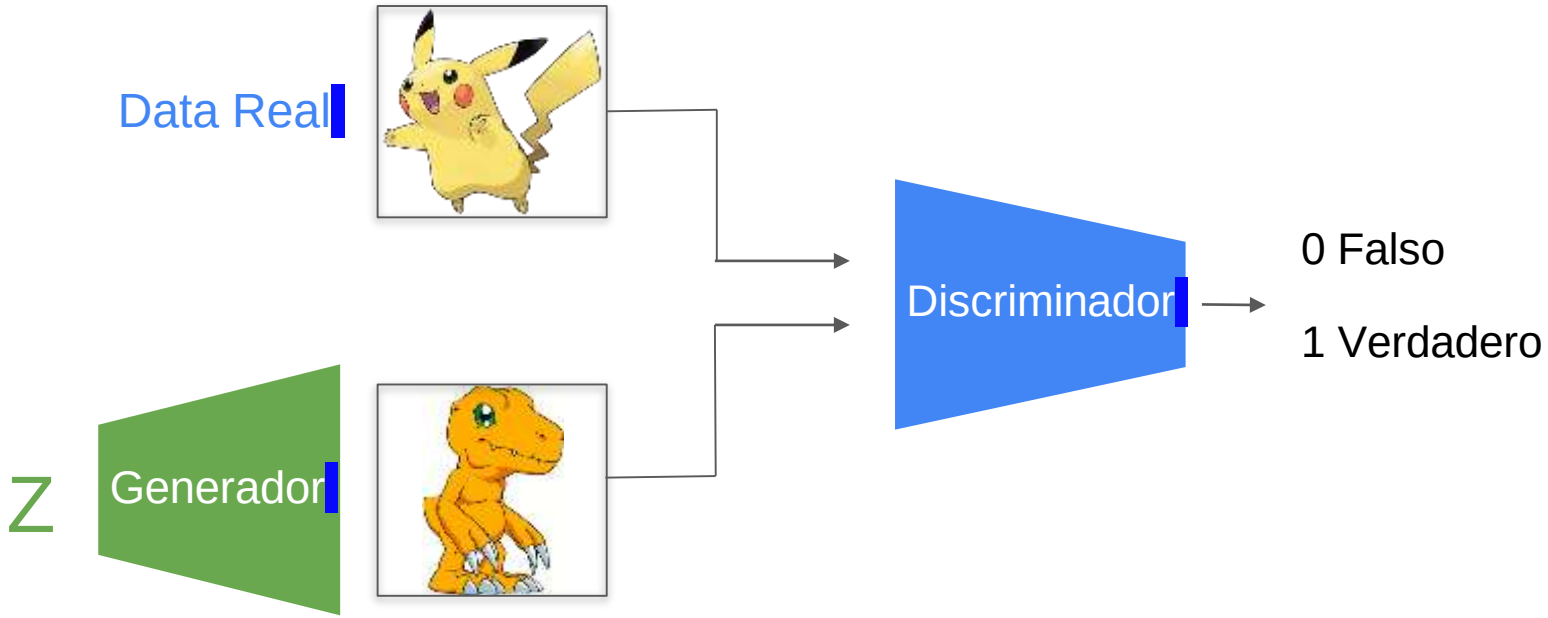
Deep Learning en Imágenes con GANs y Modelos de Difusión - Parte II

