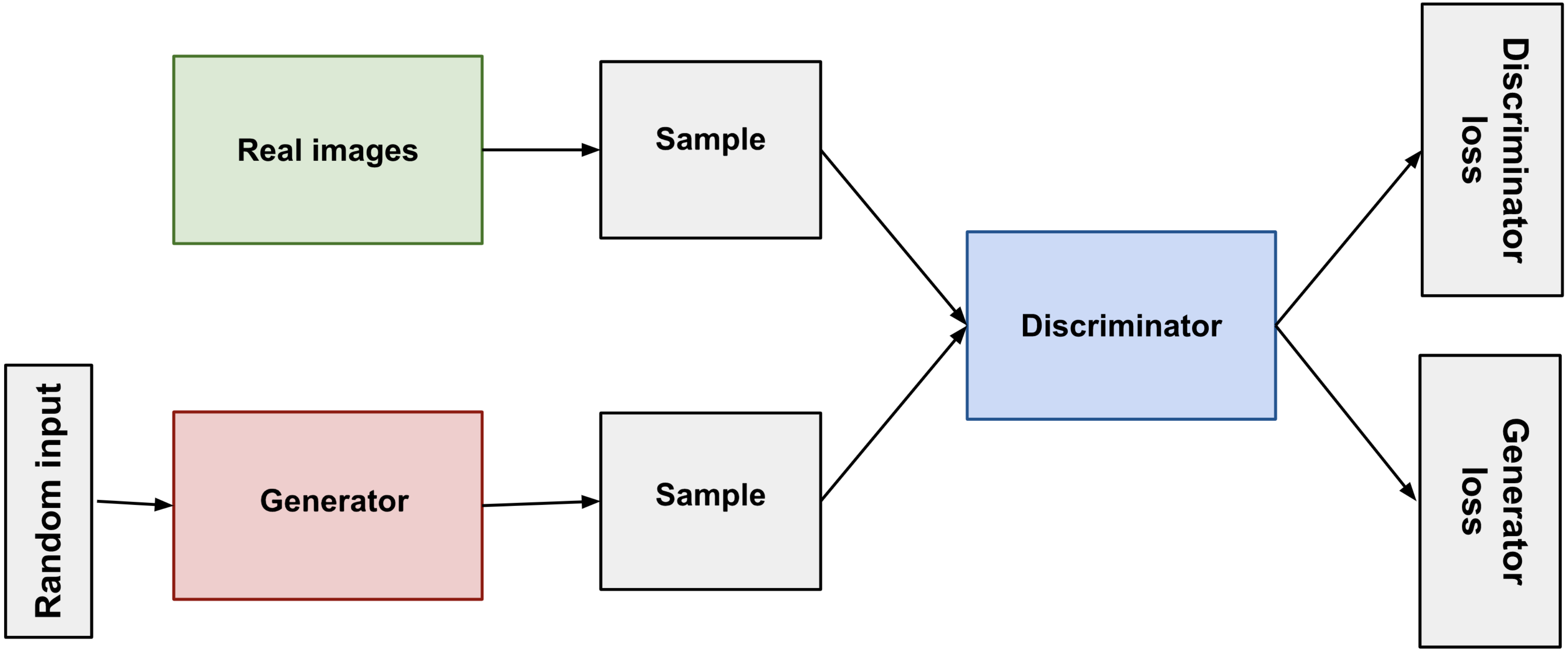
Generative Adversarial Networks

Sergey Kim

GAN – Generative Adversarial Network

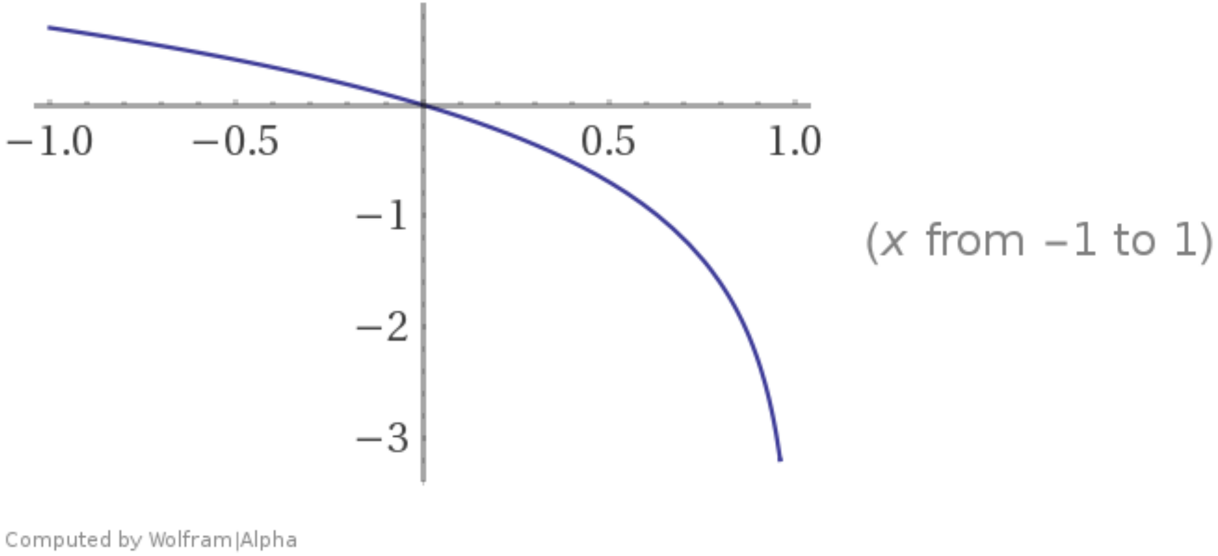


