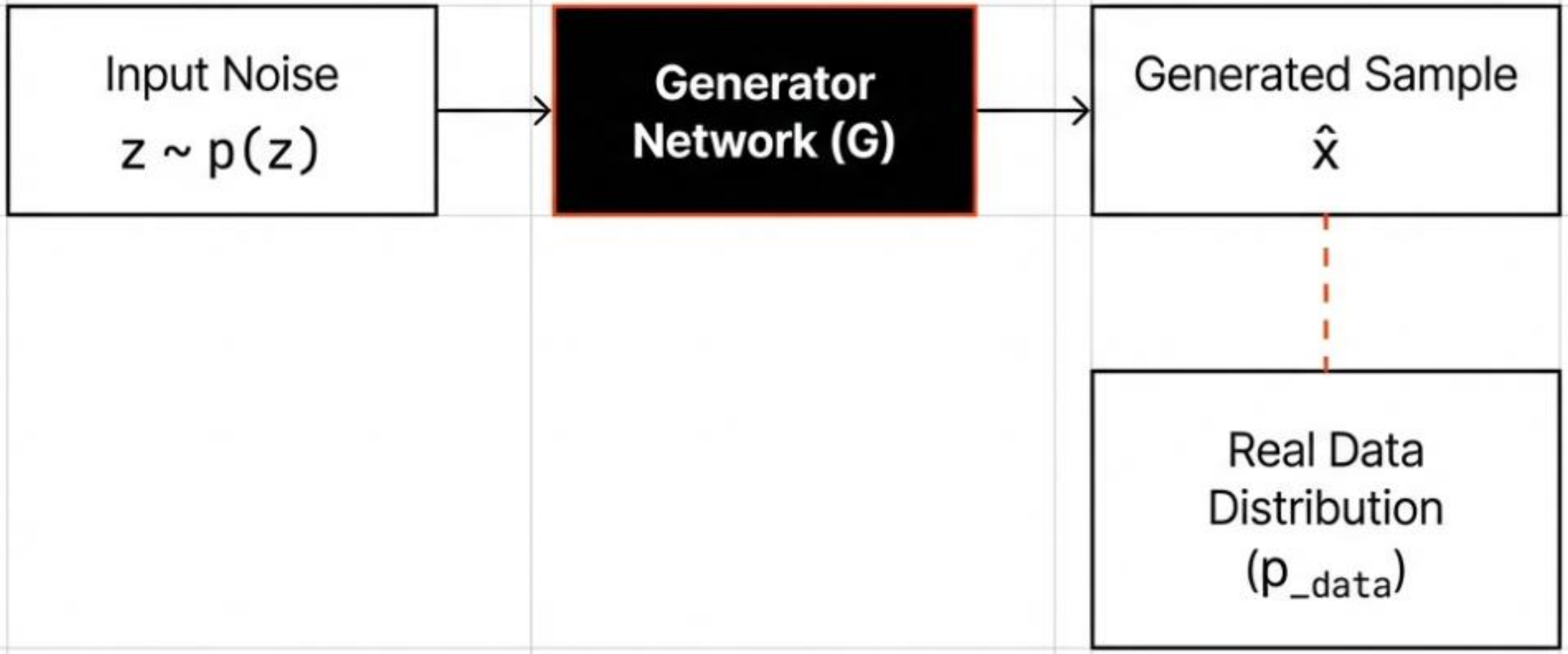
Deep Learning Frameworks

Generative Adversarial Network (GAN), Conditional GAN,
Wasserstein Loss

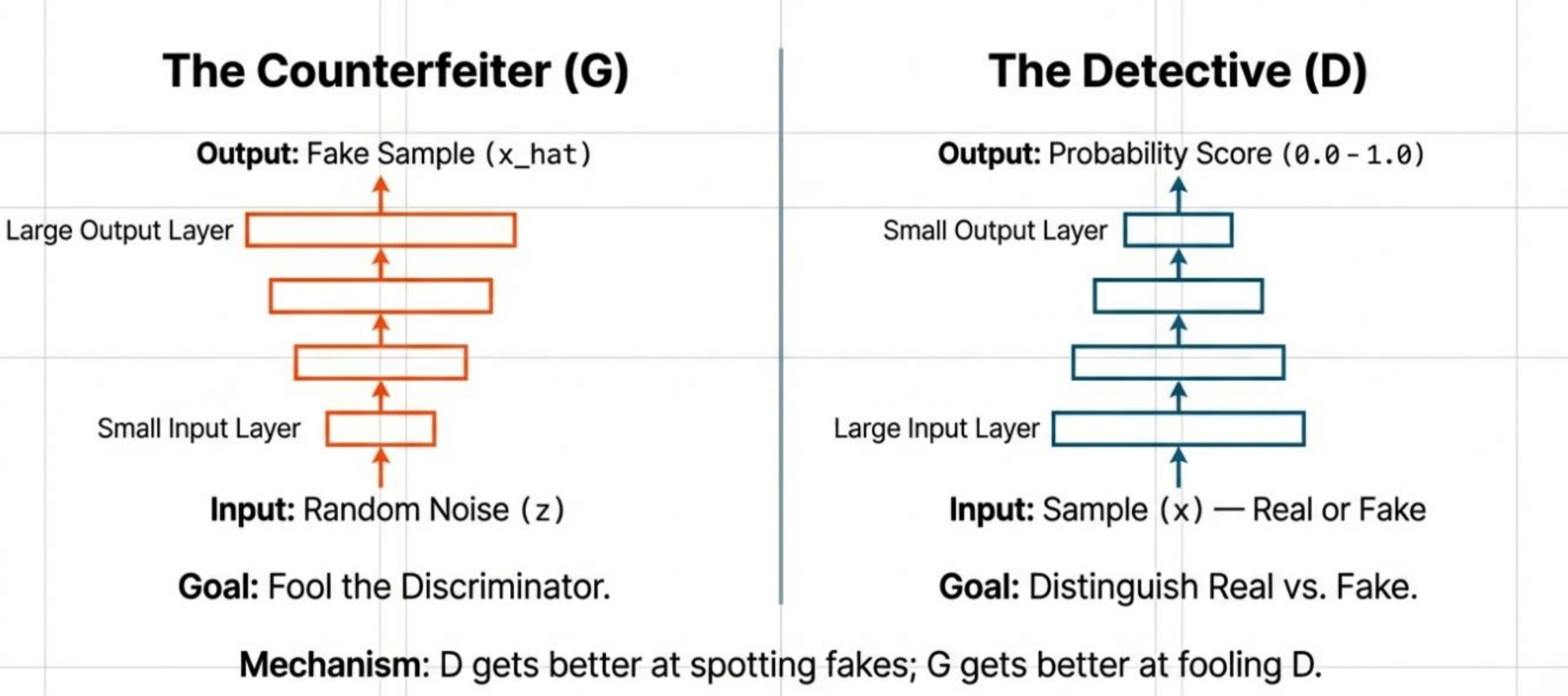
https://tinyurl.com/dlframeworks
https://github.com/sakharamg/DeepLearningFrameworks

