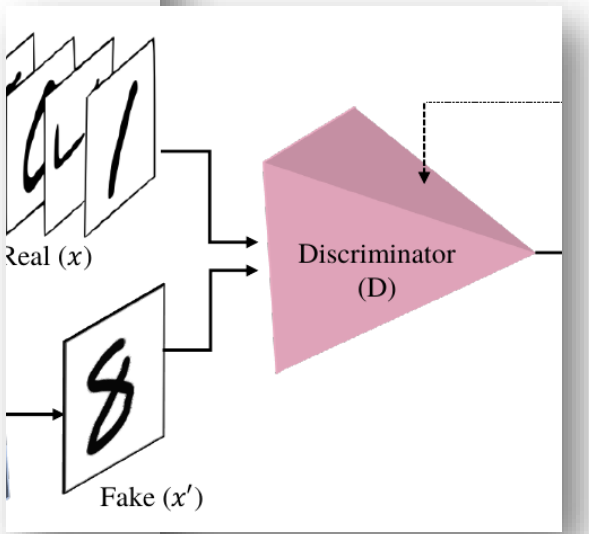
5 Experiments

We trained adversarial nets an a range of datasets including MNIST[21], the Toronto Face Database (TFD) [27], and CIFAR-10 [19]. The generator nets used a mixture of rectifier linear activations [17, 8] and sigmoid activations, while the discriminator net used maxout [9] activations. Dropout [16] was applied in training the discriminator net. While our theoretical framework permits the use of dropout and other noise at intermediate layers of the generator, we used noise as the input to only the bottommost layer of the generator network.

https://www.researchgate.net/publication/341425292_Throwing_light_on_black_boxes_emergence_of_visual_categories_from_deep_learning
https://arxiv.org/abs/1406.2661

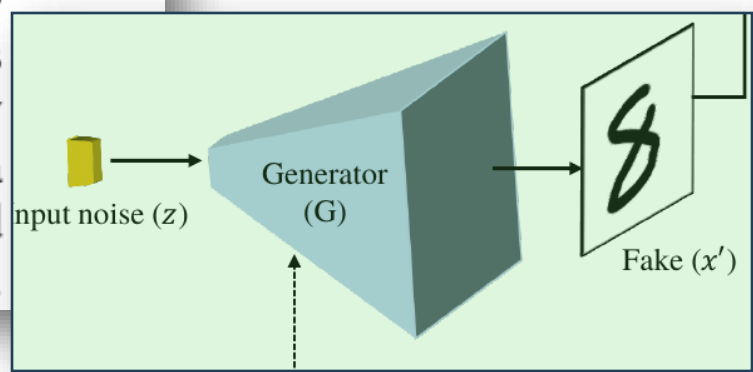
Ian J. Goodfellow, Jean Pouget-Abadie,* Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair,† Aaron Courville, Yoshua Bengio‡

Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7



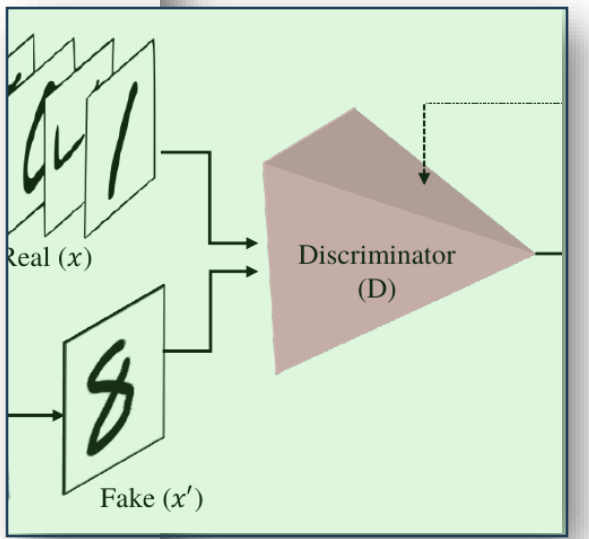
Abstract

We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation.



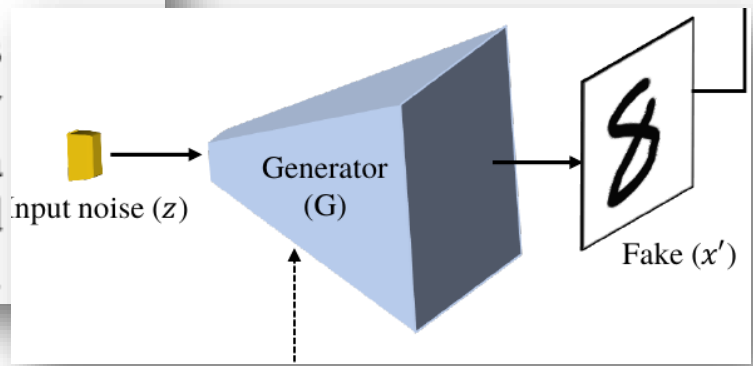
https://www.researchgate.net/publication/341425292_Throwing_light_on_black_boxes_emergence_of_visual_categories_from_deep_learning
https://arxiv.org/abs/1406.2661

Ian J. Goodfellow, Jean Pouget-Abadie,* Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair,† Aaron Courville, Yoshua Bengio‡
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7



Abstract

We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation.



https://www.researchgate.net/publication/341425292_Throwing_light_on_black_boxes_emergence_of_visual_categories_from_deep_learning
https://arxiv.org/abs/1406.2661