GAN – Generative Adversarial Network

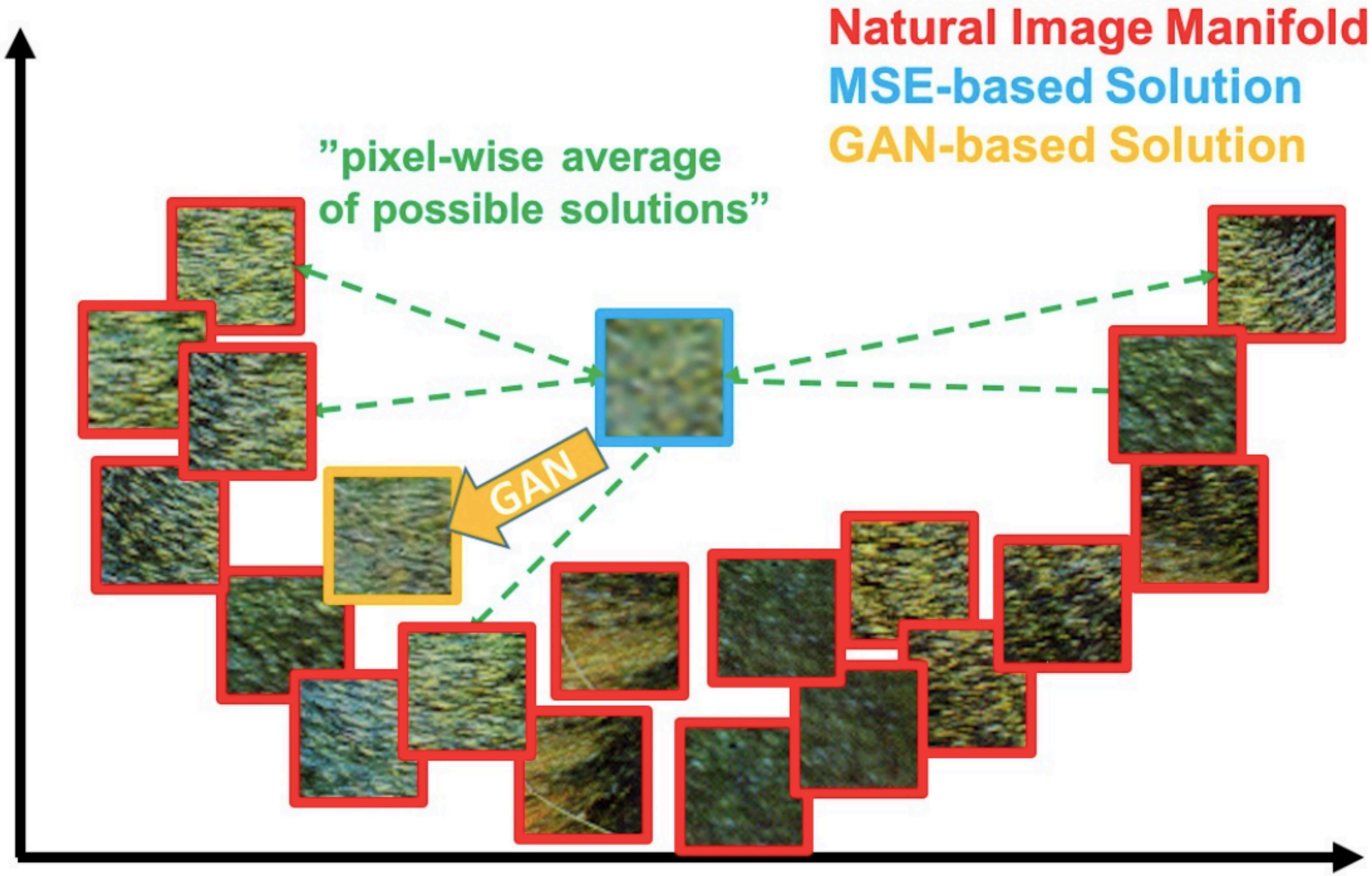
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Smart Discriminator:

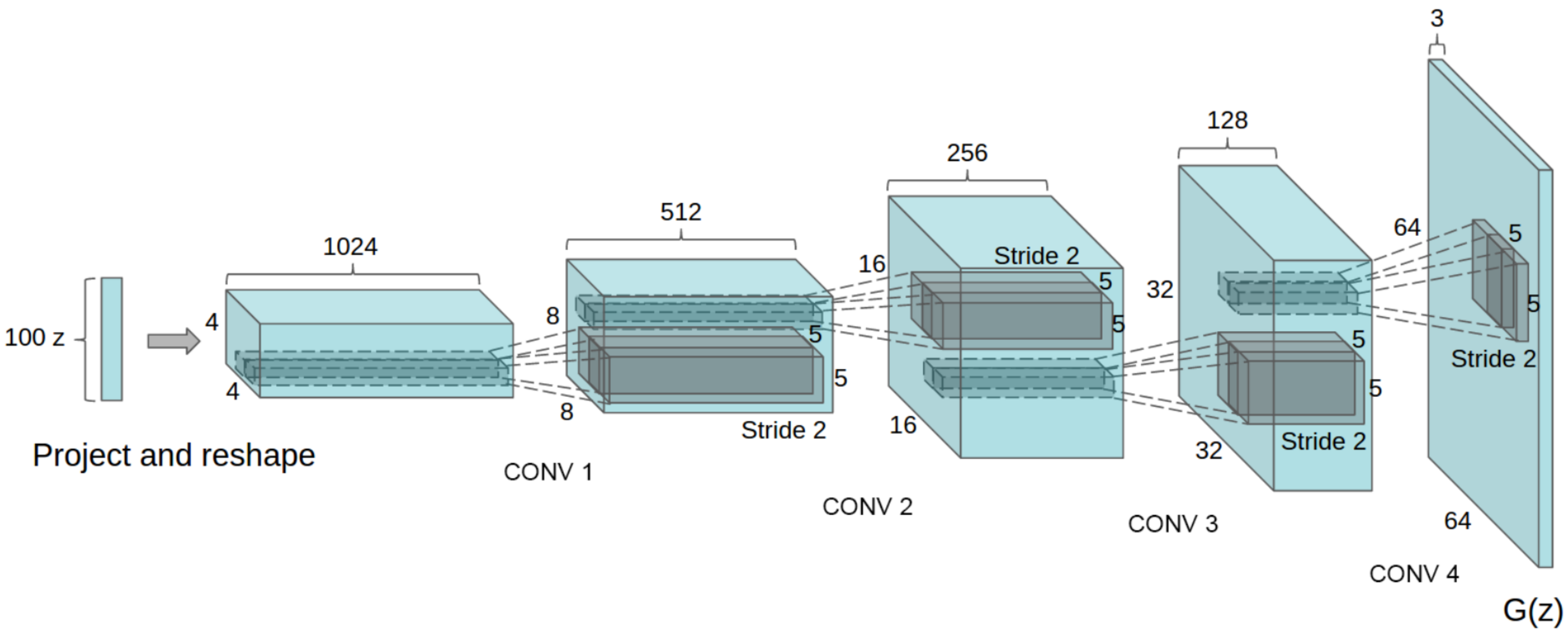
D(x) = 1, D(G(z)) = 0



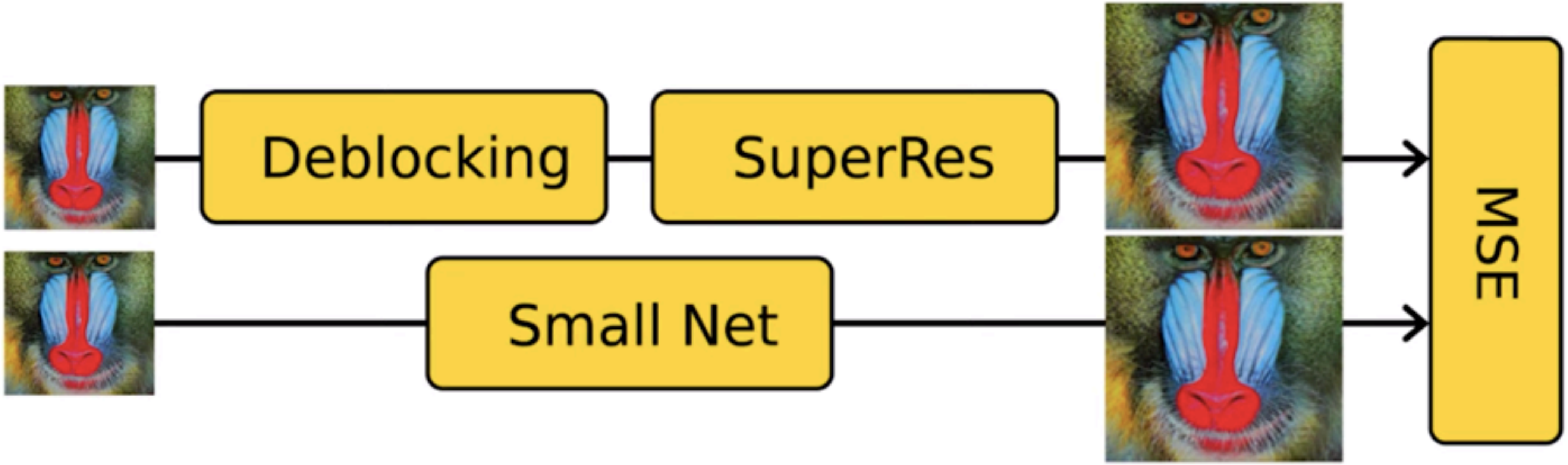
GAN – Generative Adversarial Network



DCGAN – Deep Convolutional GAN



Dark Knowledge



Tricks

Modify the Loss Function: *max log(D)* instead of *min log(1 - D)*

Sample from a Gaussian Distribution, not Uniform

Adam for G, SGD for D

Add some gaussian noise to inputs to D

Use Dropouts in G both test and train phase

GAN Loss

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

$$\begin{aligned} V(G,D) &= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x}))dx + \int_z p_{\boldsymbol{z}}(\boldsymbol{z}) \log(1 - D(g(\boldsymbol{z})))dz \\ &= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x})) + p_g(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))dx \end{aligned}$$