Generation

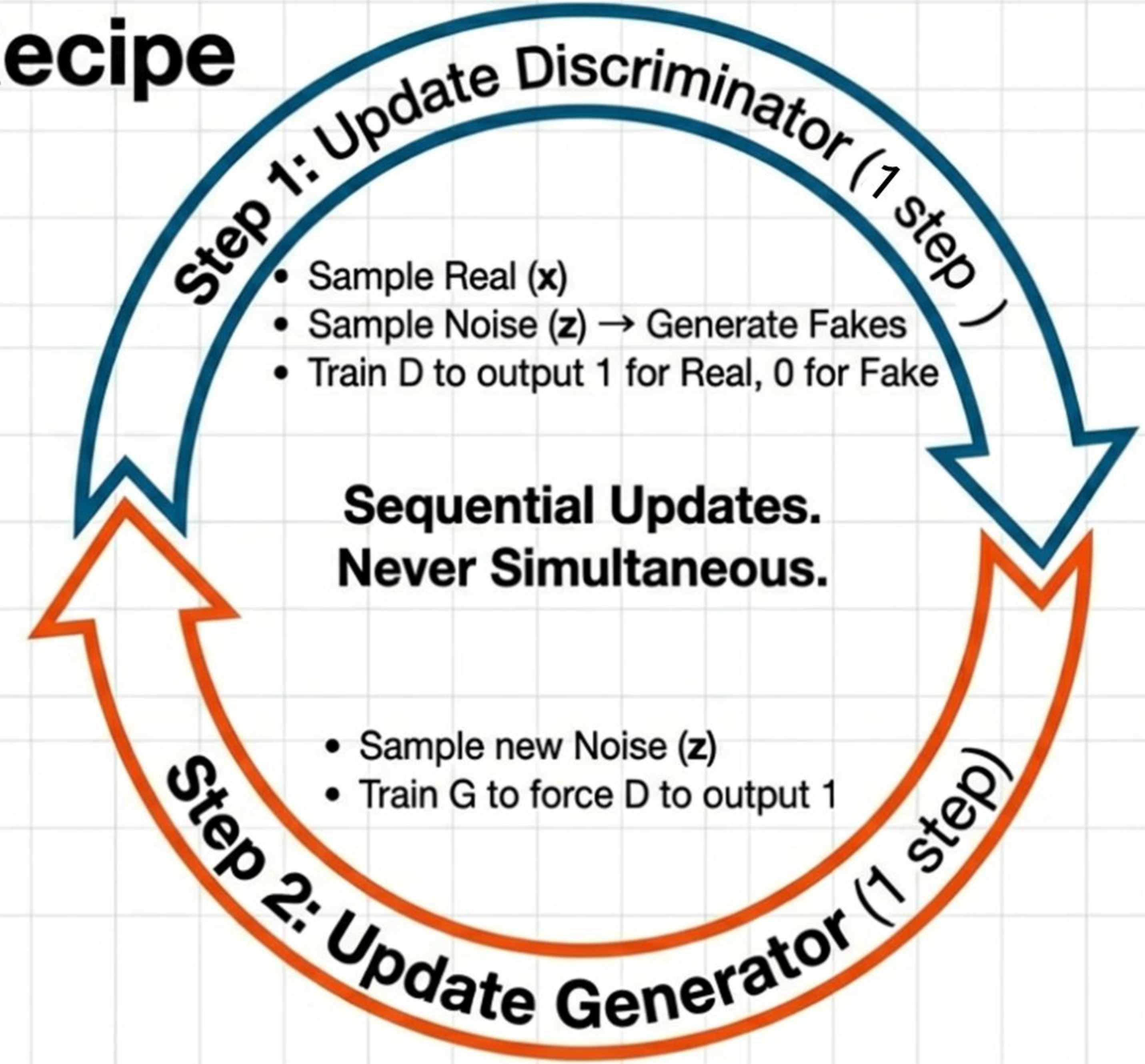
A GAN learns to generate new samples (images, audio, etc.) that mimic a real dataset. Instead of explicitly modeling a probability distribution, the system trains via a game between two neural networks.



Generator and Discriminator



The Training Loop Recipe



The min max objective

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

The Discriminator (D)	**The Generator (G)**
Maximizes the probability of assigning the correct label to both training examples and generated samples. Goal: $D(x) \to 1$ for real, $D(G(z)) \to 0$ for fake.	Minimizes the probability that D is correct. Classic Objective: $\min \log(1 - D(G(z)))$.

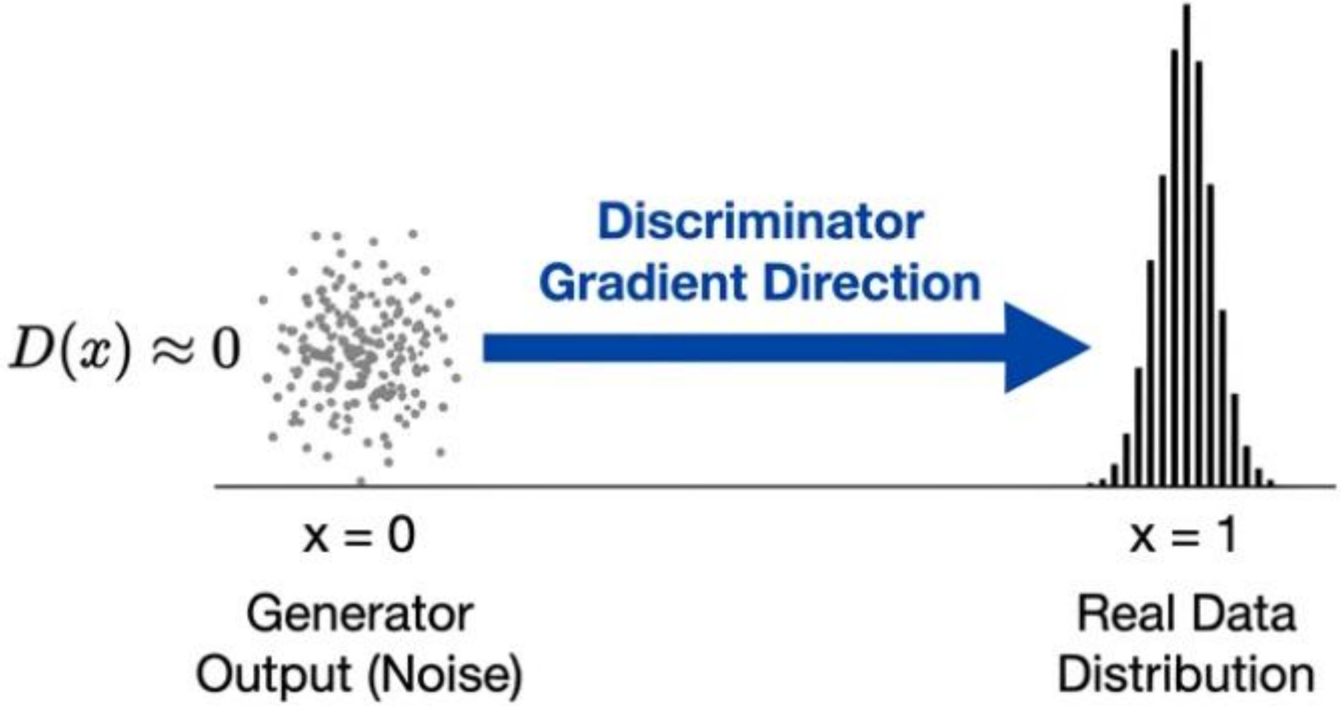
Epoch 0 Context:

- The Generator is incompetent, outputting random noise.
- The Discriminator is easily confident in distinguishing real data from random noise.

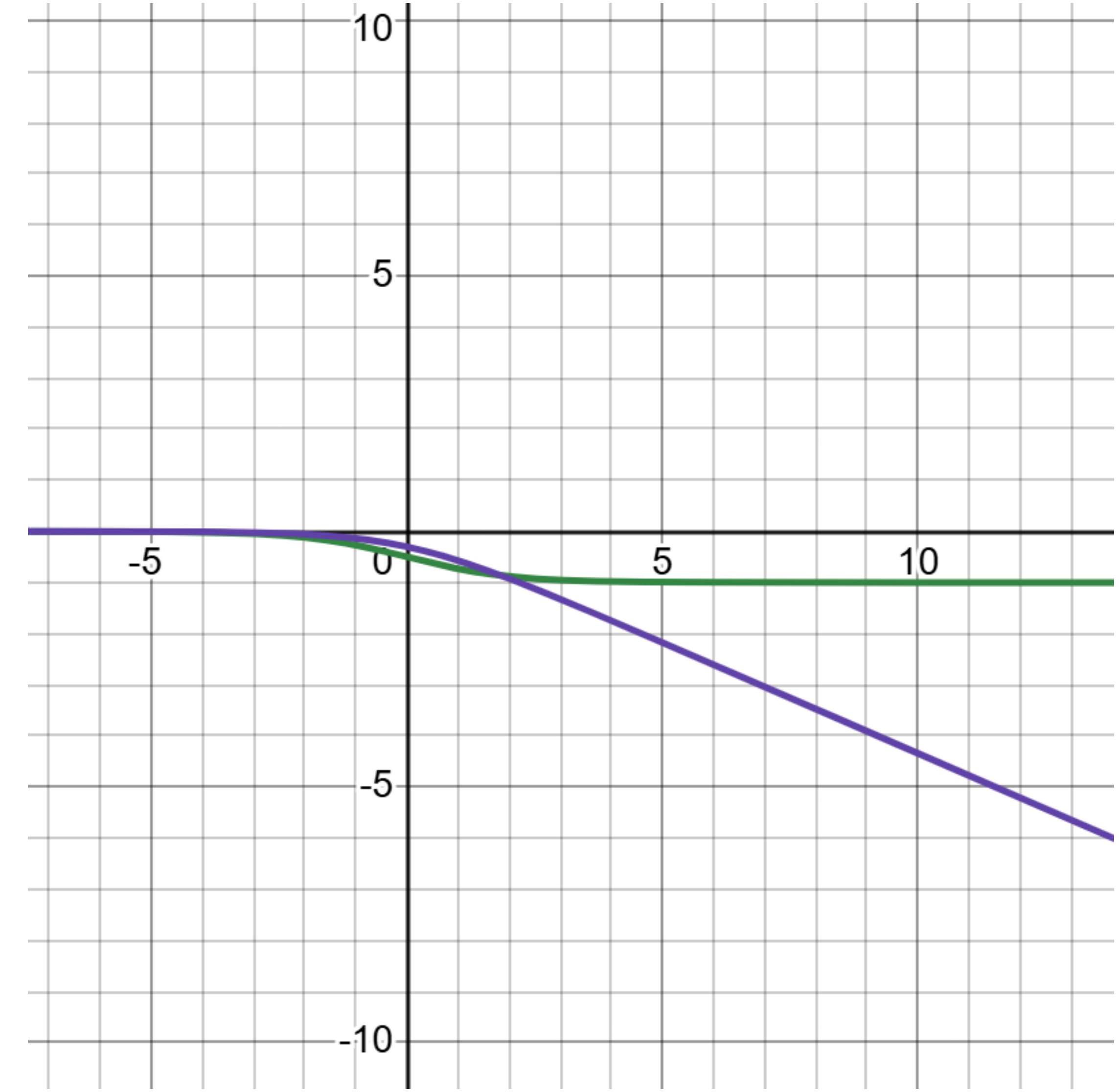
The Key Condition:

For fake samples $x_f = G(z)$, the Discriminator output is near zero.

$$D(G(z)) \approx 0$$



$d = \frac{1}{\left(1 + e^{-g}\right)}$



$$\mathbb{E}[\log(1 - D(G(z)))]$$

https://www.desmos.com/calculator/7jxtfutmq2

Fix

Goodfellow's heuristic: Instead of minimizing the likelihood of being caught, maximize the likelihood of deception.

