Deep Learning – Parte 7
Redes Neurais Adversárias Generativas

Índice Geral

1 Introdução

- O conceito de redes adversárias generativas (GANs, do inglês *Generative Adversarial Networks*) (GUI et al., 2020) é bastante avançado e requer um preâmbulo para a sua melhor compreensão.
- Uma gramática de uma língua é dita generativa quando ela é capaz de gerar infinitas frases a partir de um conjunto finito de regras.
- O mesmo conceito vale para uma rede neural generativa quando, por exemplo, ela aprende a gerar um conjunto infinito de imagens sintéticas (portanto, artificiais) de algum domínio de interesse, como carros, faces humanas ou dormitórios. Evidentemente, não cabe nos restringirmos aqui a imagens, pois a proposta é extensível para qualquer tipo de padrão de entrada, como sequências sonoras e textos em linguagem natural.
- O desafio com a geração sintética de padrões é fazer com que eles não sejam (facilmente) distinguíveis de padrões reais e sejam controláveis.

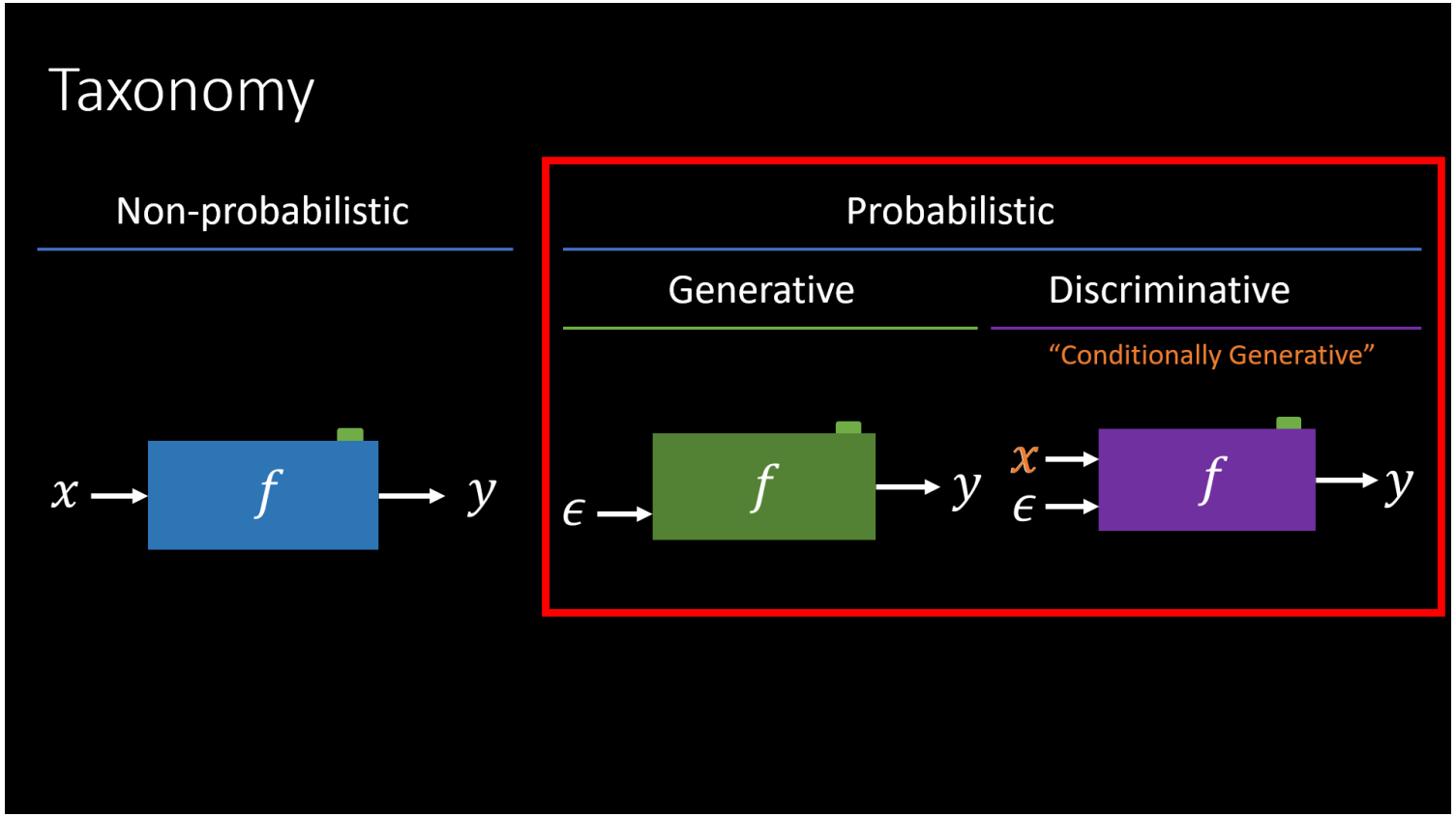
- Em outras palavras, deve-se garantir que a rede generativa não só capture a essência do padrão, mas também os seus detalhes e especificidades de estilo e forma. Além disso, a máquina generativa deve ser capaz de produzir o tipo de resultado comandado na entrada.
- É evidente que, de tudo que já foi visto ao longo do curso, havendo dados suficientes e contando com a flexibilidade dos mapeamentos produzidos por redes neurais profundas, técnicas de ajuste de pesos fundamentadas em gradiente descendente se mostram como bons candidatos.
- No entanto, por mais dados que se tenha disponível para o treinamento e por mais flexibilidade de mapeamento que possa ser explorada numa rede neural profunda, as redes generativas, quando treinadas isoladamente, não conseguiam um desempenho de alta qualidade em suas gerações sintéticas.
- O objetivo de reduzir o erro na saída estava se mostrando insuficiente para se atingir sínteses efetivamente realistas.

- Foi em meados de 2014 que um salto expressivo de qualidade no desempenho de redes generativas ocorreu (GOODFELLOW et al., 2014). Esse salto de qualidade se deu quando a rede generativa passou a ser treinada em conjunto com uma rede neural adversária.
- Essa rede neural adversária opera como um classificador binário que tem que se especializar em detectar o que é real (fornecido por um conjunto de treinamento) e o que é sintético (fornecido na saída da rede generativa).
- Essas duas redes neurais, uma generativa e a outra discriminativa, são adversárias porque o aumento de desempenho de uma rede, mantendo a outra fixa, implica na perda de desempenho da outra, de modo que ambas competem: a rede classificadora tenta discriminar cada vez melhor imagens sintéticas de imagens reais e a rede generativa tenta gerar imagens sintéticas que sejam cada vez menos discrimináveis de imagens reais.

- Ao final do treinamento conjunto dessas duas redes neurais adversárias, espera-se que o melhor que a rede classificadora pode fazer é ter um desempenho equivalente a de um classificador binário que escolhe aleatoriamente a classe de saída, com taxa de 50% para cada classe (imagem real ou imagem sintética).
- Nesta situação, fica evidente que a rede classificadora não é mais capaz de distinguir entre a fonte real e a sintética, por mais hábil que ela seja na discriminação entre real e *fake*. Logo, pode-se afirmar que a rede generativa aprendeu de fato a distribuição que está por trás da classe de padrões que se quer gerar.
- Até este ponto, já deve ter ficado claro que podemos interpretar modelos generativos como tendo um papel oposto ao de modelos discriminativos:
  - ✓ Um modelo discriminativo parte de um padrão completo e o mapeia numa classe, ao detectar nesse padrão de entrada atributos característicos da classe;

✓ Um modelo generativo parte de uma classe e procura gerar padrões completos que exibem atributos característicos daquela classe.

- Outra interpretação válida:



Fonte: https://github.com/nowozin/mlss2018-madrid-gan

- Capturar a distribuição que está por trás da classe de padrões que se quer gerar envolve descobrir e mapear corretamente o *manifold*, espaço de dimensão reduzida em que se manifestam as variações presentes nos padrões de entrada.
- O potencial dessas máquinas de aprendizado chamadas redes adversárias generativas (GANs) (do inglês *Generative Adversarial Networks*) é ilimitado, pois a rede generativa pode, em princípio, aprender qualquer distribuição de dados, se tornando uma "imitadora fiel" de qualquer padrão que se queira gerar.
- Numa área pródiga em ideias interessantes ao longo dessa última década, as GANs foram chamadas por Yann LeCun de "a ideia mais interessante nos últimos 10 anos em aprendizado de máquina".
- Vamos então abrir a "caixa preta" e ver como as GANs operam e são treinadas.

2 Princípio de operação das GANs

- Nesta seção, vamos nos basear no vídeo do ArXiv Insights: "*Face editing with Generative Adversarial Networks*": https://www.youtube.com/watch?v=dCKbRCUyop8

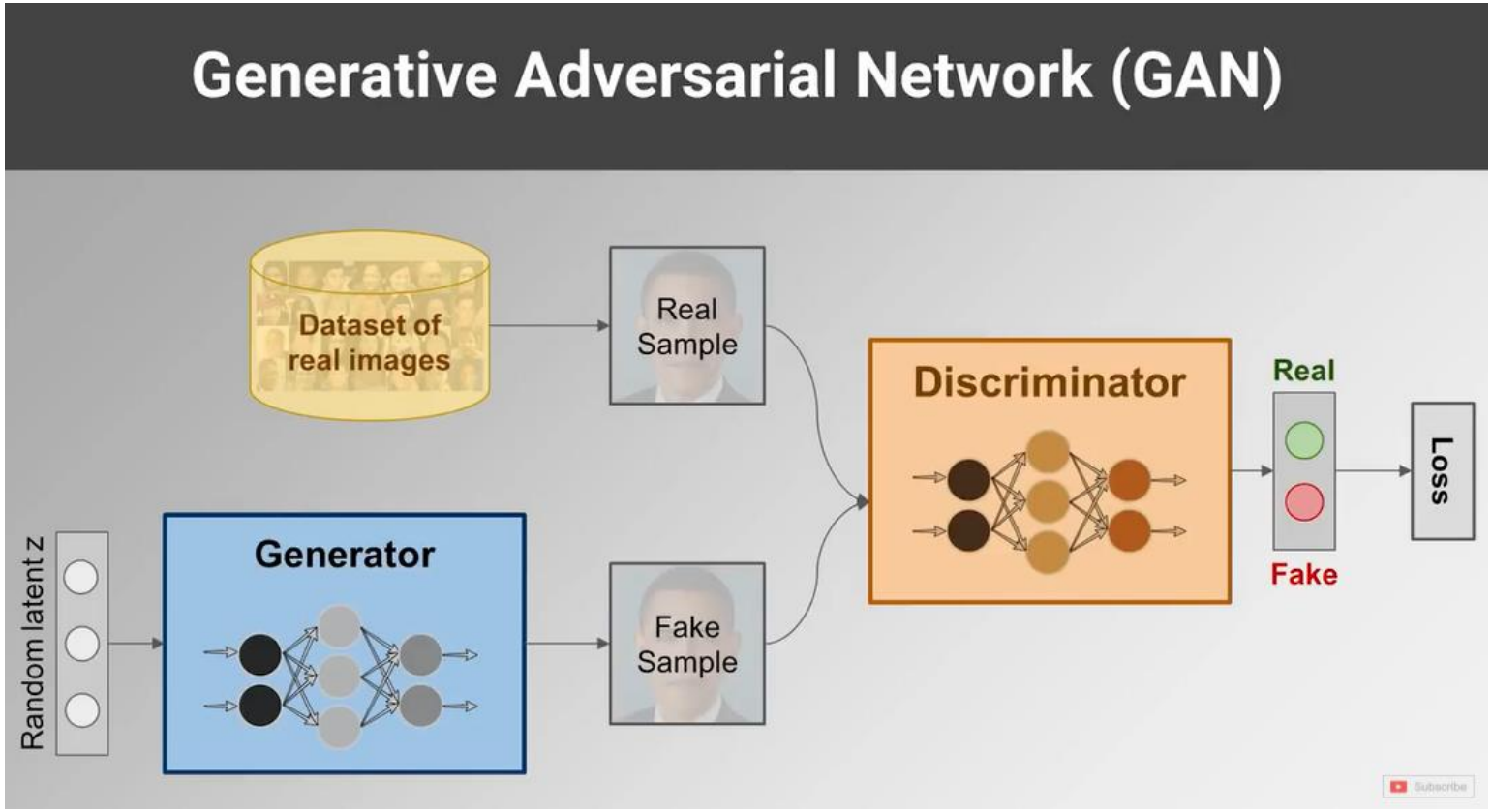


Figura 1 – Princípio de operação de uma GAN (Perspectiva 1).

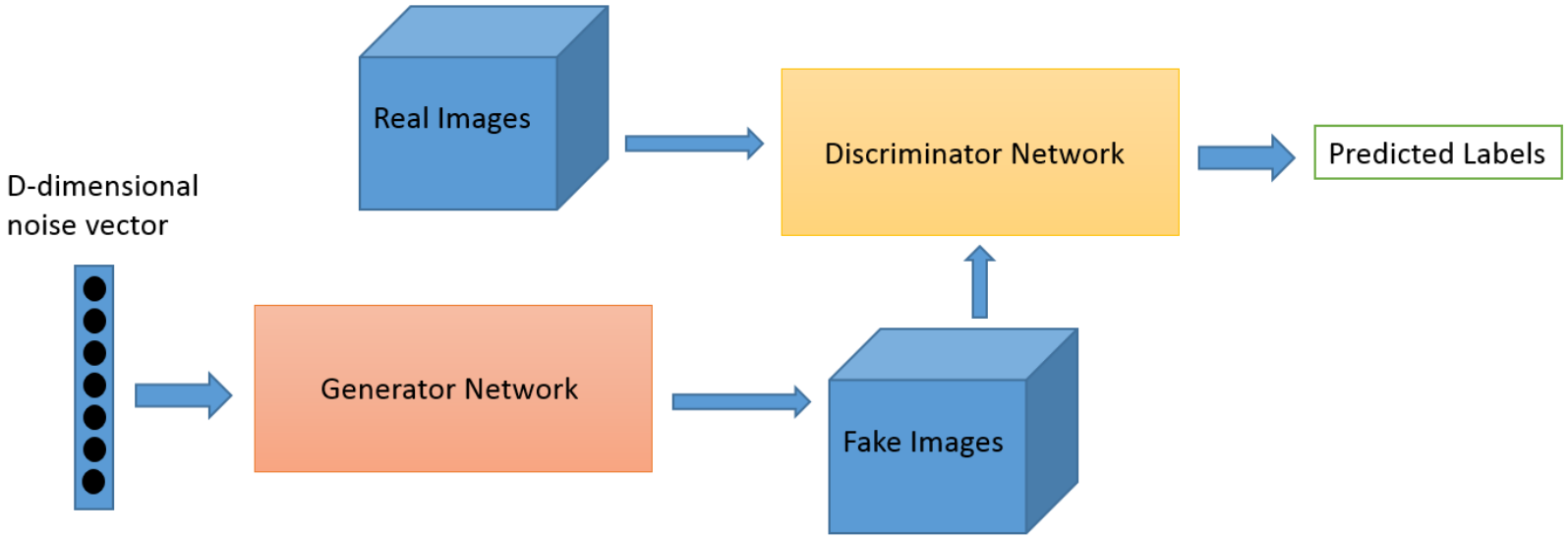


Figura 2 – Princípio de operação de uma GAN (Perspectiva 2).

Fonte: [http://deeplearningbook.com.br/introducao-as-redes-adversarias-generativas-gans-generative-adversarial-networks/]

- Vamos tomar como tarefa gerar de forma sintética dígitos manuscritos compatíveis com aqueles presentes na base MNIST.

- A Figura 3 mostra como especificar esse princípio de operação para poder lidar com a base MNIST. Evidentemente, é necessário capturar a estrutura fundamental que está por trás dos dígitos manuscritos, na distribuição dos dados.