Prof. Peter Montalvo

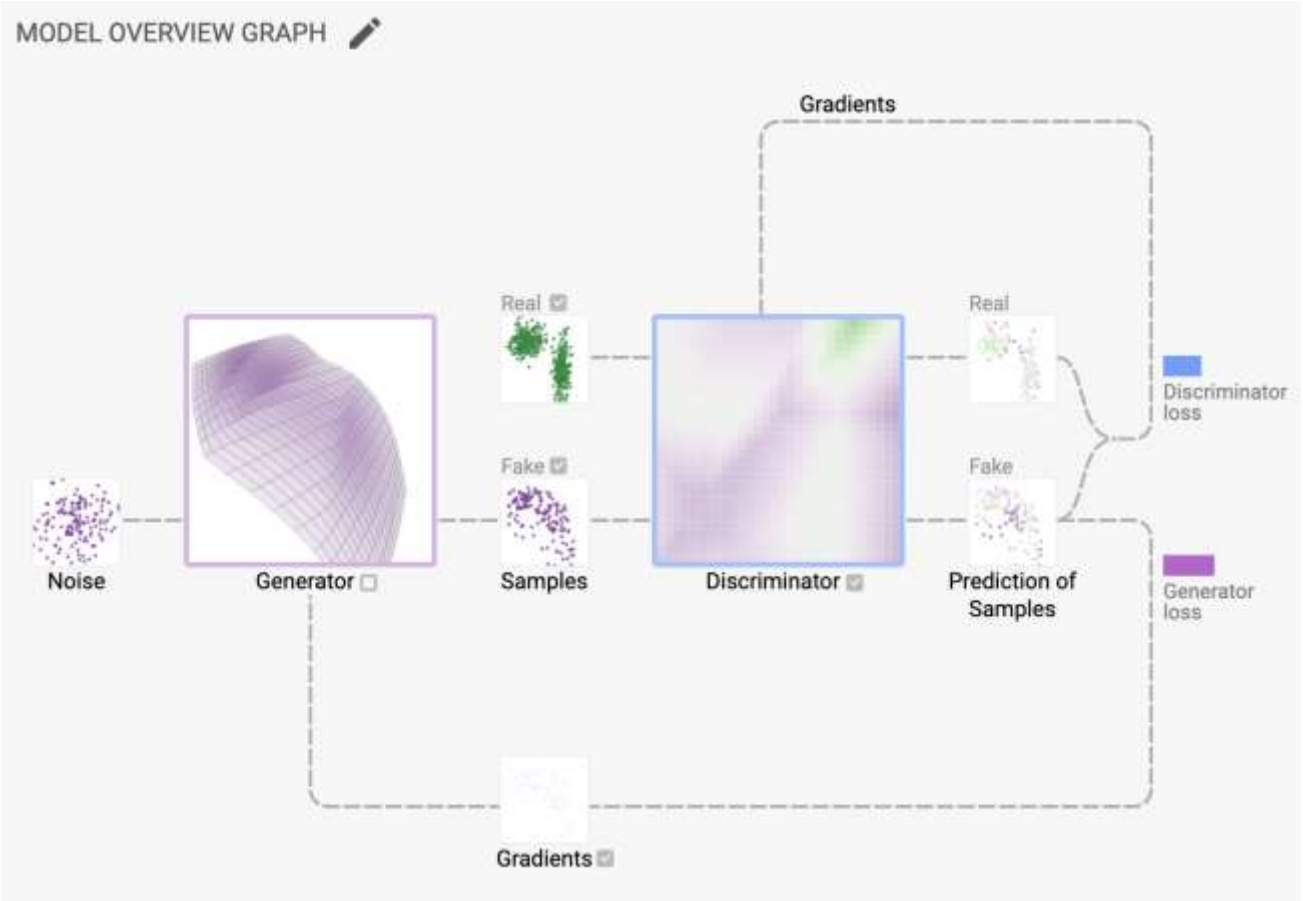
Agenda

- Introducción
- GANs
- Función de Pérdida
- Aplicaciones
- Wasserstein GAN

GAN - Generative Adversarial Nets

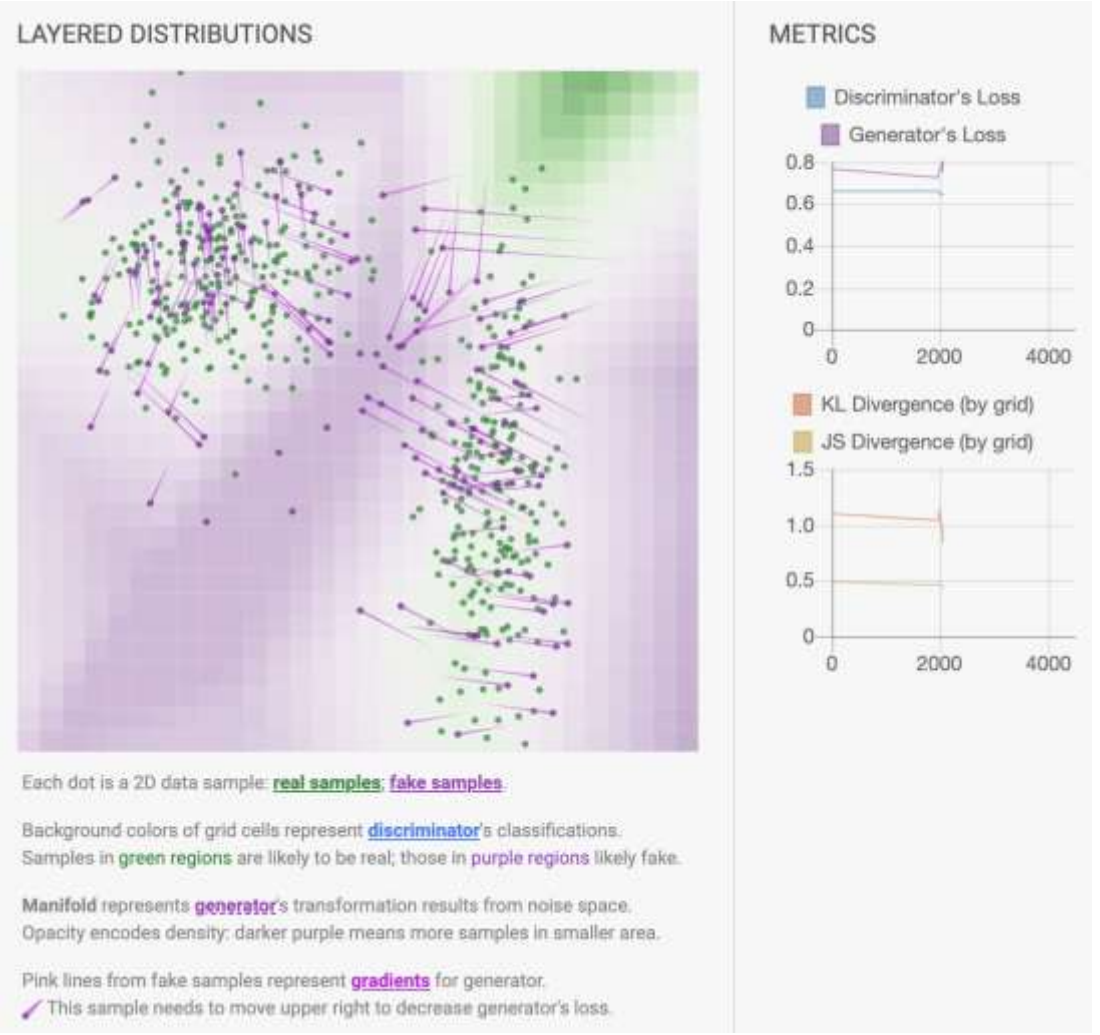


Introducción



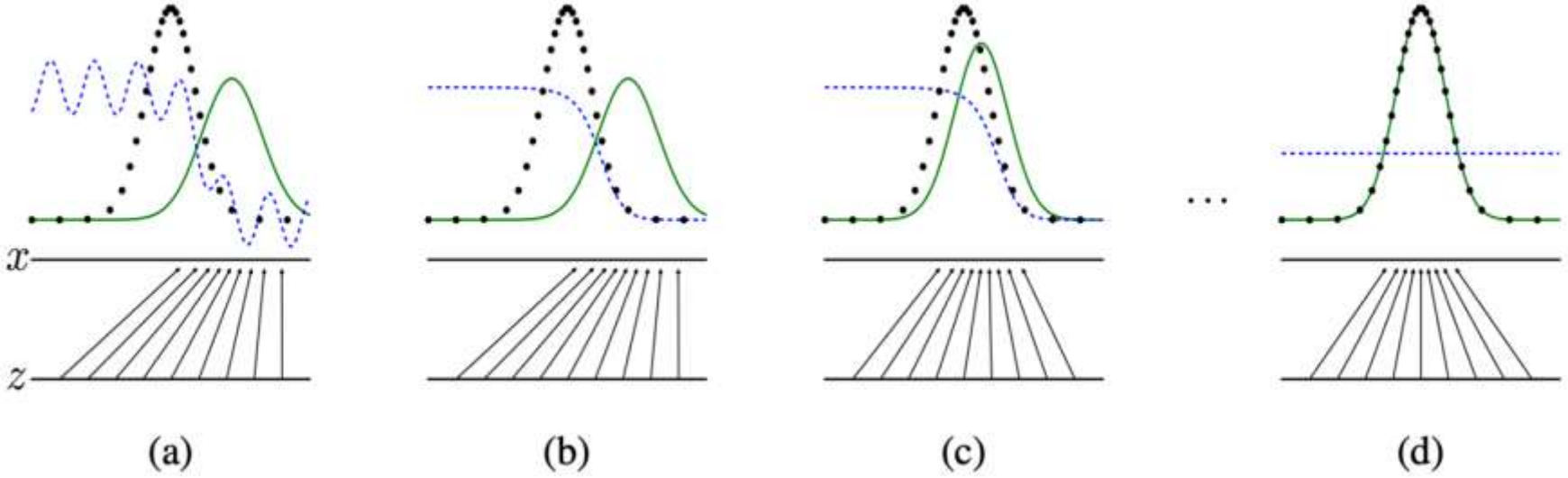
https://poloclub.github.io/ganlab/

Introducción

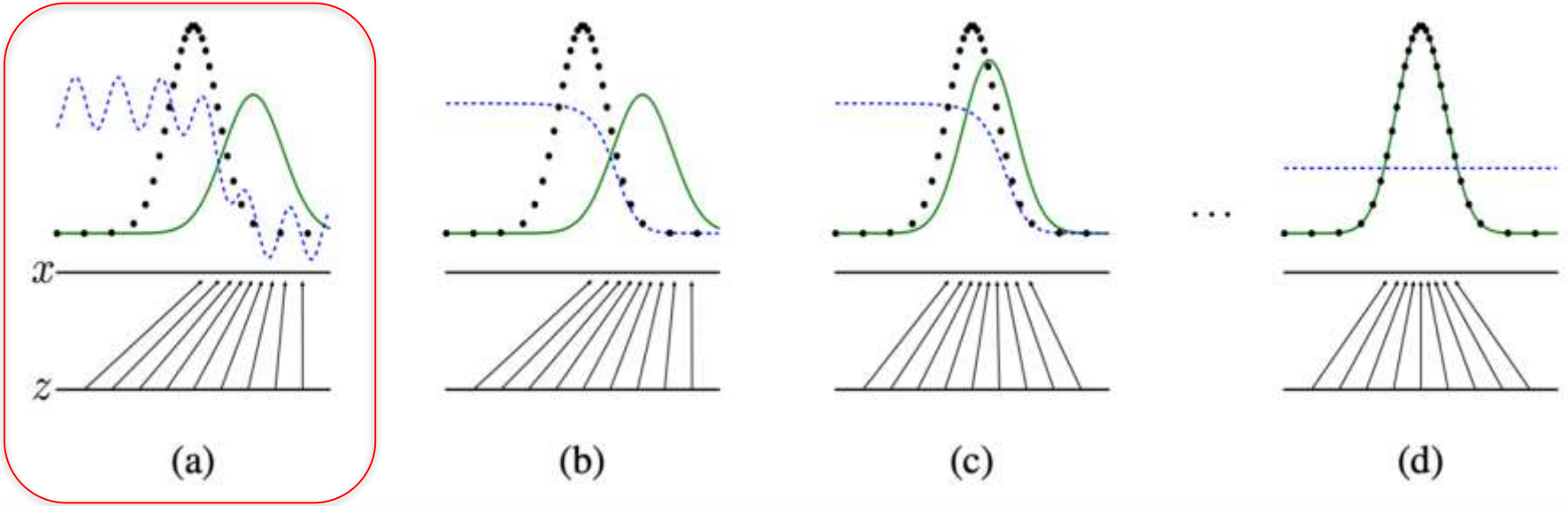


https://poloclub.github.io/ganlab/

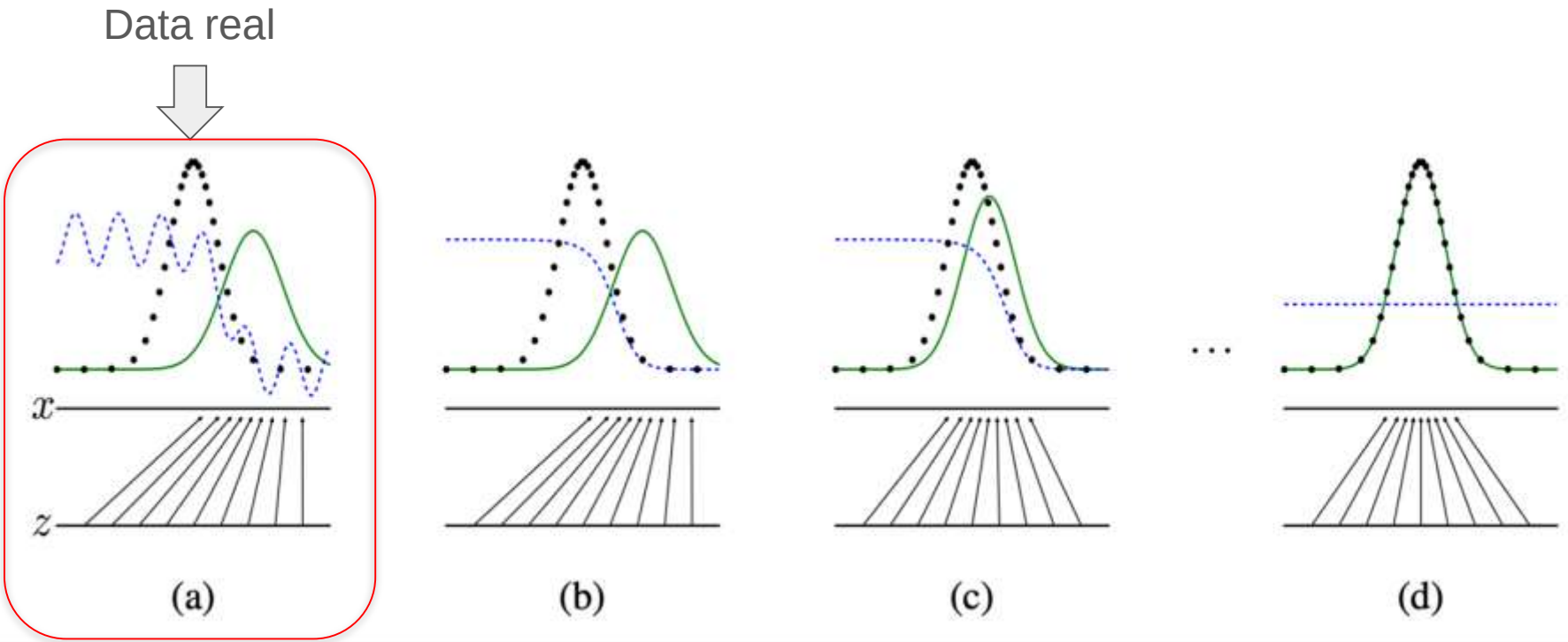
Función de pérdida



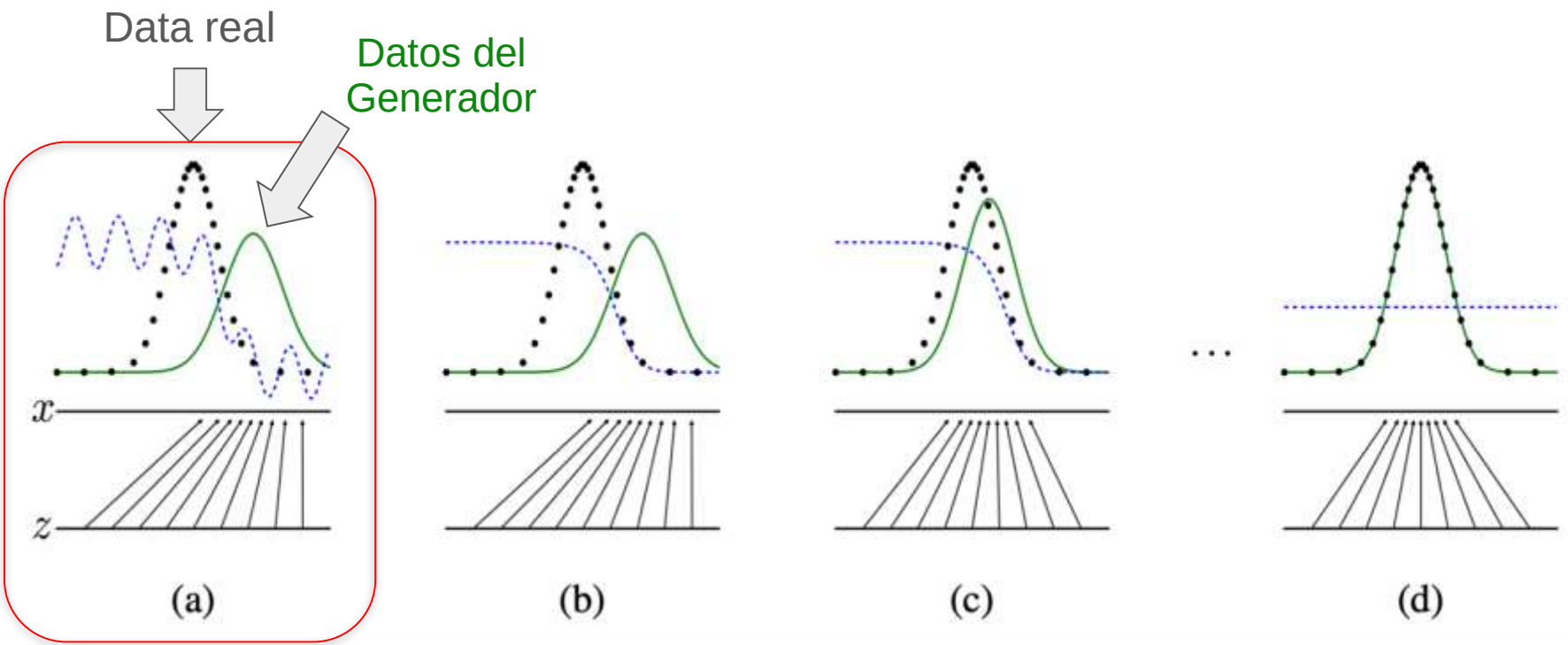
Función de pérdida



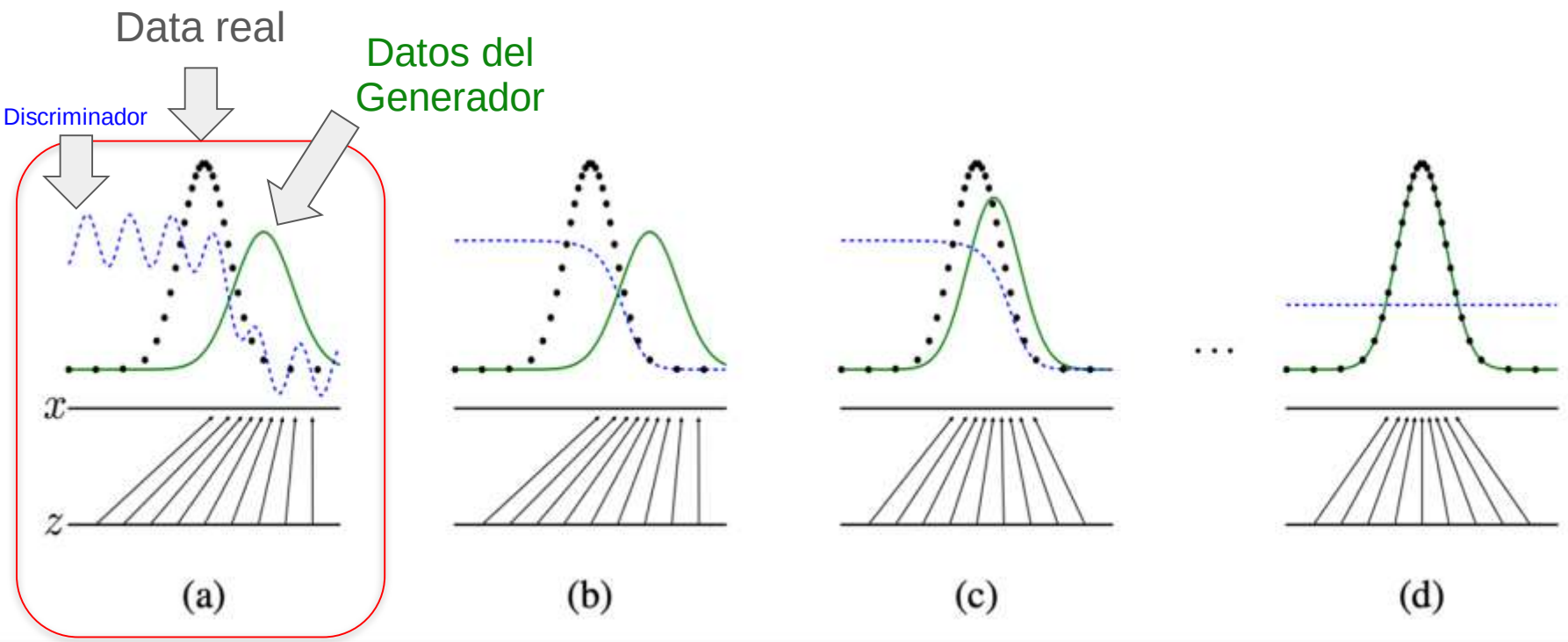
Función de pérdida



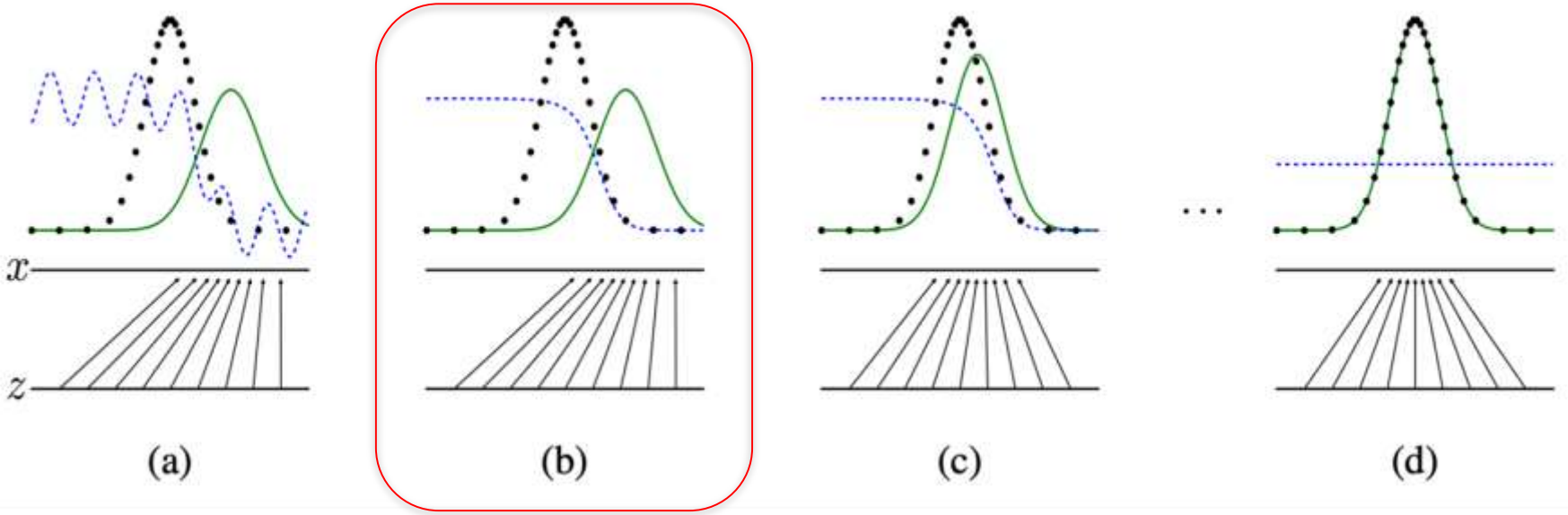
Función de pérdida



Función de pérdida

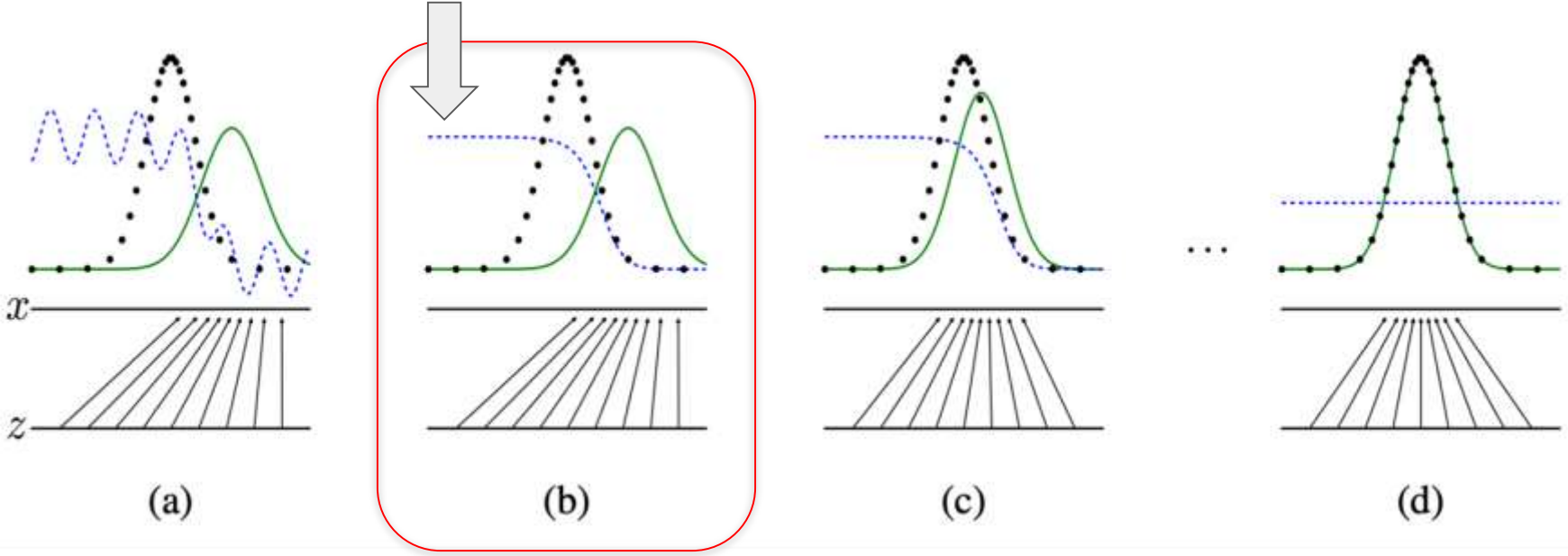


Función de pérdida



Función de pérdida

