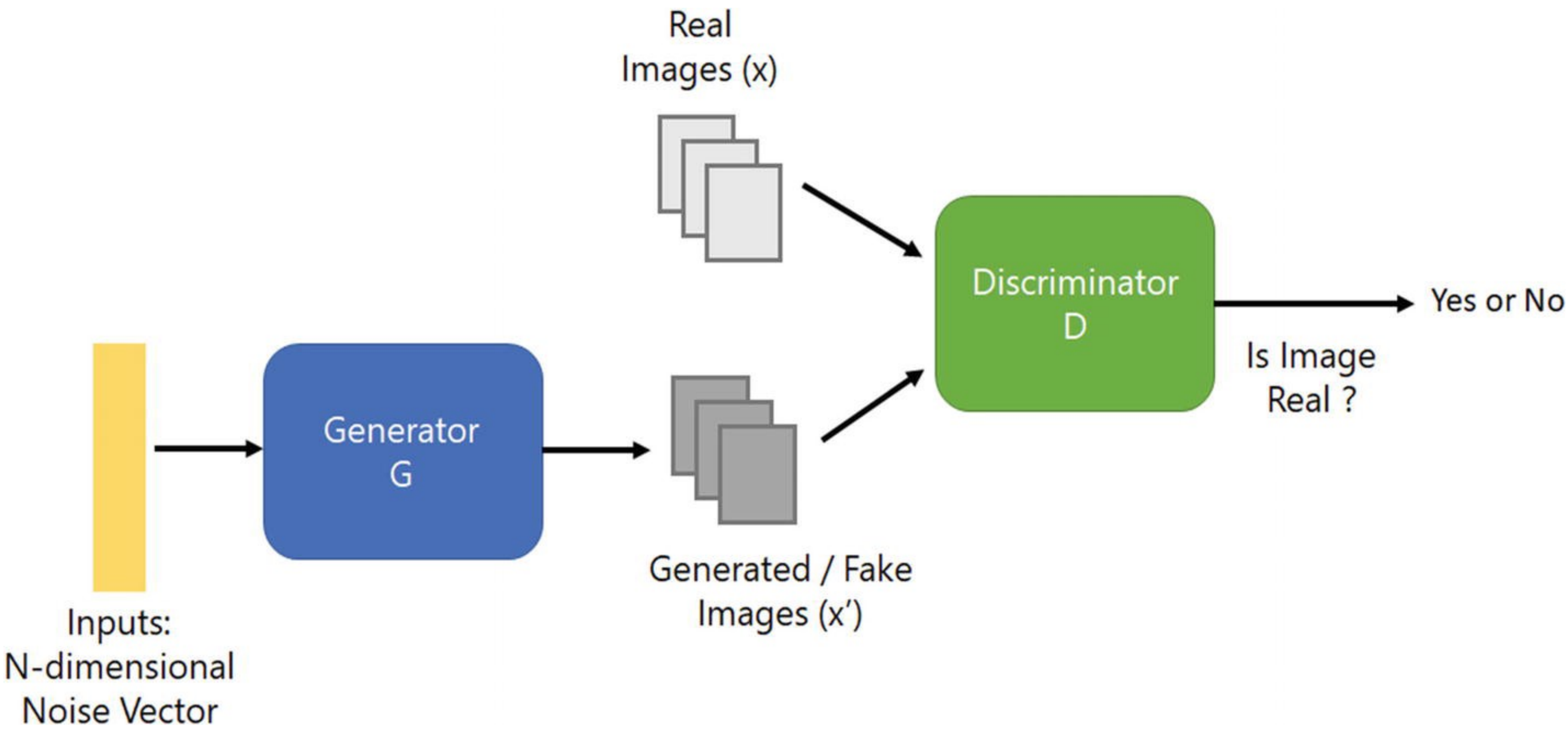
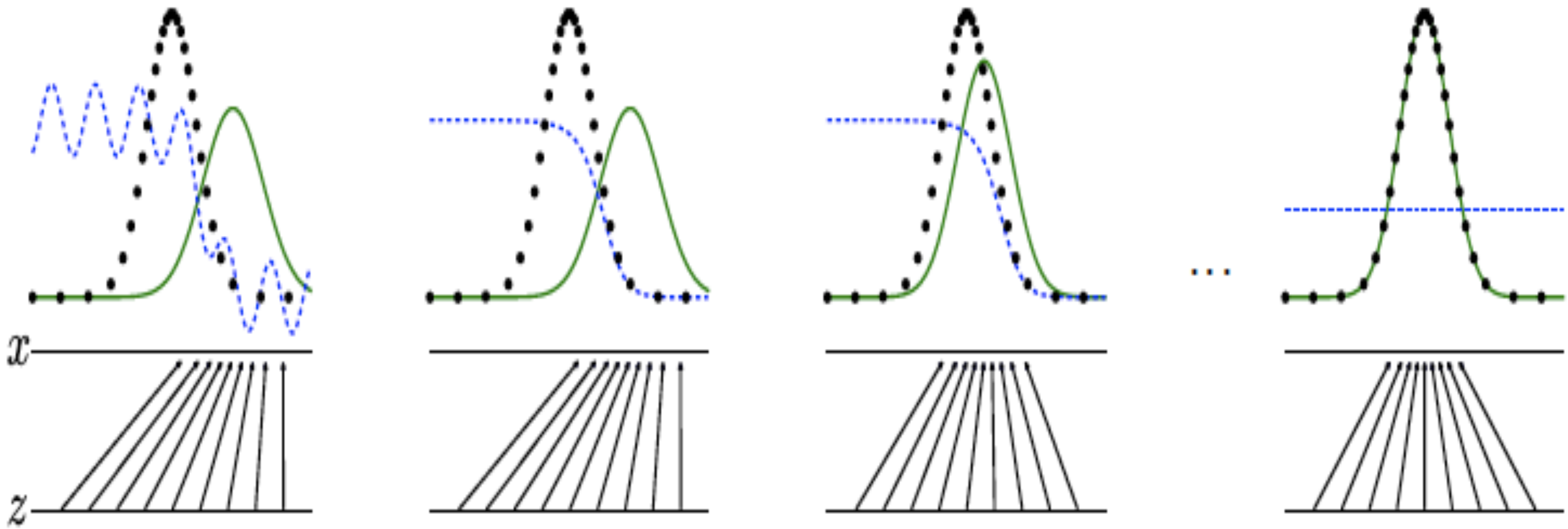
Generative Adversarial Net