~~Minimax (Classic)~~

$$\cancel{\min_G \mathbb{E}[\log(1 - D(G(z)))]}$$

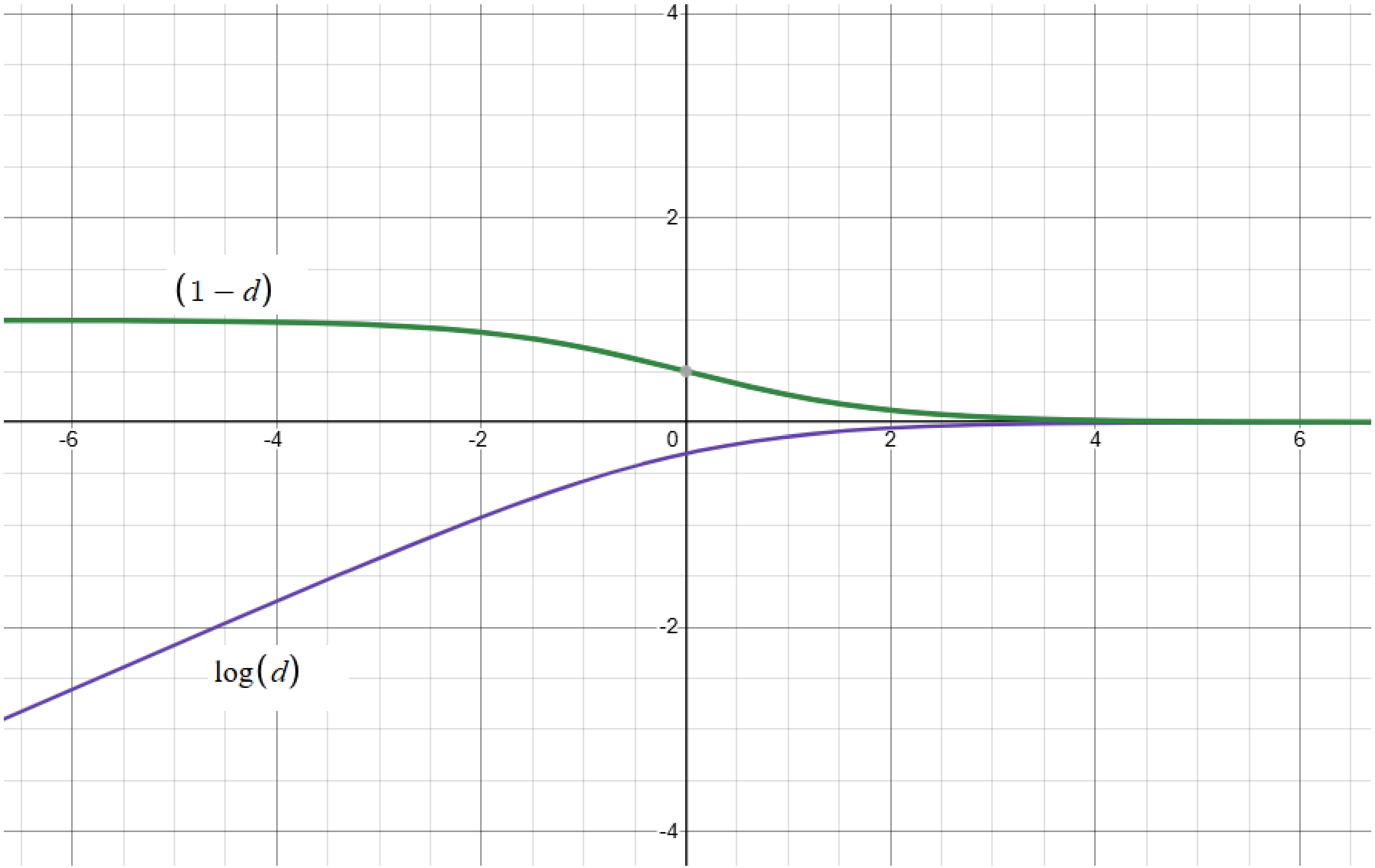
Non-Saturating (Heuristic)

$$\max_G \mathbb{E}[\log D(G(z))]$$

Equivalently: minimize $-\log D(G(z))$

$$d = \frac{1}{\left(1 + e^{-g}\right)}$$

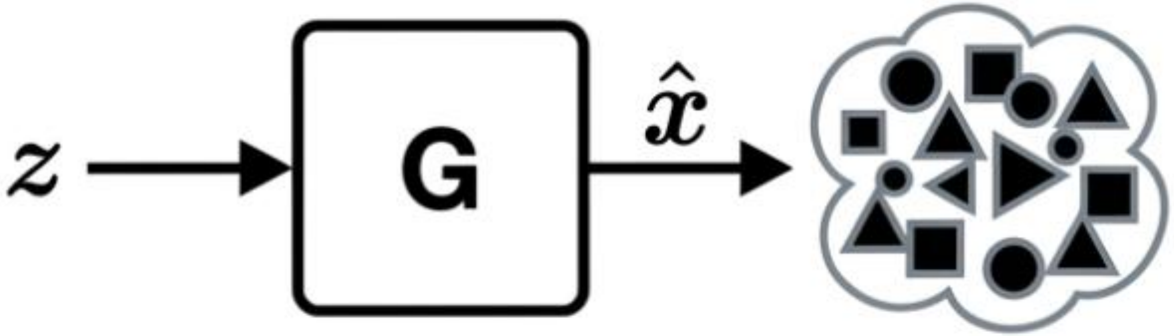
$$\log D(G(z))$$



https://www.desmos.com/calculator/ho6slpznc2

Conditional GAN

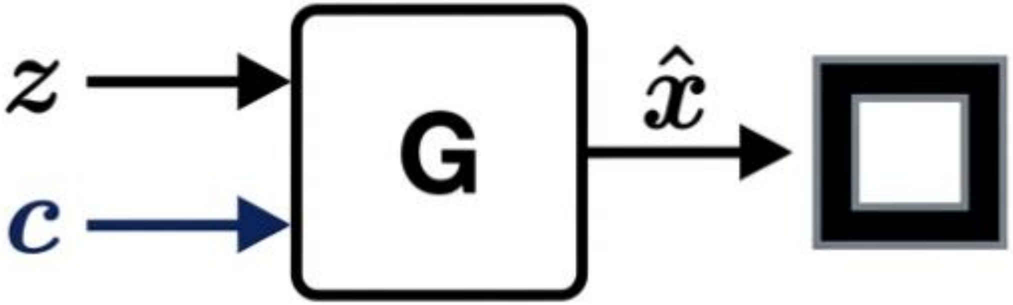
Standard GAN



$$\hat{x} = G(z)$$

Learns to map a latent distribution to the data marginal distribution. Output is stochastic and uncontrolled.