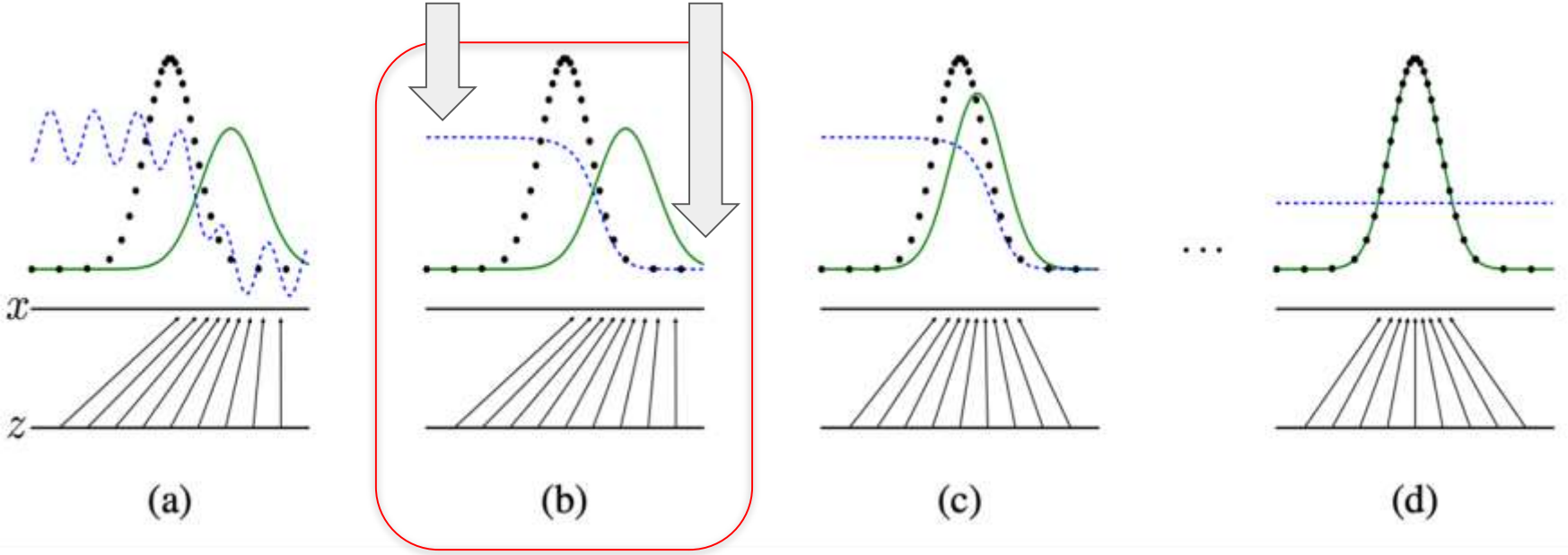
Discriminador da
1 si la data es real



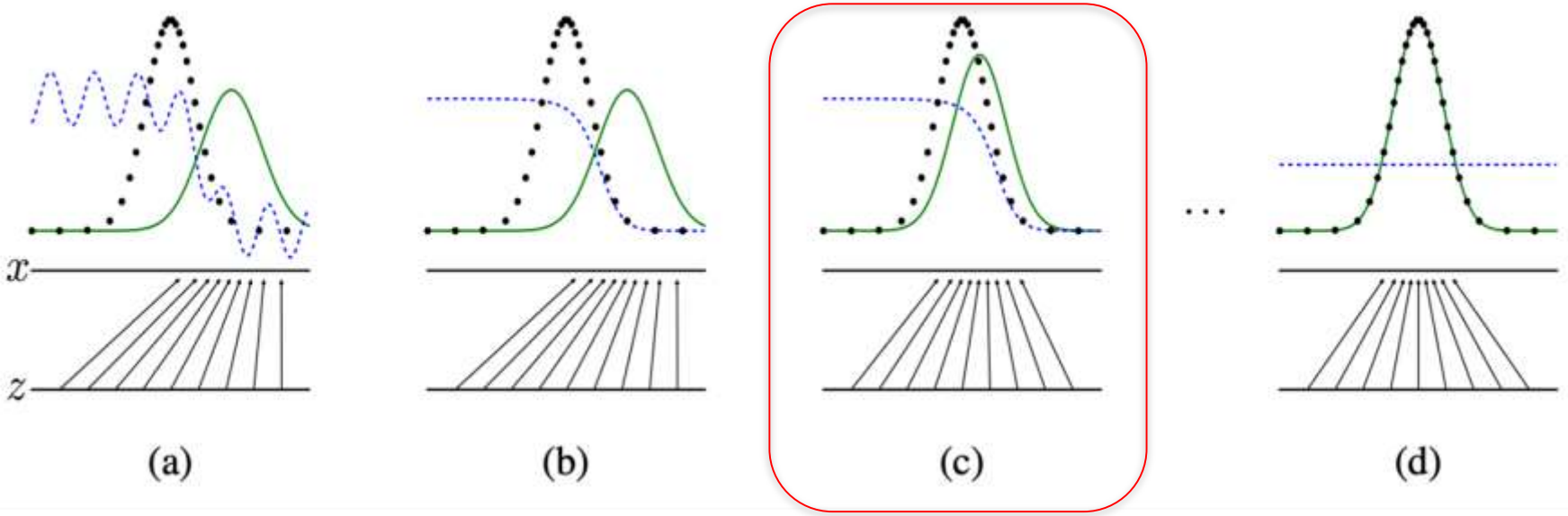
Función de pérdida

Discriminador da 1 si la data es real

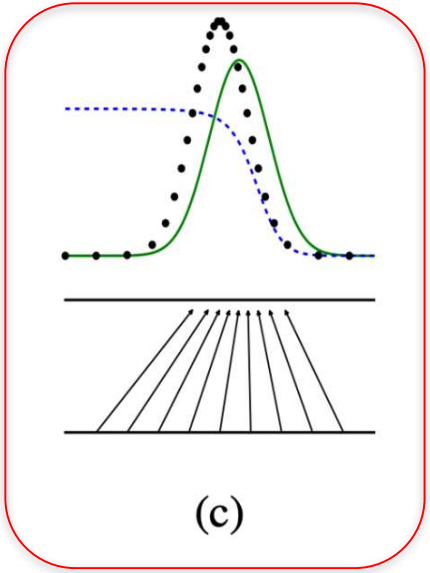
Discriminador da 0 si la data es falsa



Función de pérdida

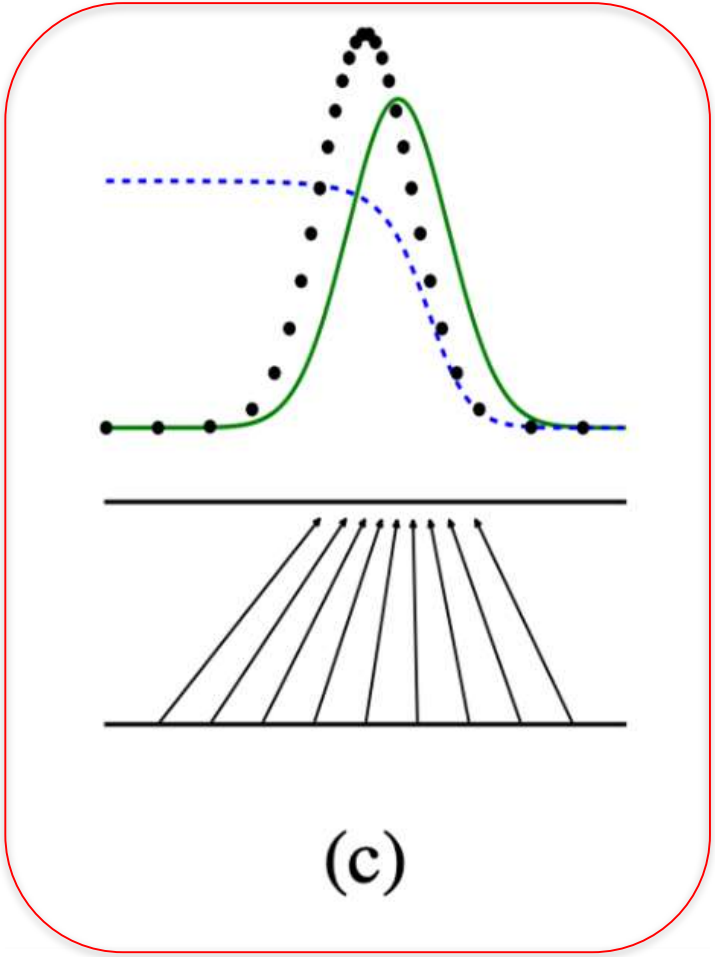


Función de pérdida



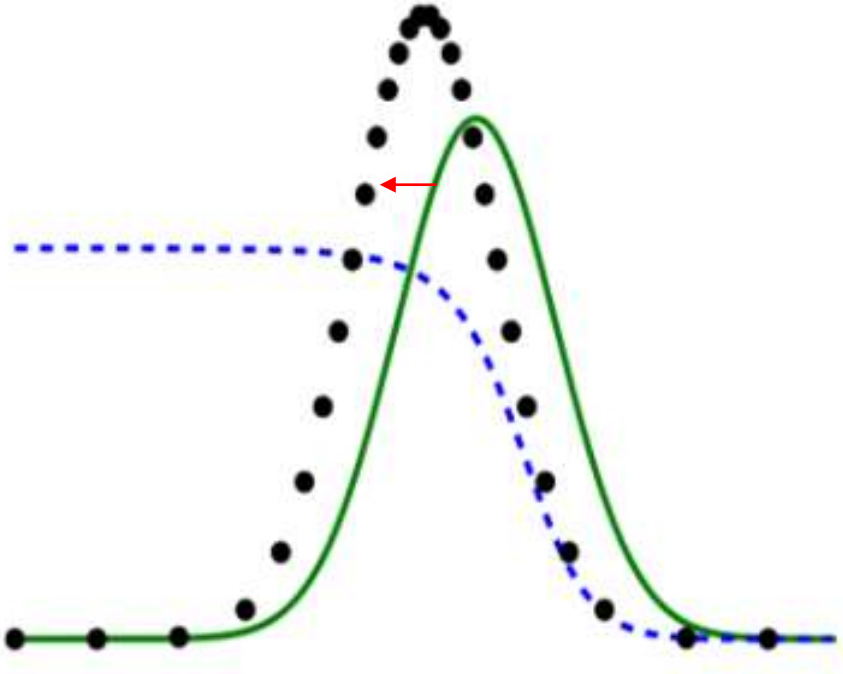
Función de pérdida

La distribución de datos falsos se comienza a acercar a la distribución de datos original



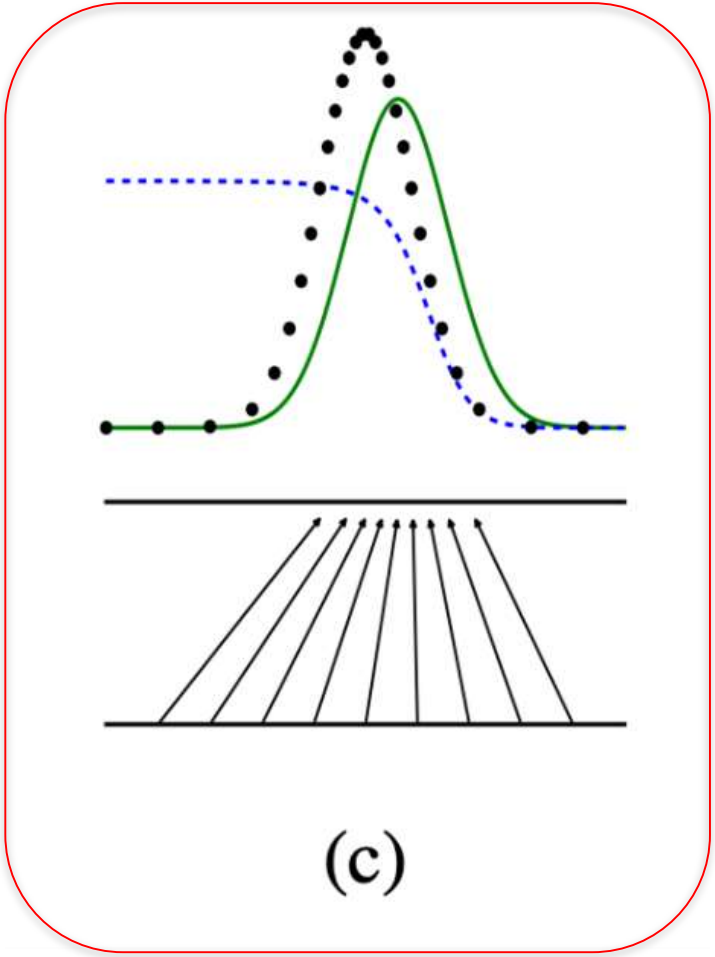
Función de pérdida

La distribución de datos falsos se comienza a acercar a la distribución de datos original



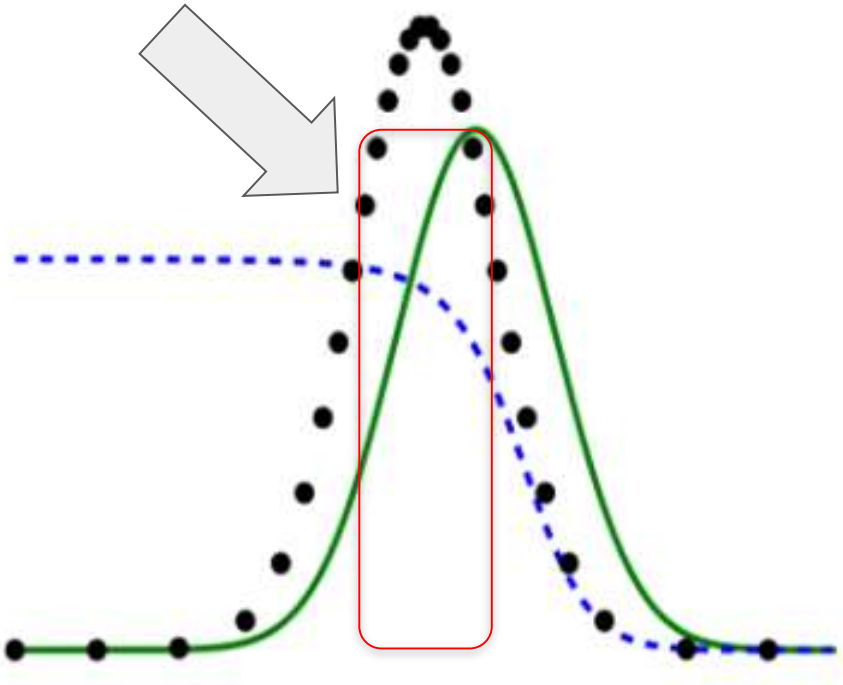
Función de pérdida

El discriminador comienza a clasificar como 1 a la distribución de data falsa.