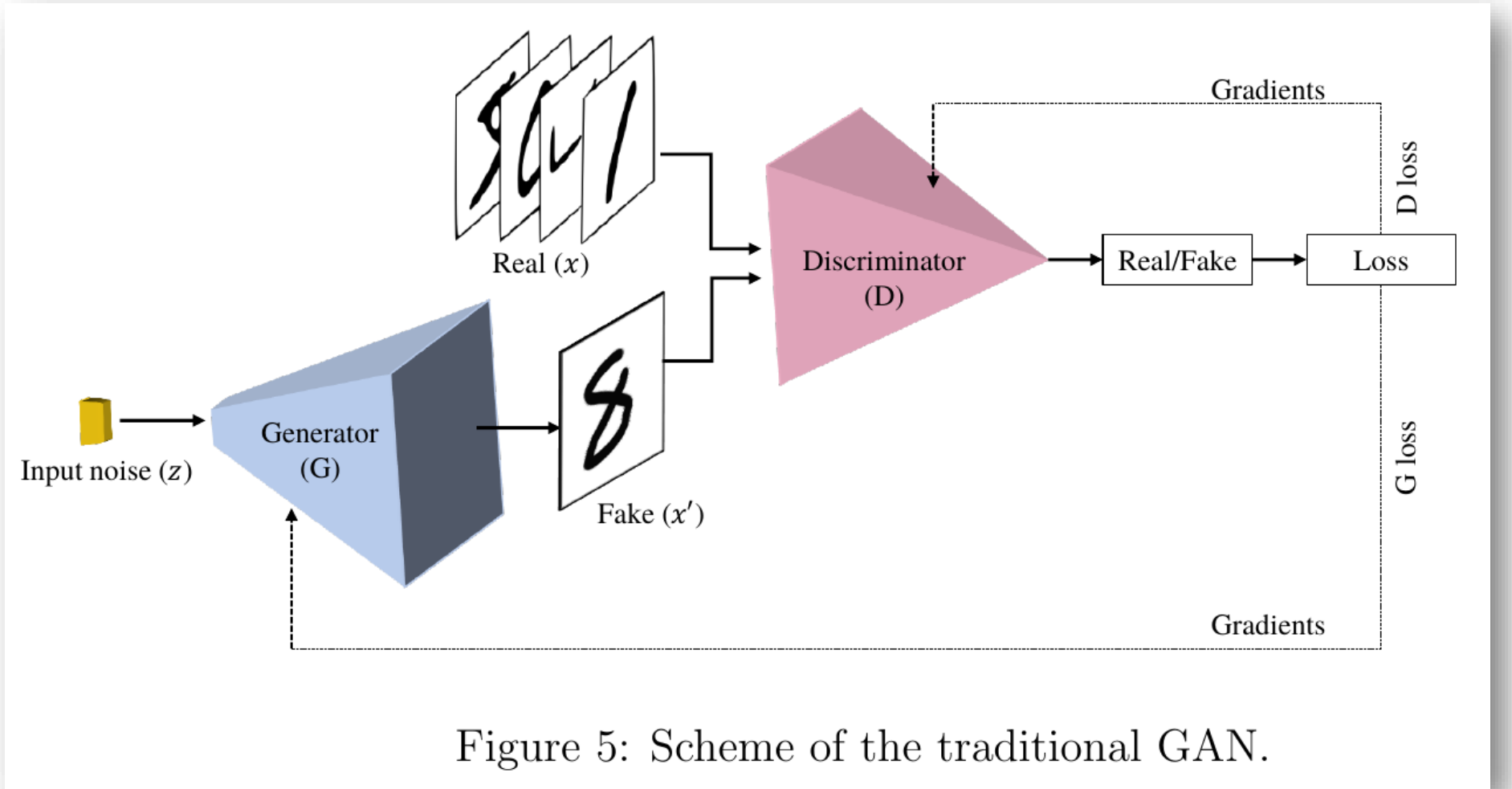


Figure 5: Scheme of the traditional GAN.

The discriminator is trying to be better and better and at the same time the generator is trying to be better and better to outsmart the discriminator.

https://www.researchgate.net/publication/341425292_Throwing_light_on_black_boxes_emergence_of_visual_categories_from_deep_learning
https://arxiv.org/abs/1406.2661

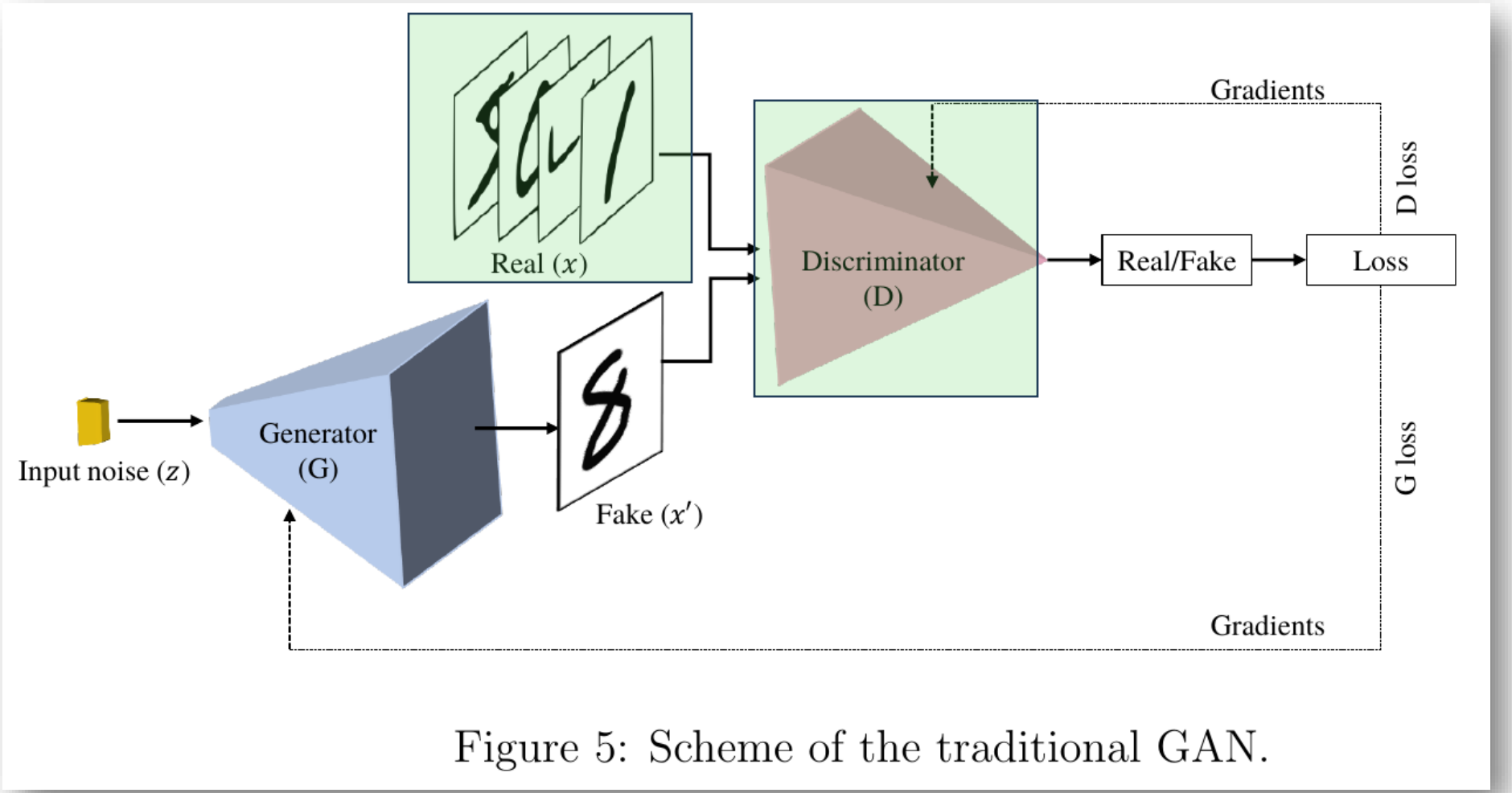


Figure 5: Scheme of the traditional GAN.

Discriminator Training Phase:

- supervised training
- real data with labels 1

https://www.researchgate.net/publication/341425292_Throwing_light_on_black_boxes_emergence_of_visual_categories_from_deep_learning
https://arxiv.org/abs/1406.2661