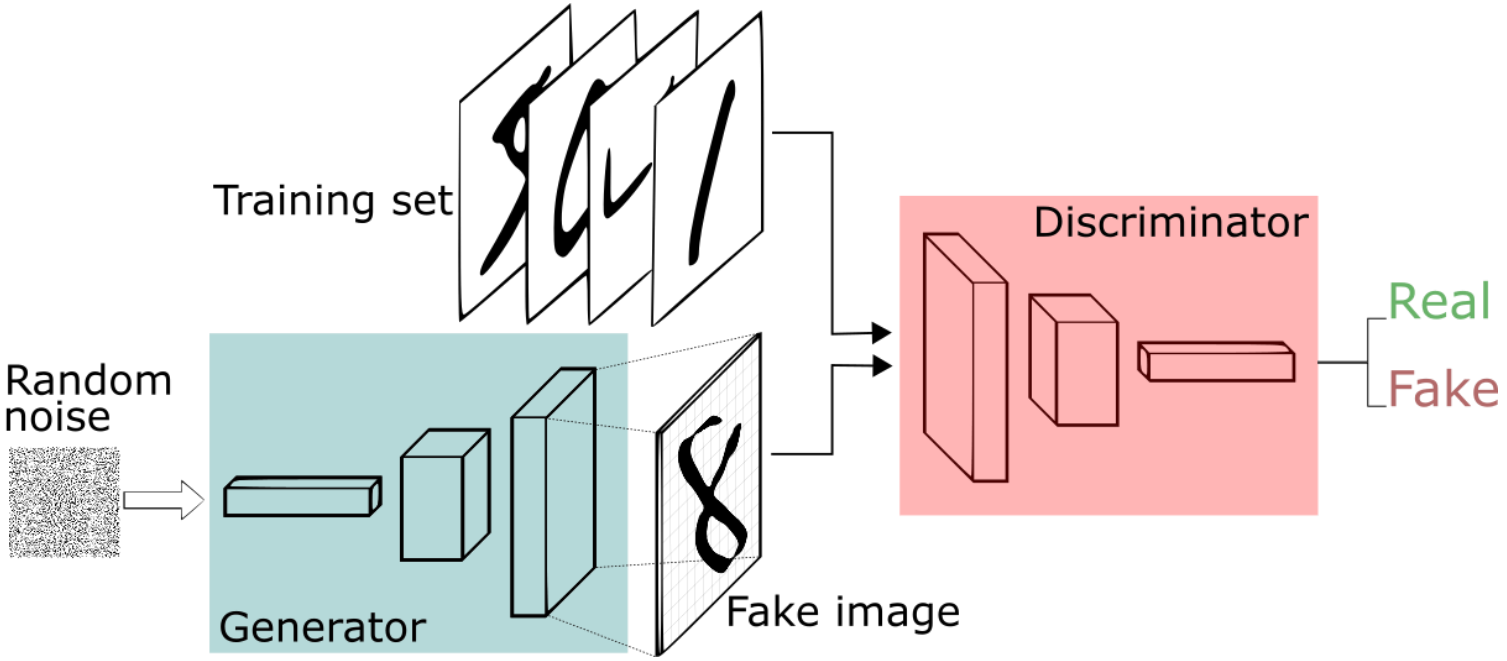


Figura 3 – GAN aplicada na síntese de dígitos manuscritos.

Fonte: [http://deeplearningbook.com.br/introducao-as-redes-adversarias-generativas-gans-generative-adversarial-networks/]

- Durante o treinamento, a rede generativa aprende a gerar imagens *fake* cada vez mais próximas de imagens reais, caso contrário, a rede discriminativa vai conseguir desempenhar o papel dela, como ilustrado nas figuras a seguir.

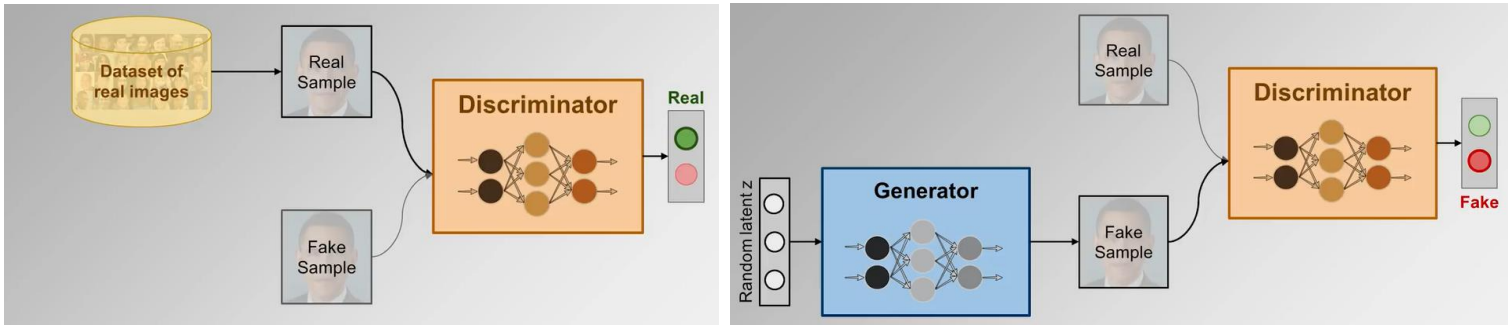


Figura 4 – Princípio de operação do classificador (rede discriminativa).

Fonte: [https://www.youtube.com/watch?v=dCKbRCUyop8]

- O potencial de imitar tão bem a realidade pode ser usado para os mais variados fins, nem todos virtuosos e com riscos para a identidade das pessoas. Assistam ao vídeo no link a seguir, envolvendo um discurso *fake* de Barack Obama.

https://www.youtube.com/watch?v=AmUC4m6w1wo

3 Estratégia de treinamento de uma GAN

- Embora a configuração da GAN seja bastante inovadora, o treinamento segue as mesmas estratégias já usadas para outras redes neurais profundas, a menos da proposição de uma função de custo mais elaborada, que reflete explicitamente os papéis a serem desempenhados pelas redes generativa e discriminativa.

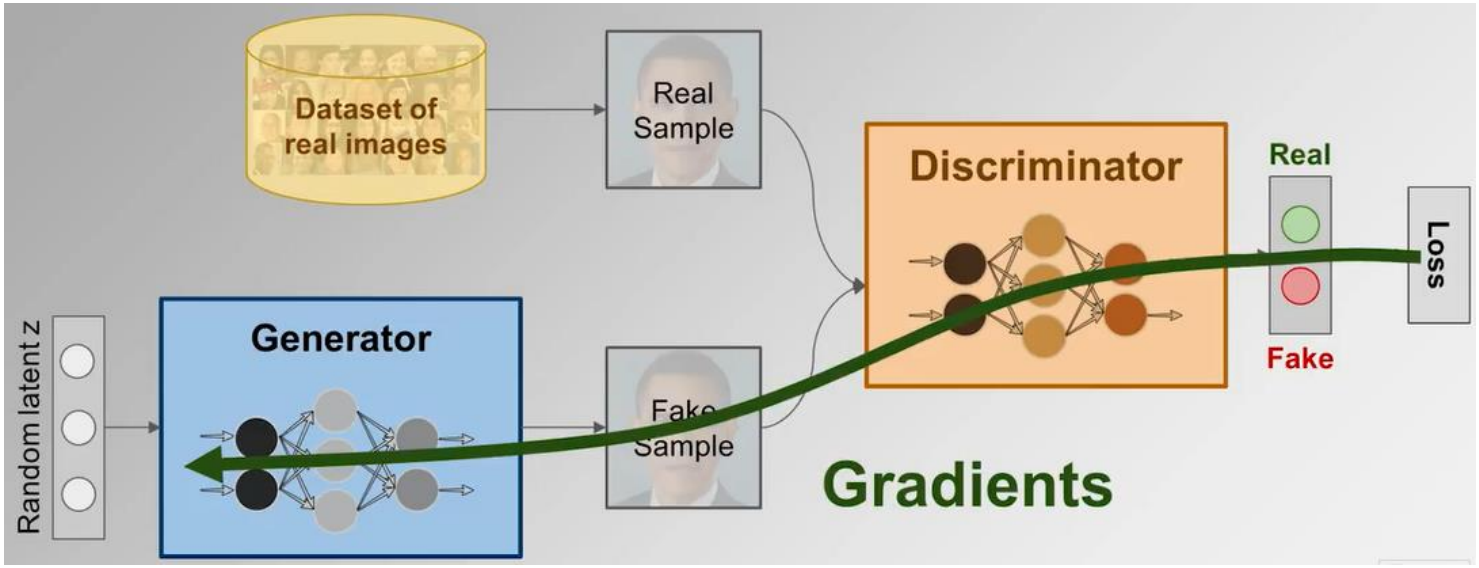


Figura 5 – Retropropagação do erro para se chegar ao vetor gradiente.

Fonte: [https://www.youtube.com/watch?v=dCKbRCUyop8]

- Durante o treinamento, é necessário evitar que o aprendizado de uma rede avance mais rápido que o da outra, ou seja, é necessário promover um certo equilíbrio no progresso do treinamento de ambas as redes neurais, na forma de um *mini-max game*.

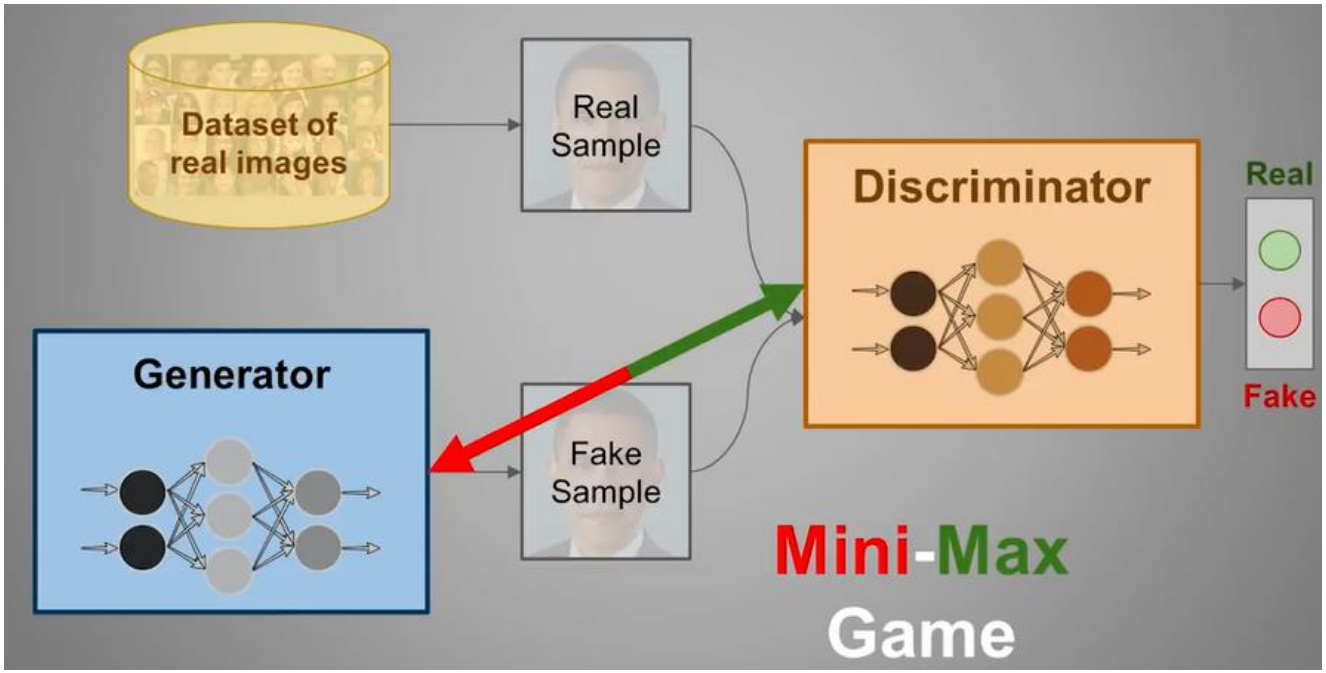
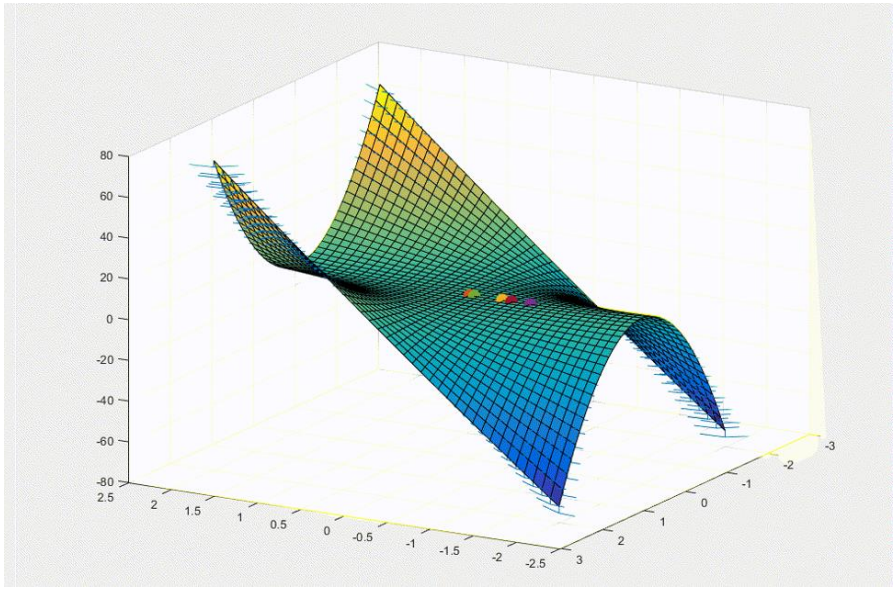
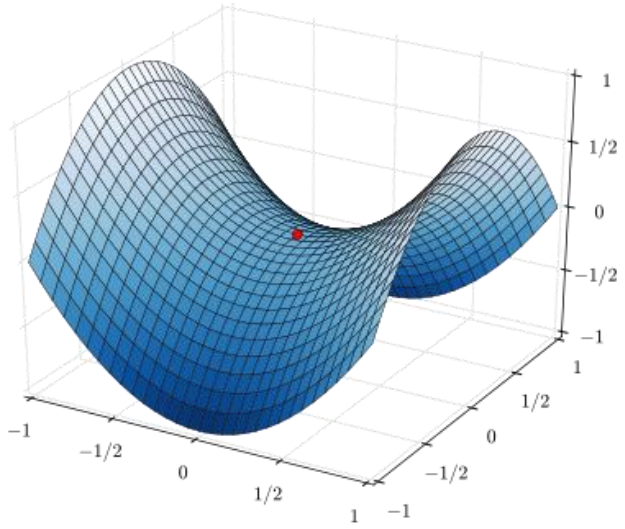


Figura 6 – Treinamento da GAN como um *mini-max game*.

Fonte: [https://www.youtube.com/watch?v=dCKbRCUyop8]

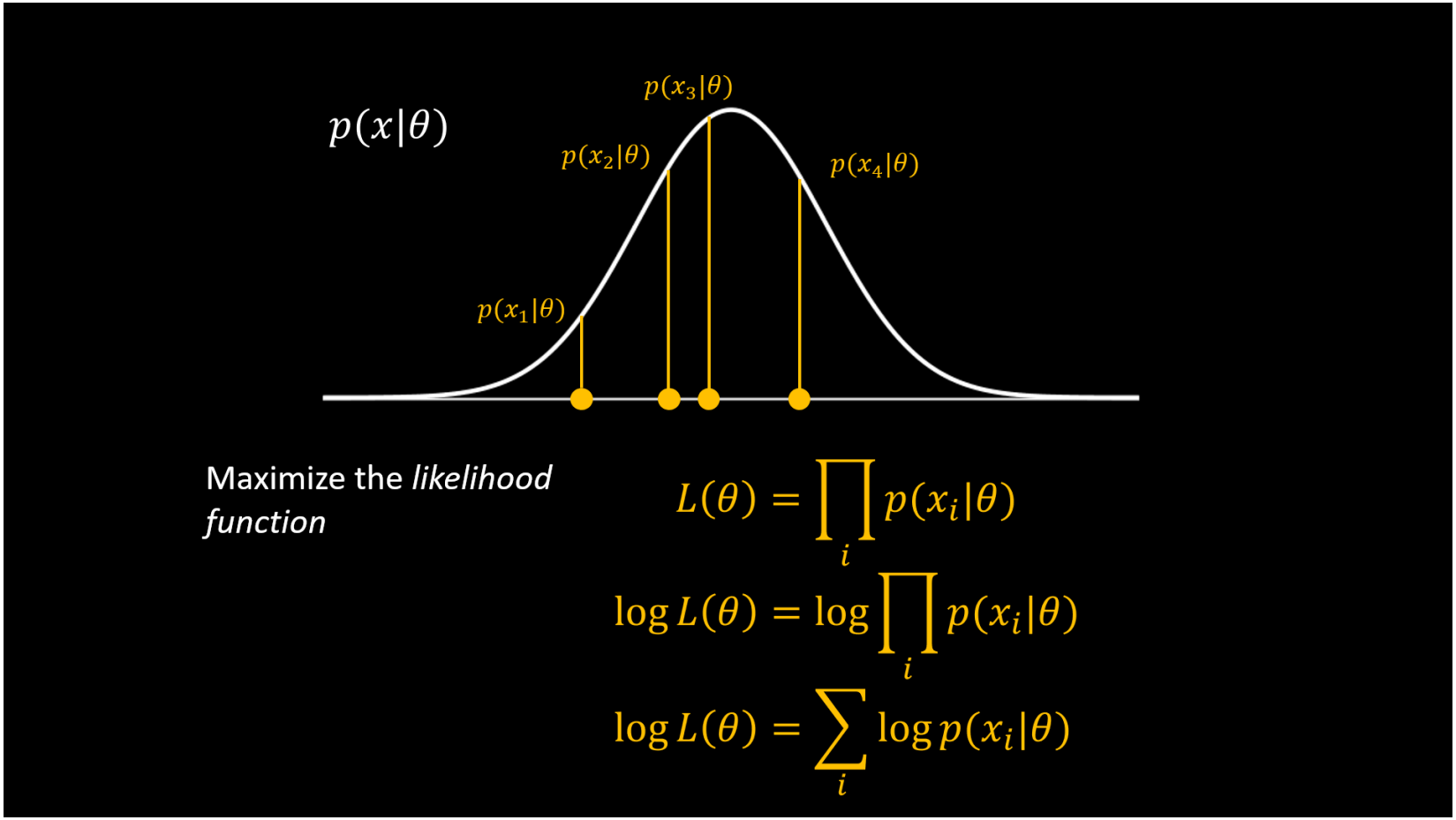




Fontes: https://github.com/nowozin/mlss2018-madrid-gan

https://en.wikipedia.org/wiki/Saddle_point

- Otimização sujeita a pontos de sela.