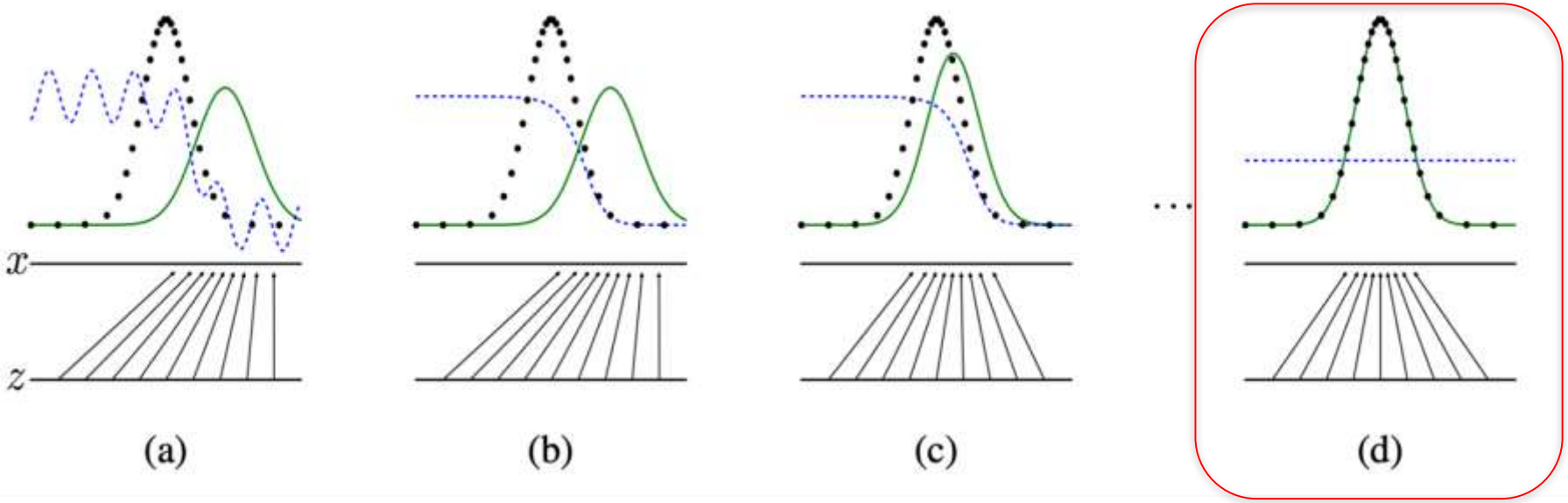


Función de pérdida

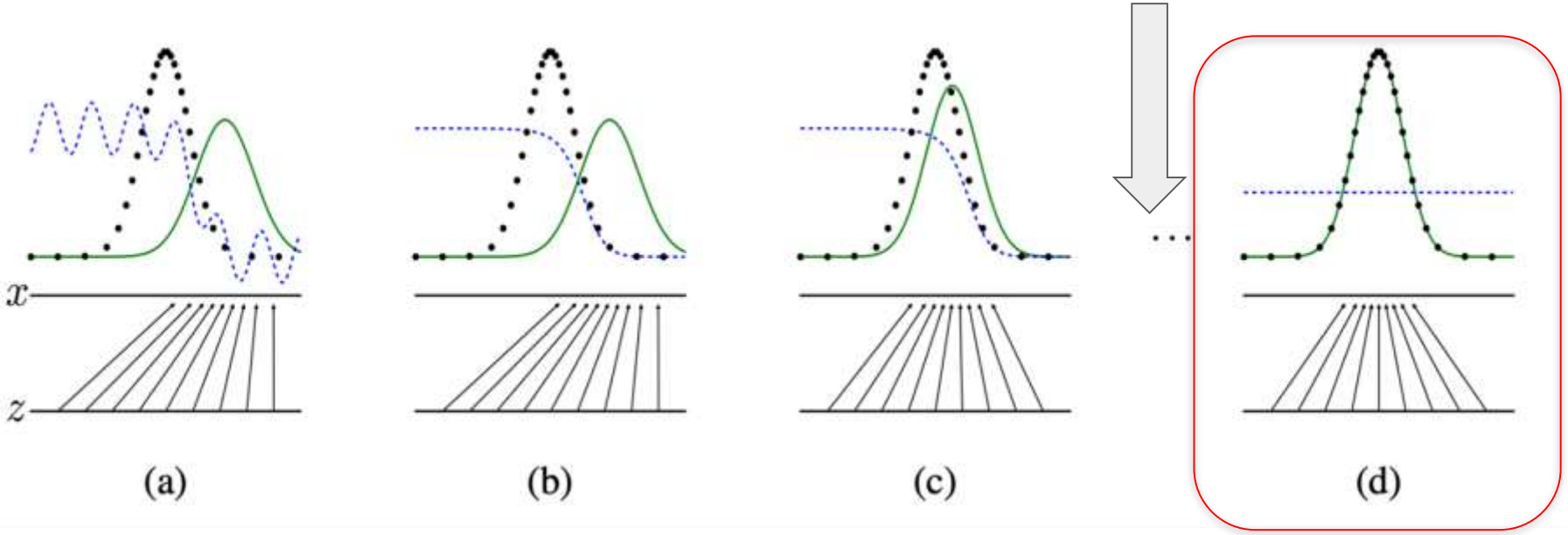
El discriminador comienza a clasificar como 1 a la distribución de data falsa.



Función de pérdida

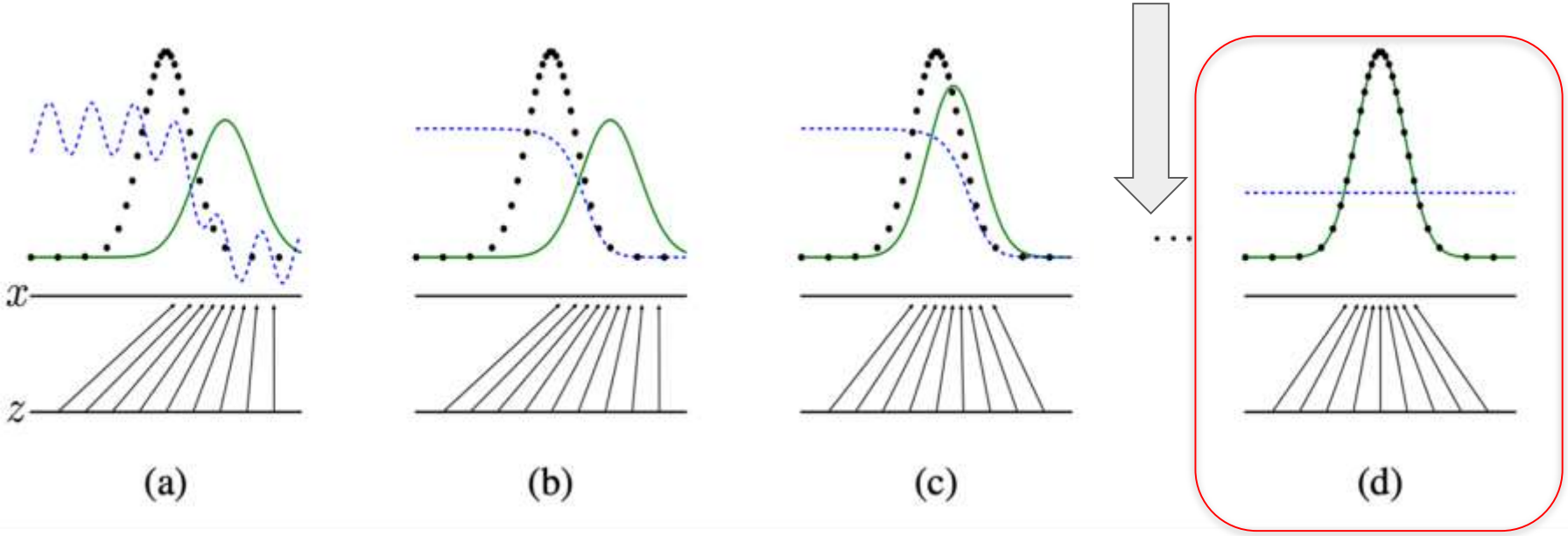


Función de pérdida



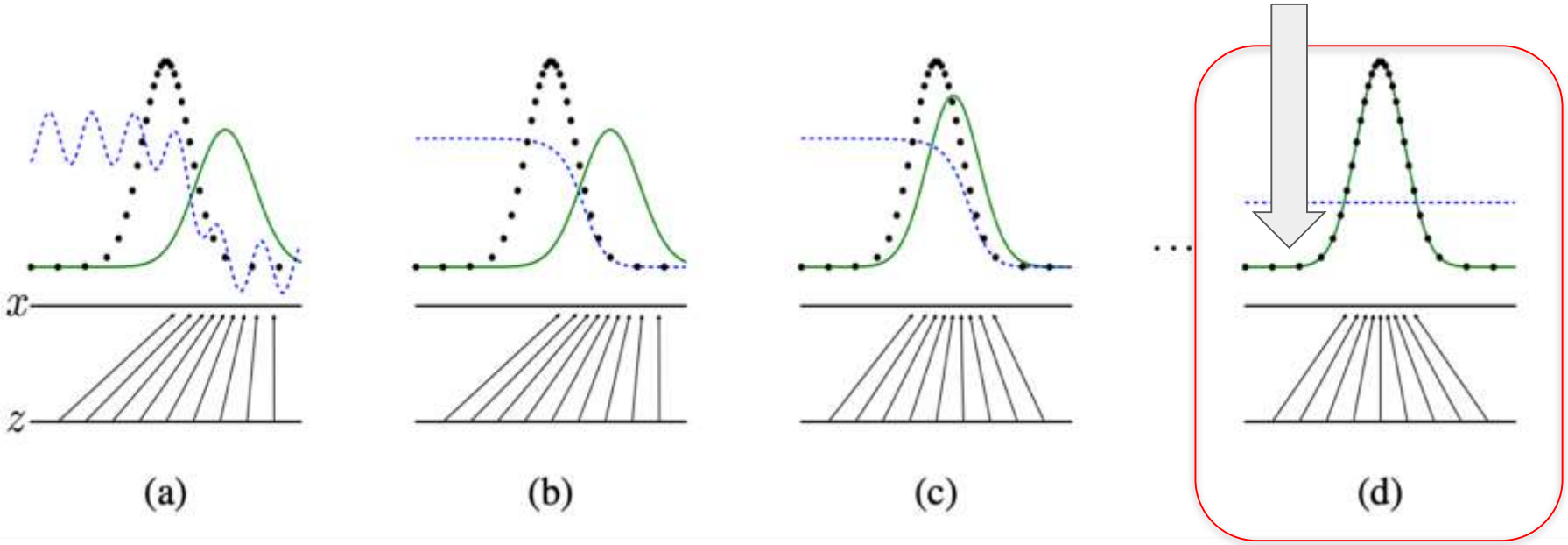
Función de pérdida

Finalmente, luego de algunas iteraciones…



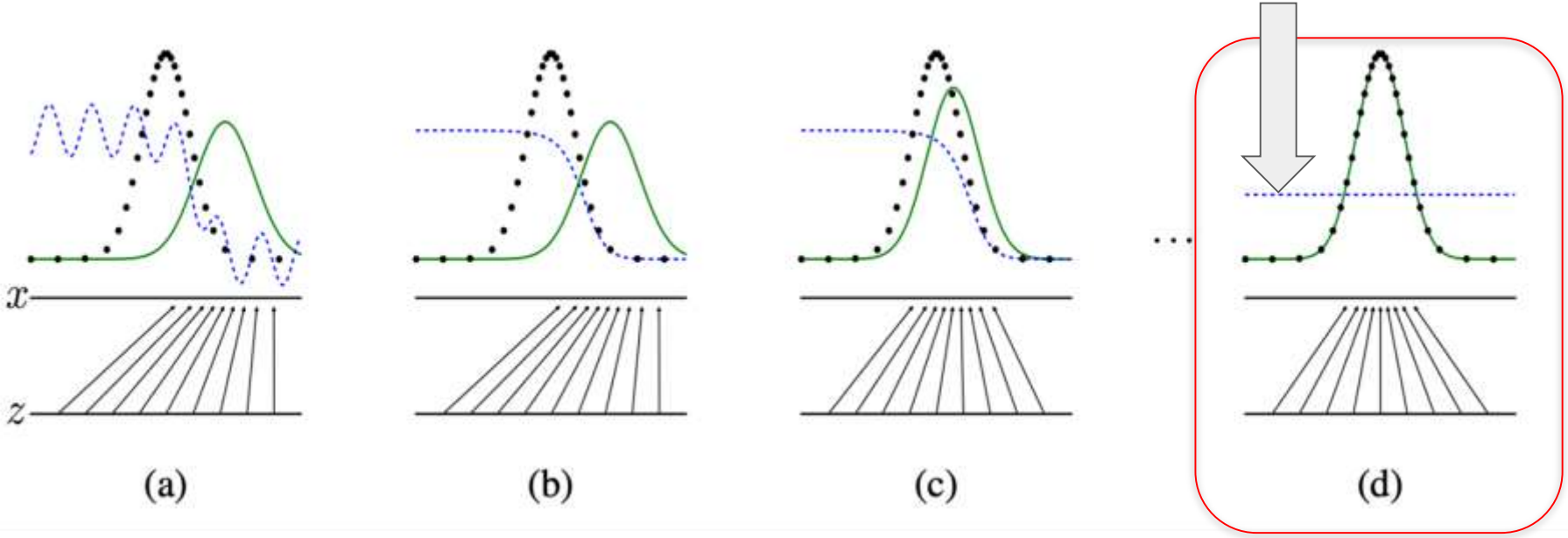
Función de pérdida

El Generador converge generando data similar a la distribución de data real.