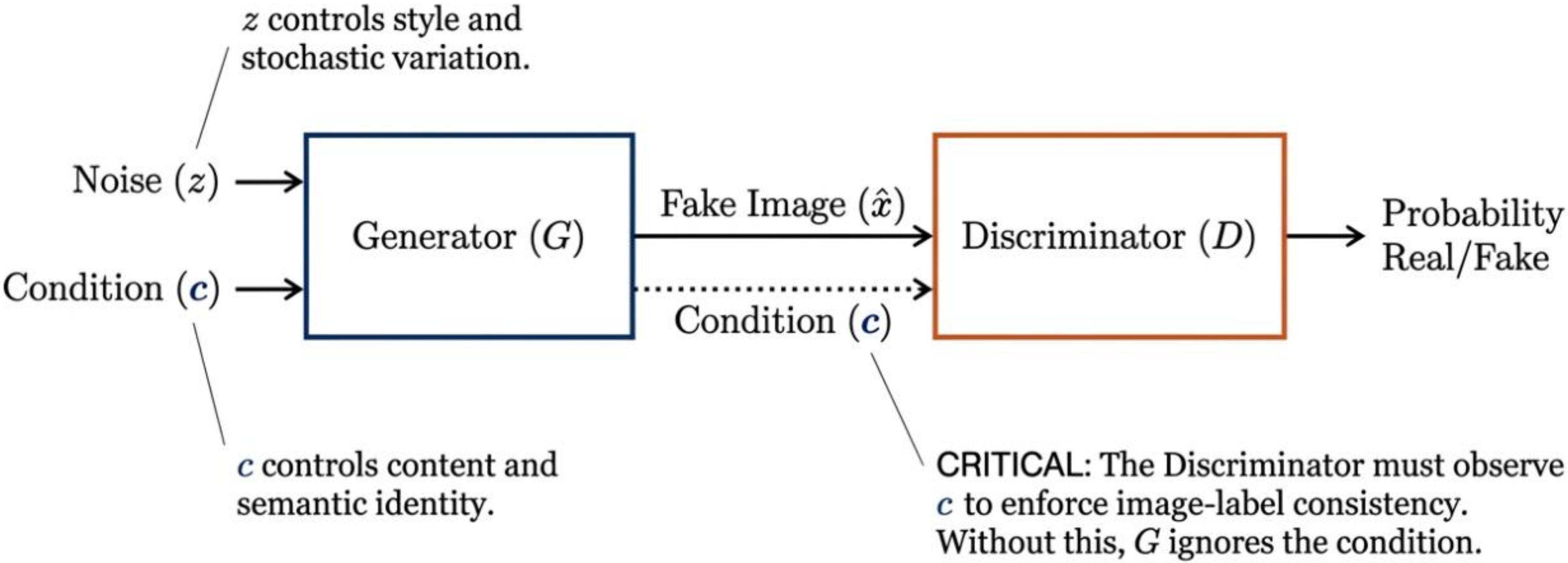
Conditional GAN



$$\hat{x} = G(z, \boldsymbol{c})$$

Extends the framework by conditioning on external information $\boldsymbol{c}$. The model learns the conditional distribution $P(\boldsymbol{X}|\boldsymbol{c})$.

Injecting Condition



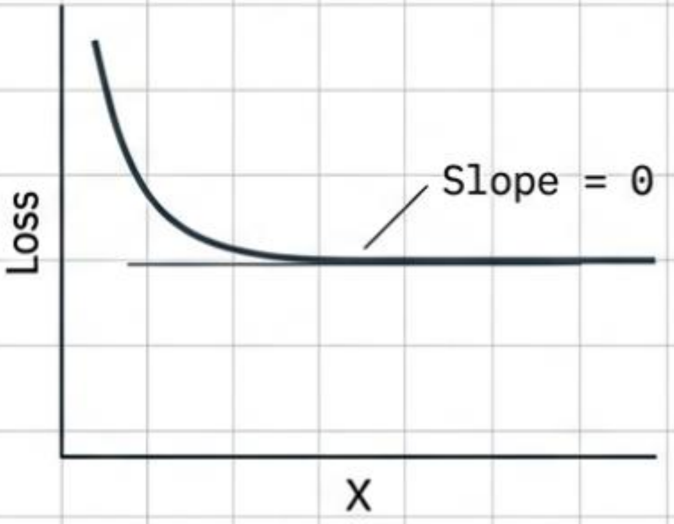
Lab

https://tinyurl.com/dlframeworks
https://github.com/sakharamg/DeepLearningFrameworks

Instability

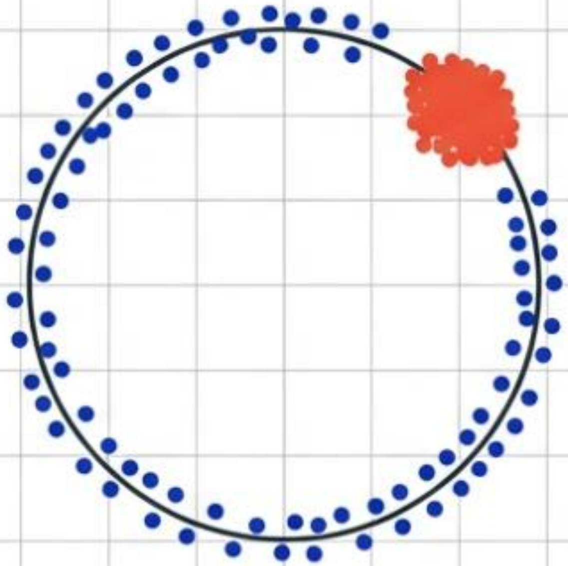
The delicate balance between G and D leads to three primary instabilities.

Vanishing Gradients



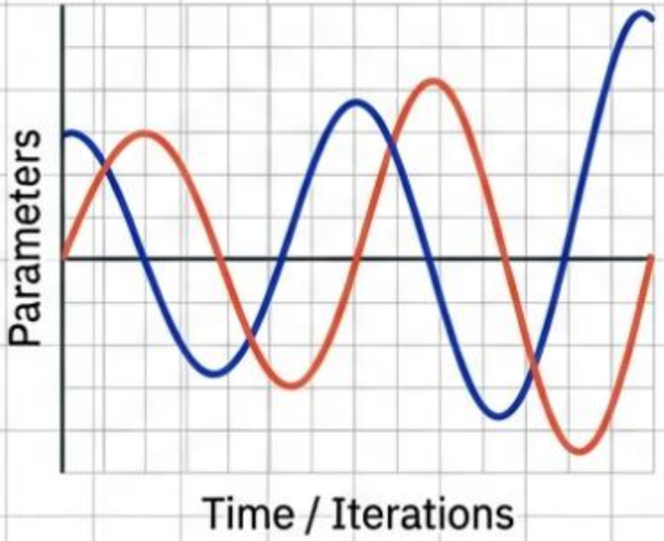
If D is too perfect, it distinguishes real/fake with 100% confidence. The loss function flattens, G receives gradient 0, and learning stops.

Mode Collapse



G finds a single sample that fools D and repeats it endlessly, ignoring the diversity of the real distribution.

Oscillation



The parameters never settle. The players chase each other in circles, preventing convergence.

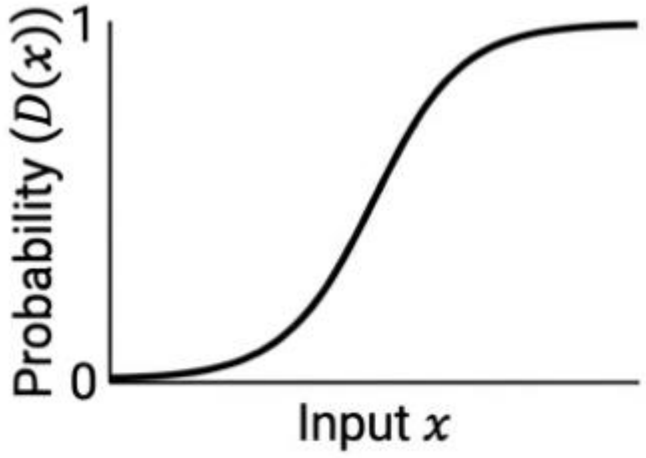
The Cure: From Classifier to Critic

Vanilla GAN

Discriminator is a Classifier.