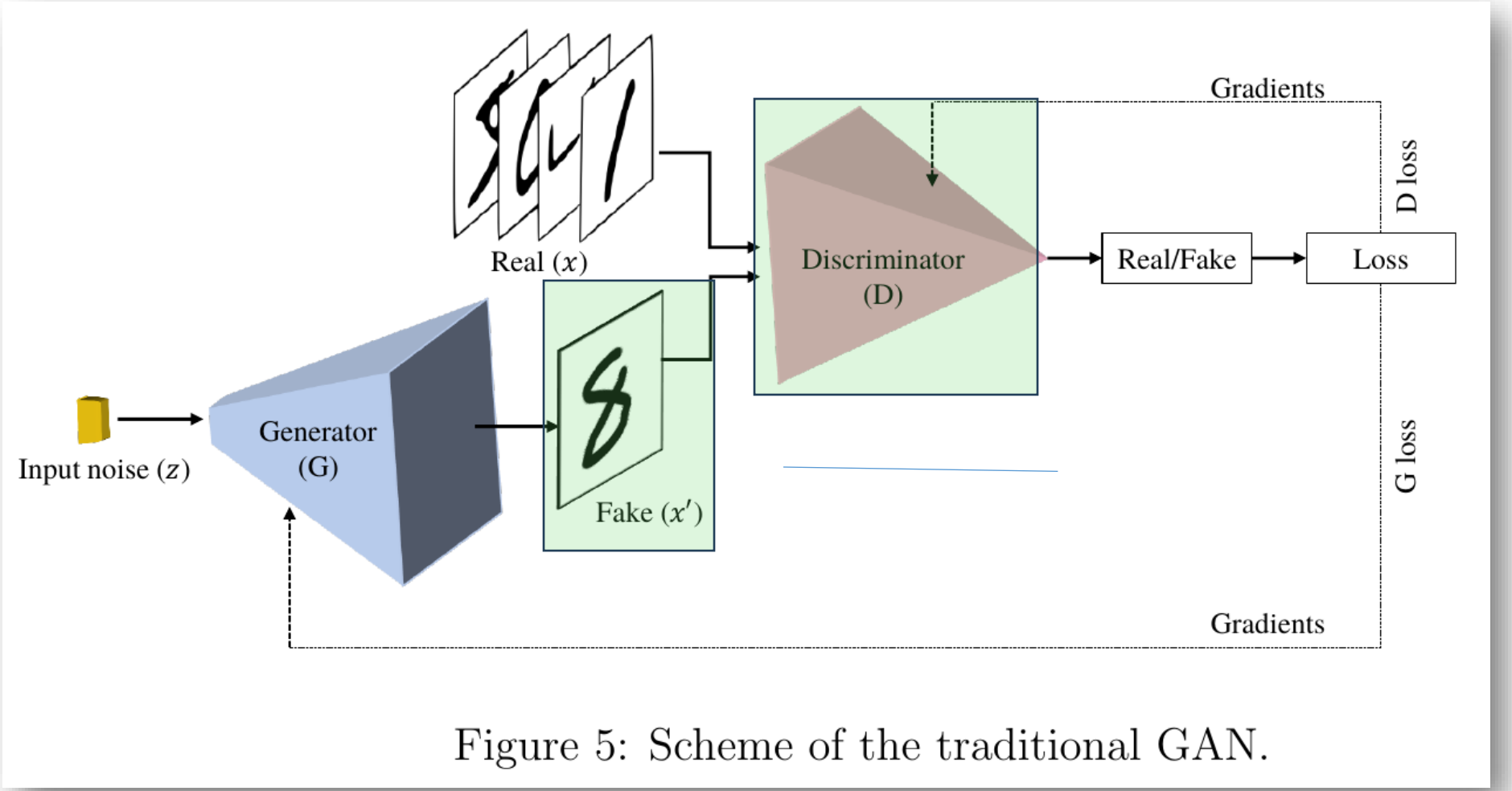


Figure 5: Scheme of the traditional GAN.

Discriminator Training Phase:

- supervised training
- real data with labels 1

- supervised training
- fake data with labels 0

https://www.researchgate.net/publication/341425292_Throwing_light_on_black_boxes_emergence_of_visual_categories_from_deep_learning
https://arxiv.org/abs/1406.2661

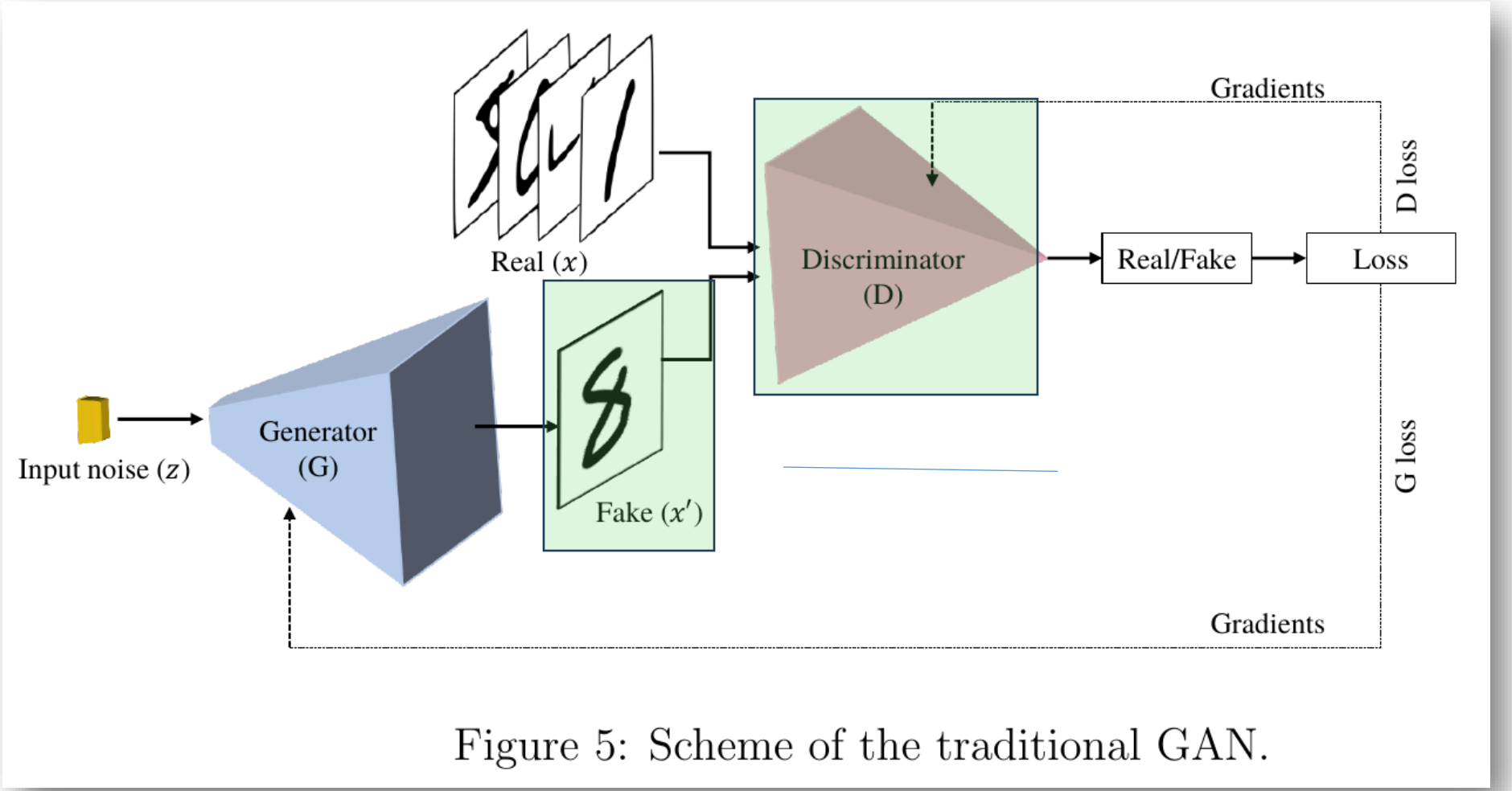


Figure 5: Scheme of the traditional GAN.

Discriminator Training Phase:

- supervised training
- real data with labels 1

- supervised training
- fake data with labels 0

The goal is (for the discriminator) to learn to distinguish between real and fake images.

https://www.researchgate.net/publication/341425292_Throwing_light_on_black_boxes_emergence_of_visual_categories_from_deep_learning
https://arxiv.org/abs/1406.2661