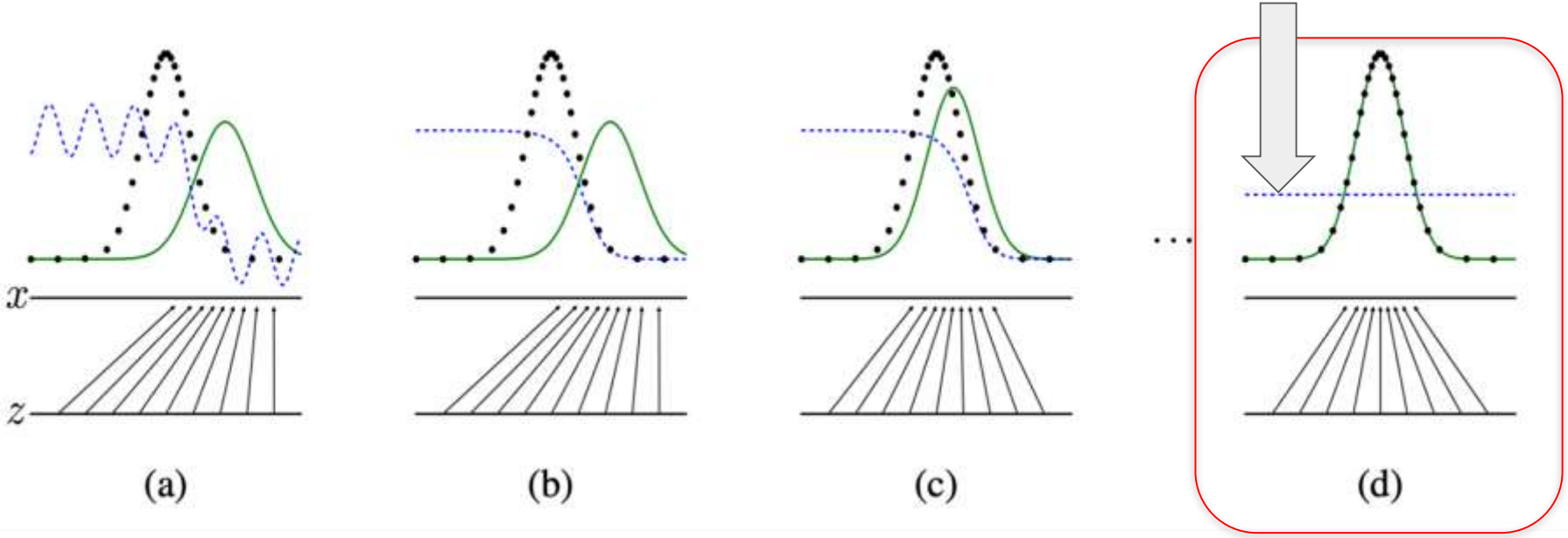
Función de pérdida

El discriminador converge en 0.5

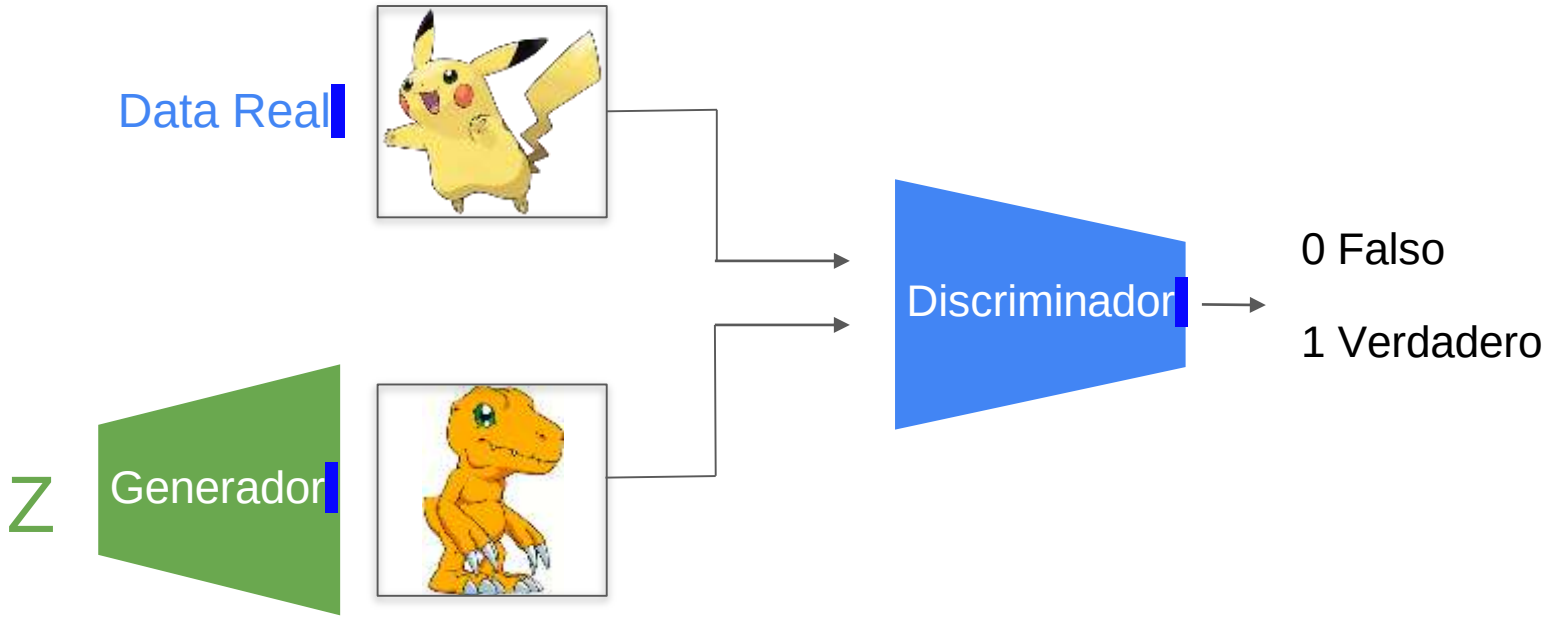


Función de pérdida

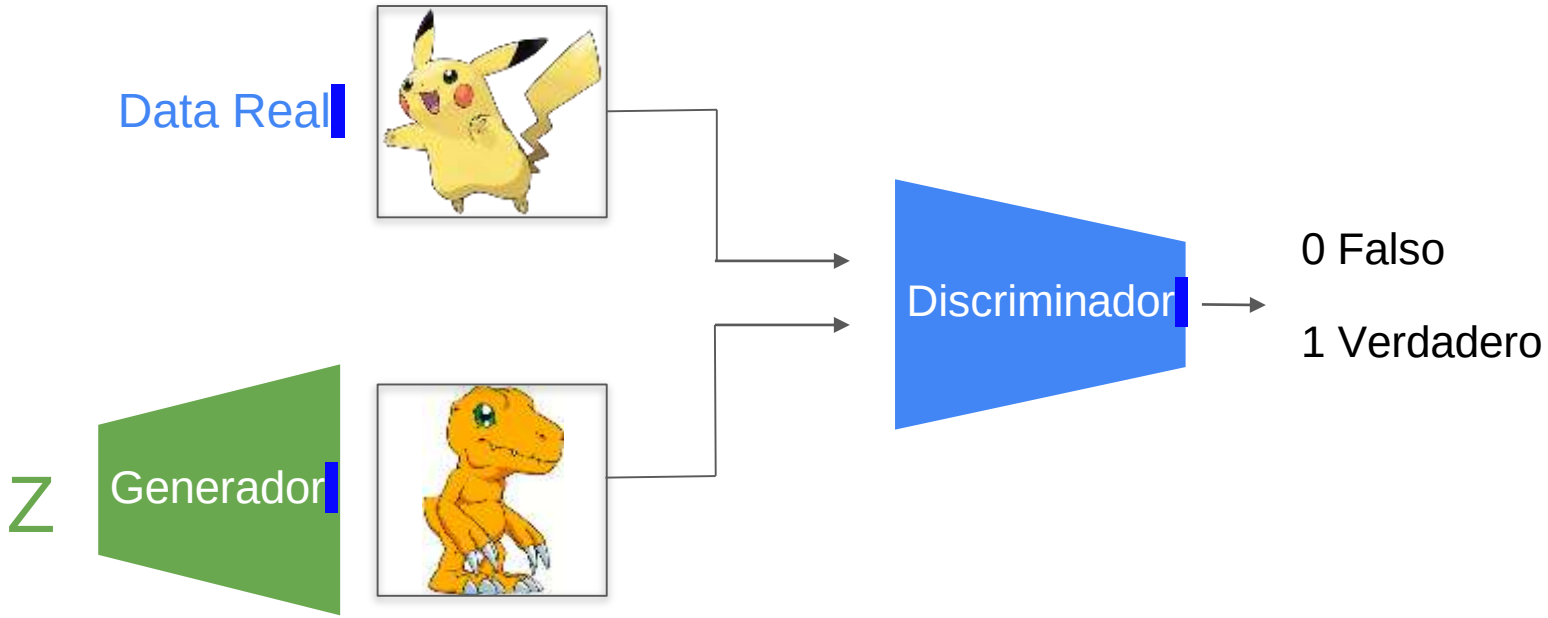
El generador converge en 0.5 (No puede diferenciar entre real y falso)



Función de pérdida

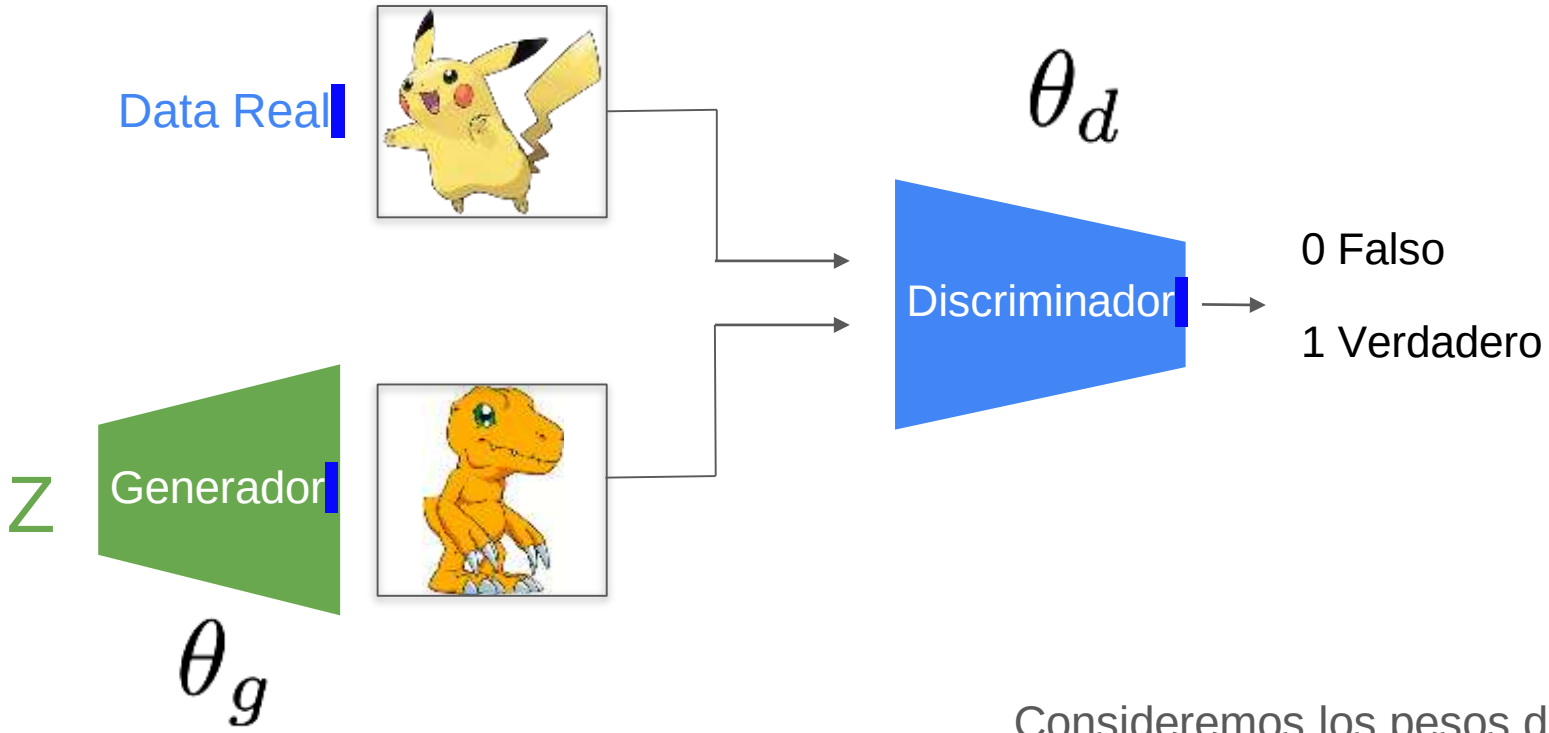


Función de pérdida



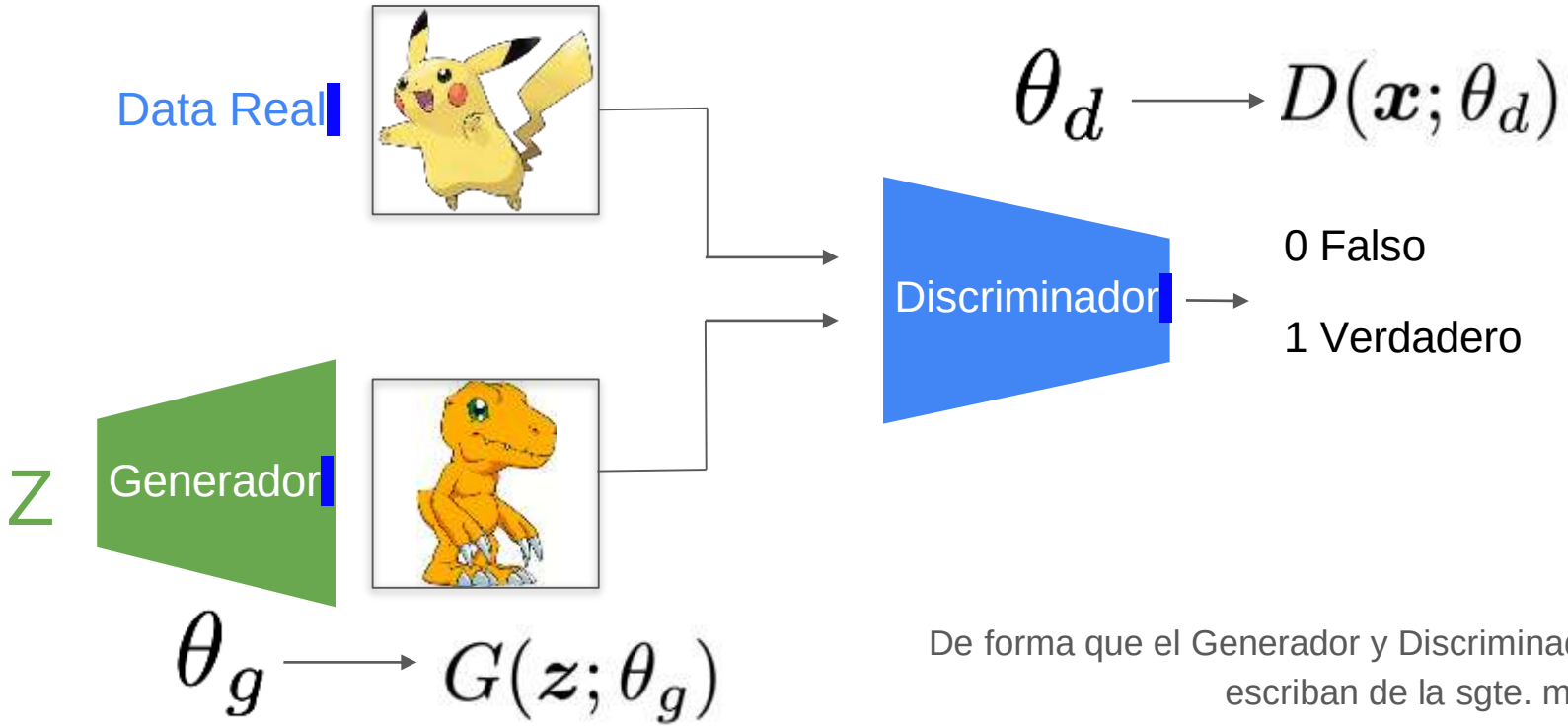
Consideremos los pesos de cada red de la siguiente forma…

Función de pérdida



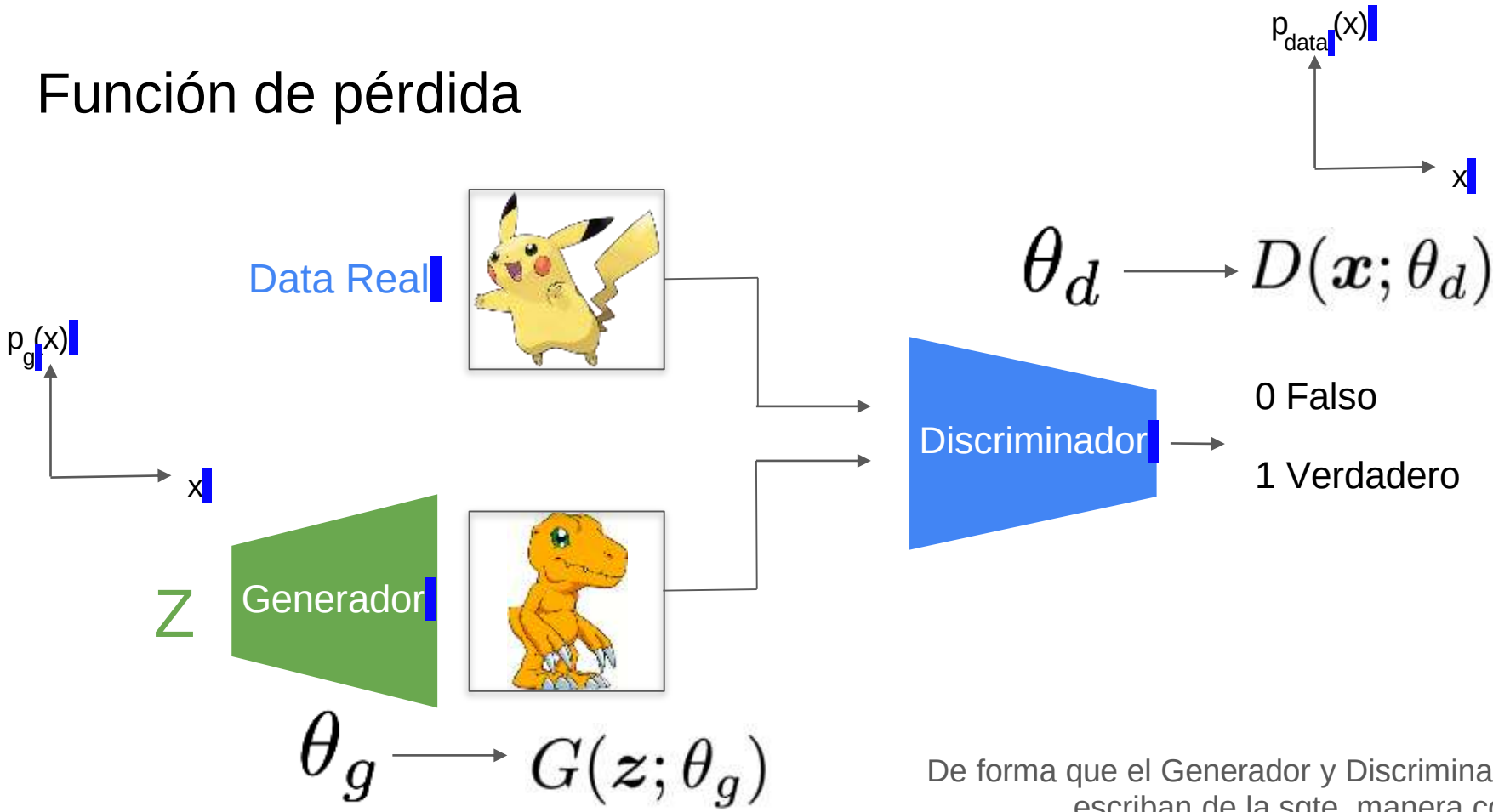
Consideremos los pesos de cada red de la siguiente forma…