Output: Probability $D(x) \in (0, 1)$

Goal: Distinguish Real vs. Fake.

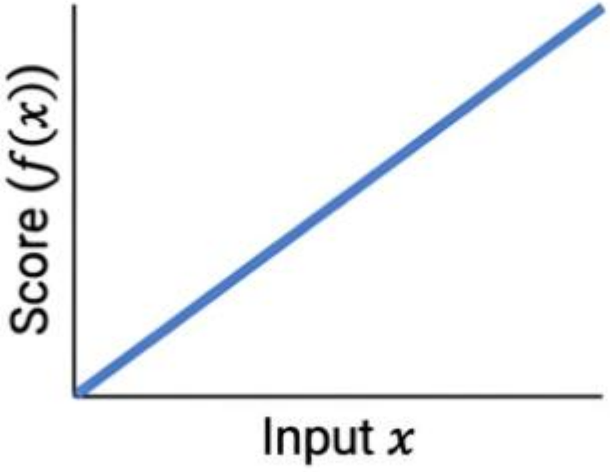


Wasserstein GAN

Discriminator is a Critic.

Output: Scalar Score $f(x) \in \mathbb{R}$

Goal: Measure Earth Mover's Distance.



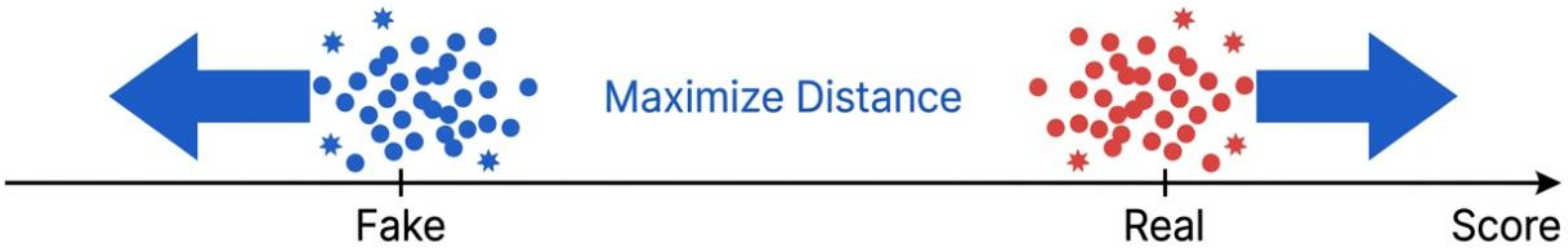
The Critic learns a smooth score function indicating "how real" an image is, rather than a binary probability.

The WGAN Objective

The Critic maximizes the gap between the scores of real images and fake images. This provides a continuous signal of progress even when distributions don't overlap.

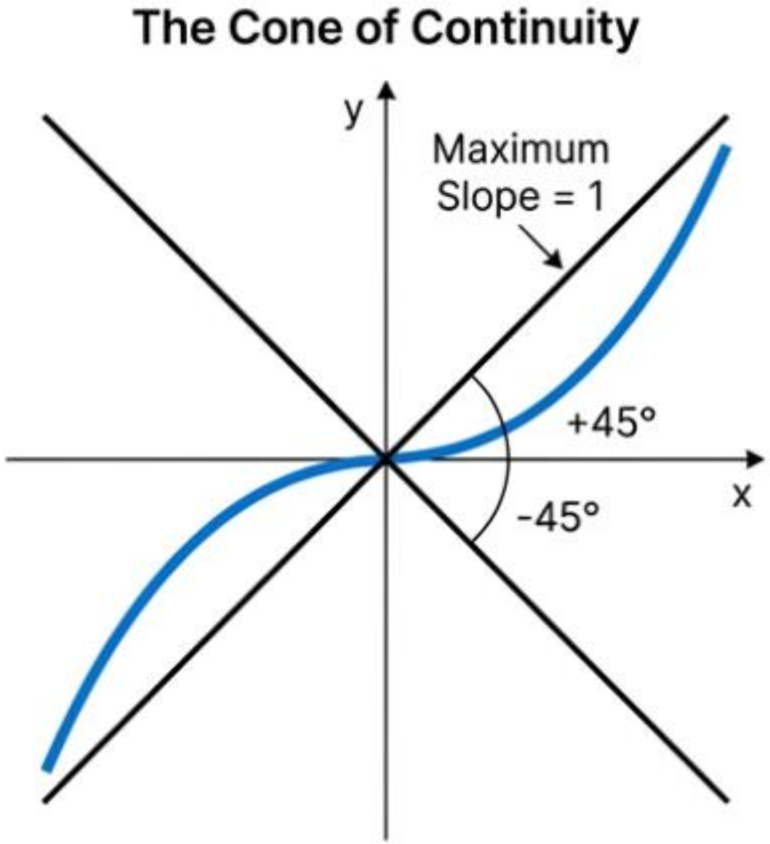
$$\max_{D} D \left[\mathrm{E}_{x\sim\mathrm{real}}[f(x)] - \mathrm{E}_{x\sim\mathrm{fake}}[f(x)] \right]$$