Discriminator as a Loss Function

$$D_G^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})}$$

$$\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{P_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x}\sim p_g}\left[\log \frac{p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right]$$

$$C(G) = -\log(4) + KL\left(p_{\text{data}} \left\| \frac{p_{\text{data}} + p_g}{2}\right.\right) + KL\left(p_g \left\| \frac{p_{\text{data}} + p_g}{2}\right.\right)$$

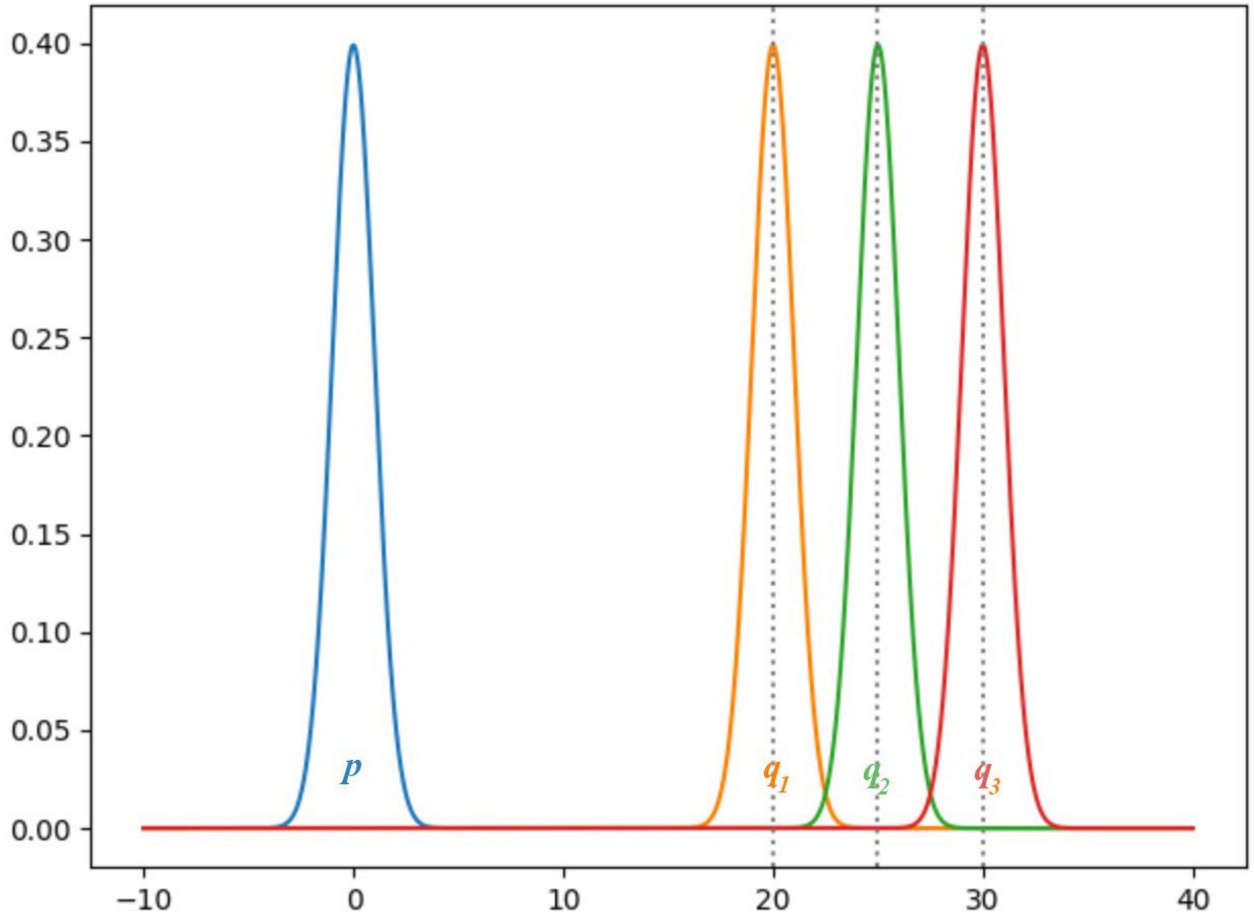
$$C(G) = -\log(4) + 2 \cdot JSD\left(p_{\text{data}} \| p_g\right)$$

Distance axioms

Let d is our distance function.