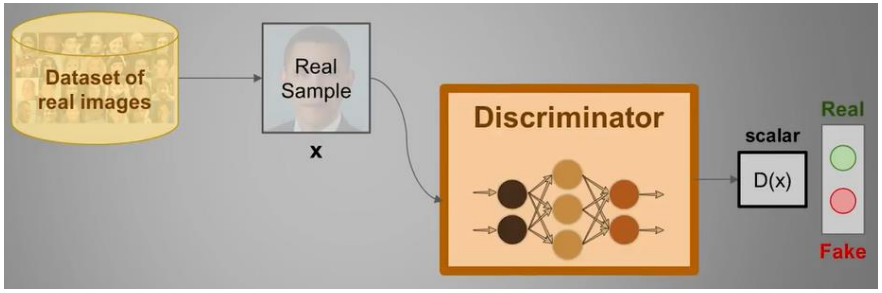
Fonte: https://github.com/nowozin/mlss2018-madrid-gan

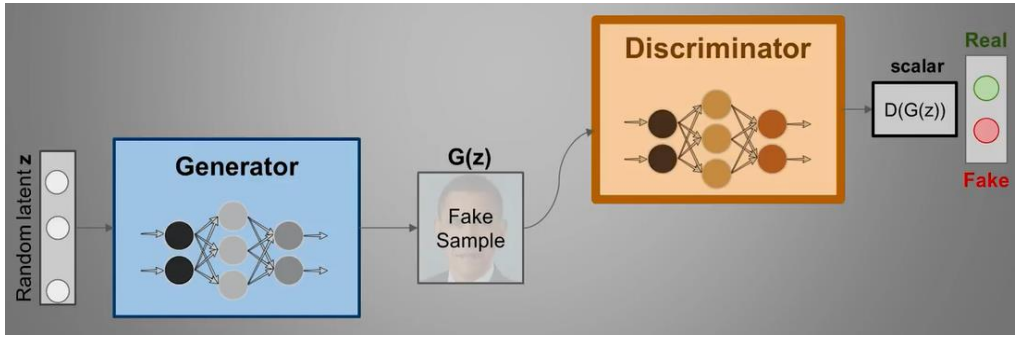
- A função objetivo a ser otimizada é a seguinte:

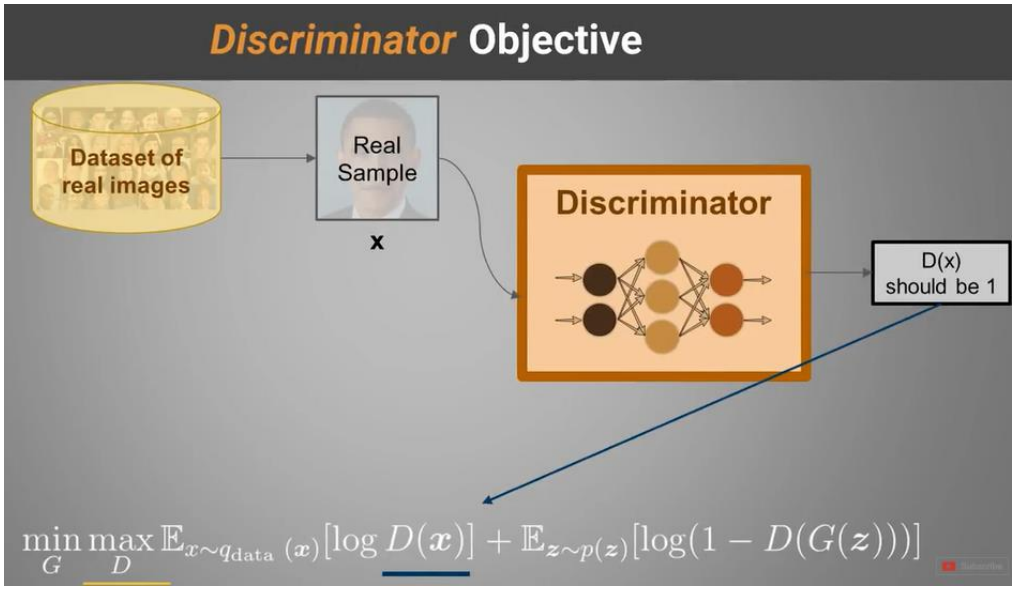
$$\min_{G} \max_{D} \mathbb{E}_{x \sim q_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z)))]$$

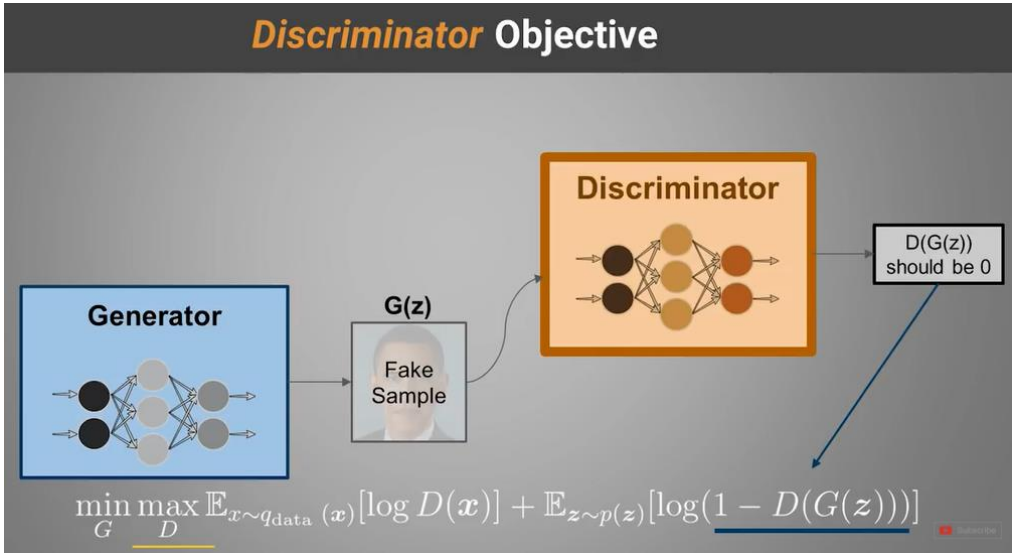
Training Loss

Mini-Max game:
- Minimize this loss wrt the Generator parameters
- Maximize this loss wrt the Discriminator parameters






Discriminator Objective
Dataset of real images
Real Sample
x
Discriminator
D(x) should be 1
$\min_G \max_D \mathbb{E}_{x \sim q_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z)))]$


Discriminator Objective
Discriminator
D(G(z)) should be 0
Generator
G(z)
Fake Sample
$\min_G \max_D \mathbb{E}_{x \sim q_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z)))]$

$$\min_G \max_D \mathbb{E}_{x \sim q_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z)))]$$

We want the **Discriminator** to:

- Recognize real images x as 'real' → output a high value(close to 1)
- Recognize fake images G(z) as 'fake' → output a low value (close to 0)

$$\min_G \max_D \mathbb{E}_{x \sim q_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z)))]$$

We want the Generator to:

- Generate fake images G(z) that look real to the Discriminator

$$\min_G \max_D \mathbb{E}_{x \sim q_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z)))]$$

This part does not depend on the parameters of the Generator...

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

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Fonte: GOODFELLOW et al. (2014).

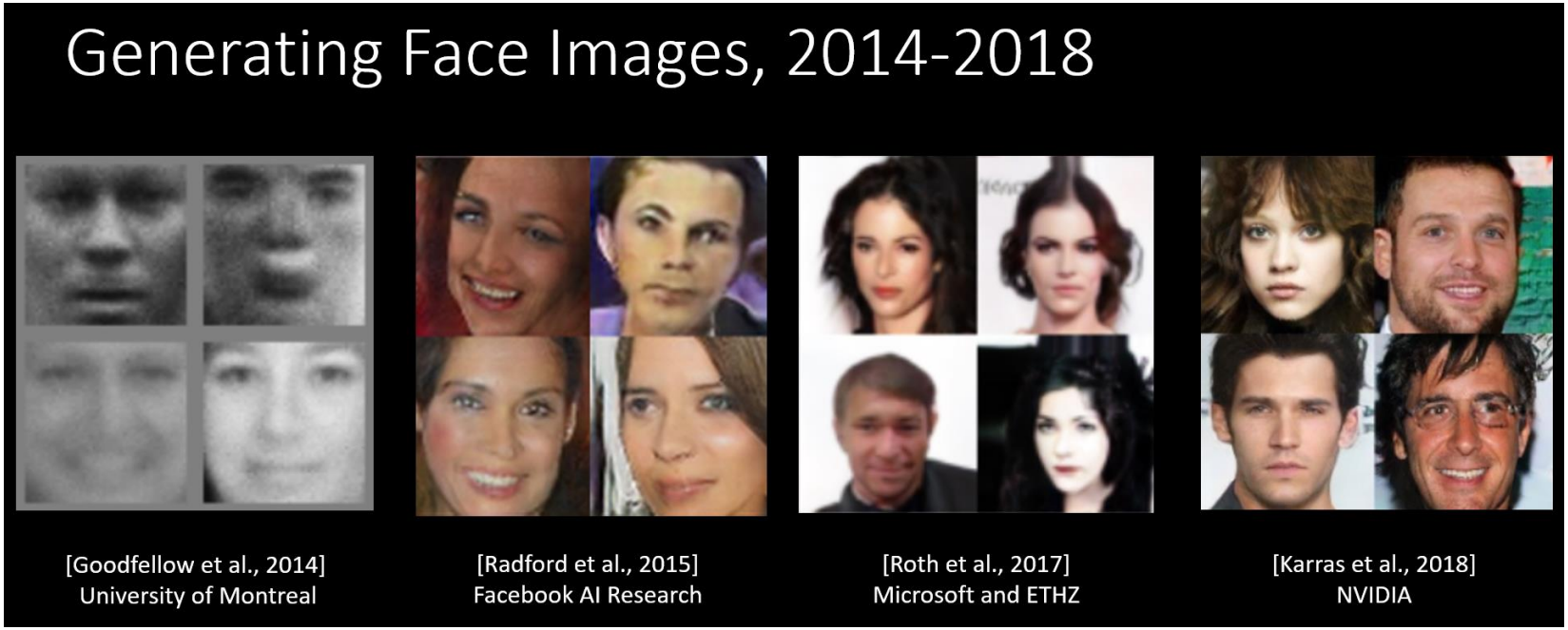
- Após a proposição inicial de GOODFELLOW et al. (2014), já foram concebidas variações que seguem o mesmo princípio operacional, conforme lista a seguir. Para mais detalhes, consultar: https://towardsdatascience.com/gan-objective-functions-gans-and-their-variations-ad77340bce3c

GAN Type	Key Take-Away
GAN	The original (JSD divergence)
WGAN	EM distance objective
Improved WGAN	No weight clipping on WGAN
LSGAN	L2 loss objective
RWGAN	Relaxed WGAN framework
McGAN	Mean/covariance minimization objective
GMMN	Maximum mean discrepancy objective
MMD GAN	Adversarial kernel to GMMN
Cramer GAN	Cramer distance
Fisher GAN	Chi-square objective
EBGAN	Autoencoder instead of discriminator
BEGAN	WGAN and EBGAN merged objectives
MAGAN	Dynamic margin on hinge loss from EBGAN

- Um link para centenas de extensões de GANs pode ser encontrado em:

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

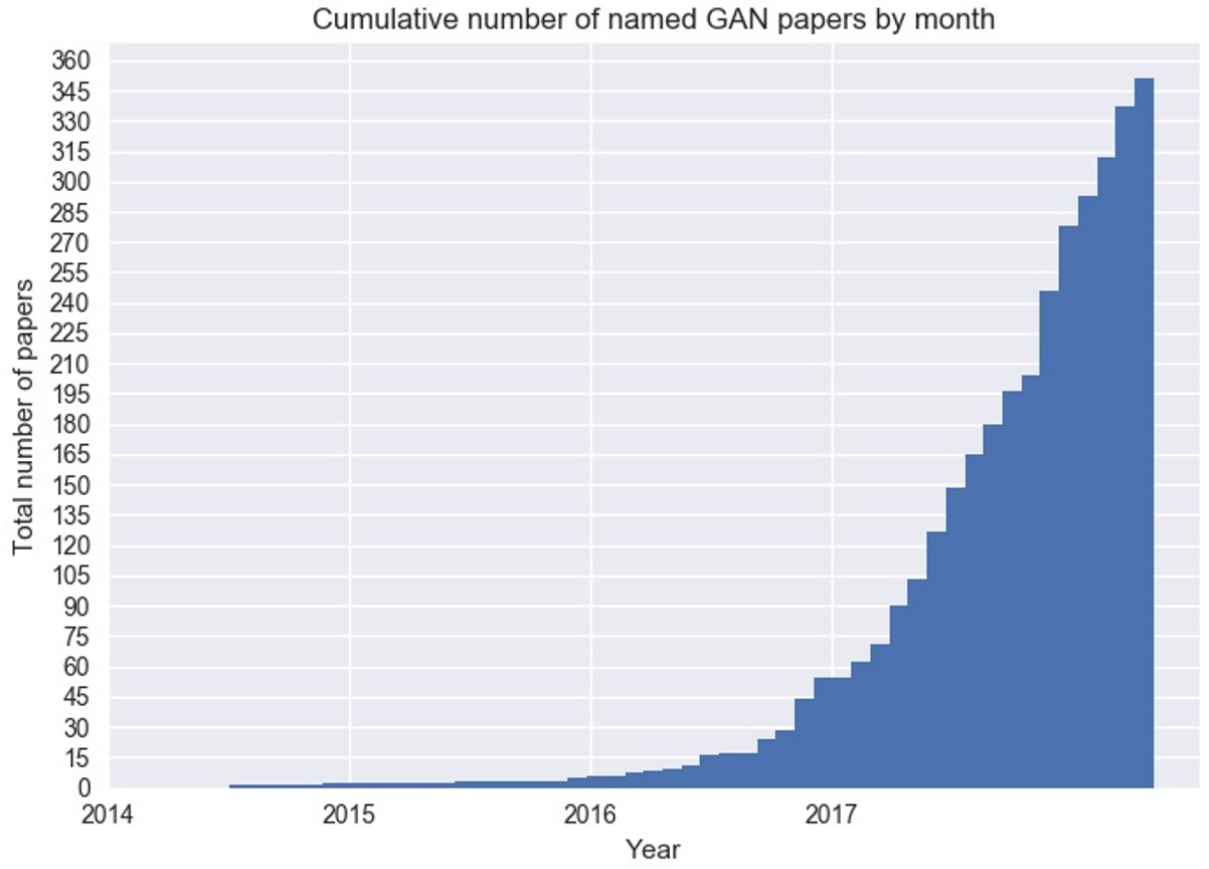
- Os resultados atuais são impressionantes, justamente porque a trajetória até chegar ao estado-da-arte foi árdua e fundamentada em muita pesquisa e inovação.