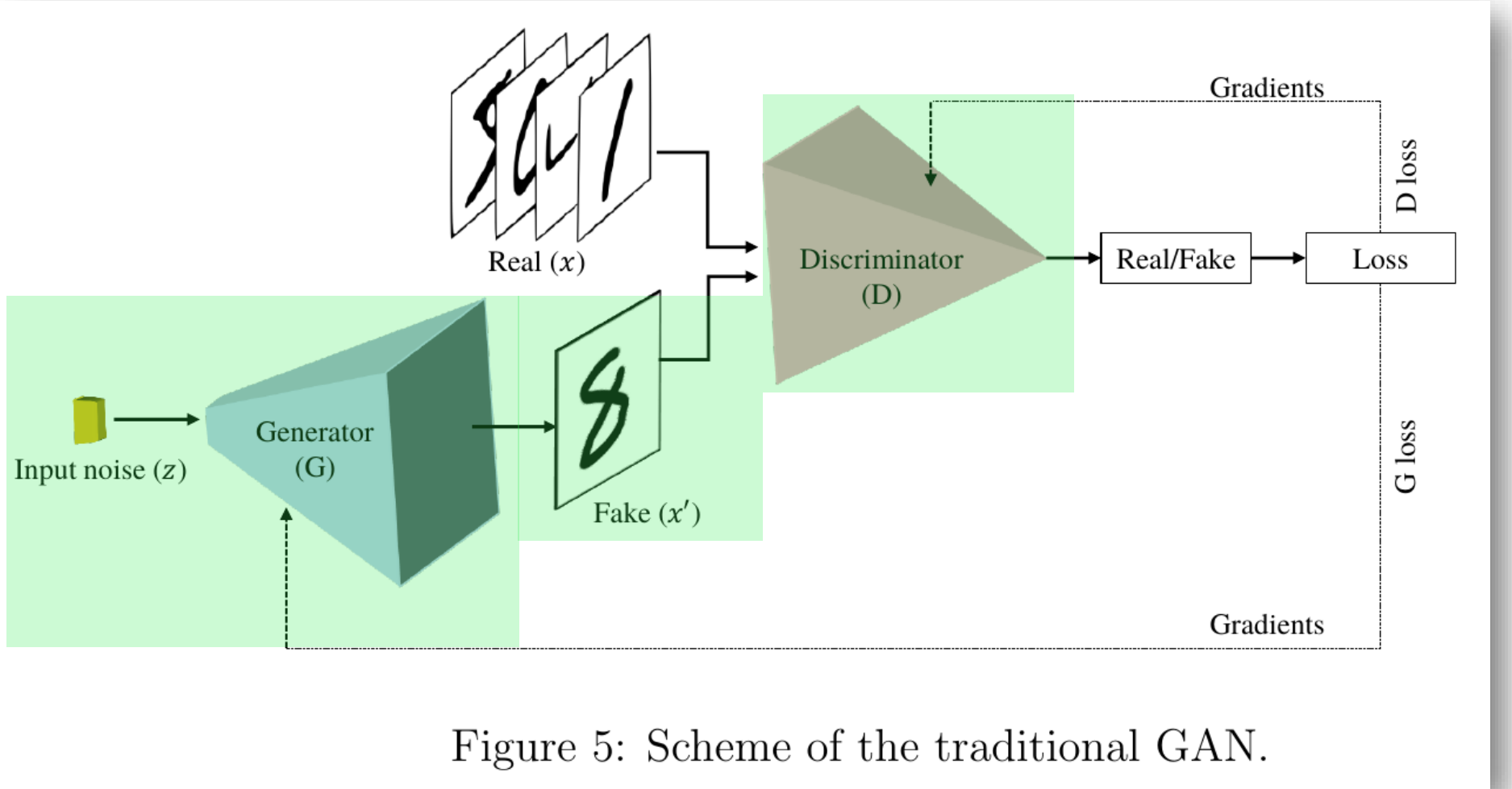


Figure 5: Scheme of the traditional GAN.

Generator Training Phase:

- We feed the generated data to the input of the discriminator.
- Tell the discriminator that they are real even though they are generated.
- What does the discriminator tell us, are they true or false?
- The discriminator tells us they are false -> that is an error signal that is propagated back to the generator.

https://www.researchgate.net/publication/341425292_Throwing_light_on_black_boxes_emergence_of_visual_categories_from_deep_learning
https://arxiv.org/abs/1406.2661

In other words, D and G play the following two-player minimax game with value function $V(G, D)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \quad (1)$$

From the discriminator's point of view:
The best possible result of D(G(z)) is close to 0.
This means that the discriminator won't be fooled.

https://www.researchgate.net/publication/341425292_Throwing_light_on_black_boxes_emergence_of_visual_categories_from_deep_learning
https://arxiv.org/abs/1406.2661

In other words, D and G play the following two-player minimax game with value function $V(G, D)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \quad (1)$$

From the discriminator's point of view:
The best possible result of D(G(z)) is close to 0.
This means that the discriminator won't be fooled.
What about D(x)?

https://www.researchgate.net/publication/341425292_Throwing_light_on_black_boxes_emergence_of_visual_categories_from_deep_learning
https://arxiv.org/abs/1406.2661

In other words, D and G play the following two-player minimax game with value function $V(G, D)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \quad (1)$$

From the discriminator's point of view:
The best possible result of D(G(z)) is close to 0.
This means that the discriminator won't be fooled.
What about D(x)?

The best possible result of D(x) is close to 1, from the discriminator's point of view.

https://www.researchgate.net/publication/341425292_Throwing_light_on_black_boxes_emergence_of_visual_categories_from_deep_learning
https://arxiv.org/abs/1406.2661

In other words, D and G play the following two-player minimax game with value function $V(G, D)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \quad (1)$$

From the discriminator's point of view:
The best possible result of D(G(z)) is close to 0.
This means that the discriminator won't be fooled.
What about D(x)?
The best possible result of D(x) is close to 1, From the discriminator's point of view.

From the generator's point of view:
The best possible result of D(G(z)) is 1.

This means that the generator outsmarted the discriminator.

https://www.researchgate.net/publication/341425292_Throwing_light_on_black_boxes_emergence_of_visual_categories_from_deep_learning
https://arxiv.org/abs/1406.2661

In other words, D and G play the following two-player minimax game with value function $V(G, D)$:

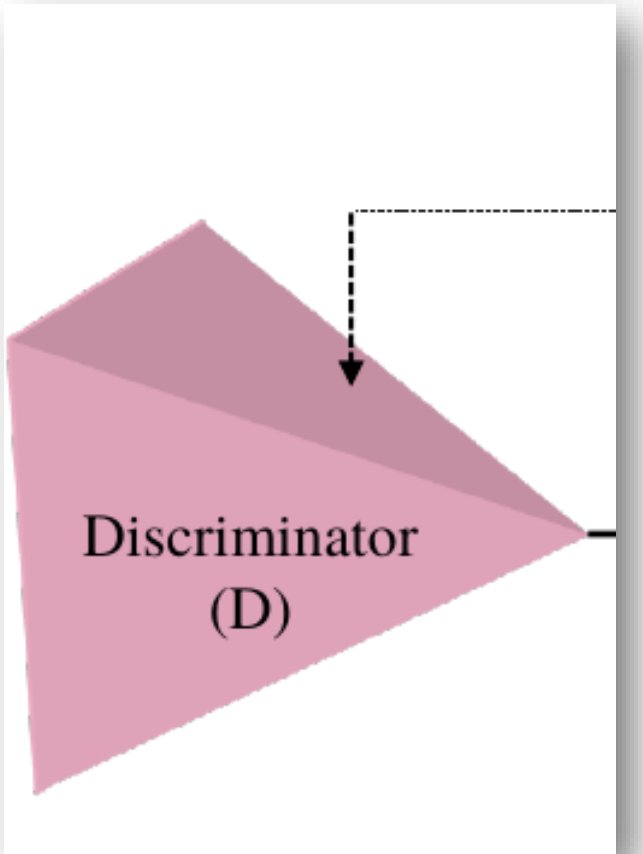
$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \quad (1)$$

From the discriminator's point of view:
The best possible result of D(G(z)) is close to 0.
This means that the discriminator won't be fooled.
What about D(x)?
The best possible result of D(x) is close to 1, From the discriminator's point of view.

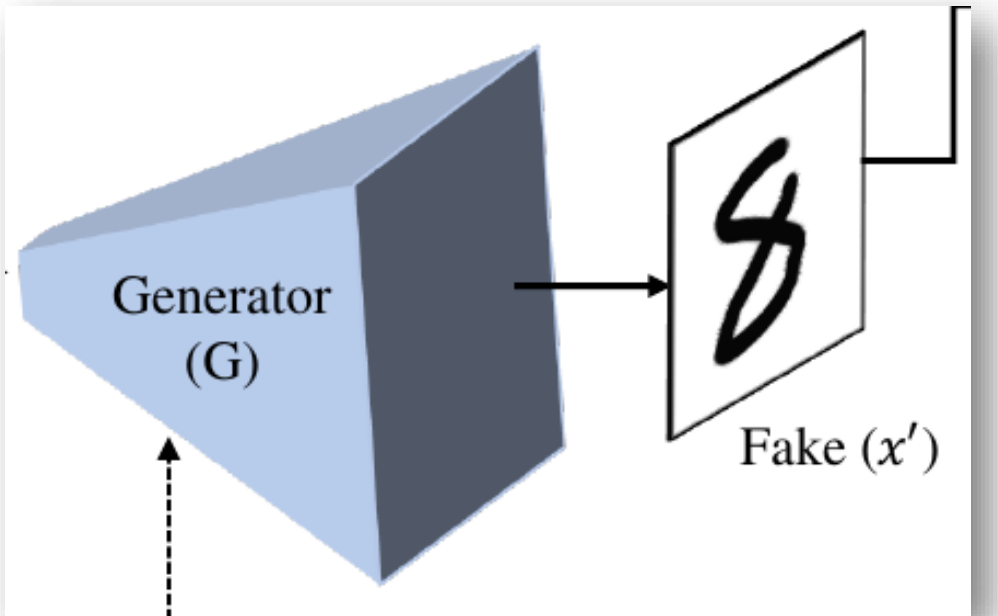
From the generator's point of view:
The best possible result of D(G(z)) is 1.
This means that the generator outsmarted the discriminator.