1) $d(x, y) > 0$ ($d(x, y) = 0 \Leftrightarrow x = y$)
2) $d(x, y) = d(y, x)$
3) $d(x, z) \leq d(x, y) + d(y, z)$

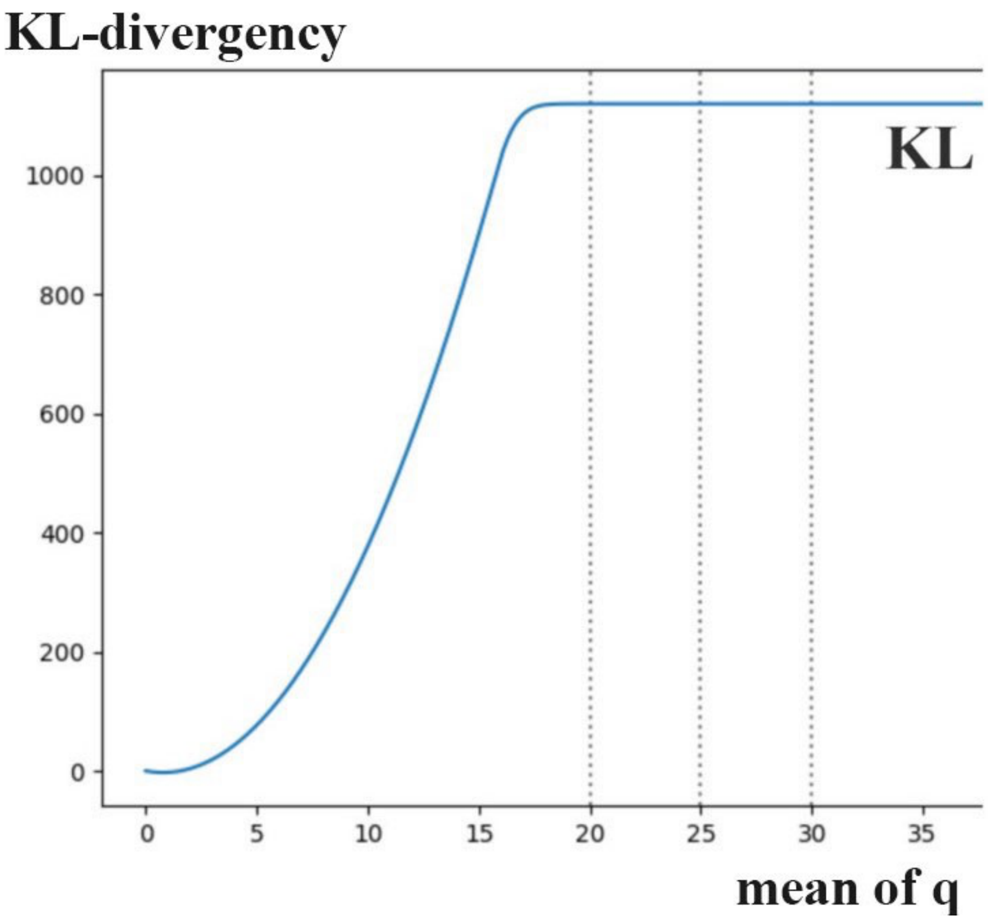
JSD – Jensen-Shannon Divergence

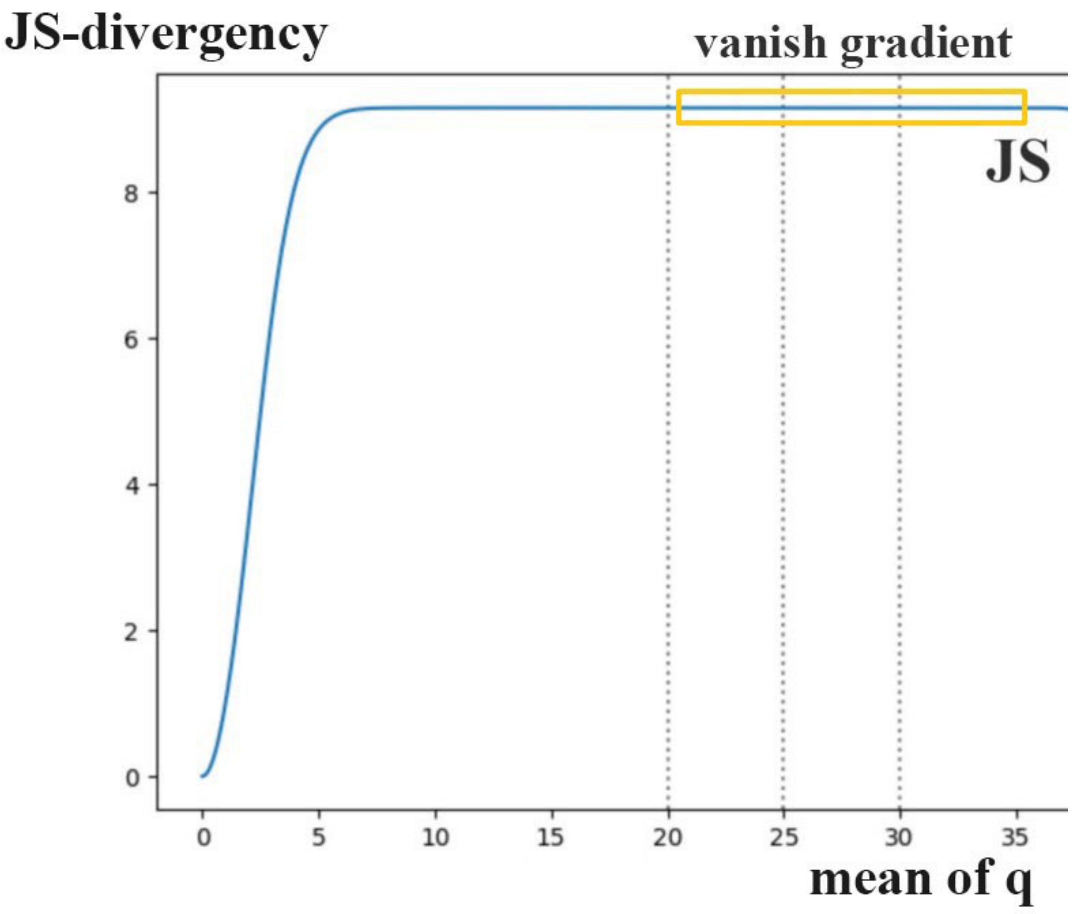


$$KL(\mathbb{P}_r \| \mathbb{P}_g) = \int \log\left(\frac{P_r(x)}{P_g(x)}\right) P_r(x) d\mu(x)$$