$$L_G = -\mathrm{E}_{x\sim\mathrm{fake}}[f(x)]$$



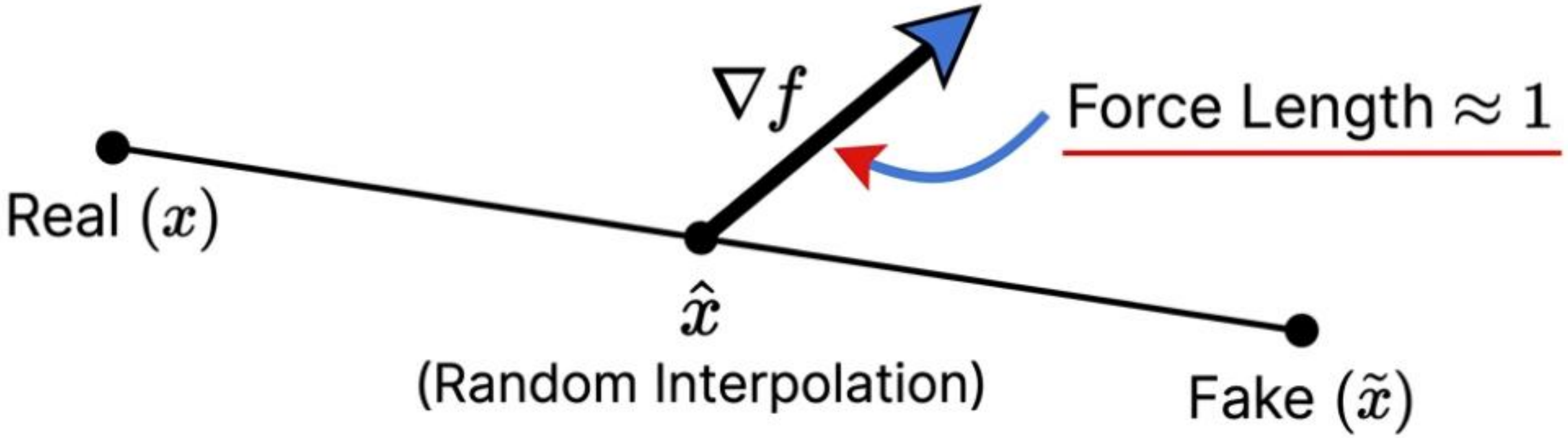
For the WGAN theory to hold, the Critic function f must be 1-Lipschitz. The gradient cannot change abruptly.

$$|f(x_1) - f(x_2)| \leq |x_1 - x_2|$$



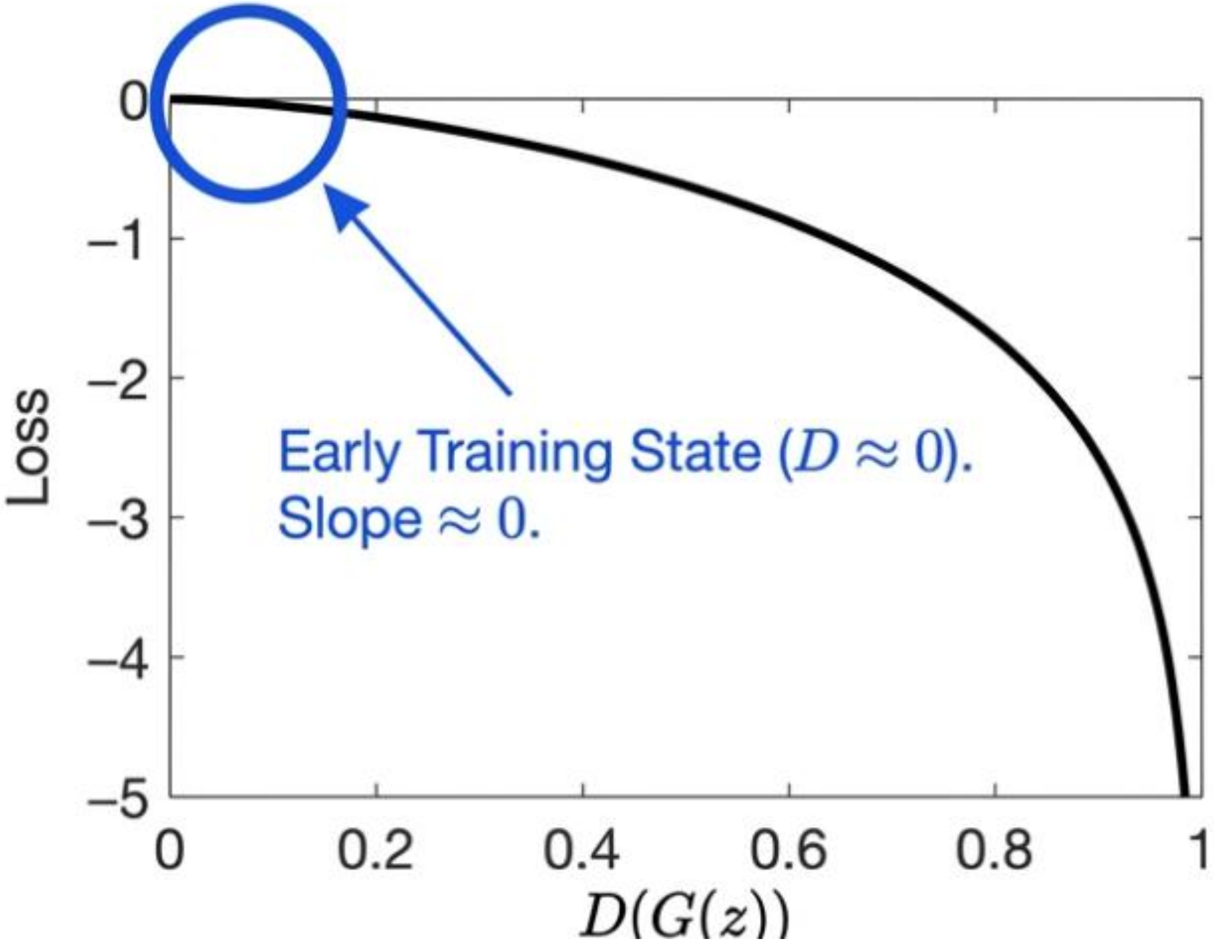
We penalize the gradient norm for deviating from 1.

$$\lambda * \mathbb{E}_{\hat{x}}\left[\left(\|\nabla_{\hat{x}} f(\hat{x})\|_2 - 1\right)^2\right]$$



Thank You

Appendix



Generator Loss Function:

$J = \log(1 - D(G(z)))$

If $D(G(z)) \approx 0$, the loss value is $\log(1) = 0$.

Numerically stable, but the gradient (slope) is effectively flat.

Optimization requires a slope to slide down. Here, there is no slope.

To understand the gradient behavior, we analyze the derivative with respect to the discriminator's logit (a).

The Logit (a): Let a be the pre-activation output of the discriminator.