Fonte: https://github.com/nowozin/mlss2018-madrid-gan

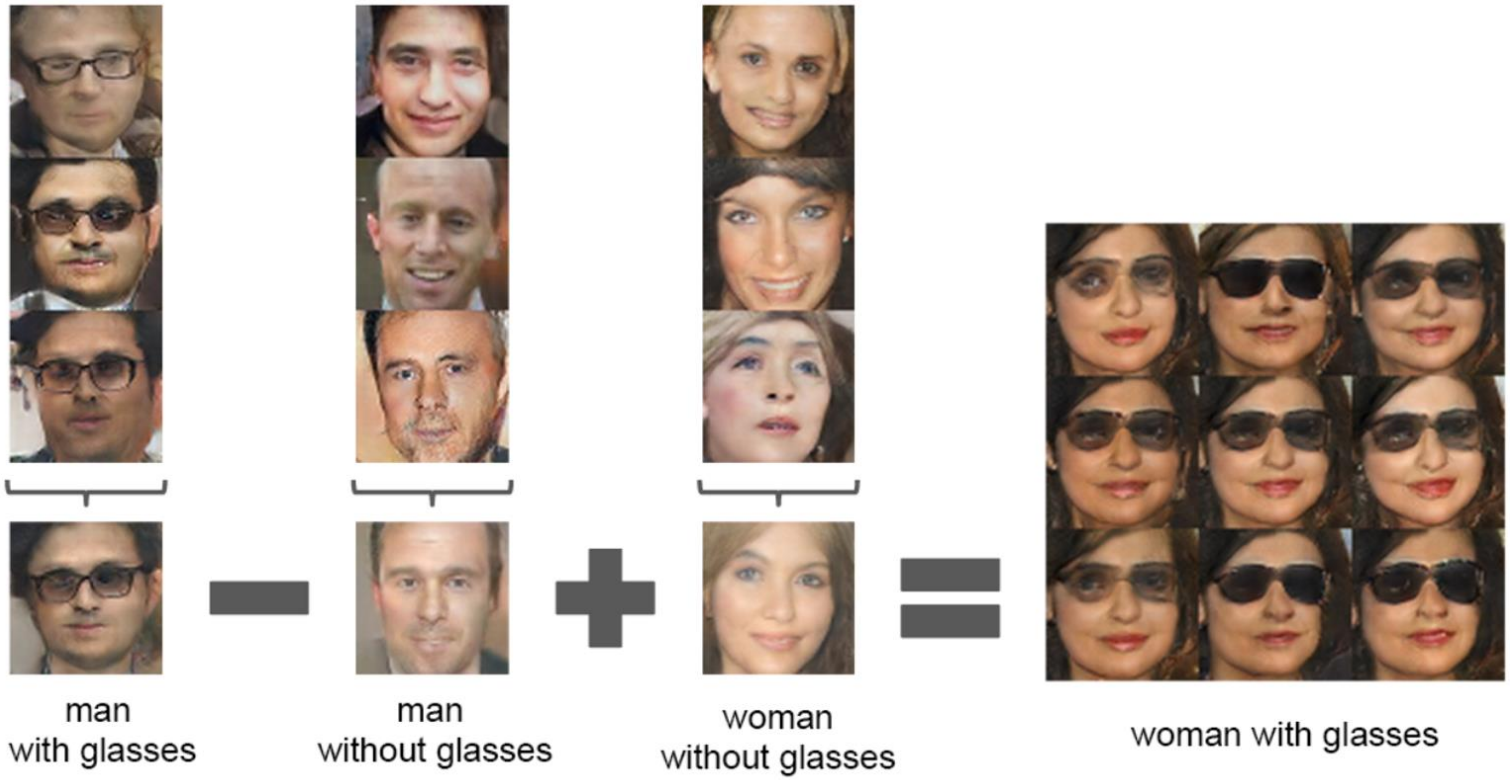
- Sugestão: consultar a webpage:

https://thispersondoesnotexist.com/



Cumulative number of GAN models (Credit: Bruno Gavranović)

Fonte (ano de 2018): https://github.com/nowozin/mlss2018-madrid-gan



Fonte: RADFORD et al. (2016)

4 GAN construtiva

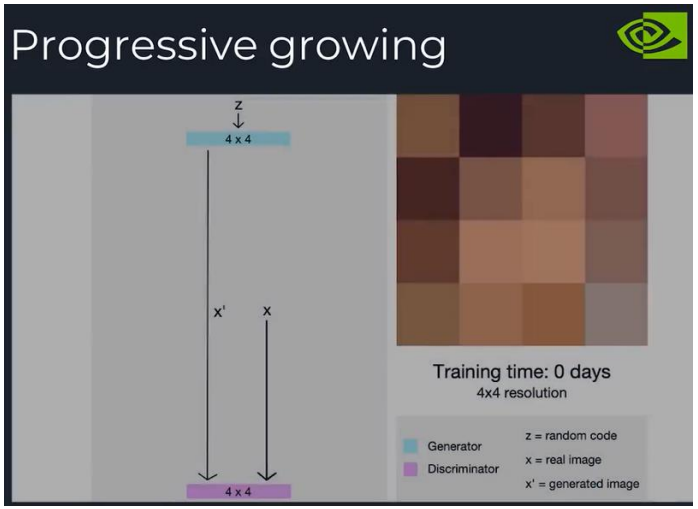
Published as a conference paper at ICLR 2018

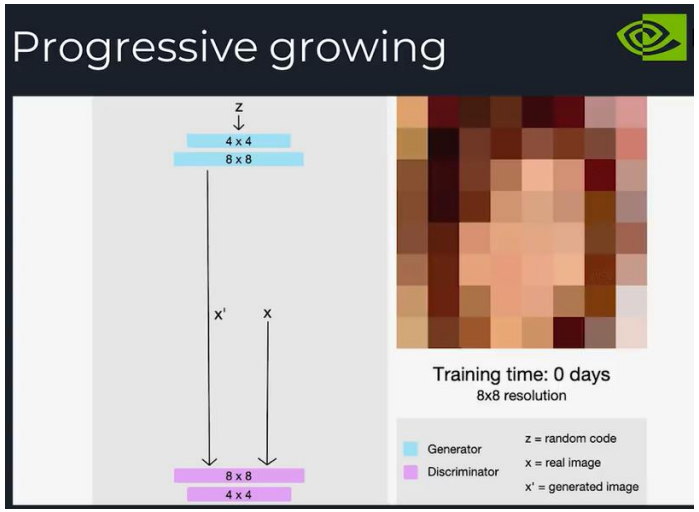
PROGRESSIVE GROWING OF GANS FOR IMPROVED QUALITY, STABILITY, AND VARIATION

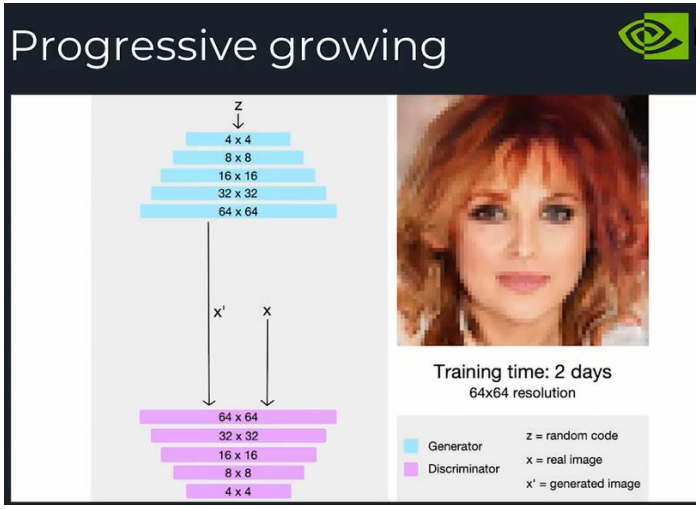
Tero Karras NVIDIA
Timo Aila NVIDIA
Samuli Laine NVIDIA
Jaakko Lehtinen NVIDIA and Aalto University
{tkarras,taila,slaine,jlehtinen}@nvidia.com

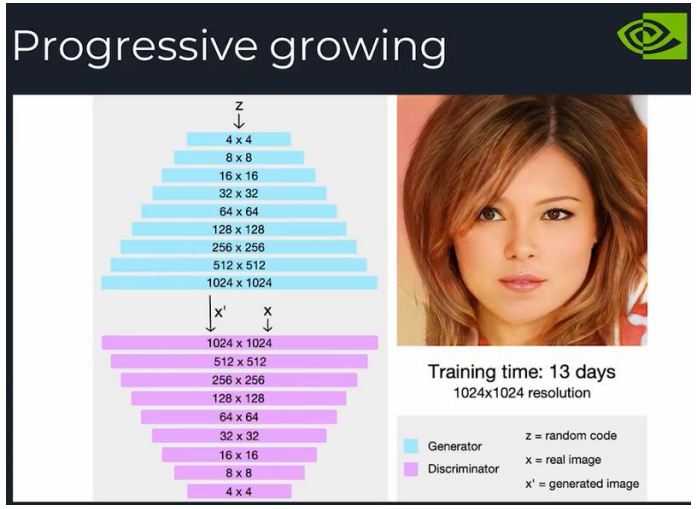
ABSTRACT

We describe a new training methodology for generative adversarial networks. The key idea is to grow both the generator and discriminator progressively: starting from a low resolution, we add new layers that model increasingly fine details as training progresses. This both speeds the training up and greatly stabilizes it, allowing us to produce images of unprecedented quality, e.g., CELEBA images at 1024^2. We also propose a simple way to increase the variation in generated images, and achieve a record inception score of 8.80 in unsupervised CIFAR10. Additionally, we describe several implementation details that are important for discouraging unhealthy competition between the generator and discriminator. Finally, we suggest a new metric for evaluating GAN results, both in terms of image quality and variation. As an additional contribution, we construct a higher-quality version of the CELEBA dataset.


Progressive growing
z
4 x 4
x'
x
4 x 4
Training time: 0 days
4x4 resolution
Generator
Discriminator
z = random code
x = real image
x' = generated image


Progressive growing
z
4 x 4
8 x 8
x'
x
8 x 8
4 x 4
Training time: 0 days
8x8 resolution
Generator
Discriminator
z = random code
x = real image
x' = generated image


Progressive growing
z
4 x 4
8 x 8
16 x 16
32 x 32
64 x 64
x'
x
64 x 64
32 x 32
16 x 16
8 x 8
4 x 4
Training time: 2 days
64x64 resolution
Generator
Discriminator
z = random code
x = real image
x' = generated image


Progressive growing
z
4 x 4
8 x 8
16 x 16
32 x 32
64 x 64
128 x 128
256 x 256
512 x 512
1024 x 1024
x'
x
1024 x 1024
512 x 512
256 x 256
128 x 128
64 x 64
32 x 32
16 x 16
8 x 8
4 x 4
Training time: 13 days
1024x1024 resolution
Generator
Discriminator
z = random code
x = real image
x' = generated image

5 StyleGAN

A Style-Based Generator Architecture for Generative Adversarial Networks

Tero Karras
NVIDIA
tkarras@nvidia.com

Samuli Laine
NVIDIA
slaine@nvidia.com

Timo Aila
NVIDIA
taila@nvidia.com

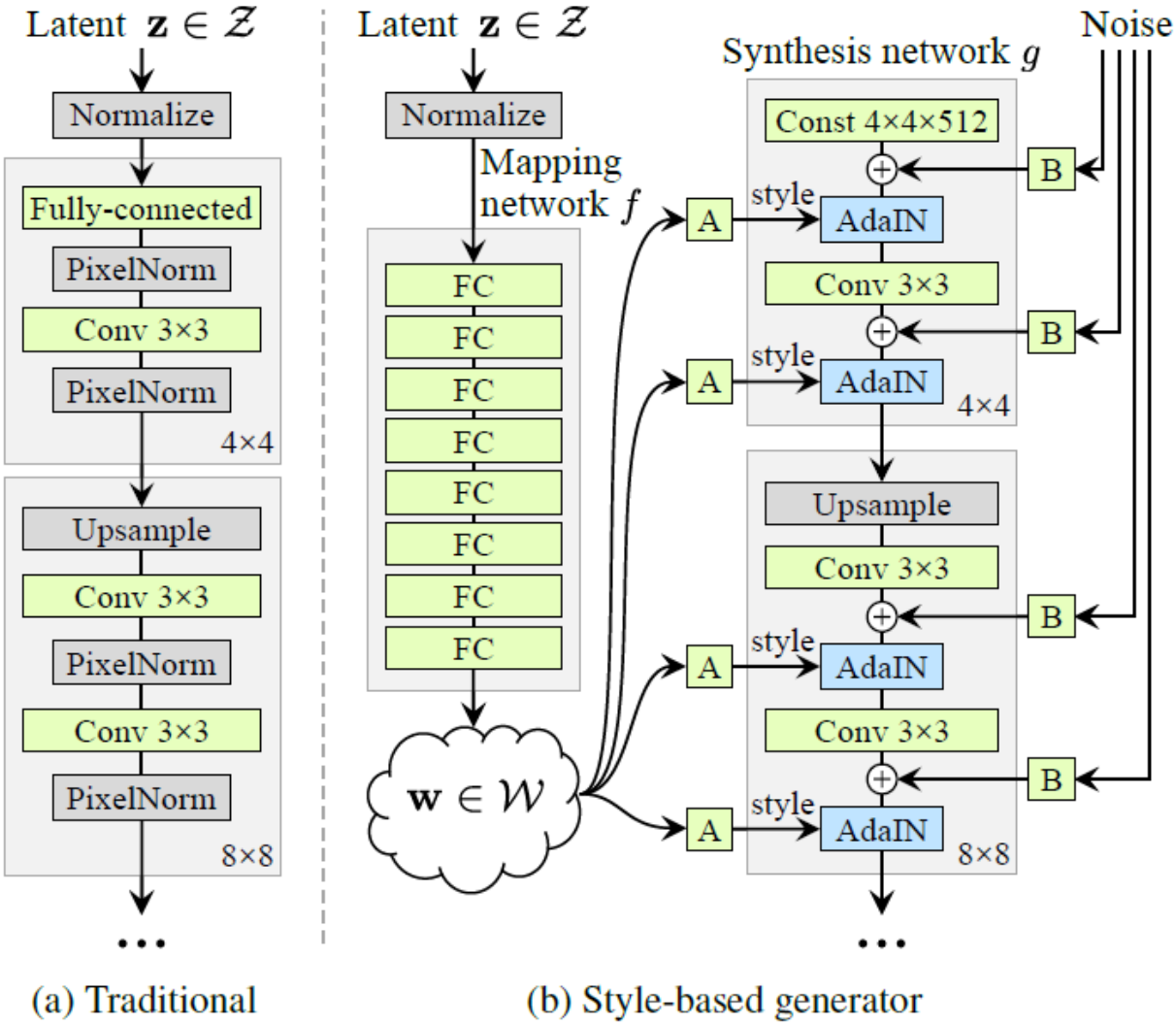
Abstract

We propose an alternative generator architecture for generative adversarial networks, borrowing from style transfer literature. The new architecture leads to an automatically learned, unsupervised separation of high-level attributes (e.g., pose and identity when trained on human faces) and stochastic variation in the generated images (e.g., freckles, hair), and it enables intuitive, scale-specific control of the synthesis. The new generator improves the state-of-the-art in terms of traditional distribution quality metrics, leads to demonstrably better interpolation properties, and also better disentangles the latent factors of variation. To quantify interpolation quality and disentanglement, we propose two new, automated methods that are applicable to any generator architecture. Finally, we introduce a new, highly varied and high-quality dataset of human faces.

(e.g., pose, identity) from stochastic variation (e.g., freckles, hair) in the generated images, and enables intuitive scale-specific mixing and interpolation operations. We do not modify the discriminator or the loss function in any way, and our work is thus orthogonal to the ongoing discussion about GAN loss functions, regularization, and hyperparameters [24, 45, 5, 40, 44, 36].