Función de pérdida



De forma que el Generador y Discriminador se escriban de la sgte. manera

Función de pérdida



De forma que el Generador y Discriminador se escriban de la sgte. manera con sus distribuciones

Función de pérdida

Función de valor

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

De manera que G y D juegan un MinMax de 2 jugadores

Función de pérdida

$$-\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

Función de BCE

Esto es similar a la función de

Binary Cross Entropy

Función de pérdida

$$-\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

Función de BCE

Cuando $y = 1$ $\quad \hat{y} = D(x) \Rightarrow \mathcal{L} = \ln[Dx]$

Cuando $y = 0$ $\quad \hat{y} = D(G(z)) \Rightarrow \mathcal{L} = \ln[1 - D(G(z))]$

Asimismo, cuando analizamos los valores de y=1 o y=0, vemos que…

Función de pérdida

$$\mathcal{L} = \ln[D(x)] + \ln[1 - D(G(z))]$$

De forma que la función de valor queda de la sgte. forma

Función de pérdida

$$\mathcal{L} = \ln[D(x)] + \ln[1 - D(G(z))]$$

Pero esto es para una única muestra.

Se debe considerar la Esperanza

Función de pérdida

$$E(x) = \sum x p(x) \rightarrow \qquad (1+2+3+4+5+6)$$

La Esperanza…

Función de pérdida

$$E(x) = \sum x p(x) \rightarrow \frac{1}{6}(1+2+3+4+5+6)$$

La Esperanza…

Función de pérdida

$$E(x) = \sum x p(x) \rightarrow 3.5$$

La Esperanza…

Función de pérdida

$$\sum p_{data}(x) \ln[D(x)] + \sum p_z(z) \ln[1 - D(G(z))]$$

Aplicando la Esperanza a nuestra función de valor

Función de pérdida

$$\int p_{data}(x) \ln[D(x)]\,dx + \int p_z(z) \ln[1 - D(G(z))]\,dz$$

Aplicando la Esperanza a nuestra función de valor

Función de pérdida

$$V(G,D) = E_{x \sim p_{data}}[\ln(D(x))] + E_{z \sim p_z}[\ln(1 - D(G(z)))]$$

Aplicando la Esperanza a nuestra función de valor

Función de pérdida

MINIMIZAR
D

$$\frac{\partial}{\partial \theta_d} \frac{1}{m} \cdot \left[\ln\left[1 - D(G(z))\right]\right]$$

MAXIMIZAR
G

$$\frac{\partial}{\partial \theta_g} \frac{1}{m} \left[\ln\left[1 - D(G(z))\right]\right]$$

cant. de muestras

El algoritmo de entrenamiento quedaría de la siguiente forma

Función de pérdida

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.

El algoritmo de entrenamiento quedaría de la siguiente forma

Aplicaciones

Crear personajes de anime

https://arxiv.org/pdf/1708.05509.pdf

Figure 7: Generated samples

Aplicaciones

Crear personajes de anime

https://arxiv.org/pdf/1708.05509.pdf

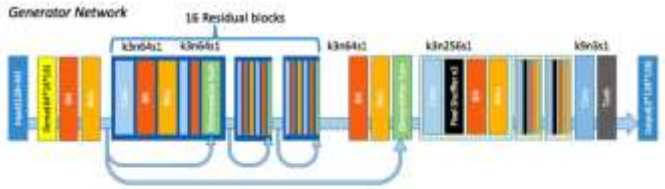


Figure 3: Generator Architecture

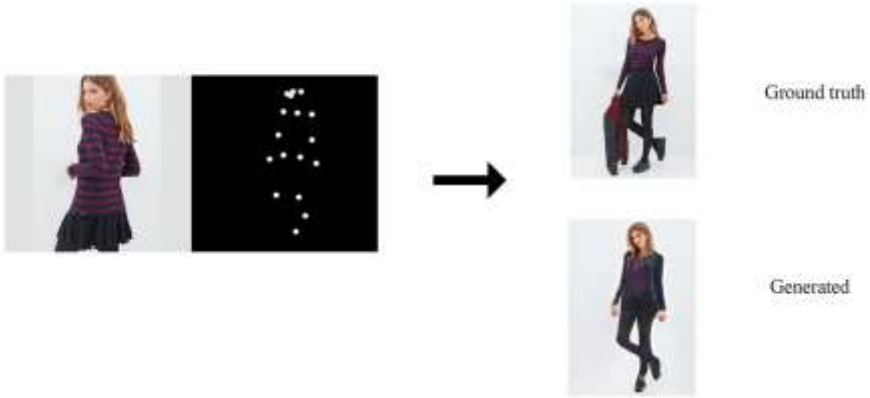
Figure 4: Discriminator Architecture

Figure 7: Generated samples

Aplicaciones

Personas posando

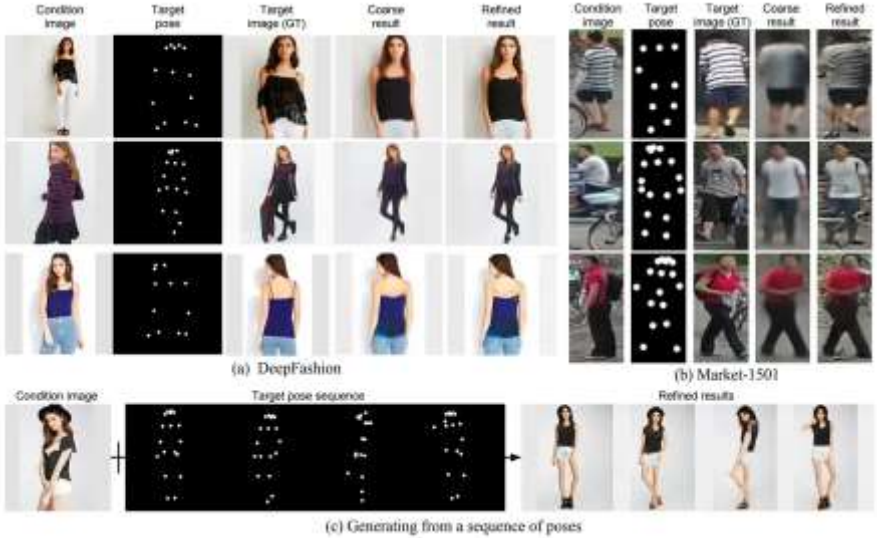
https://arxiv.org/pdf/1705.09368.pdf

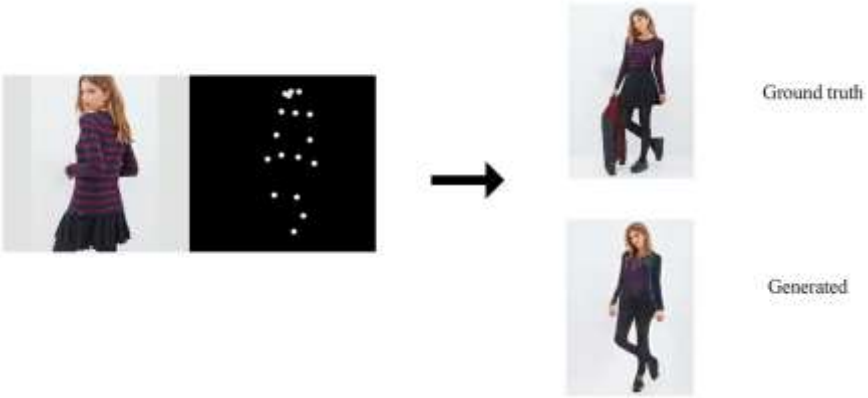


Aplicaciones

Personas posando

https://arxiv.org/pdf/1705.09368.pdf

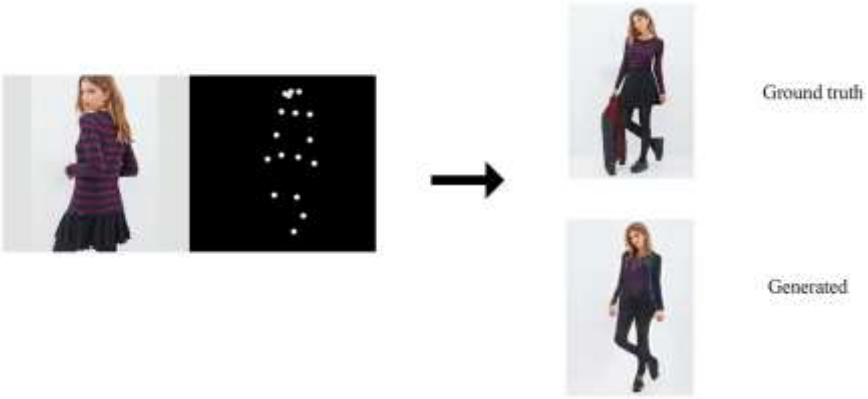


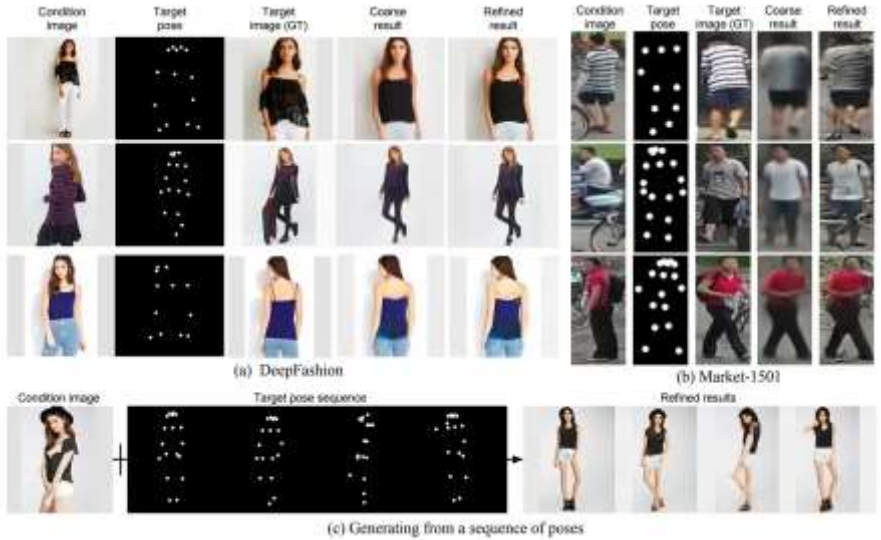


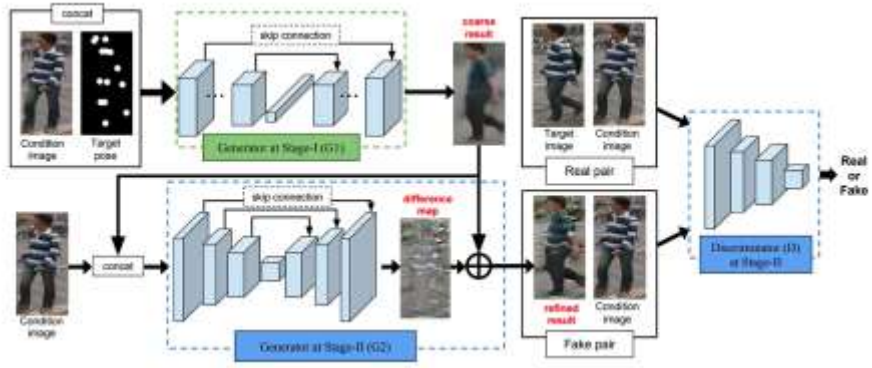
Aplicaciones

Personas posando

https://arxiv.org/pdf/1705.09368.pdf



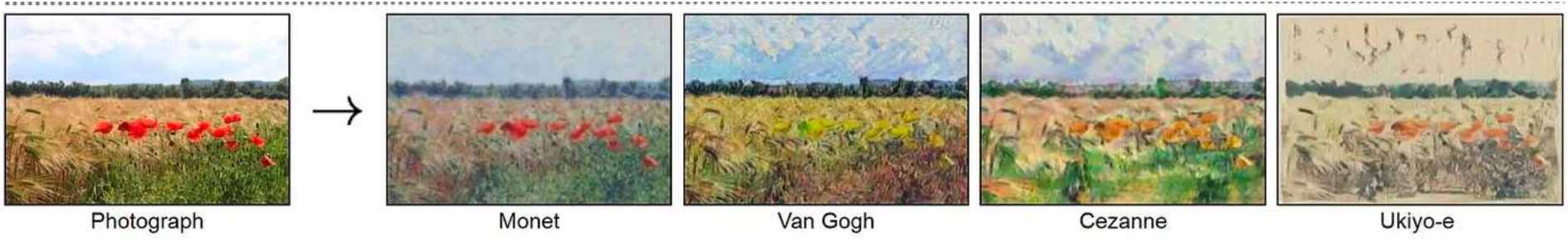




Aplicaciones

CycleGAN

https://github.com/junyanz/CycleGAN



Aplicaciones

CycleGAN

https://github.com/junyanz/CycleGAN

Aplicaciones

StarGAN

https://arxiv.org/pdf/1711.09020.pdf

StarGAN es una traducción de imagen a imagen de un dominio a otro.

Por ejemplo, dada una cara feliz, queremos transformarla en una cara de miedo.

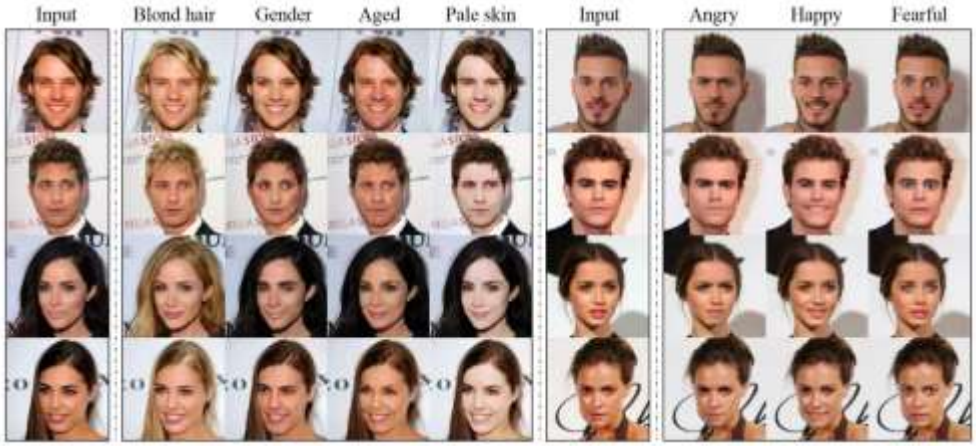
Aplicaciones

StarGAN

https://arxiv.org/pdf/1711.09020.pdf

StarGAN es una traducción de imagen a imagen de un dominio a otro.