https://www.researchgate.net/publication/341425292_Throwing_light_on_black_boxes_emergence_of_visual_categories_from_deep_learning
https://arxiv.org/abs/1406.2661

```python
class Discriminator_0(nn.Module):
    def __init__(self):
        super().__init__()
        self.model = nn.Sequential(
            nn.Flatten(),
            nn.Linear(IMG_SIZE*IMG_SIZE, 2048),
            nn.LeakyReLU(0.2),
            nn.Dropout(0.3),
            nn.Linear(2048, 1024),
            nn.LeakyReLU(0.2),
            nn.Dropout(0.3),
            nn.Linear(1024, 512),
            nn.LeakyReLU(0.2),
            nn.Dropout(0.3),
            nn.Linear(512, 256),
            nn.LeakyReLU(0.2),
            nn.Dropout(0.3),
            nn.Linear(256, 1),
            nn.Sigmoid()
        )

    def forward(self, x):
        output = self.model(x)
        return output
```



```python
class Generator_0(nn.Module):
    def __init__(self):
        super().__init__()
        self.model = nn.Sequential(
            nn.Linear(100, 256),
            nn.LeakyReLU(0.2),
            nn.Linear(256, 512),
            nn.LeakyReLU(0.2),
            nn.Linear(512, 1024),
            nn.LeakyReLU(0.2),
            nn.Linear(1024, IMG_SIZE*IMG_SIZE),
            nn.Tanh(),
        )

    def forward(self, x):
        output = self.model(x)
        return output
```



EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

Training phase

```
real_img, labels = data[0].to(device), data[1].to(device)
fake_img = g_net(torch.randn(BATCH_SIZE, 100).to(device))

real_labels = torch.ones(BATCH_SIZE,1).to(device)
fake_labels = torch.zeros(BATCH_SIZE,1).to(device)
```

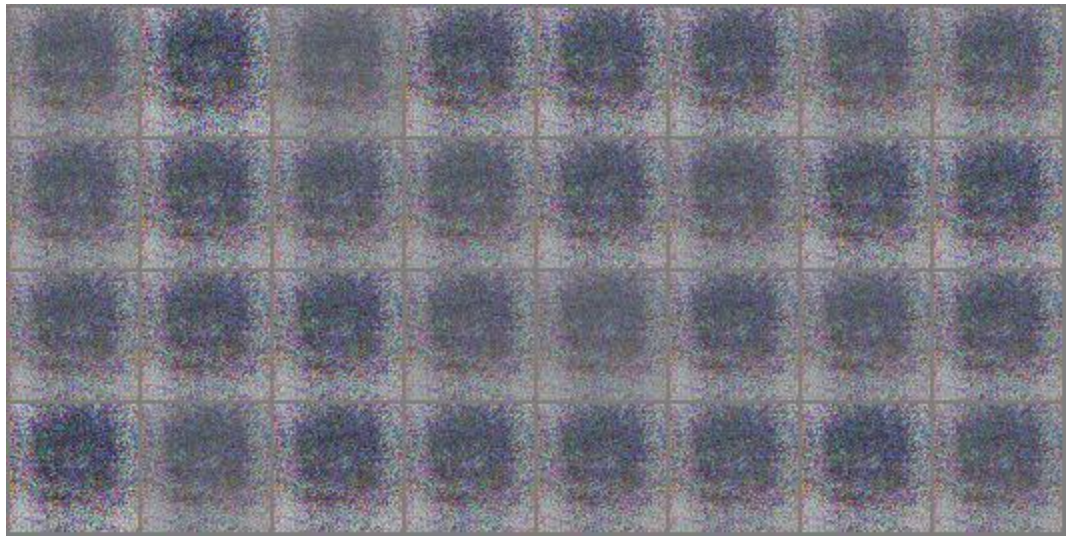
```
# REAL IMG - all labels are 1
pred_real   = d_net(real_img)
d_loss_real = lossfun(pred_real,real_labels)

# FAKE IMG - all labels are 0
pred_fake   = d_net(fake_img)
d_loss_fake = lossfun(pred_fake,fake_labels)

# combined losses
d_loss = d_loss_real + d_loss_fake

# backprop
d_optimizer.zero_grad()
d_loss.backward()
d_optimizer.step()
```

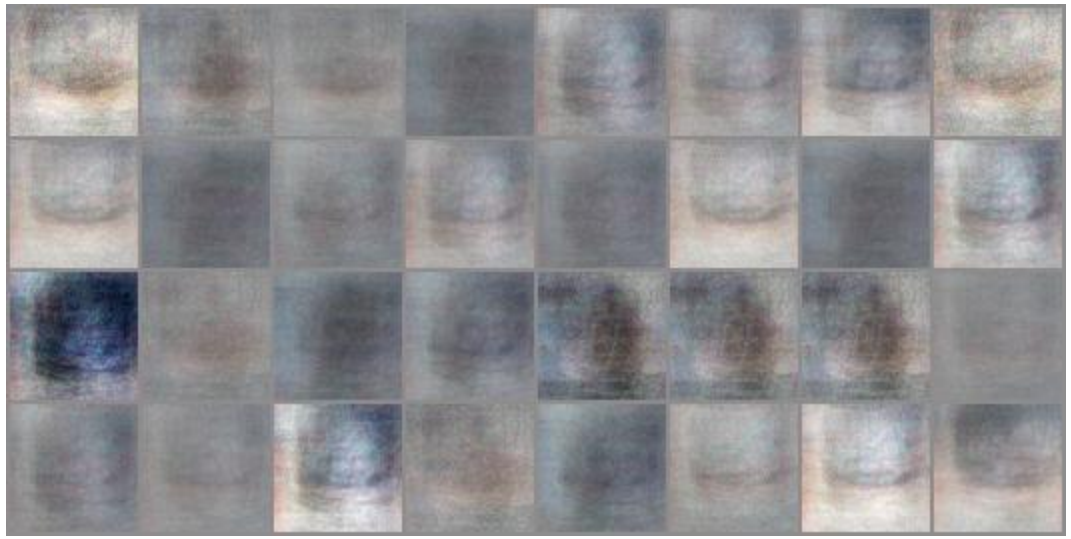
Training phase

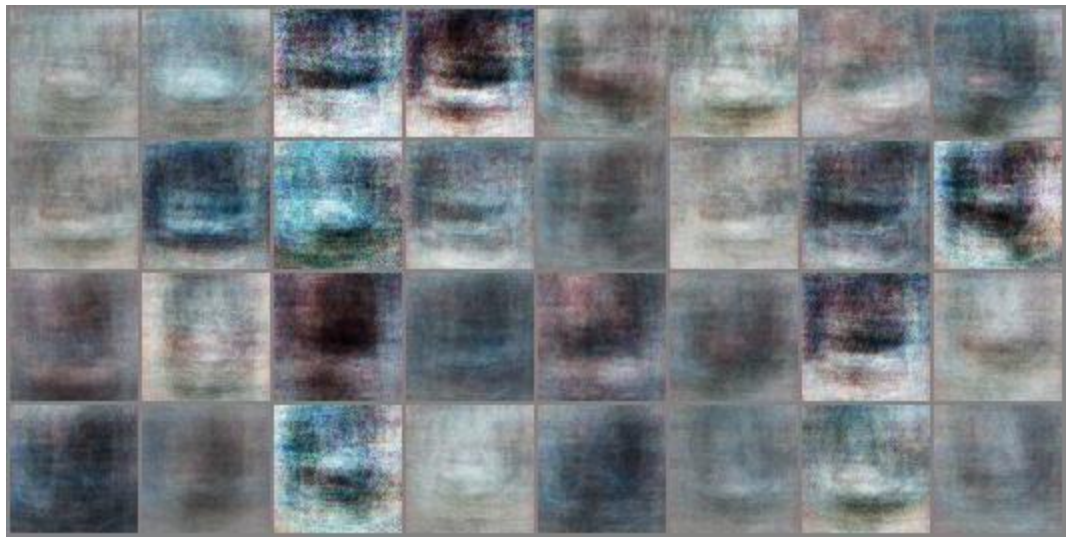
```
# create fake images
fake_images = g_net( torch.randn(BATCH_SIZE, 100).to(device) )
pred_fake   = d_net(fake_images)

# compute loss
g_loss = lossfun(pred_fake,real_labels)

# backprop
g_optimizer.zero_grad()
g_loss.backward()
g_optimizer.step()
```