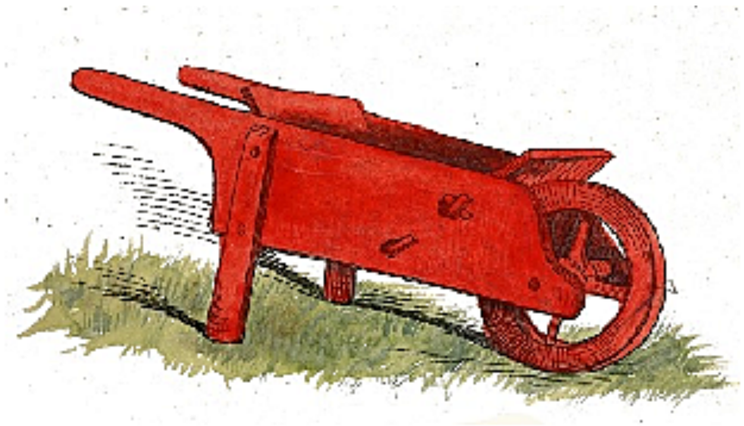
$$JS(\mathbb{P}_r, \mathbb{P}_g) = KL(\mathbb{P}_r \| \mathbb{P}_m) + KL(\mathbb{P}_g \| \mathbb{P}_m)$$