Por ejemplo, dada una cara feliz, queremos transformarla en una cara de miedo.

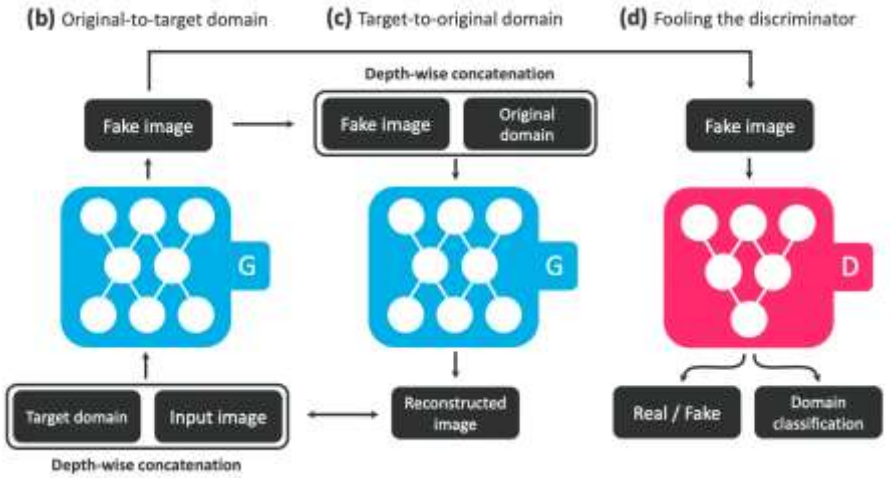


Aplicaciones

StarGAN

https://arxiv.org/pdf/1711.09020.pdf

En (b), el generador genera una imagen falsa basada en una imagen de entrada y una etiqueta de dominio de destino (por ejemplo, enojado).

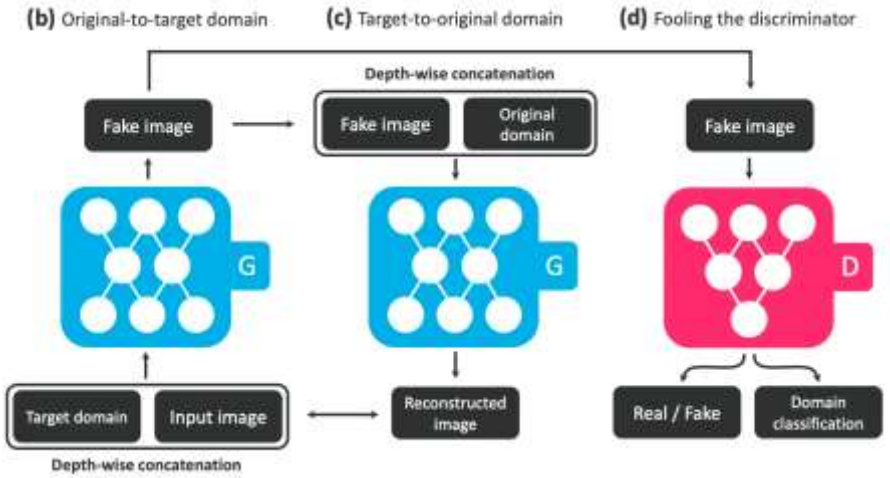


Aplicaciones

StarGAN

https://arxiv.org/pdf/1711.09020.pdf

En (d), alimentamos imágenes reales y falsas al discriminador para etiquetarlo como real o no, así como su clasificación de dominio.

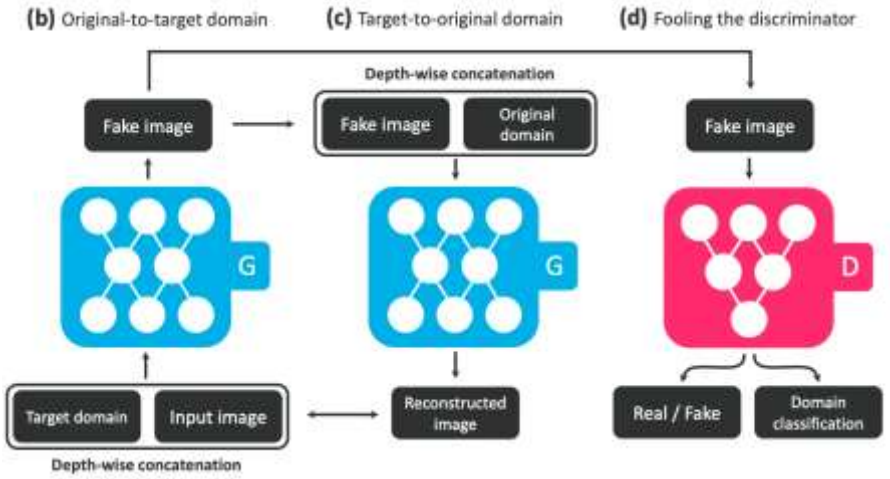


Aplicaciones

StarGAN

https://arxiv.org/pdf/1711.09020.pdf

La función de costo involucrará errores de reconstrucción así como el costo del discriminador en la identificación de las imágenes y sus etiquetas.



Aplicaciones

PixelDTGAN

https://github.com/fxia22/PixelDTGAN

Aplicaciones

PixelDTGAN

https://github.com/fxia22/PixelDTGAN

A source image. Possible target images.

Aplicaciones

PixelDTGAN

https://github.com/fxia22/PixelDTGAN

Aplicaciones

PixelDTGAN

https://github.com/fxia22/PixelDTGAN

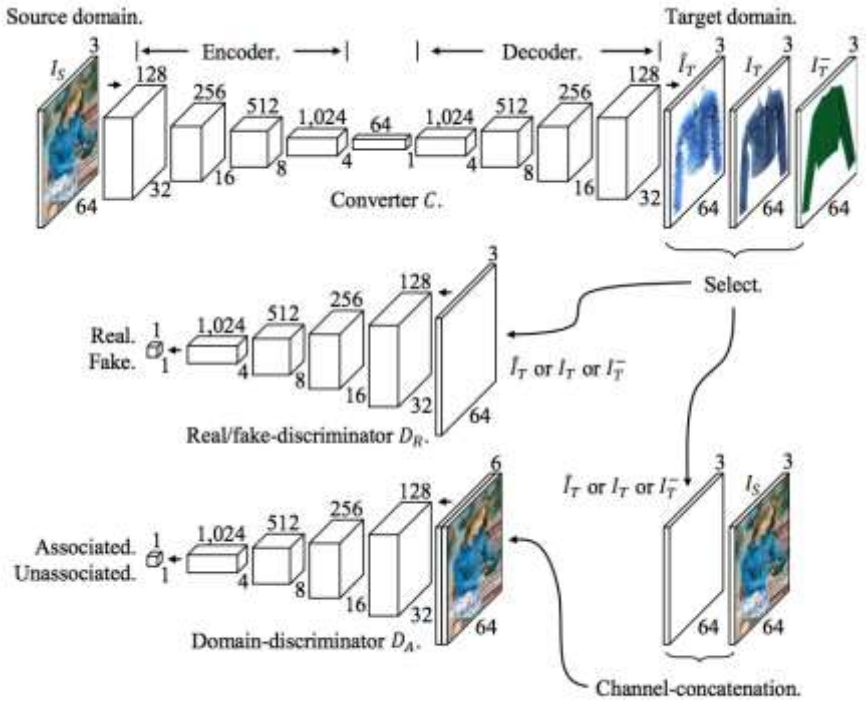


Fig. 2. Whole architecture for pixel-level domain transfer.

Aplicaciones

SuperResolution GAN

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

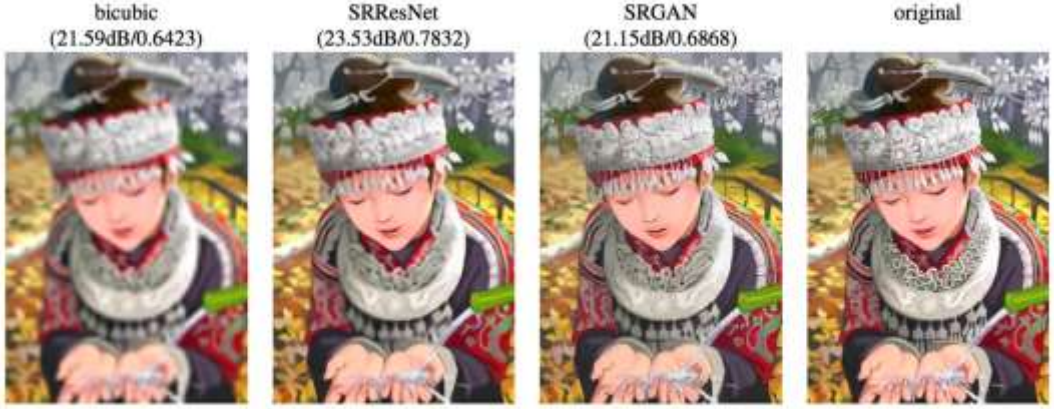


Figure 2: From left to right: bicubic interpolation, deep residual network optimized for MSE, deep residual generative adversarial network optimized for a loss more sensitive to human perception, original HR image. Corresponding PSNR and SSIM are shown in brackets. [4× upscaling]

Aplicaciones

StyleGAN 2

https://arxiv.org/pdf/1912.04958.pdf

Aplicaciones

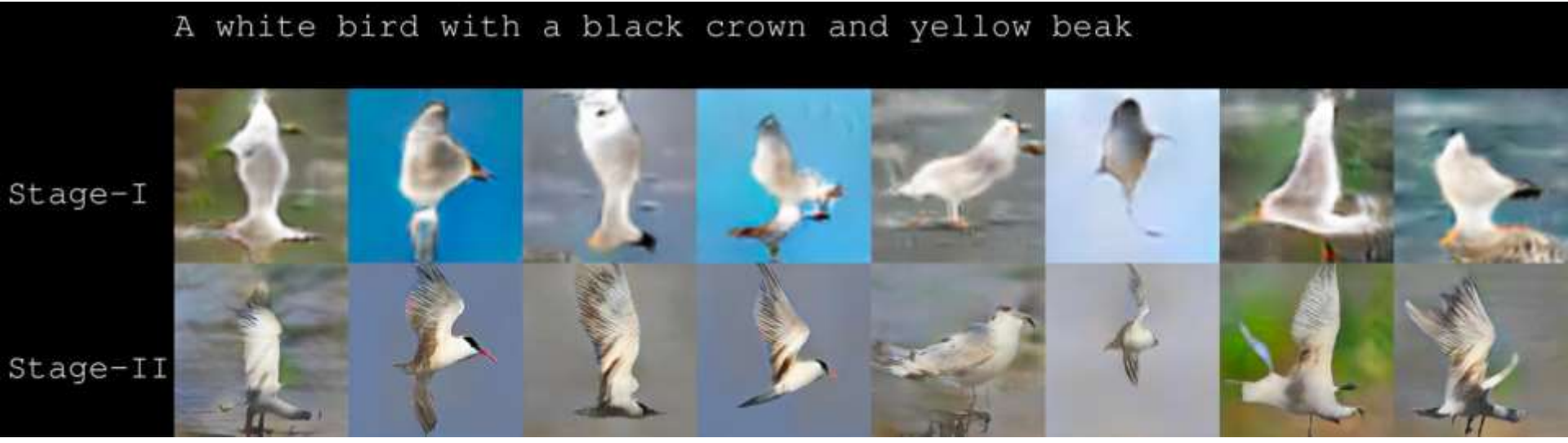
Text 2 Imagen GAN (StackGAN)

https://github.com/hanzhanggit/StackGAN

Aplicaciones

Text 2 Imagen GAN (StackGAN)

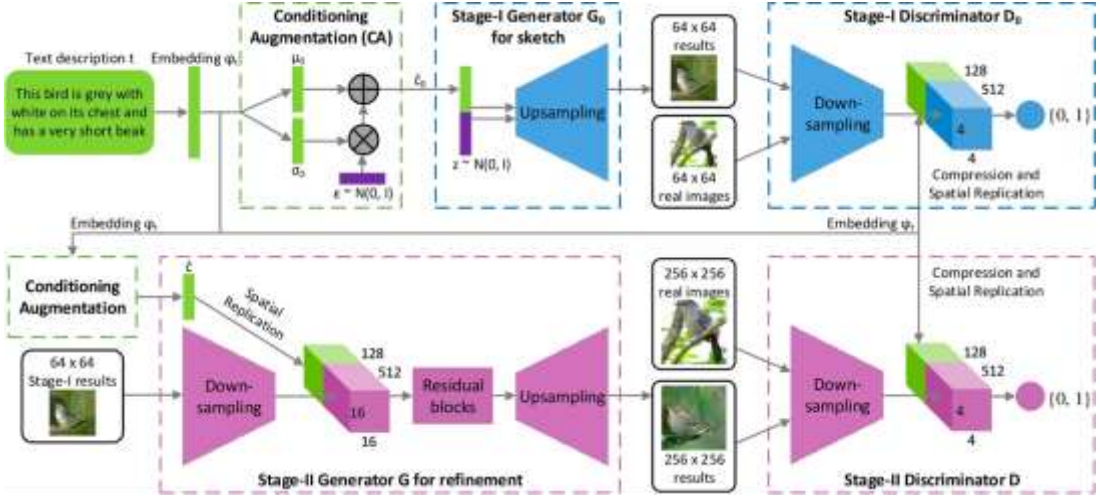
https://github.com/hanzhanggit/StackGAN



Aplicaciones

Text 2 Imagen GAN (StackGAN)

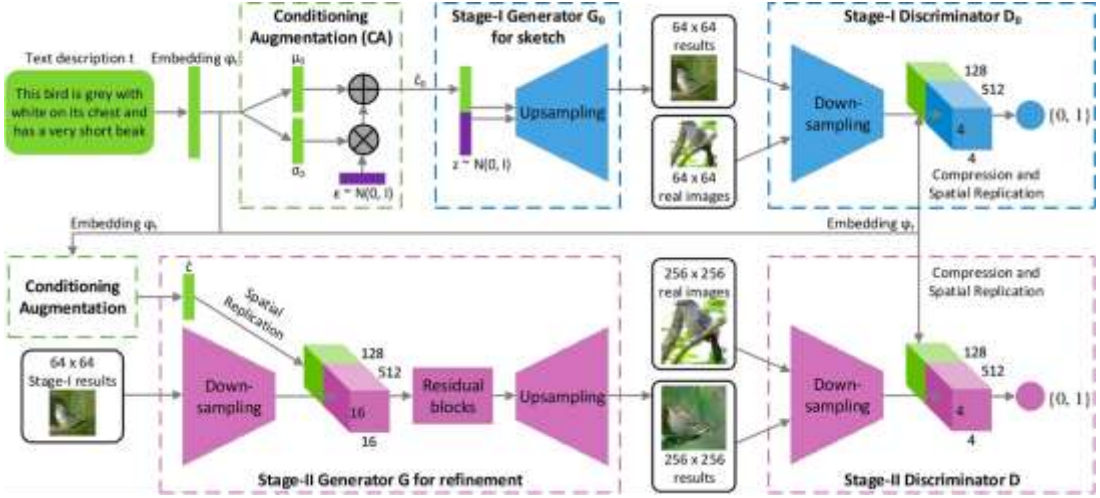
https://github.com/hanzhanggit/StackGAN



Aplicaciones

Text 2 Imagen GAN (StackGAN)