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY







EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS

Alec Radford & Luke Metz
indico Research
Boston, MA
{alec,luke}@indico.io

Soumith Chintala
Facebook AI Research
New York, NY
soumith@fb.com

ABSTRACT

In recent years, supervised learning with convolutional networks (CNNs) has seen huge adoption in computer vision applications. Comparatively, unsupervised learning with CNNs has received less attention. In this work we hope to help bridge the gap between the success of CNNs for supervised learning and unsupervised learning. We introduce a class of CNNs called deep convolutional generative adversarial networks (DCGANs), that have certain architectural constraints, and demonstrate that they are a strong candidate for unsupervised learning. Training on various image datasets, we show convincing evidence that our deep convolutional adversarial pair learns a hierarchy of representations from object parts to scenes in both the generator and discriminator. Additionally, we use the learned features for novel tasks - demonstrating their applicability as general image representations.

Architecture guidelines for stable Deep Convolutional GANs

- Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm in both the generator and the discriminator.
- Remove fully connected hidden layers for deeper architectures.
- Use ReLU activation in generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

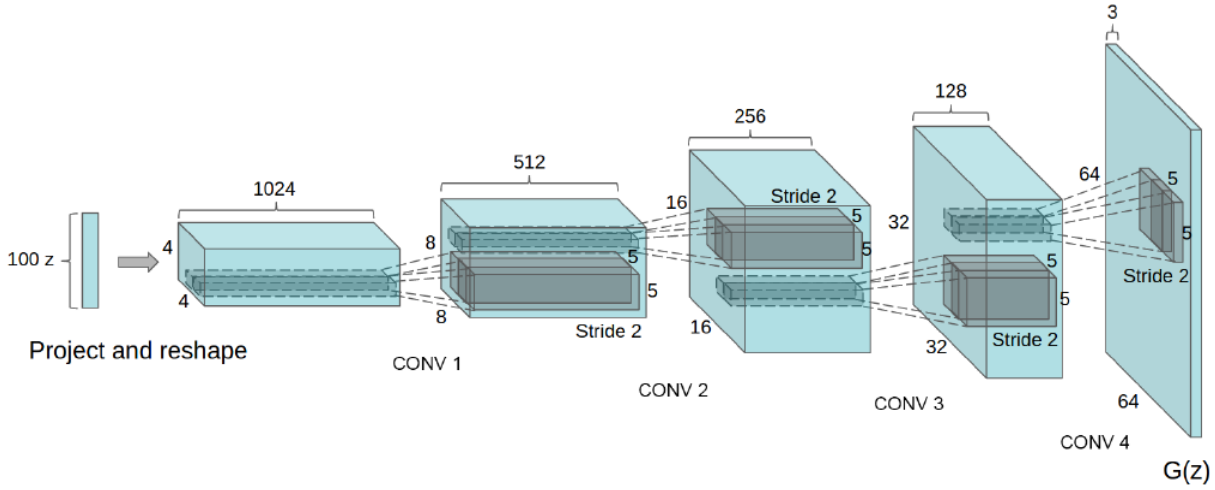


Figure 1: DCGAN generator used for LSUN scene modeling. A 100 dimensional uniform distribution Z is projected to a small spatial extent convolutional representation with many feature maps. A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions) then convert this high level representation into a 64×64 pixel image. Notably, no fully connected or pooling layers are used.

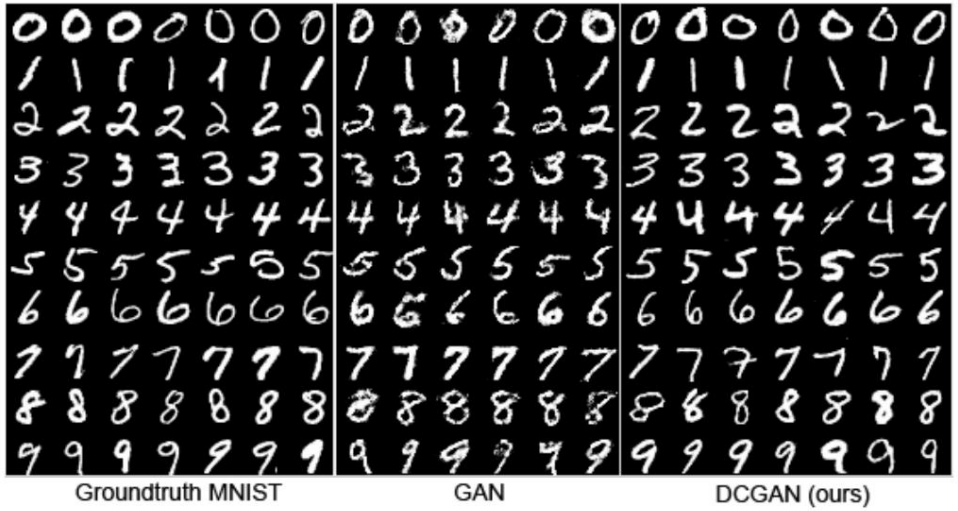
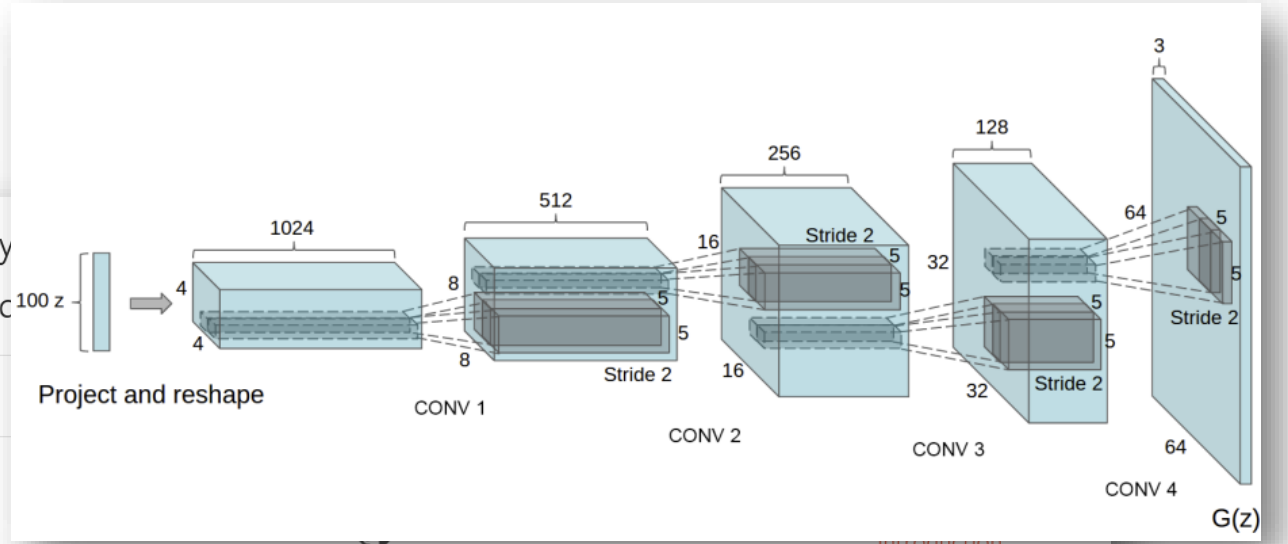