Our generator embeds the input latent code into an intermediate latent space, which has a profound effect on how the factors of variation are represented in the network. The input latent space must follow the probability density of the training data, and we argue that this leads to some degree of unavoidable entanglement. Our intermediate latent space is free from that restriction and is therefore allowed to be disentangled. As previous methods for estimating the degree of latent space disentanglement are not directly applicable in our case, we propose two new automated metrics —

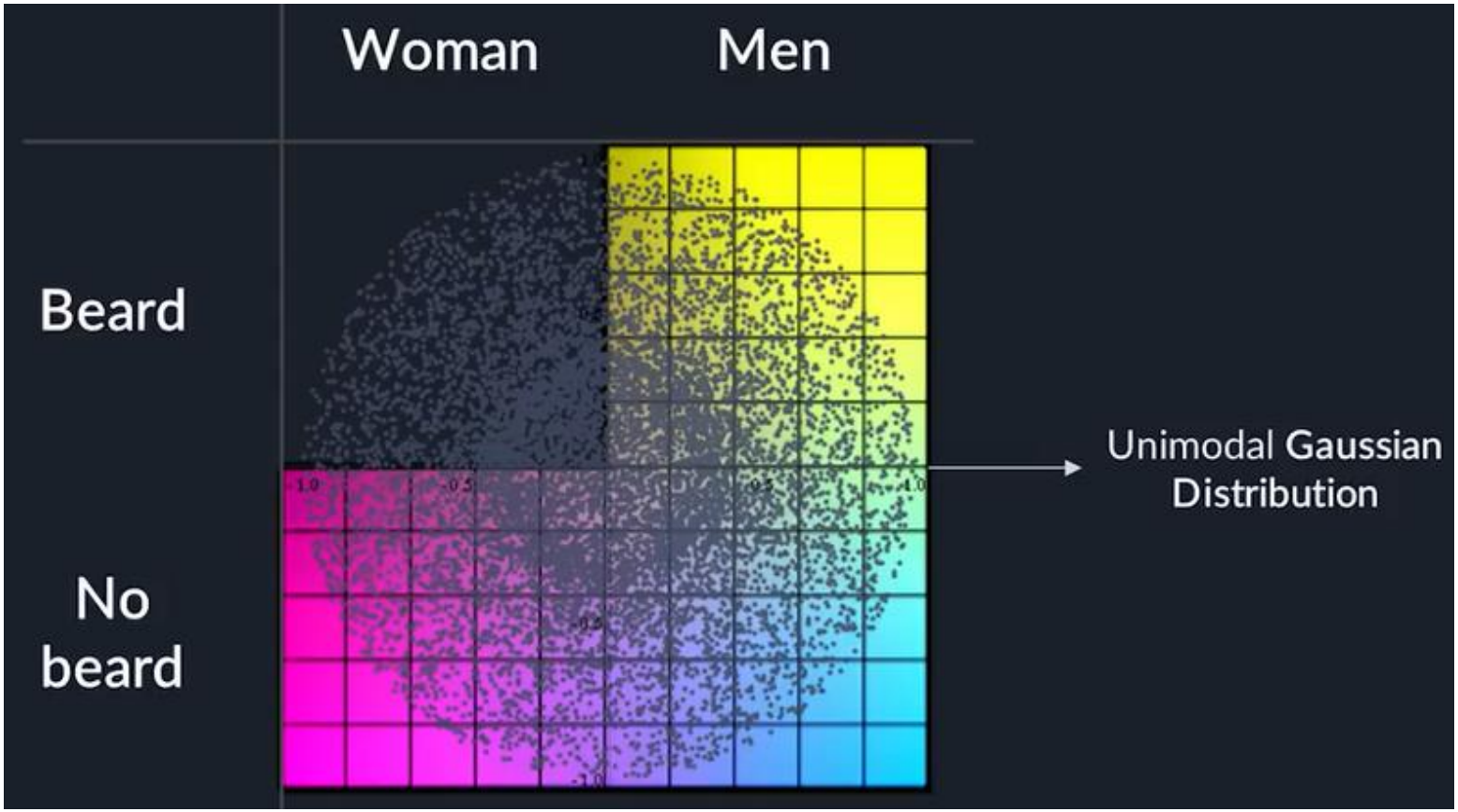
- Há 3 tipos de estilos: *course*, *middle* & *fine*.

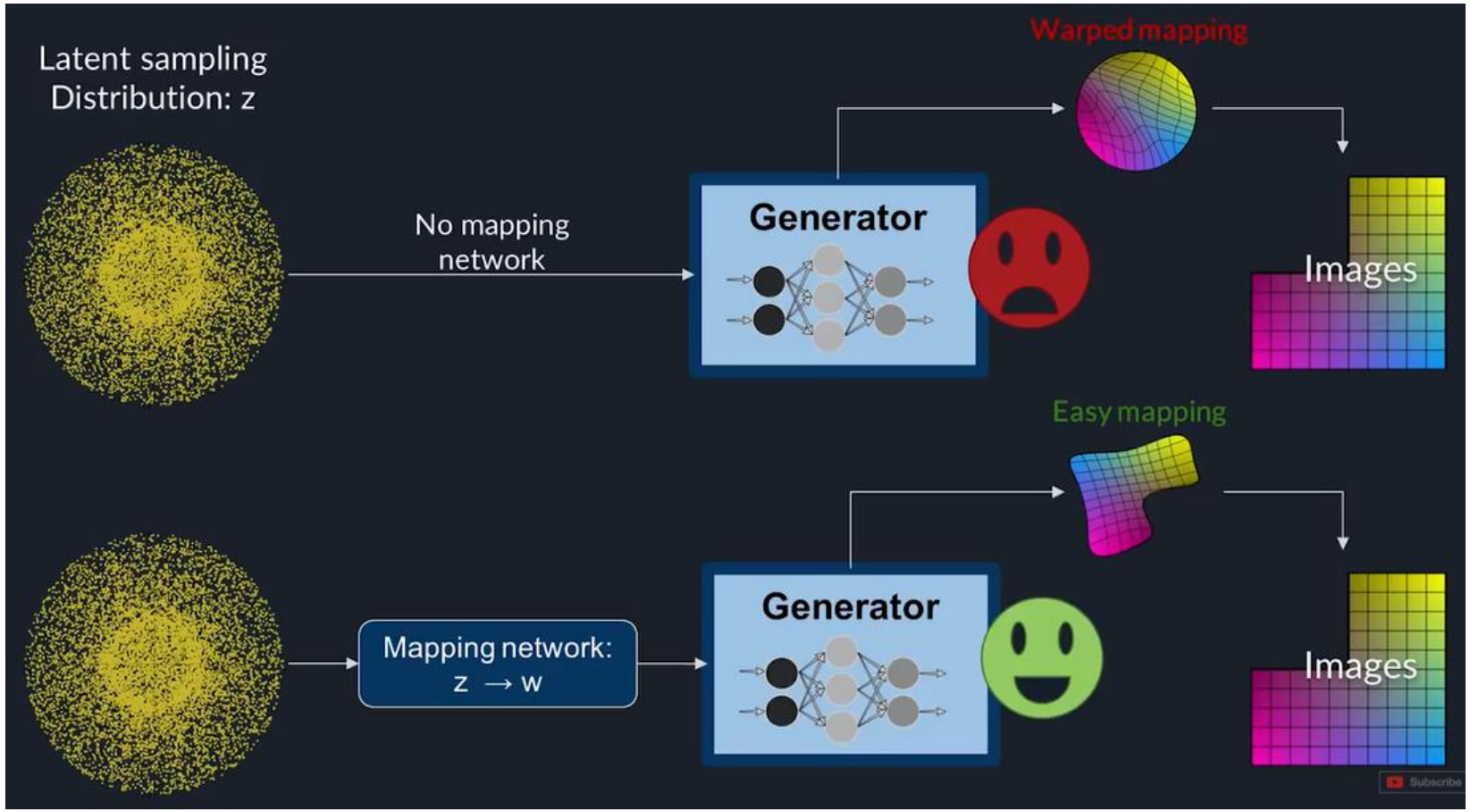


(a) Traditional (b) Style-based generator

- *Traditional*: 23,1 milhões; *Style-based*: 26,2 milhões de pesos sinápticos.

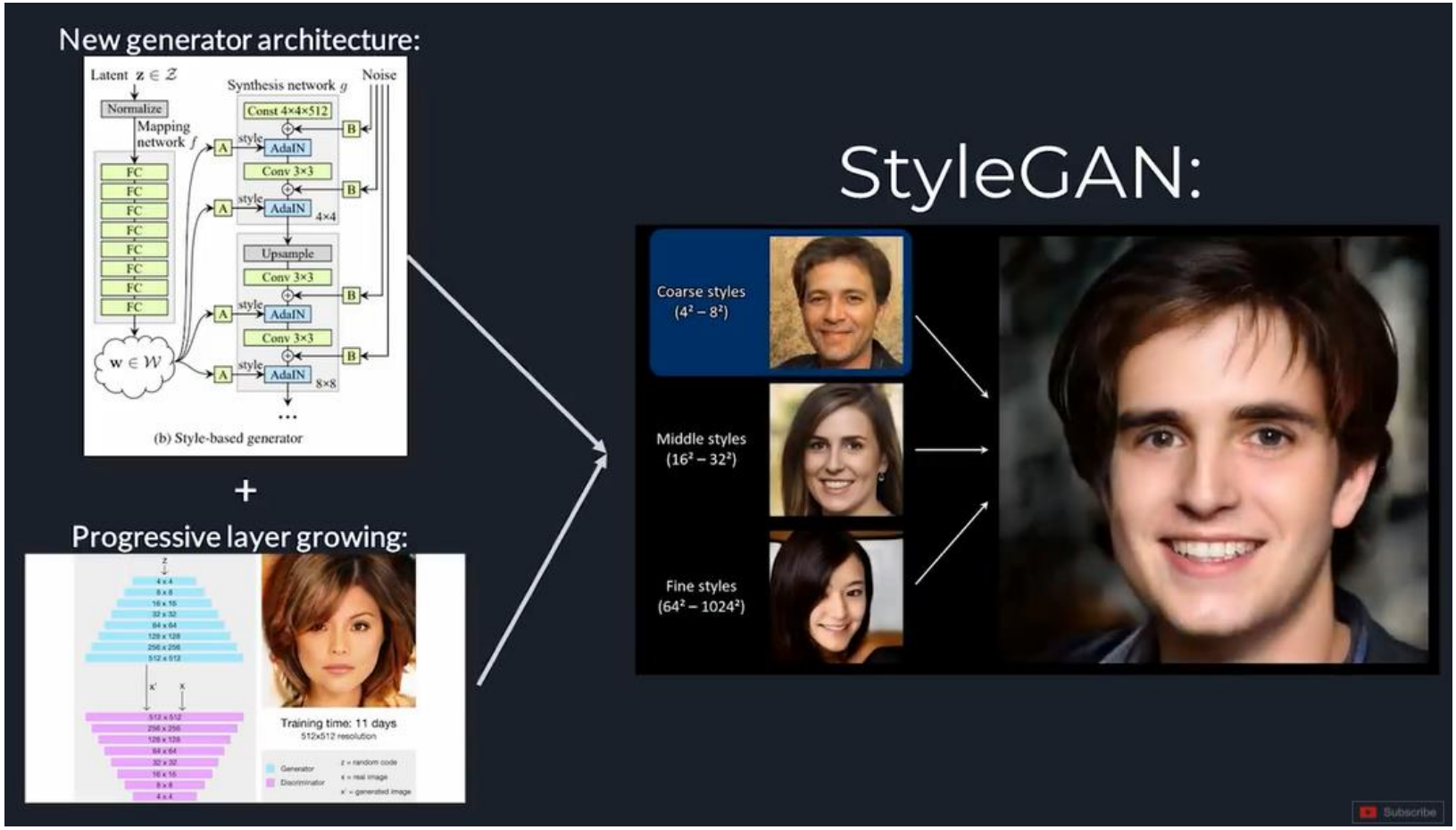
- Motivação para o mapeamento $\mathbf{z} \rightarrow \mathbf{w}$:





- Com o mapeamento $\mathbf{z} \rightarrow \mathbf{w}$, a tarefa da rede generativa tende a ficar mais simples, o que produz melhores resultados sintéticos.

- Cabe ressaltar que a proposta de GAN construtiva está presente em StyleGAN

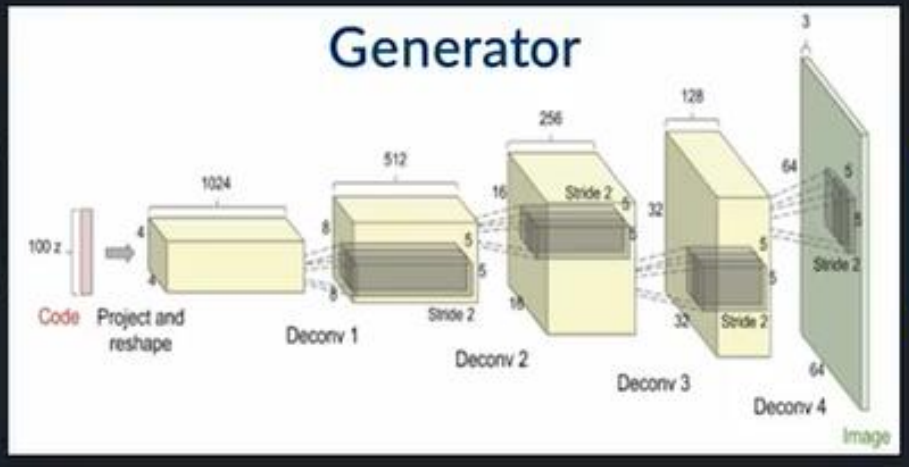


6 Como obter o código latente em StyleGAN

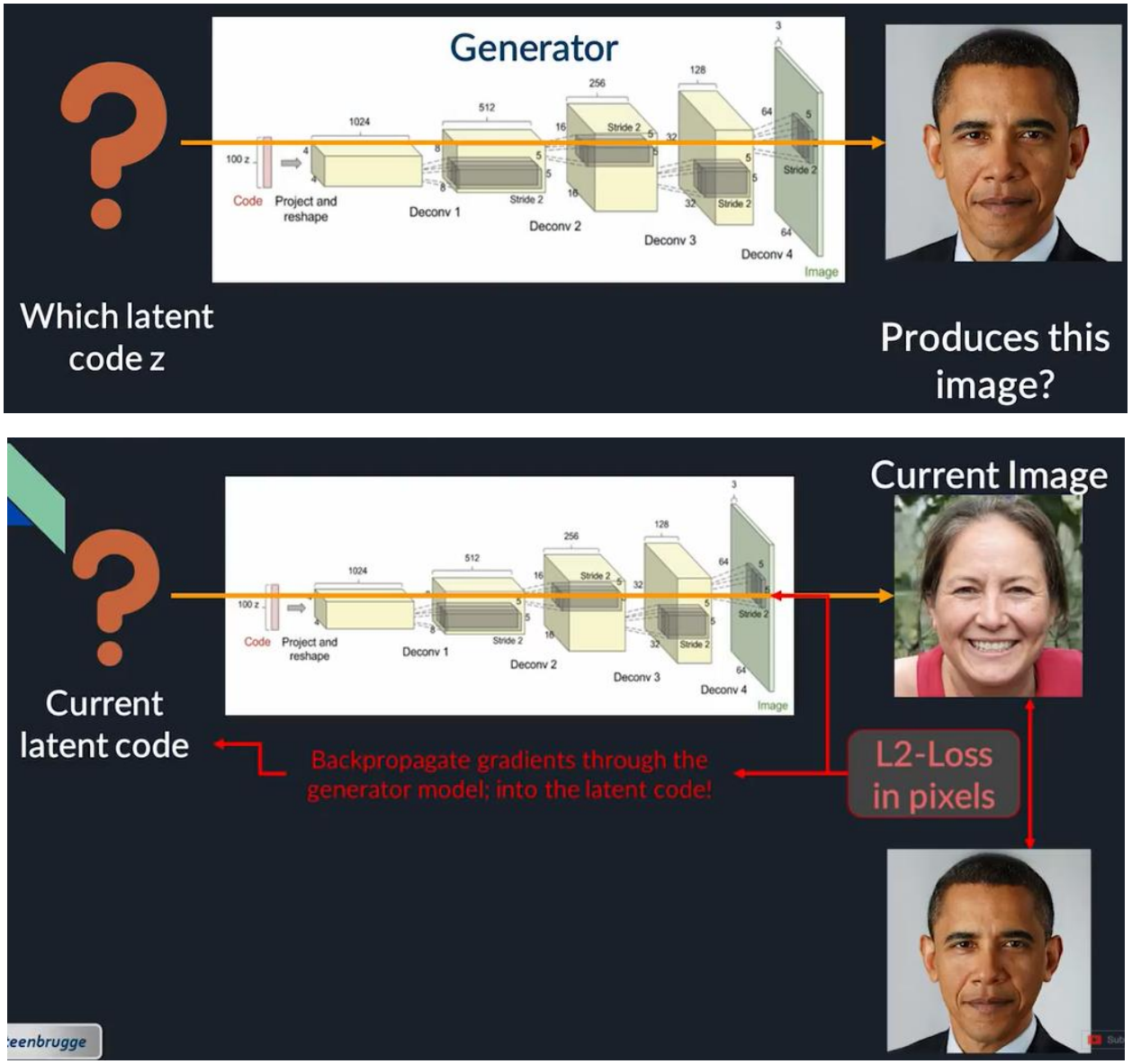
1. Instead of manipulating images in the pixel domain, let's manipulate them in the latent space!