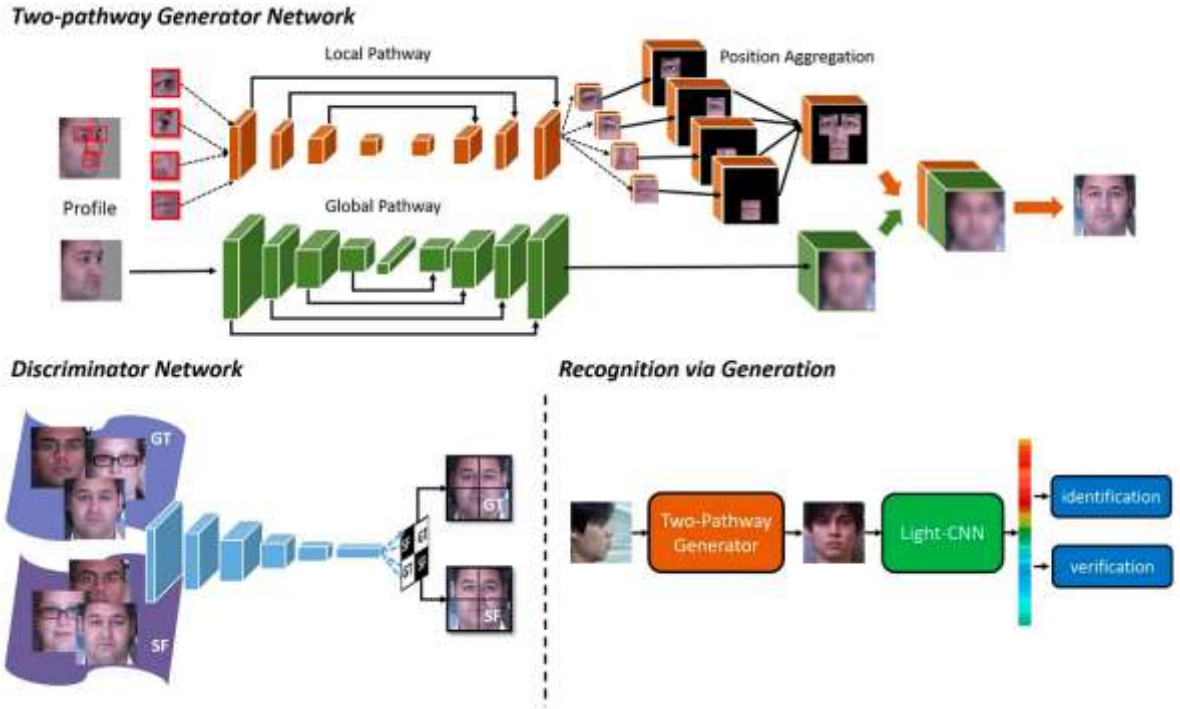
https://github.com/hanzhanggit/StackGAN - Demo: https://www.craiyon.com/



Aplicaciones

TPGAN

https://arxiv.org/pdf/1704.04086.pdf

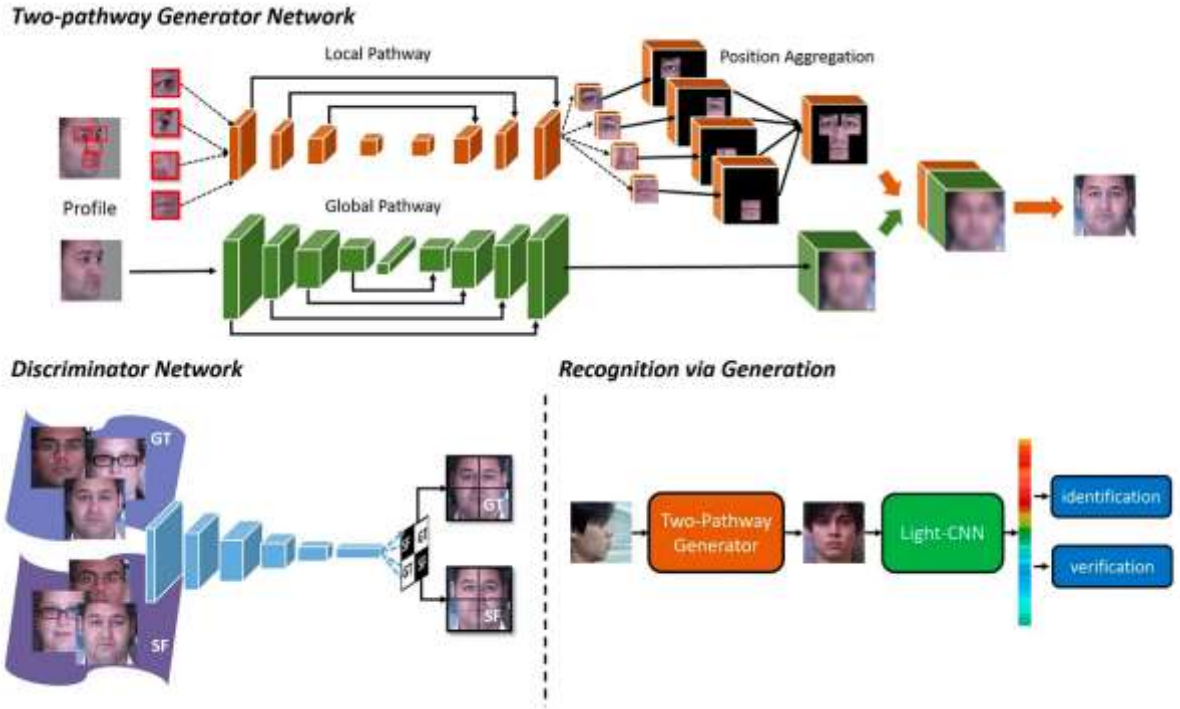


Aplicaciones

TPGAN

https://arxiv.org/pdf/1704.04086.pdf

Caras de síntesis en diferentes poses: Con una única imagen de entrada, crea caras en diferentes ángulos de visión.

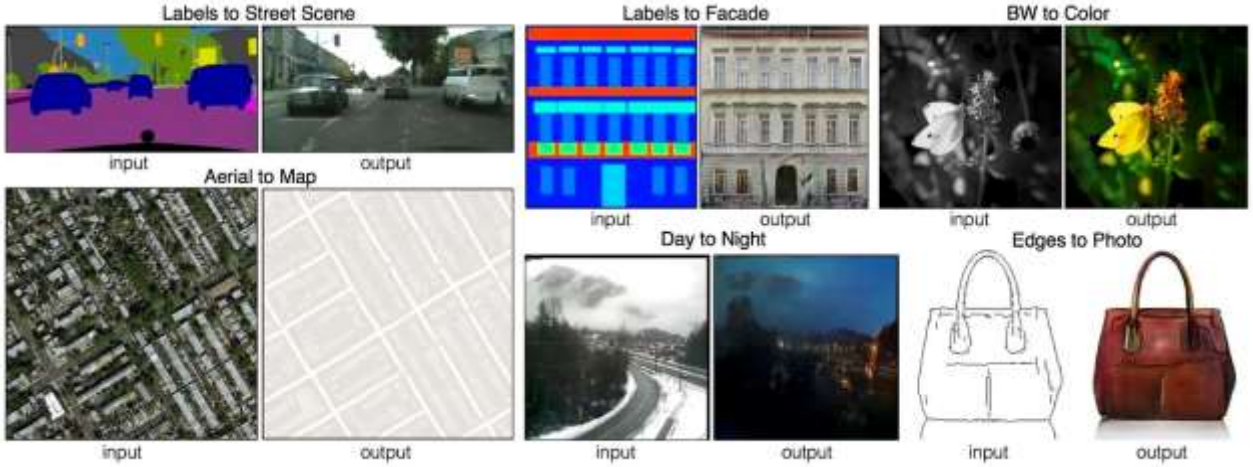


Aplicaciones

Pix2Pix

https://github.com/phillipi/pix2pix

Traducción de imágenes en diferentes tareas.



Aplicaciones

DeBlur

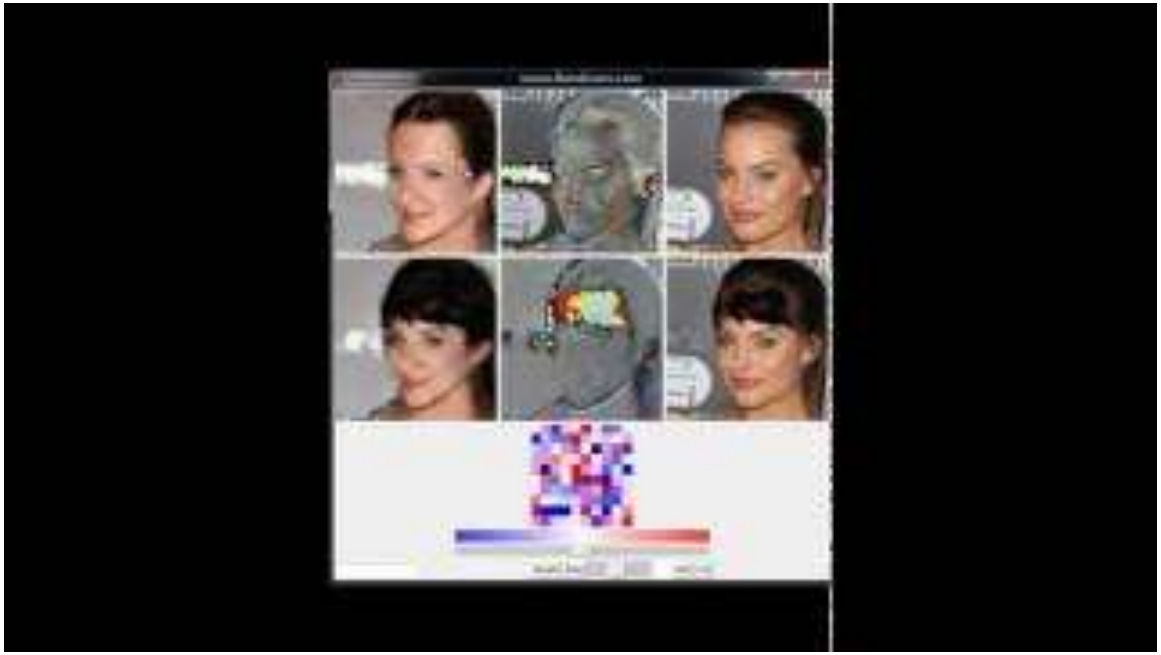
https://arxiv.org/pdf/1711.07064.pdf

Figure 2: GoPro images [25] processed by DeblurGAN. Blurred – left, DeblurGAN – center, ground truth sharp – right.

Aplicaciones

Neural Photo Editor

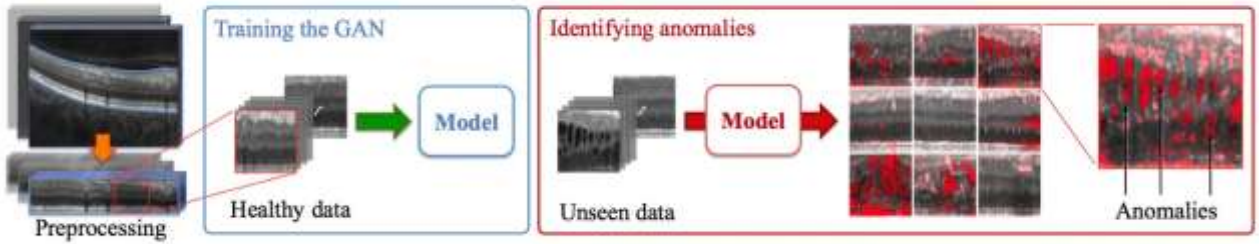
https://github.com/ajbrock/Neural-Photo-Editor



Aplicaciones

MediGAN

https://arxiv.org/pdf/1703.05921.pdf



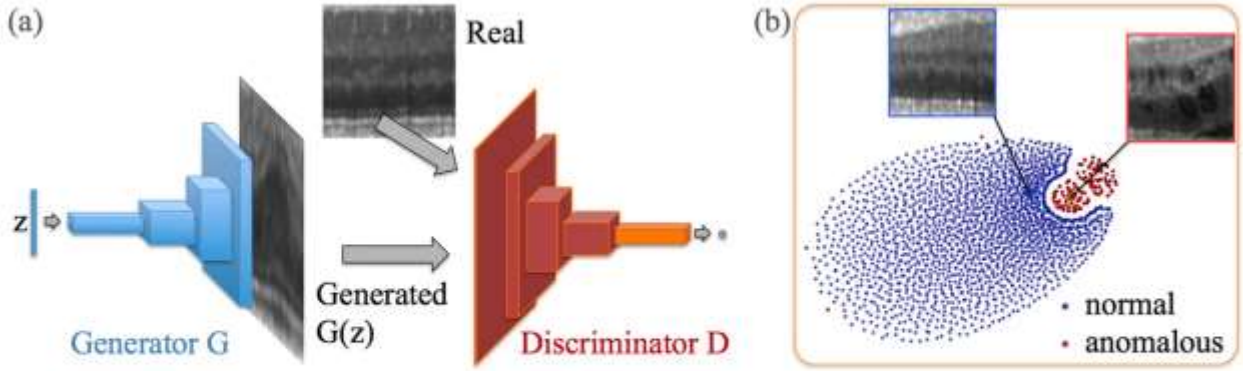


Fig. 2. (a) Deep convolutional generative adversarial network. (b) t-SNE embedding of normal (blue) and anomalous (red) images on the feature representation of the last convolution layer (orange in (a)) of the discriminator.

Aplicaciones

3D GAN

https://proceedings.neurips.cc/paper_files/paper/2016/file/44f683a84163b3523afe57c2e008bc8c-Paper.pdf

Aplicaciones

3D GAN

https://proceedings.neurips.cc/paper_files/paper/2016/file/44f683a84163b3523afe57c2e008bc8c-Paper.pdf

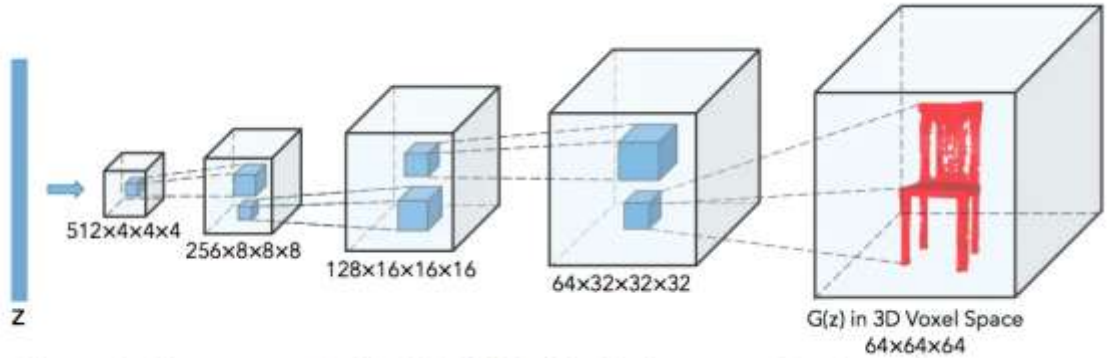


Figure 1: The generator in 3D-GAN. The discriminator mostly mirrors the generator.

Fuentes:

GAN (2014) Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio
https://arxiv.org/abs/1406.2661

https://github.com/soumith/ganhacks

https://jonathan-hui.medium.com/gan-some-cool-applications-of-gans-4c9ecca35900