Figure 9: Side-by-side illustration of (from left-to-right) the MNIST dataset, generations from a baseline GAN, and generations from our DCGAN .

https://arxiv.org/abs/1511.06434



PyTorch

Get Started | Ecosystem | Py Ec

2.1.1+cu121

Search Tutorials

PyTorch Recipes [+]

Introduction to PyTorch [-]

- Learn the Basics
- Quickstart
- Tensors
- Datasets & DataLoaders
- Transforms
- Build the Neural Network
- Automatic Differentiation with `torch.autograd`
- Optimizing Model Parameters
- Save and Load the Model

Introduction to PyTorch on YouTube [-]

Tutorials > DCGAN Tutorial

Run in Google Colab

DCGAN TUTORIAL

Author: Nathan Inkawhich

Introduction

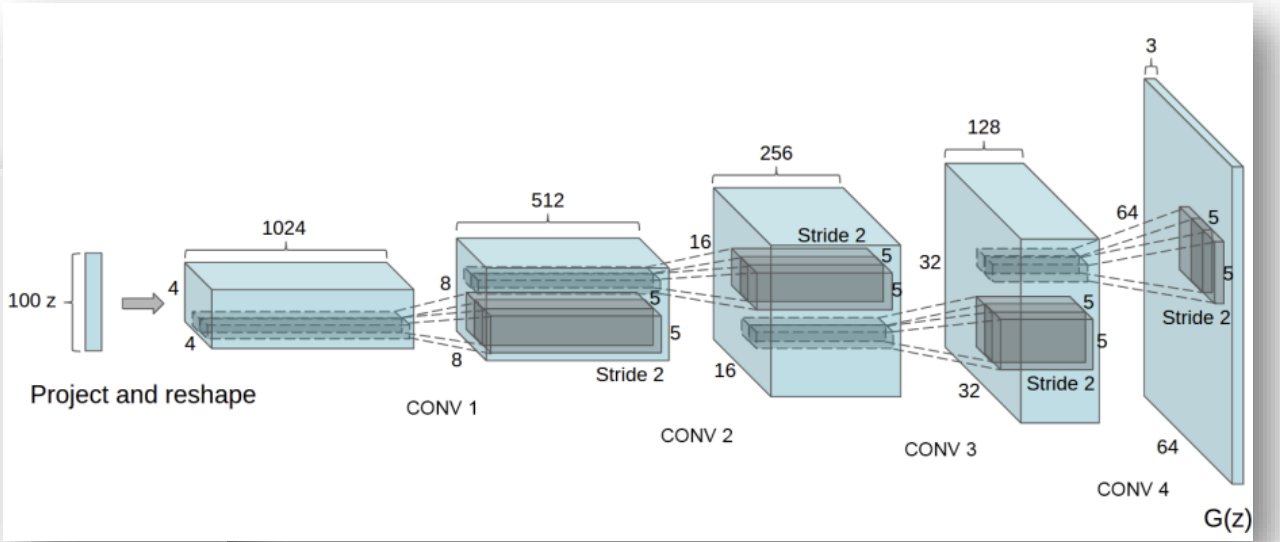
This tutorial will give an introduction to DCGANs through an example. We will train a generative adversarial network (GAN) to generate new celebrities after showing it pictures of many real celebrities. Most of the code here is from the DCGAN implementation in pytorch/examples, and this document will give a thorough explanation of the implementation and shed light on how and why this model works. But don't worry, no prior knowledge of GANs is required, but it may require a first-timer to spend some time reasoning about what is actually happening under the hood. Also, for the sake of time it will help to have a GPU, or two. Lets start from the beginning.

Generative Adversarial Networks

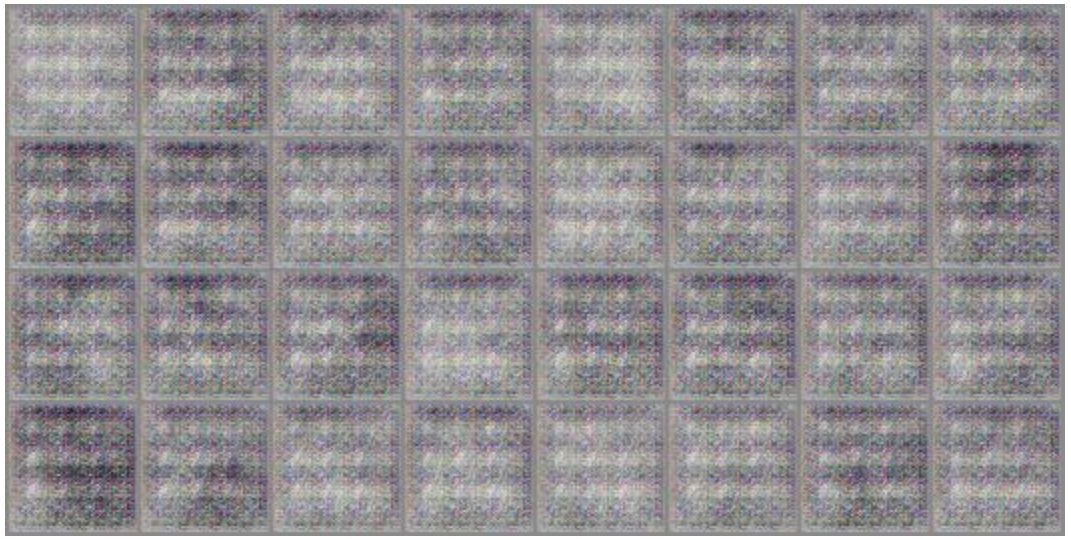
Introduction
+ Generative Adversarial Networks
Inputs
Data
+ Implementation
Results
Where to Go Next

https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html

```python
# Generator Code

class Generator(nn.Module):
    def __init__(self, ngpu):
        super(Generator, self).__init__()
        self.ngpu = ngpu
        self.main = nn.Sequential(
            # input is Z, going into a convolution
            nn.ConvTranspose2d( nz, ngf * 8, 4, 1, 0, bias=False),
            nn.BatchNorm2d(ngf * 8),
            nn.ReLU(True),
            # state size. ``(ngf*8) x 4 x 4``
            nn.ConvTranspose2d(ngf * 8, ngf * 4, 4, 2, 1, bias=False),
            nn.BatchNorm2d(ngf * 4),
            nn.ReLU(True),
            # state size. ``(ngf*4) x 8 x 8``
            nn.ConvTranspose2d( ngf * 4, ngf * 2, 4, 2, 1, bias=False),
            nn.BatchNorm2d(ngf * 2),
            nn.ReLU(True),
            # state size. ``(ngf*2) x 16 x 16``
            nn.ConvTranspose2d( ngf * 2, ngf, 4, 2, 1, bias=False),
            nn.BatchNorm2d(ngf),
            nn.ReLU(True),
            # state size. ``(ngf) x 32 x 32``
            nn.ConvTranspose2d( ngf, nc, 4, 2, 1, bias=False),
            nn.Tanh()
            # state size. ``(nc) x 64 x 64``
        )

    def forward(self, input):
        return self.main(input)
```