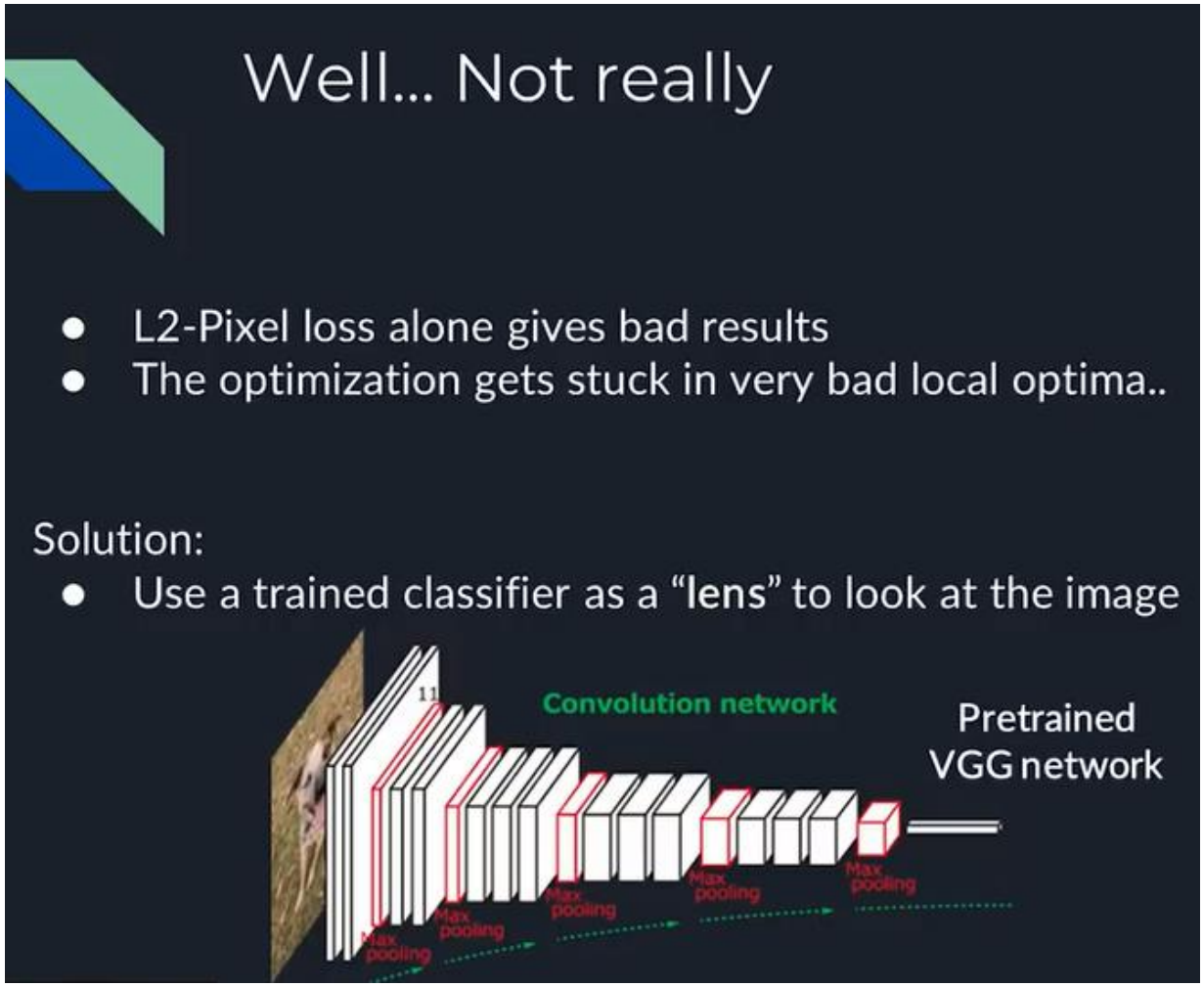
1. To do this, we first need to find a query image inside StyleGAN's latent space

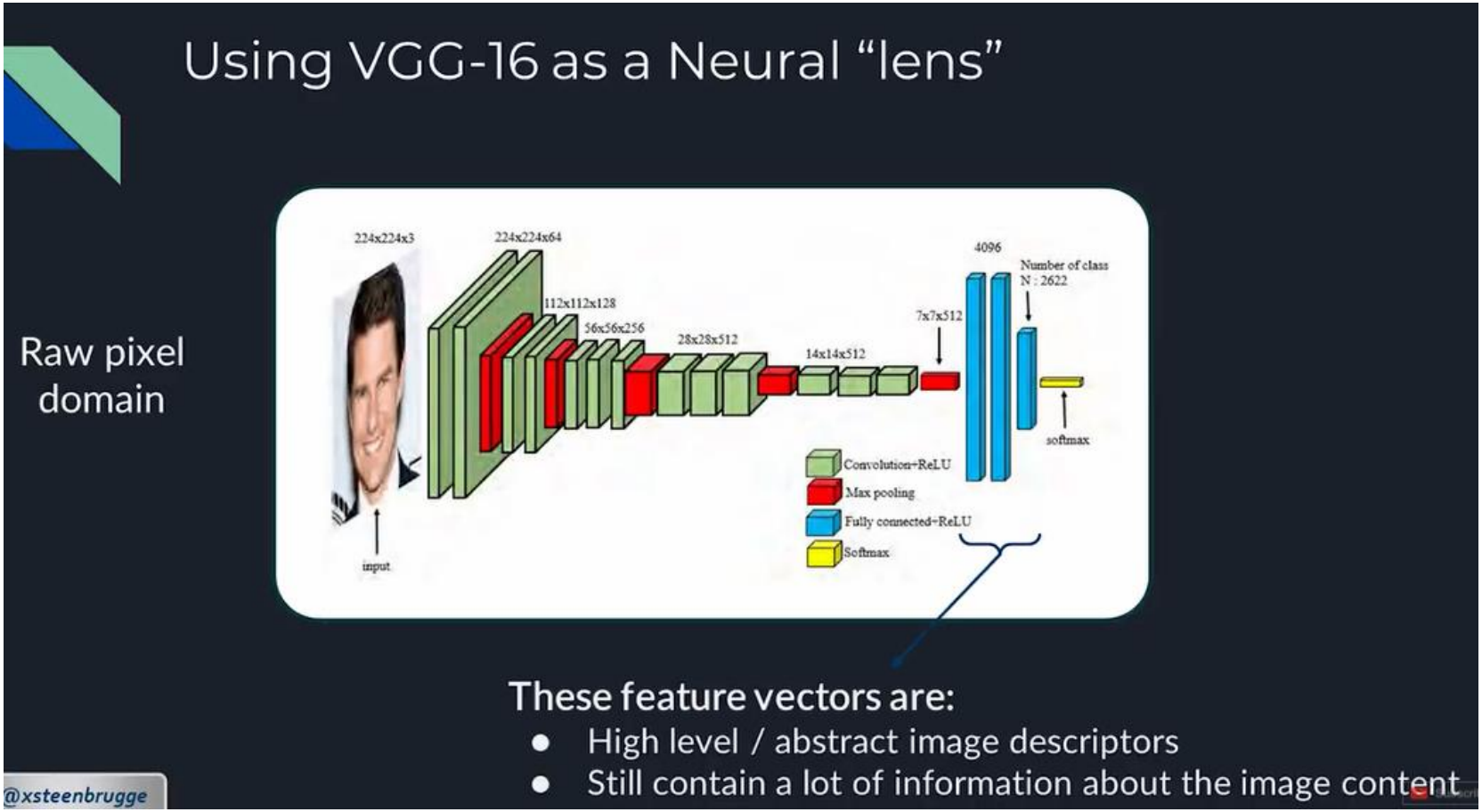
Structured latent space

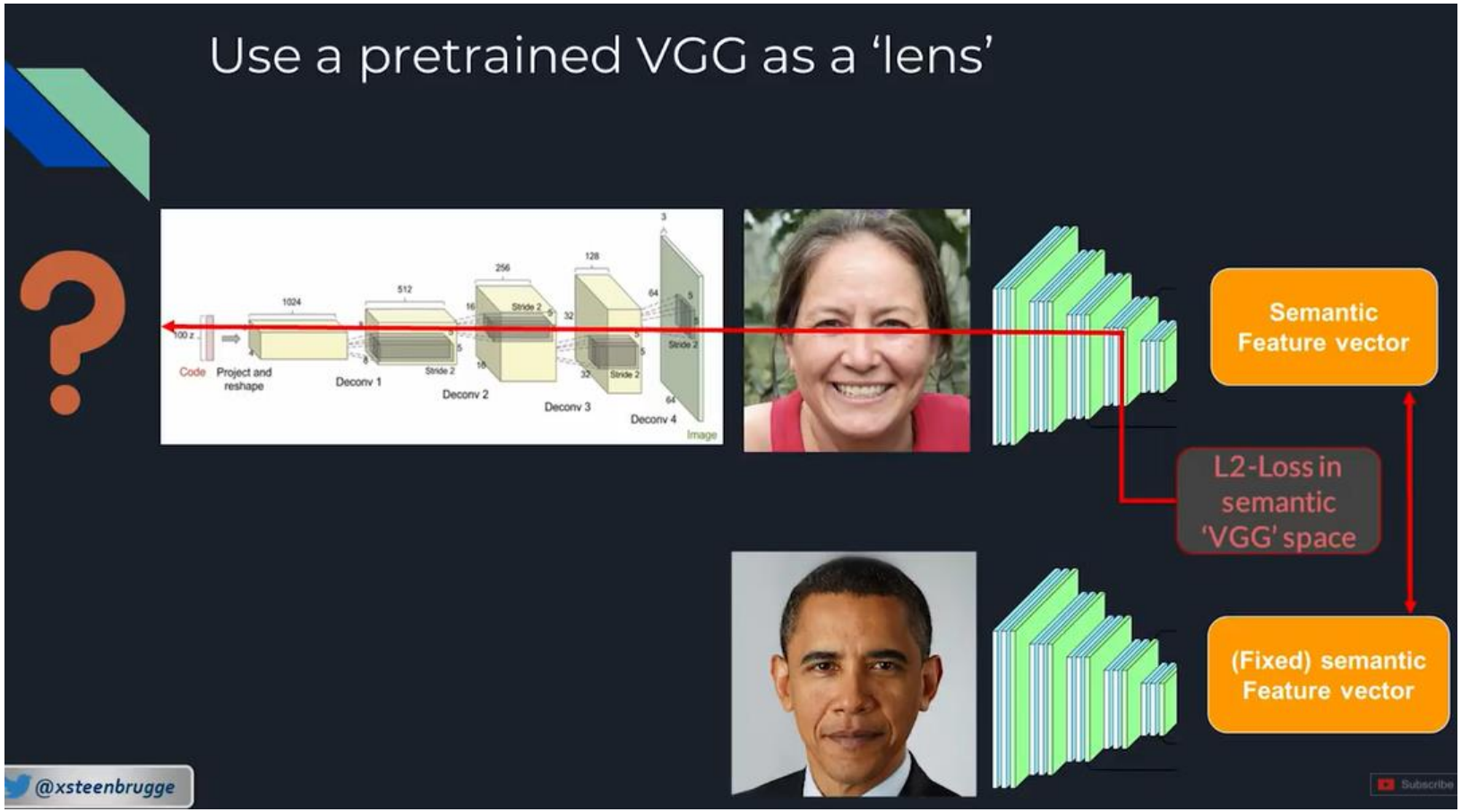


Big mess of pixels

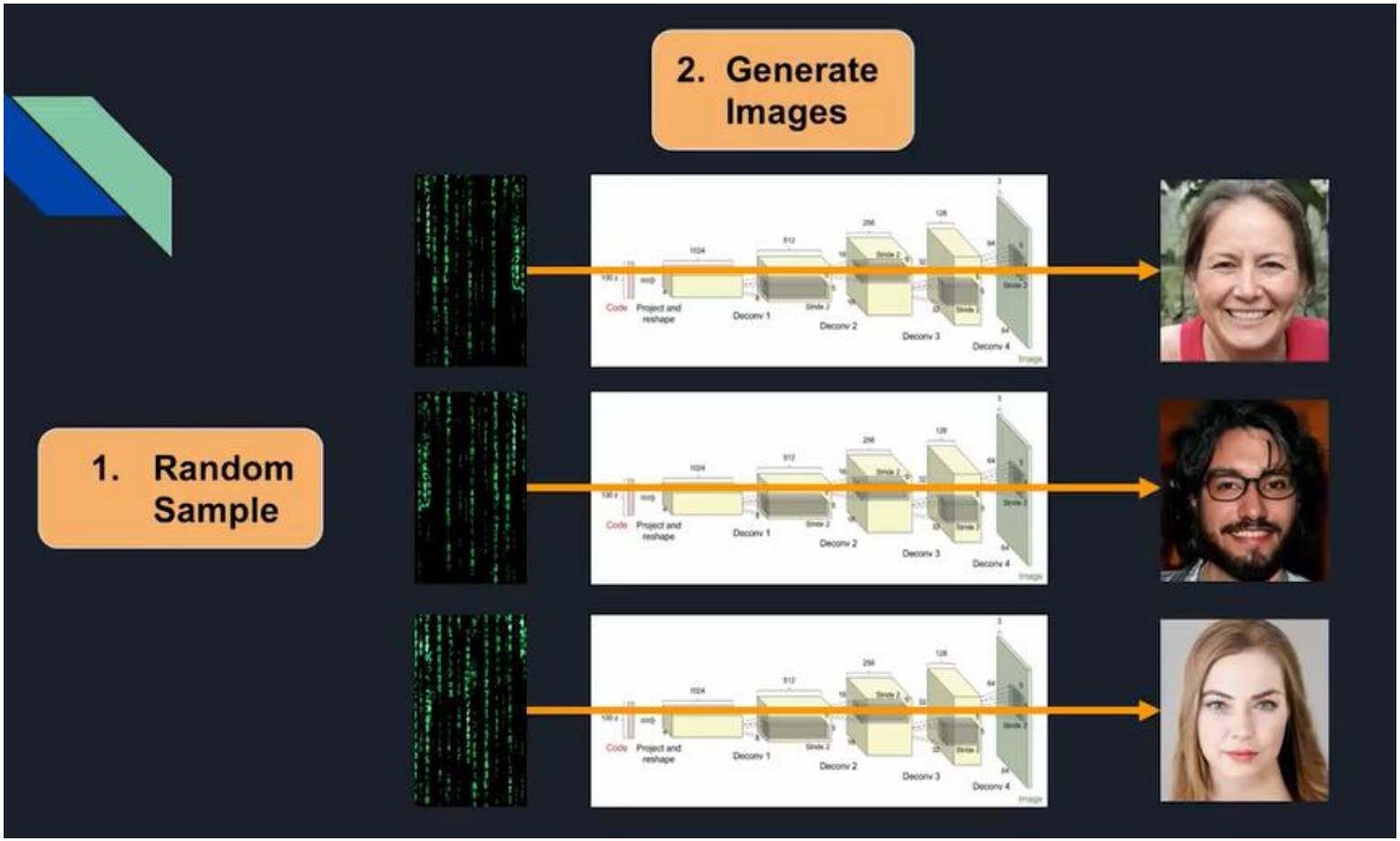

Generator
Which latent code z
Produces this image?
Current Image
Current latent code
Backpropagate gradients through the generator model; into the latent code!
L2-Loss in pixels
Code
Project and reshape
Deconv 1
Deconv 2
Deconv 3
Deconv 4
Stride 2
Image

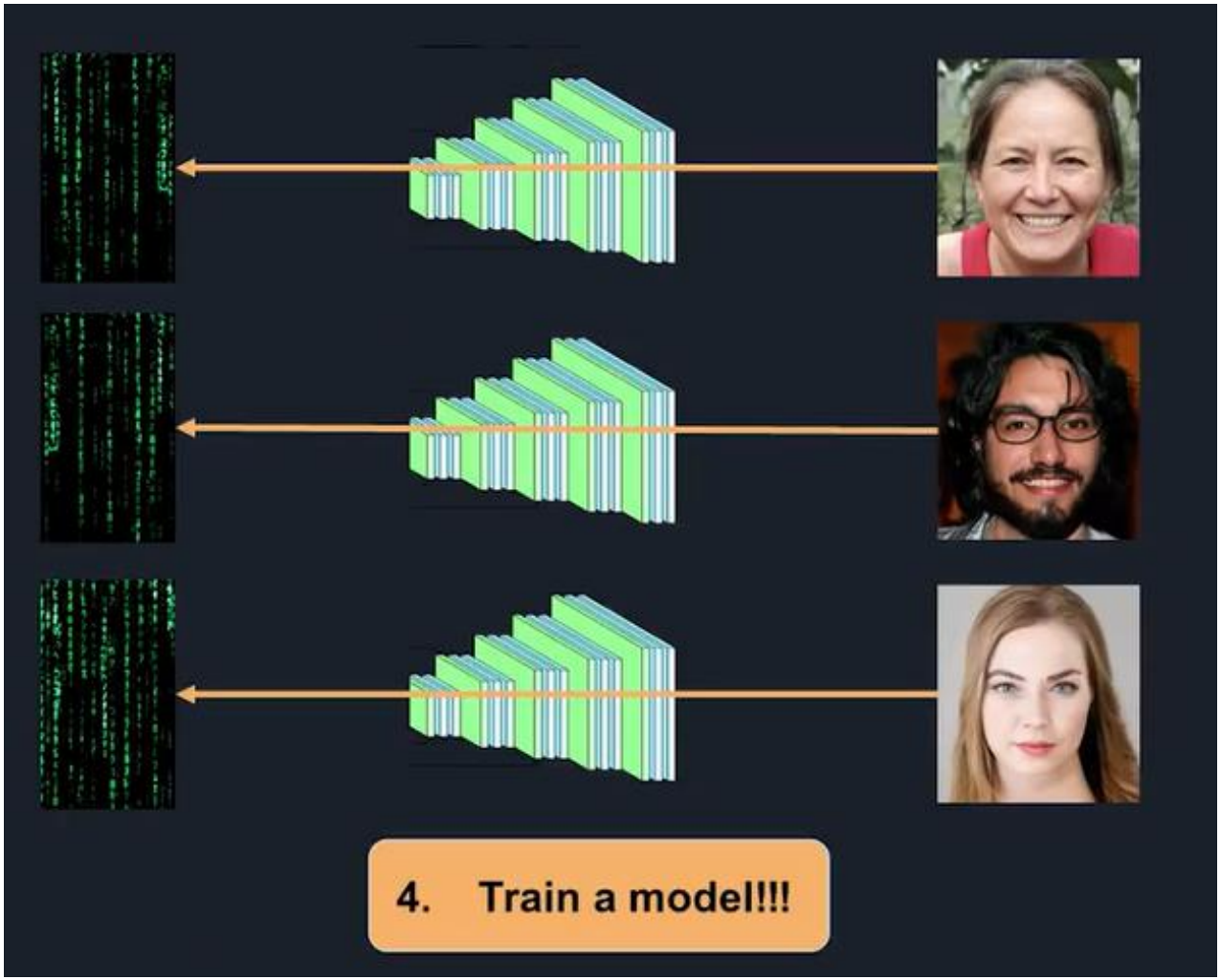

Well... Not really
L2-Pixel loss alone gives bad results
The optimization gets stuck in very bad local optima..
Solution:
Use a trained classifier as a "lens" to look at the image
Convolution network
Pretrained
VGG network
Max pooling
Max pooling
Max pooling
Max pooling
Max pooling