The Sigmoid Output (D):

$$D(x) = \sigma(a) = \frac{1}{1 + e^{-a}}$$

The Patient: The Minimax Generator Objective:

$$L_G^{minimax} = \log(1 - \sigma(a))$$

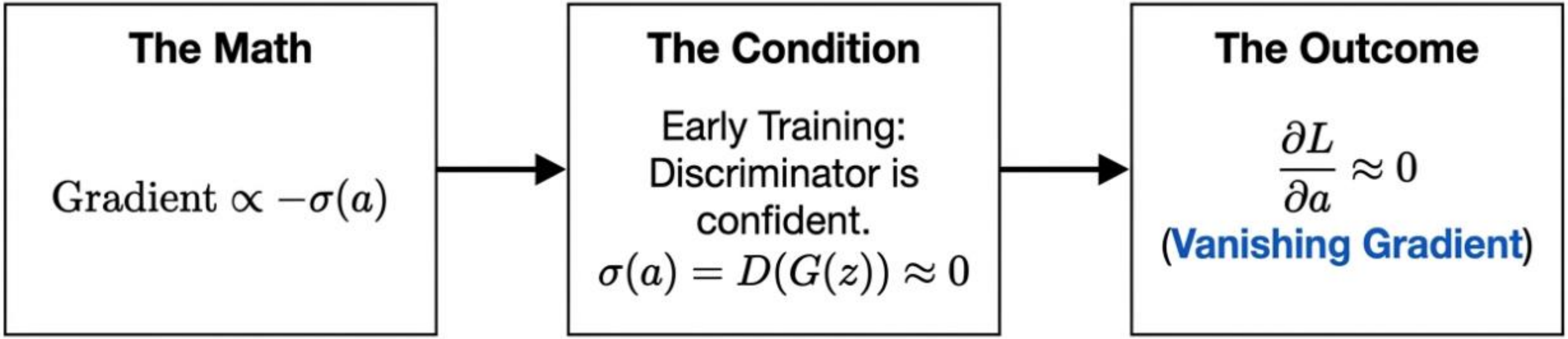
$$\frac{\partial L}{\partial a} = \frac{\partial}{\partial a} \log(1 - \sigma(a))$$

$$\text{Chain Rule: } = \frac{1}{1-\sigma(a)} \cdot \frac{\partial}{\partial a}(1 - \sigma(a))$$

$$= \frac{1}{1-\sigma(a)} \cdot (-\sigma(a)(1 - \sigma(a)))$$

Gradient magnitude $\propto \sigma(a)$

$$\text{The Result: } \boldsymbol{\frac{\partial L_G^{\text{minimax}}}{\partial a} = -\sigma(a)}$$



Conclusion: When the Discriminator is too successful, the learning signal for the **Generator** evaporates. The Generator stops learning exactly when it needs to learn the most.

Fix

Goodfellow's heuristic: Instead of minimizing the likelihood of being caught, maximize the likelihood of deception.

~~**Minimax (Classic)**~~

$$\cancel{\min_G \mathbb{E}[\log(1 - D(G(z)))]}$$

Non-Saturating (Heuristic)

$$\max_G \mathbb{E}[\log D(G(z))]$$

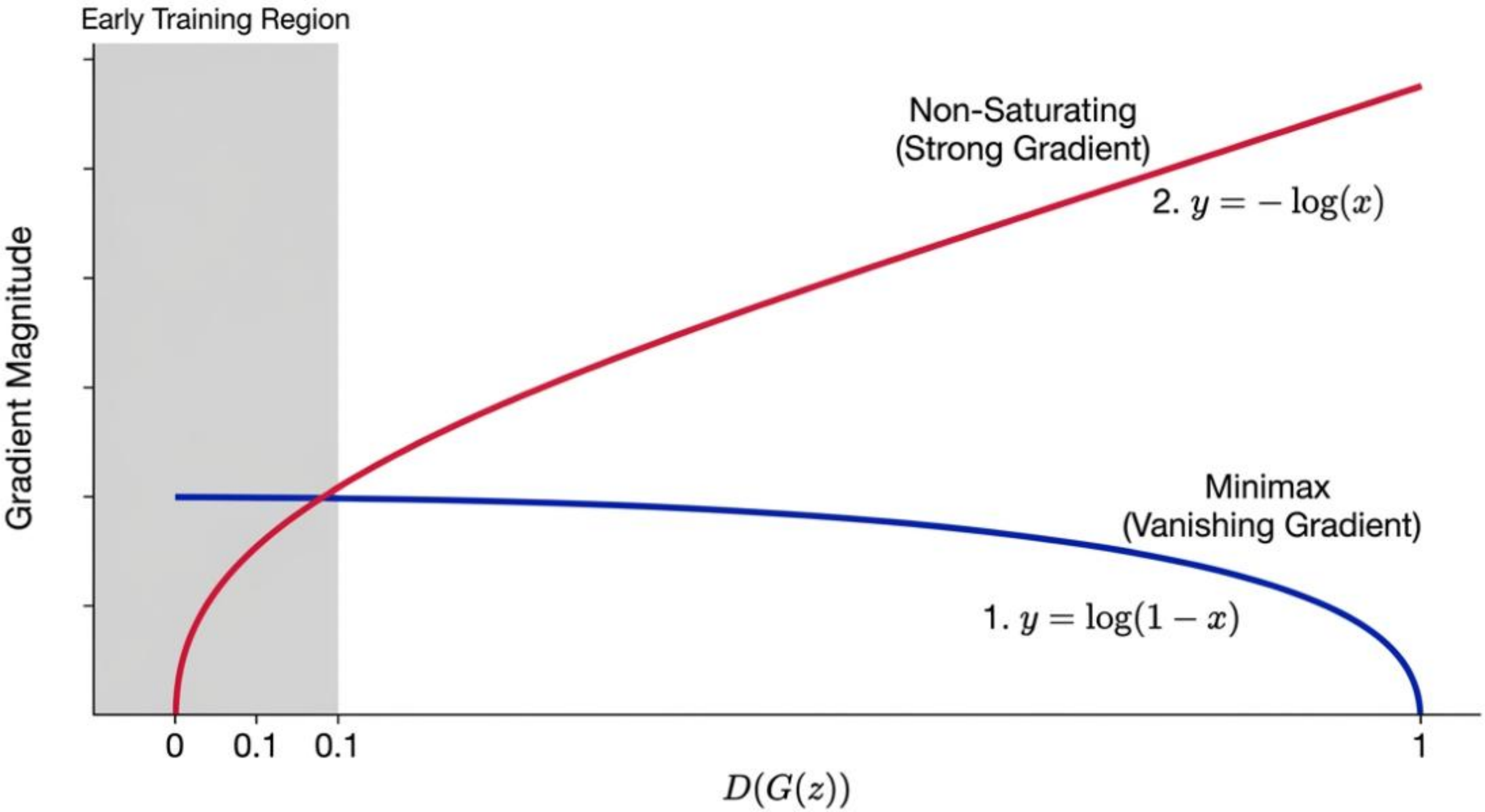
Equivalently: minimize $-\log D(G(z))$

New Objective: $L_G^{NS} = -\log(\sigma(a))$

Derivative: $\frac{\partial L}{\partial a} = \frac{\partial}{\partial a}(-\log(\sigma(a)))$

$$\boxed{\frac{\partial L_G^{NS}}{\partial a} = \sigma(a) - 1}$$

Gradient magnitude depends on $(\sigma(a) - 1)$



Early Training Region
Gradient Magnitude
Non-Saturating
(Strong Gradient)
2. $y = -\log(x)$
Minimax
(Vanishing Gradient)
1. $y = \log(1-x)$
0
0.1
0.1
1
$D(G(z))$