https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html



https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html

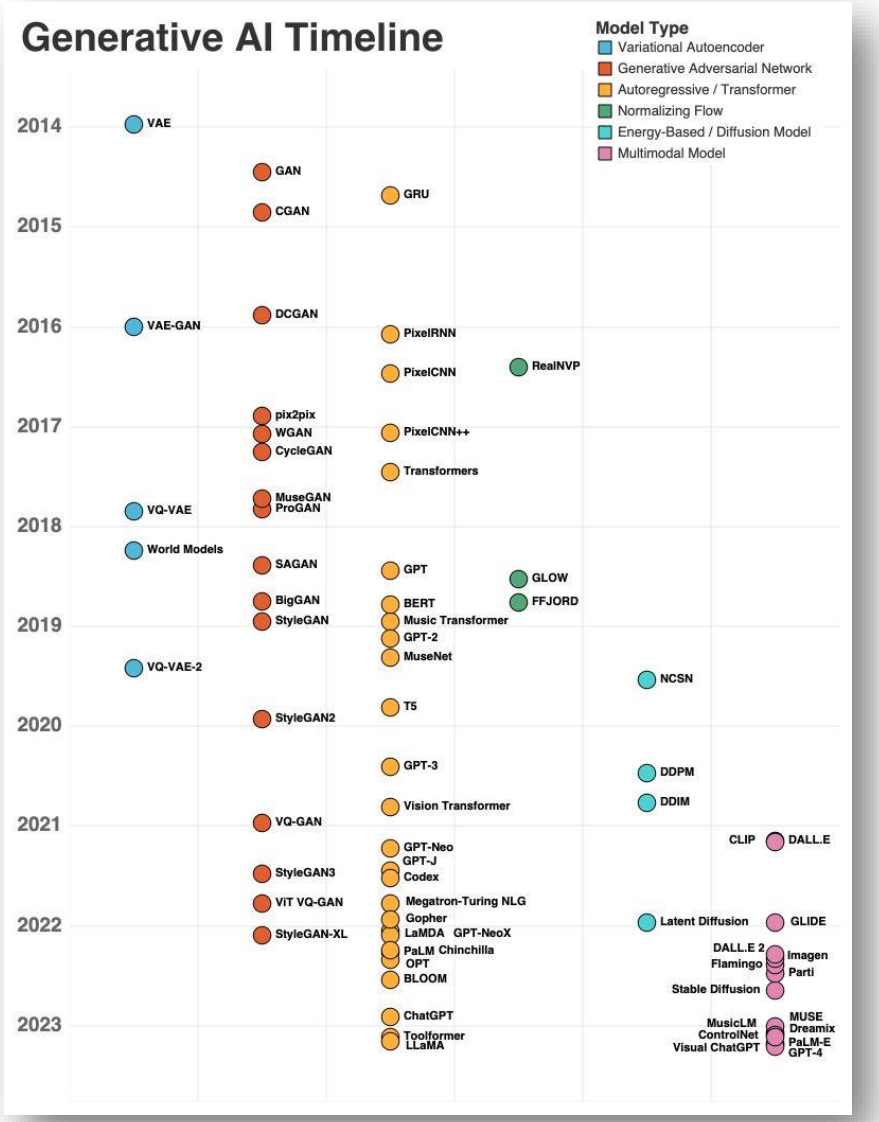
Real Images

Fake Images

https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html

Where to go from here?

https://www.kaggle.com/discussions/getting-started/397345



https://arxiv.org/pdf/2206.02262

Where to go from here?

Published as a conference paper at ICLR 2023

DIFFUSION-GAN: TRAINING GANS WITH DIFFUSION

Zhendong Wang[1,2], **Huangjie Zheng**[1,2], **Pengcheng He**[2], **Weizhu Chen**[2], **Mingyuan Zhou**[1]
[1]The University of Texas at Austin, [2]Microsoft Azure AI
{*zhendong.wang, huangjie.zheng*}*@utexas.edu,* {*penhe,wzchen*}*@microsoft.com*
mingyuan.zhou@mccombs.utexas.edu

ABSTRACT

Generative adversarial networks (GANs) are challenging to train stably, and a promising remedy of injecting instance noise into the discriminator input has not been very effective in practice. In this paper, we propose Diffusion-GAN, a novel GAN framework that leverages a forward diffusion chain to generate Gaussian-mixture distributed instance noise. Diffusion-GAN consists of three components, including an adaptive diffusion process, a diffusion timestep-dependent discriminator, and a generator. Both the observed and generated data are diffused by the same adaptive diffusion process. At each diffusion timestep, there is a different noise-to-data ratio and the timestep-dependent discriminator learns to distinguish the diffused real data from the diffused generated data. The generator learns from the discriminator's feedback by backpropagating through the forward diffusion chain, whose length is adaptively adjusted to balance the noise and data levels. We theoretically show that the discriminator's timestep-dependent strategy gives consistent and helpful guidance to the generator, enabling it to match the true data distribution. We demonstrate the advantages of Diffusion-GAN over strong GAN baselines on various datasets, showing that it can produce more realistic images with higher stability and data efficiency than state-of-the-art GANs.

https://arxiv.org/pdf/2206.02262

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

Where to go from here?

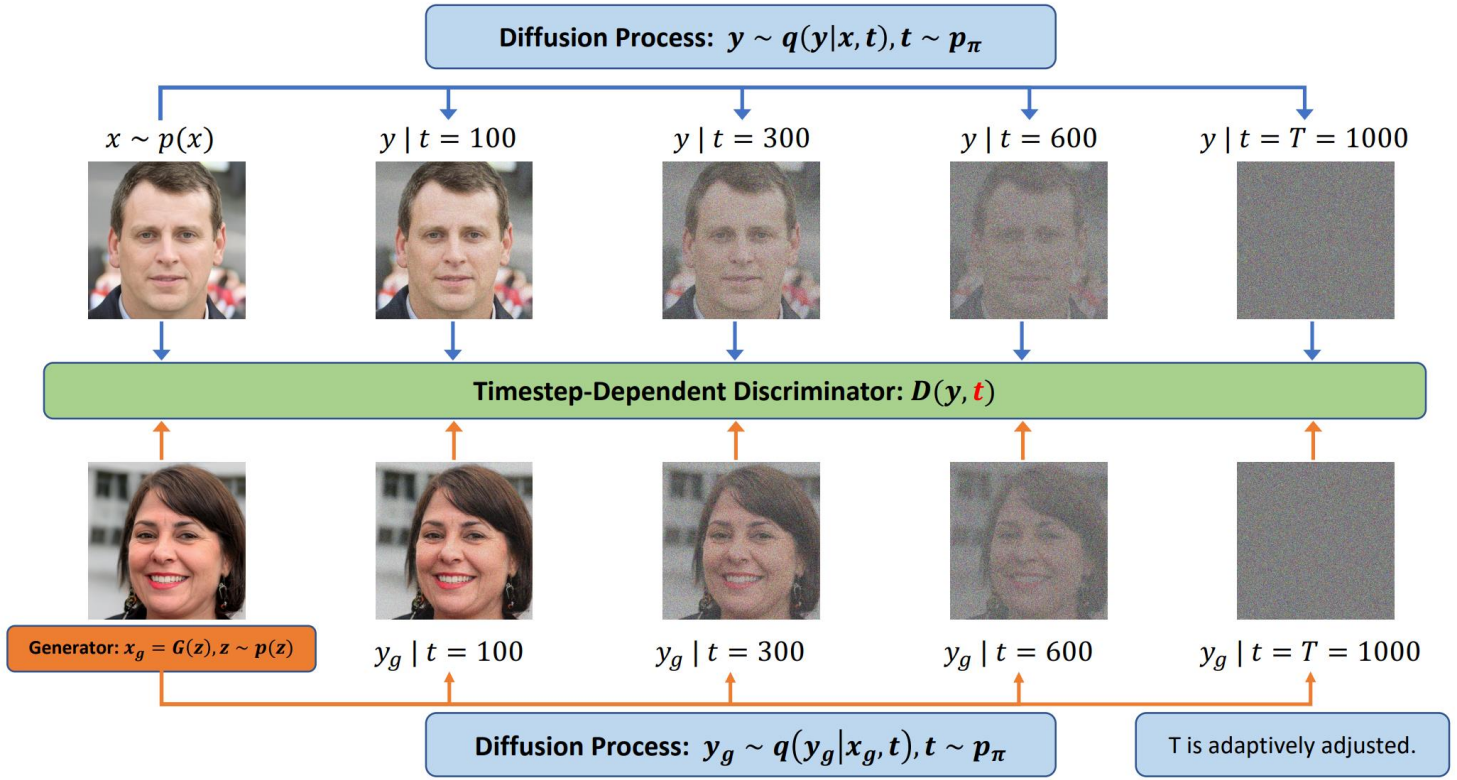


Figure 1: Flowchart for Diffusion-GAN. The top-row images represent the forward diffusion process of a real image, while the bottom-row images represent the forward diffusion process of a generated fake image. The discriminator learns to distinguish a diffused real image from a diffused fake image at all diffusion steps.

https://arxiv.org/pdf/2206.02262