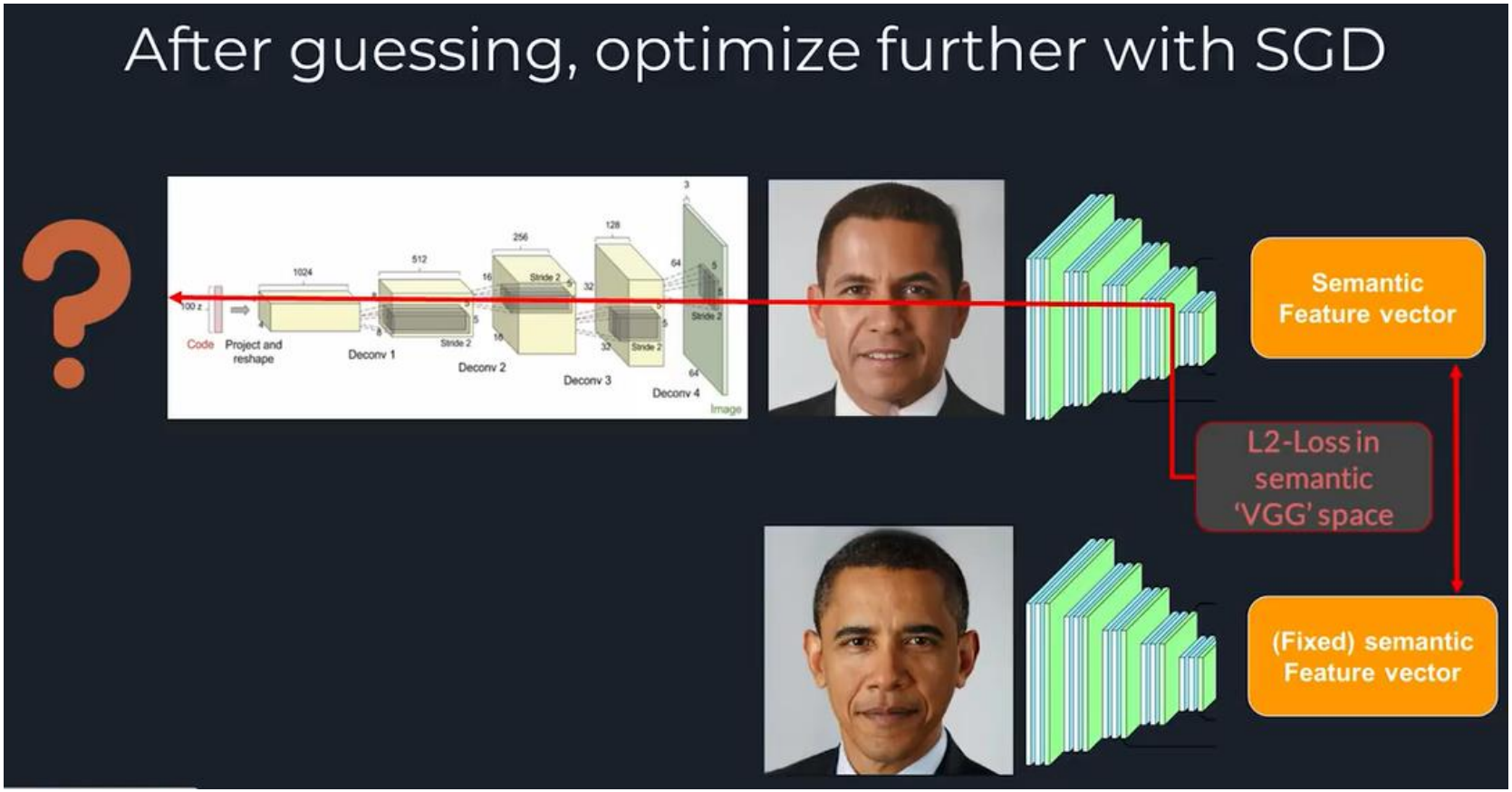

Using VGG-16 as a Neural "lens"
Raw pixel domain
224x224x3
224x224x64
112x112x128
56x56x256
28x28x512
14x14x512
7x7x512
4096
Number of class
N : 2622
softmax
input
Convolution+ReLU
Max pooling
Fully connected+ReLU
Softmax
These feature vectors are:
High level / abstract image descriptors
Still contain a lot of information about the image content
@xsteenbrugge

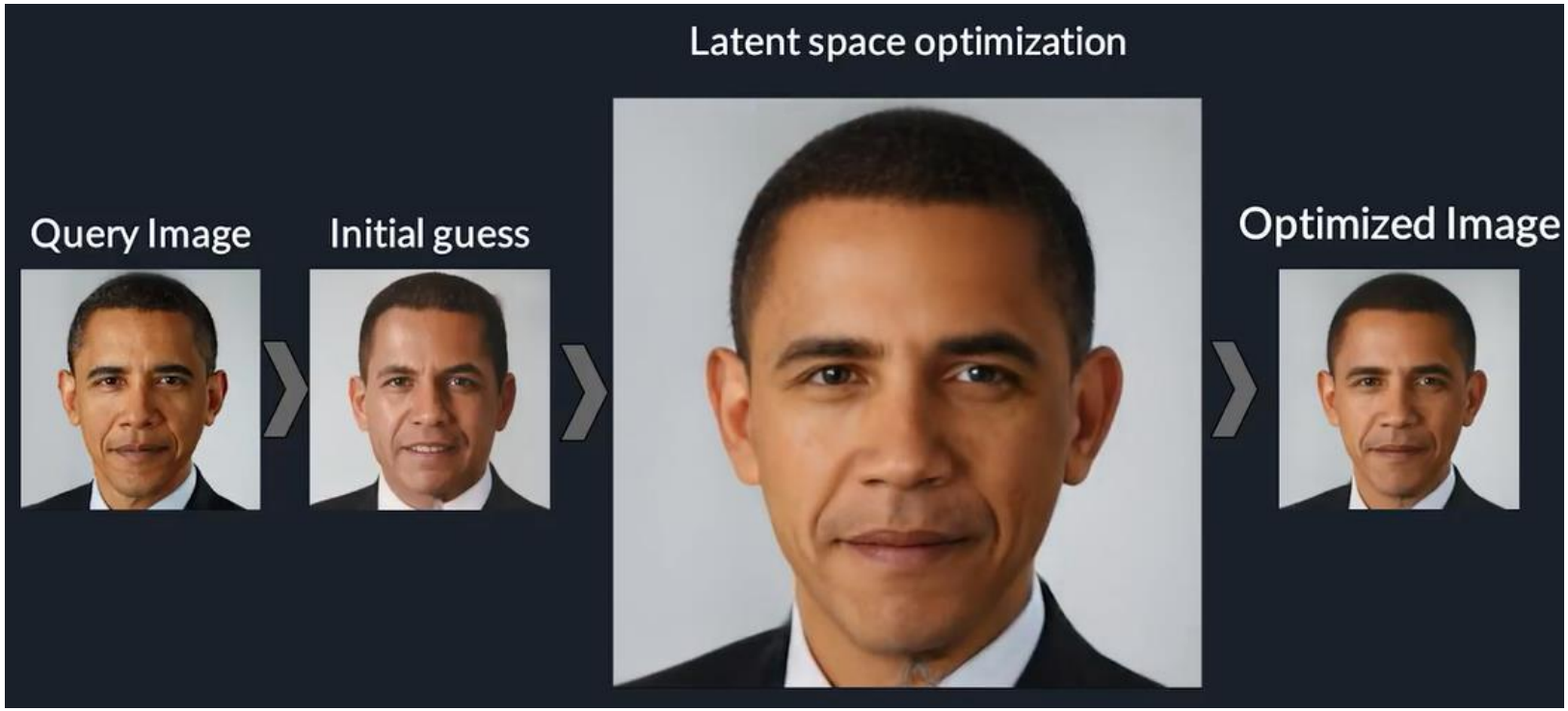


- Mesmo sendo uma ideia que funciona, esse processo é muito lento e se faz necessário mais um passo técnico para se chegar a um bom código latente.


2. Generate Images
1. Random Sample
Code
Project and reshape
Deconv 1
Deconv 2
Deconv 3
Deconv 4

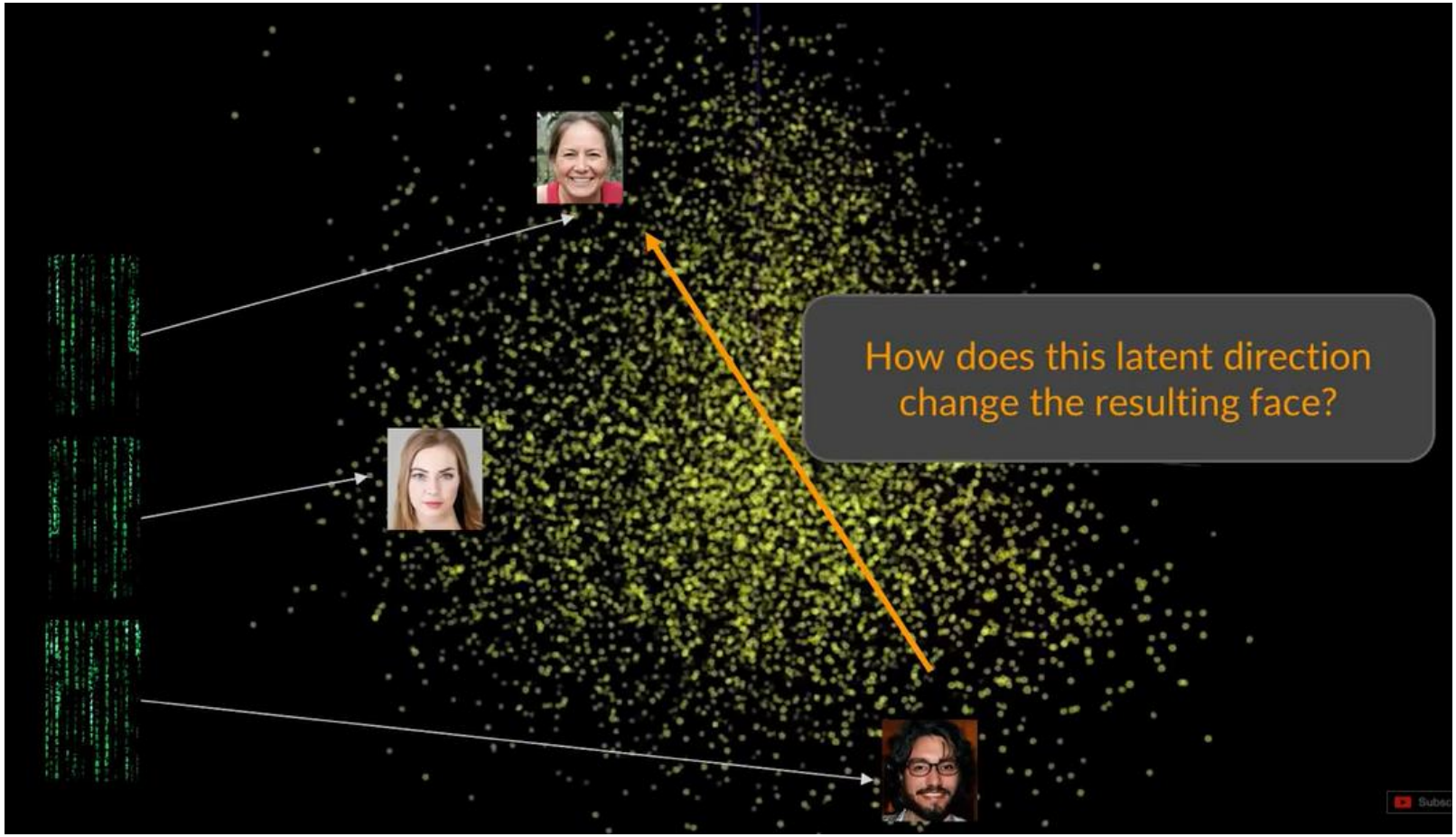

4. Train a model!!!

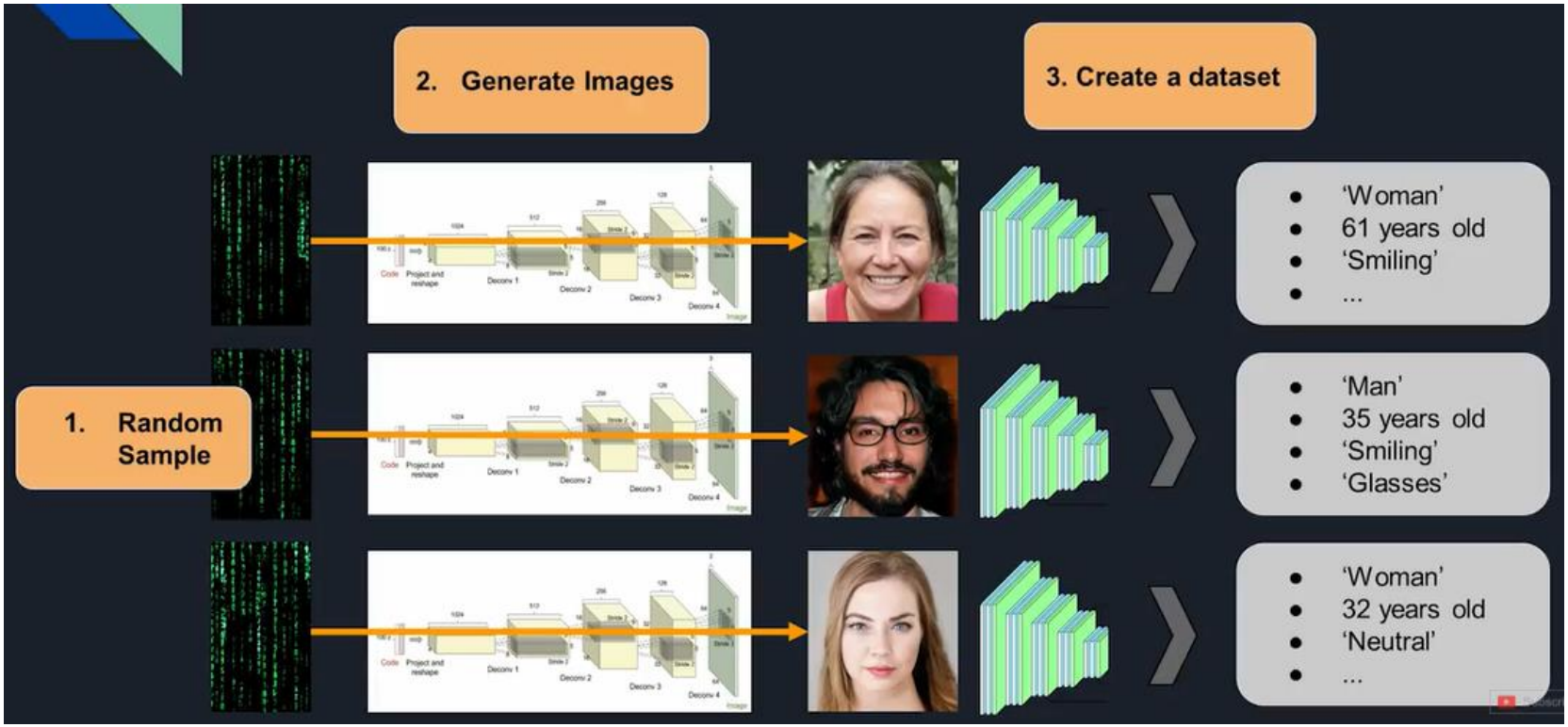

After guessing, optimize further with SGD
Code
Project and reshape
Deconv 1
Deconv 2
Deconv 3
Deconv 4
Image
Semantic Feature vector
L2-Loss in semantic 'VGG' space
(Fixed) semantic Feature vector