Generative Adversarial Training



(a) (b) (c) (d)

Generative Adversarial Training

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:

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

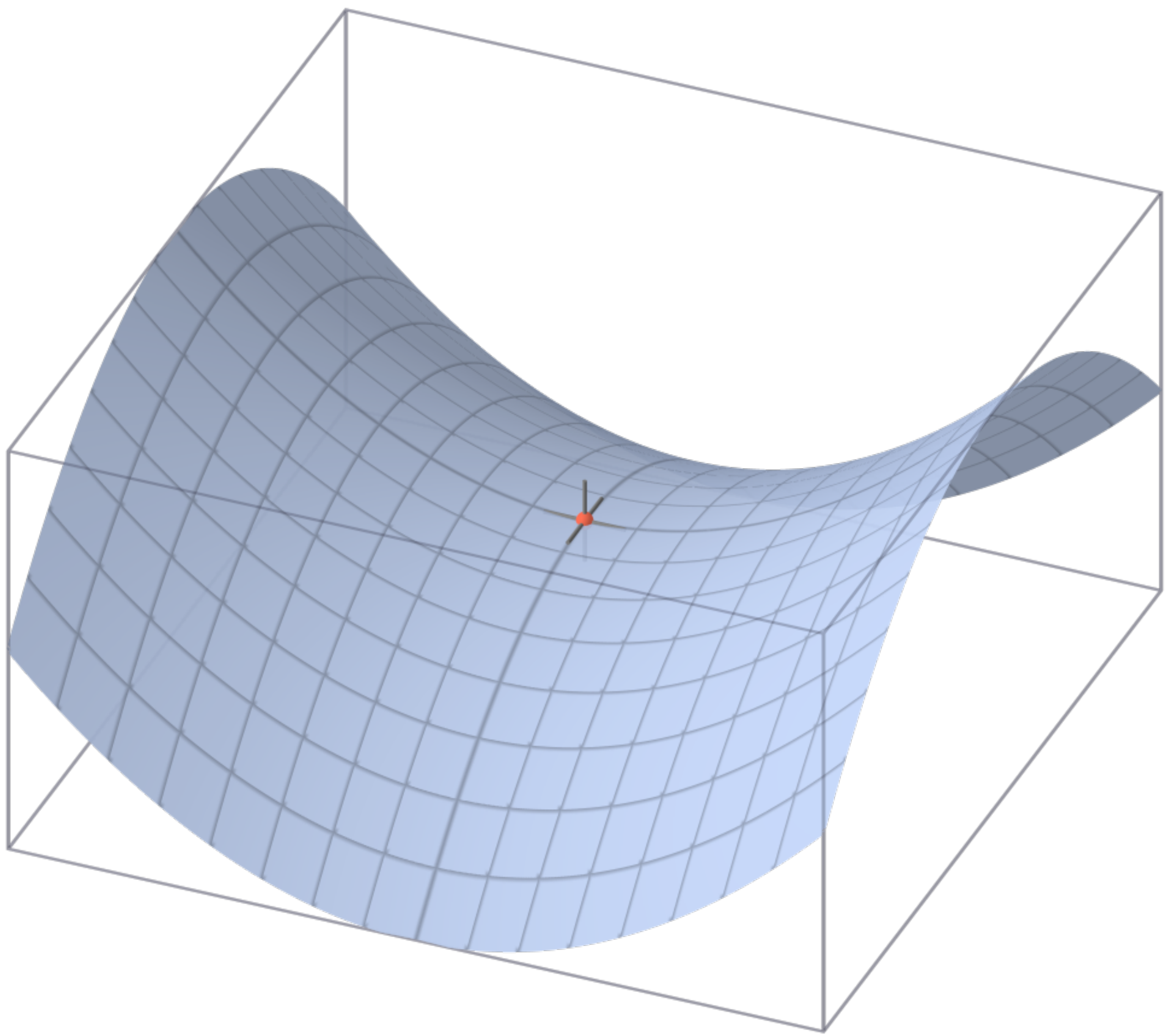
end for

Which image is real?

GAN Demo

https://www.youtube.com/watch?time_continue=16&v=G06dEcZ-QTg&feature=emb_logo

$z = x^2 - y^2$ has a saddle point



Common failure modes of GANs

- The generator outputs only a small number of realistic images, which always fool the discriminator.
- The generator outputs only 1 very realistic image.
- The system oscillates: during training, the generator outputs a single image, but this image changes over time as the discriminator adapts to it.

Wasserstein GANS (WGANs)

- Attempt to improve GANs by changing the loss function, $\mathcal{L}$(D,G)
- The problem is that $\mathcal{L}$ is the likelihood of the data, which is flat almost everywhere if the real data and the generated data do not overlap, as they initially do not.
- So gradient descent/ascent will not work, since the slope is always 0.
- Use a loss function that is not flat.