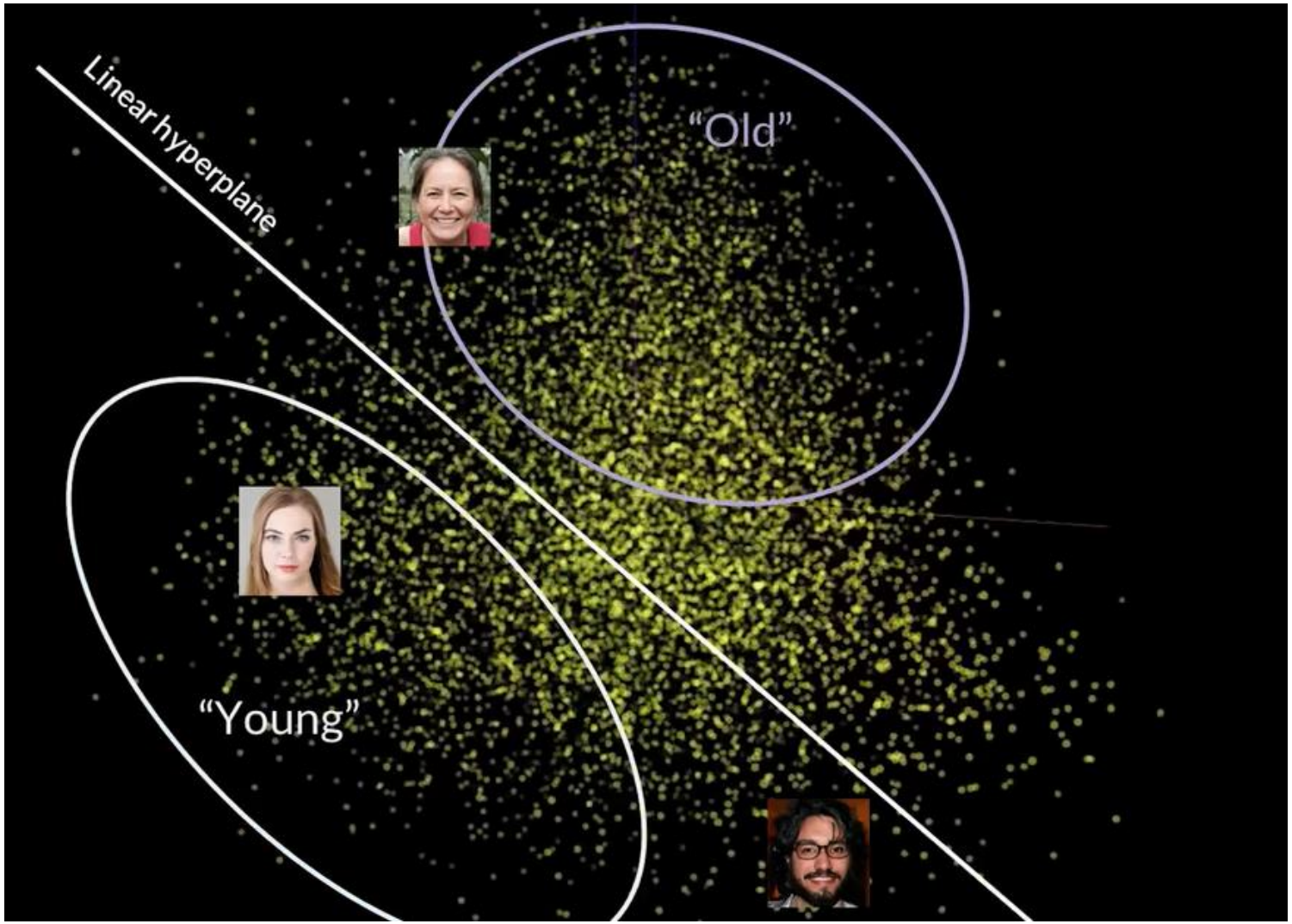

Latent space optimization
Query Image
Initial guess
Optimized Image

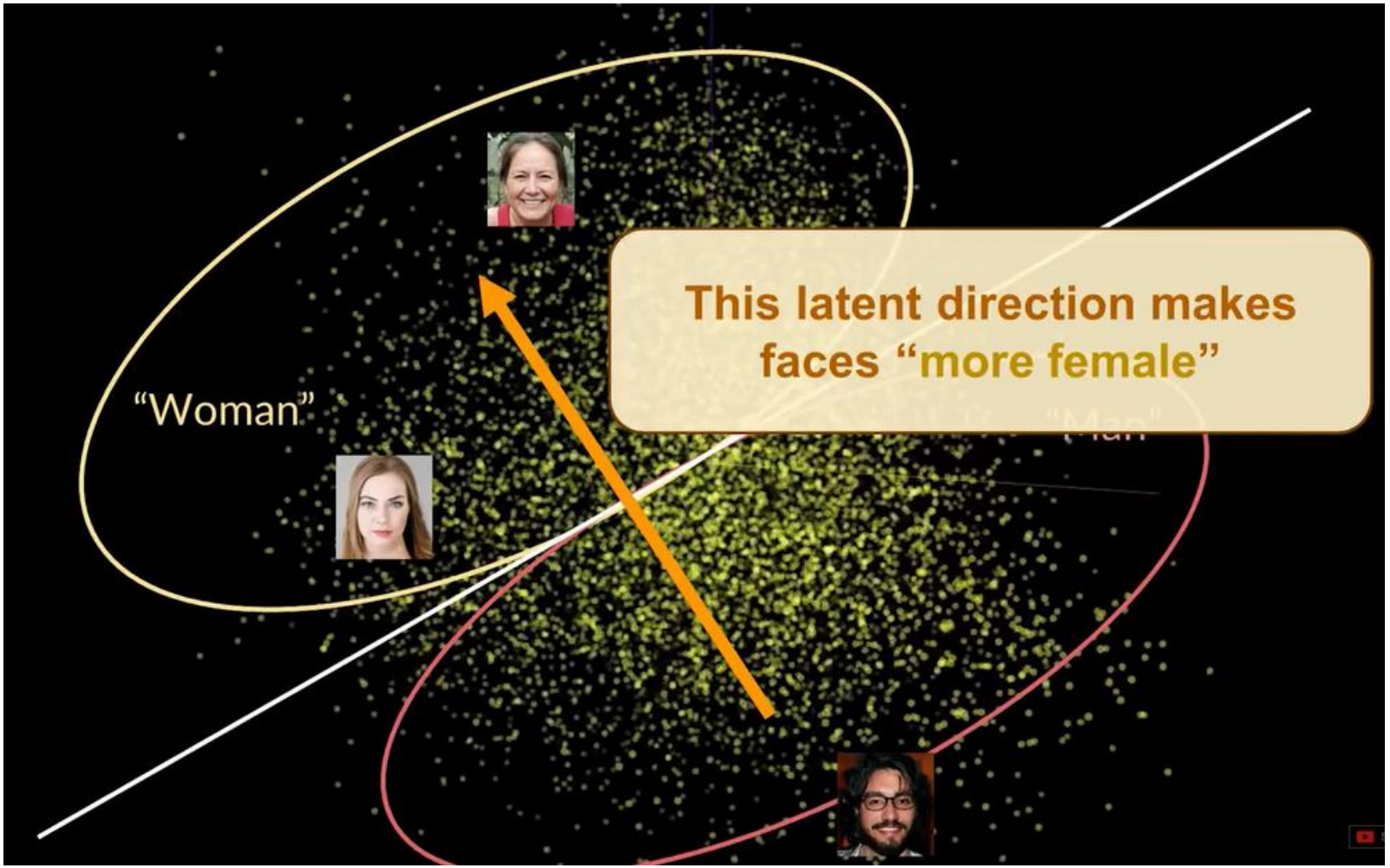
7 Desembaraçando o código latente

- O código latente tem dimensão 512. Será mostrado aqui numa projeção 2D.



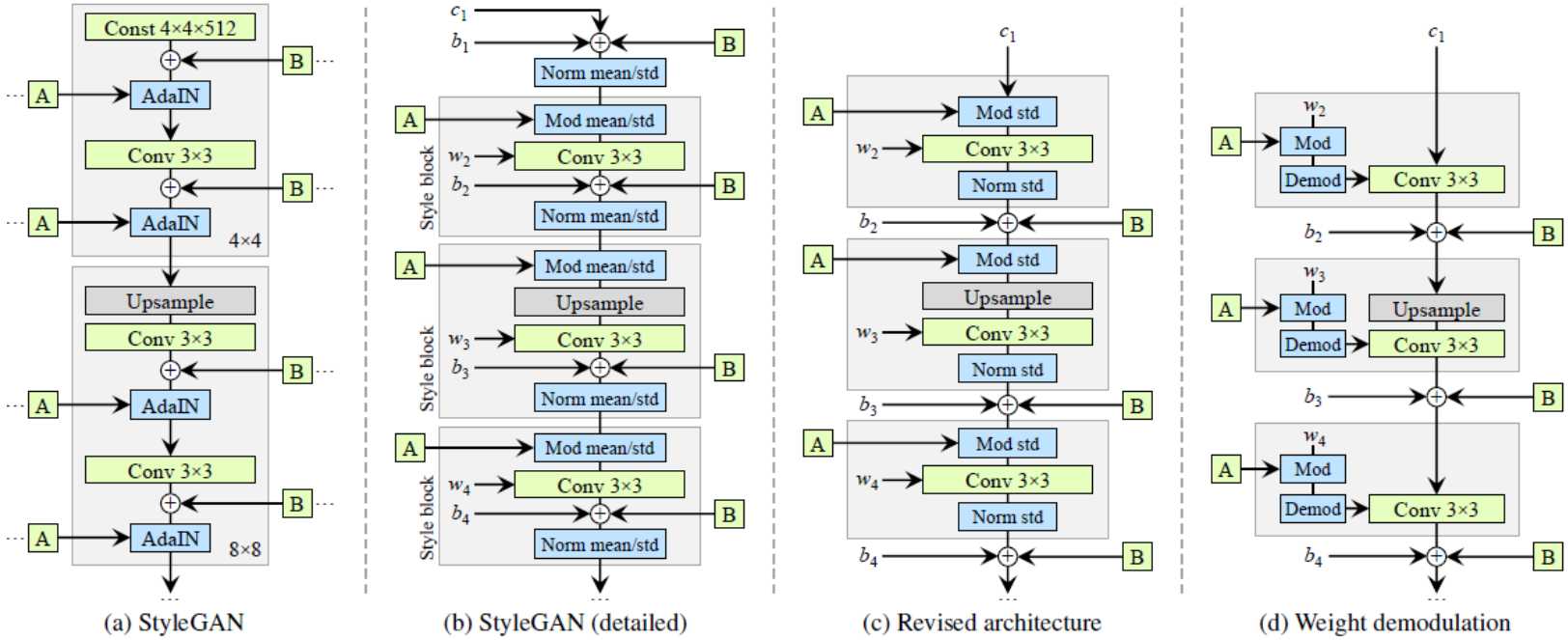

2. Generate Images
3. Create a dataset
1. Random Sample
'Woman'
61 years old
'Smiling'
...
'Man'
35 years old
'Smiling'
'Glasses'
'Woman'
32 years old
'Neutral'
...


Linear hyperplane
"Old"
"Young"


This latent direction makes faces "more female"
"Woman"

8 StyleGAN 2

- Incorporação de melhorias em vários aspectos do projeto, visando ganho de desempenho (KARRAS et al., 2020).



(a) StyleGAN (b) StyleGAN (detailed) (c) Revised architecture (d) Weight demodulation

- Rede Generativa: de 26 para 30 milhões; Rede Discriminativa: de 24 para 29 milhões de pesos sinápticos.

- Iniciativas voltadas para a detecção de imagens sintéticas, usando projeção para o espaço latente e re-síntese da imagem.

StyleGAN2 — generated images

StyleGAN2 — real images

9 Detecção de *DeepFakes*

DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection

Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales and Javier Ortega-Garcia
Biometrics and Data Pattern Analytics - BiDA Lab, Universidad Autonoma de Madrid, Spain
{ruben.tolosana, ruben.vera, julian.fierrez, aythami.morales, javier.ortega}@uam.es

***Abstract*—The free access to large-scale public databases, together with the fast progress of deep learning techniques, in particular Generative Adversarial Networks, have led to the generation of very realistic fake content with its corresponding implications towards society in this era of fake news.**

This survey provides a thorough review of techniques for manipulating face images including DeepFake methods, and methods to detect such manipulations. In particular, four types of facial manipulation are reviewed: *i)* entire face synthesis, *ii)* identity swap (DeepFakes), *iii)* attribute manipulation, and *iv)* expression swap. For each manipulation group, we provide details regarding manipulation techniques, existing public databases, and key benchmarks for technology evaluation of fake detection methods, including a summary of results from those evaluations. Among all the aspects discussed in the survey, we pay special

(MFC2018)[1] and the Deepfake Detection Challenge (DFDC)[2] launched by the National Institute of Standards and Technology (NIST) and Facebook, respectively.

Traditionally, the number and realism of facial manipulations have been limited by the lack of sophisticated editing tools, the domain expertise required, and the complex and time-consuming process involved. For example, an early work in this topic [23] was able to modify the lip motion of a person speaking using a different audio track, by making connections between the sounds of the audio track and the shape of the subject's face. However, from these early works up to date, many things have rapidly evolved in the last years. Nowadays,

The Creation and Detection of Deepfakes: A Survey

YISROEL MIRSKY, Georgia Institute of Technology and Ben-Gurion University
WENKE LEE, Georgia Institute of Technology

Generative deep learning algorithms have progressed to a point where it is difficult to tell the difference between what is real and what is fake. In 2018, it was discovered how easy it is to use this technology for unethical and malicious applications, such as the spread of misinformation, impersonation of political leaders, and the defamation of innocent individuals. Since then, these 'deepfakes' have advanced significantly.

In this paper, we explore the creation and detection of deepfakes and provide an in-depth view of how these architectures work. The purpose of this survey is to provide the reader with a deeper understanding of (1) how deepfakes are created and detected, (2) the current trends and advancements in this domain, (3) the shortcomings of the current defense solutions, and (4) the areas which require further research and attention.

CCS Concepts: • **Security and privacy → Social engineering attacks**; *Human and societal aspects of security and privacy*; • **Computing methodologies → Machine learning**.

Additional Key Words and Phrases: Deepfakes, Deep fakes, reenactment, replacement, face swap, generative AI, social engineering, impersonation

ACM Reference Format:
Yisroel Mirsky and Wenke Lee. 2020. The Creation and Detection of Deepfakes: A Survey. 1, 1 (May 2020), 36 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

10 Referências bibliográficas

GOODFELLOW, I.J.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; BENGIO, Y. "Generative Adversarial Networks", arXiv:1406.2661v1, June 2014.

GUI, J.; SUN, Z.; WEN, Y.; TAO, D.; YE, J. "A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications", arXiv:2001.06937v1, January 2020.

KARRAS, T.; AILA, T.; LAINE, S.; LEHTINEN, J. "Progressive Growing of GANs for Improved Quality, Stability, and Variation", arXiv:1710.10196v3, February 2018.

KARRAS, T.; LAINE, S.; AILA, T. "A Style-Based Generator Architecture for Generative Adversarial Networks", arXiv:1812.04948v3, March 2019.

KARRAS, T.; LAINE, S.; AITTALA, M.; HELLSTEN, J.; LEHTINEN, J.; AILA, T. "Analyzing and Improving the Image Quality of StyleGAN", arXiv:1912.04958v2, March 2020.

MIRSKY, Y.; LEE, W. "The Creation and Detection of Deepfakes: A Survey", arXiv:2004.11138v2, May 2020.

RADFORD, A.; METZ, L.; CHINTALA, S. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", arXiv:1511.06434v2, January 2016.

TOLOSANA, R.; VERA-RODRIGUEZ, R.; FIERREZ, J.; MORALES, A.; ORTEGA-GARCIA, J. "DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection", arXiv:2001.00179v2, May 2020.