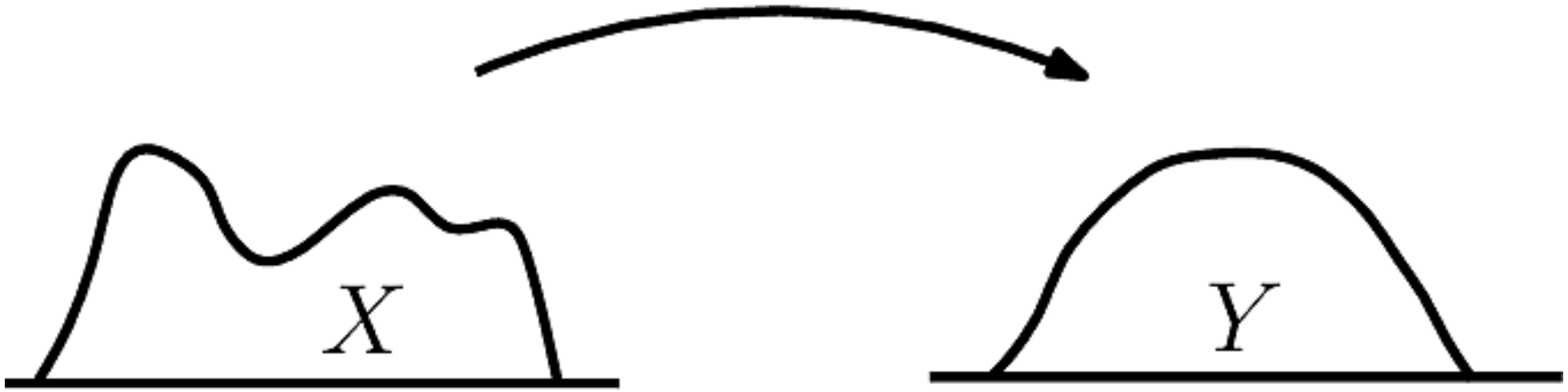
JSD – Jensen-Shannon Divergence





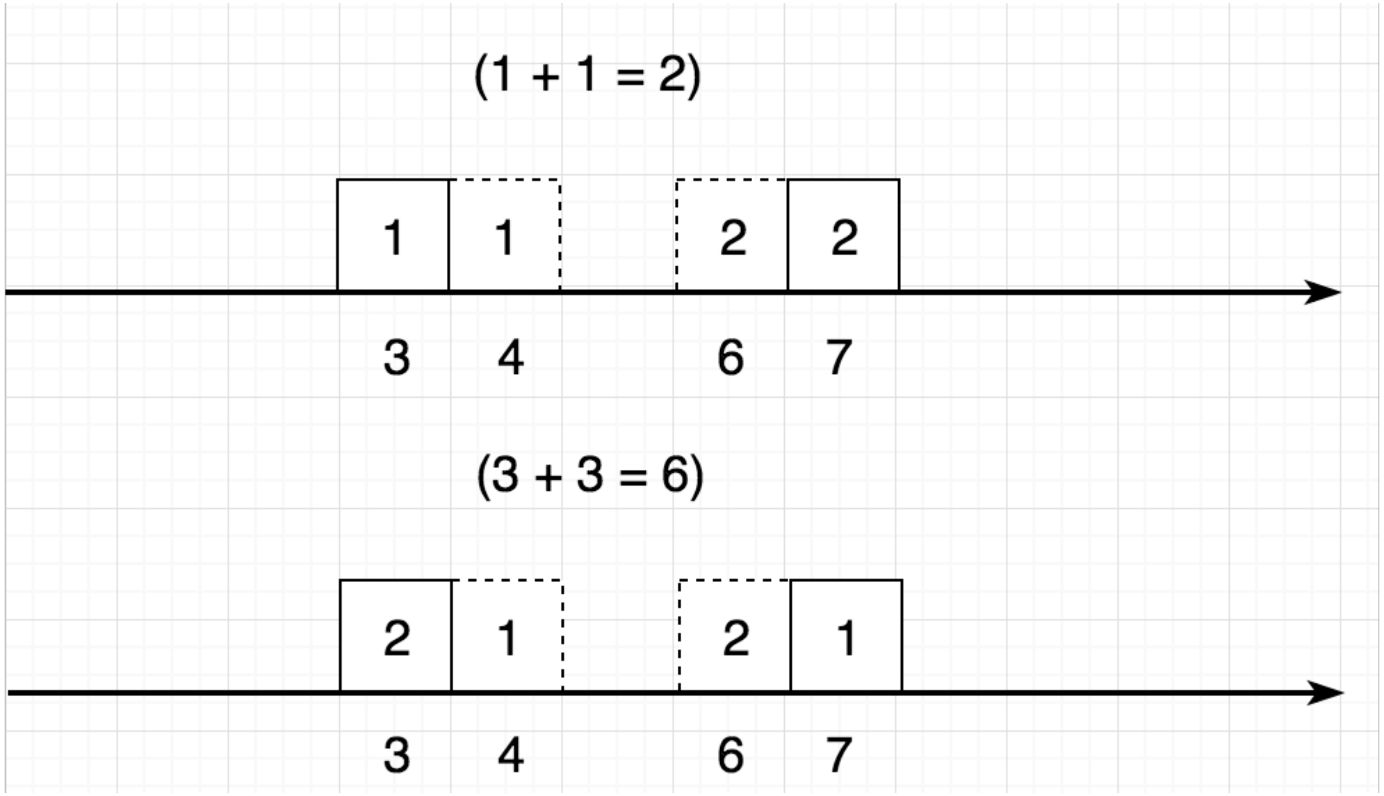
Earth Mover’s (Wasserstein) Distance



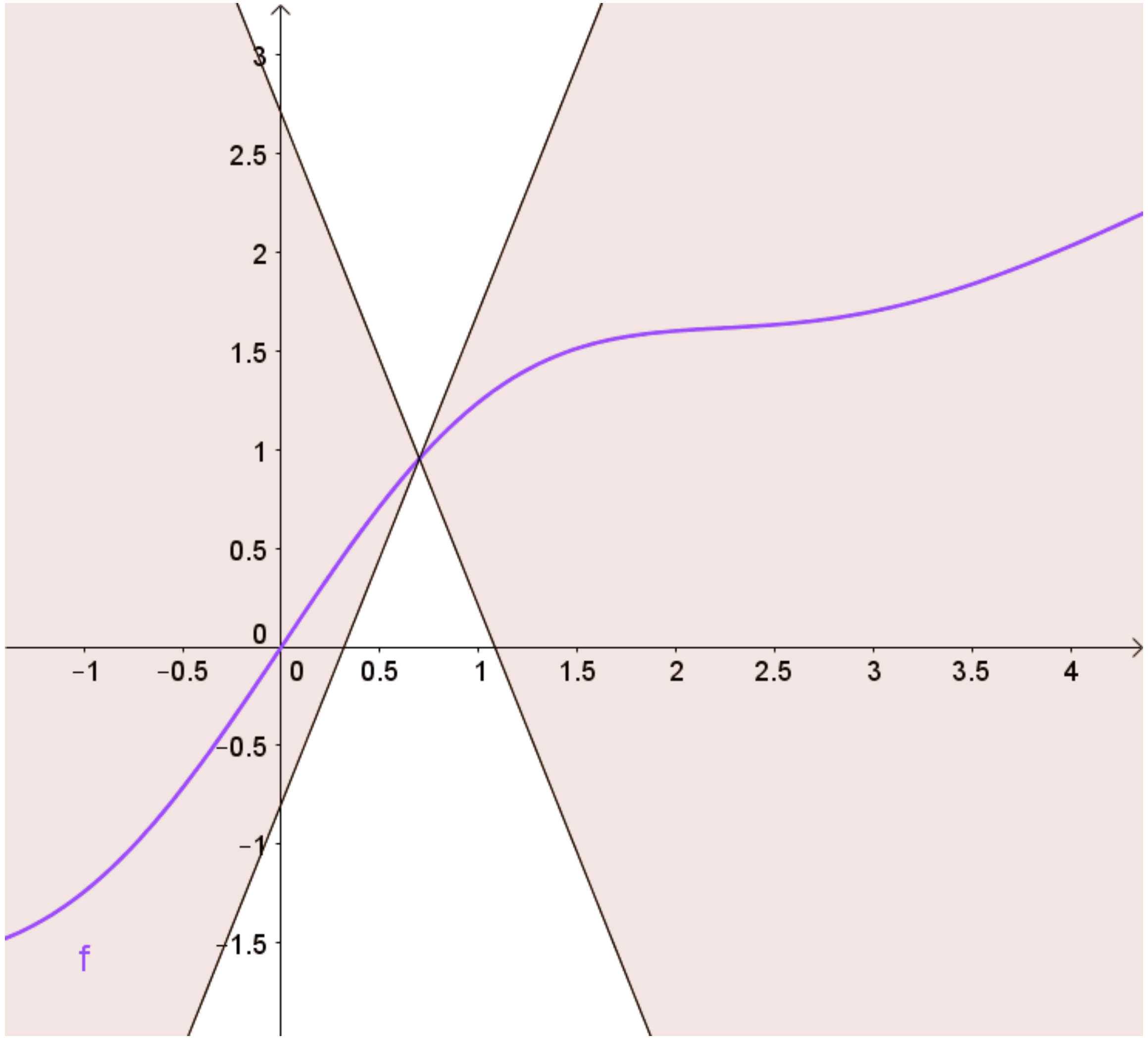


Earth Mover’s (Wasserstein) Distance

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma} \big[\|x - y\| \big] \ ,$$



Lipshitz Continuity



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

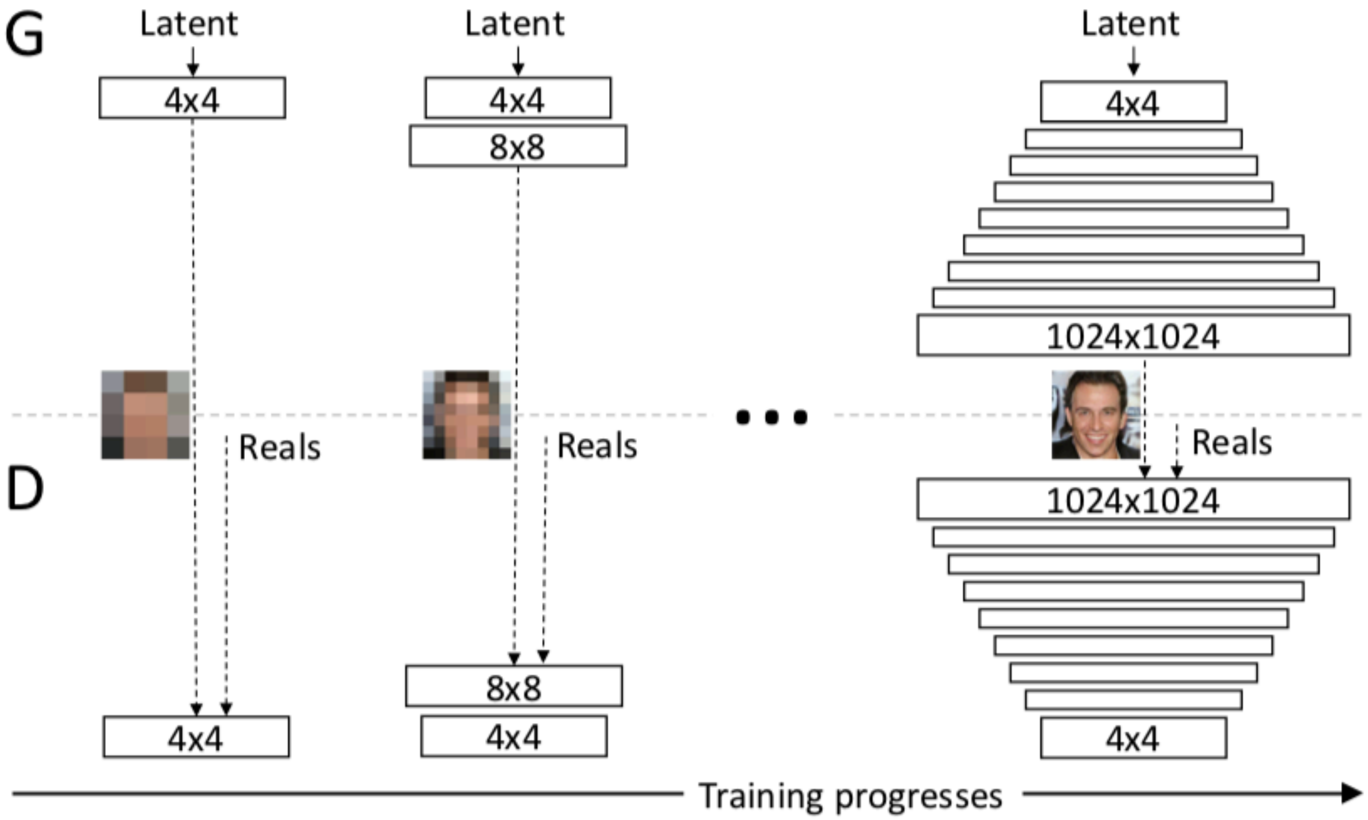
Gradient Penalty

$$L = \underbrace{\mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} \left[D(\tilde{\boldsymbol{x}}) \right] - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} \left[D(\boldsymbol{x}) \right]}_{\text{Original critic loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}} \sim \mathbb{P}_{\hat{\boldsymbol{x}}}} \left[(\| \nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}}) \|_2 - 1)^2 \right]}_{\text{Our gradient penalty}}.$$

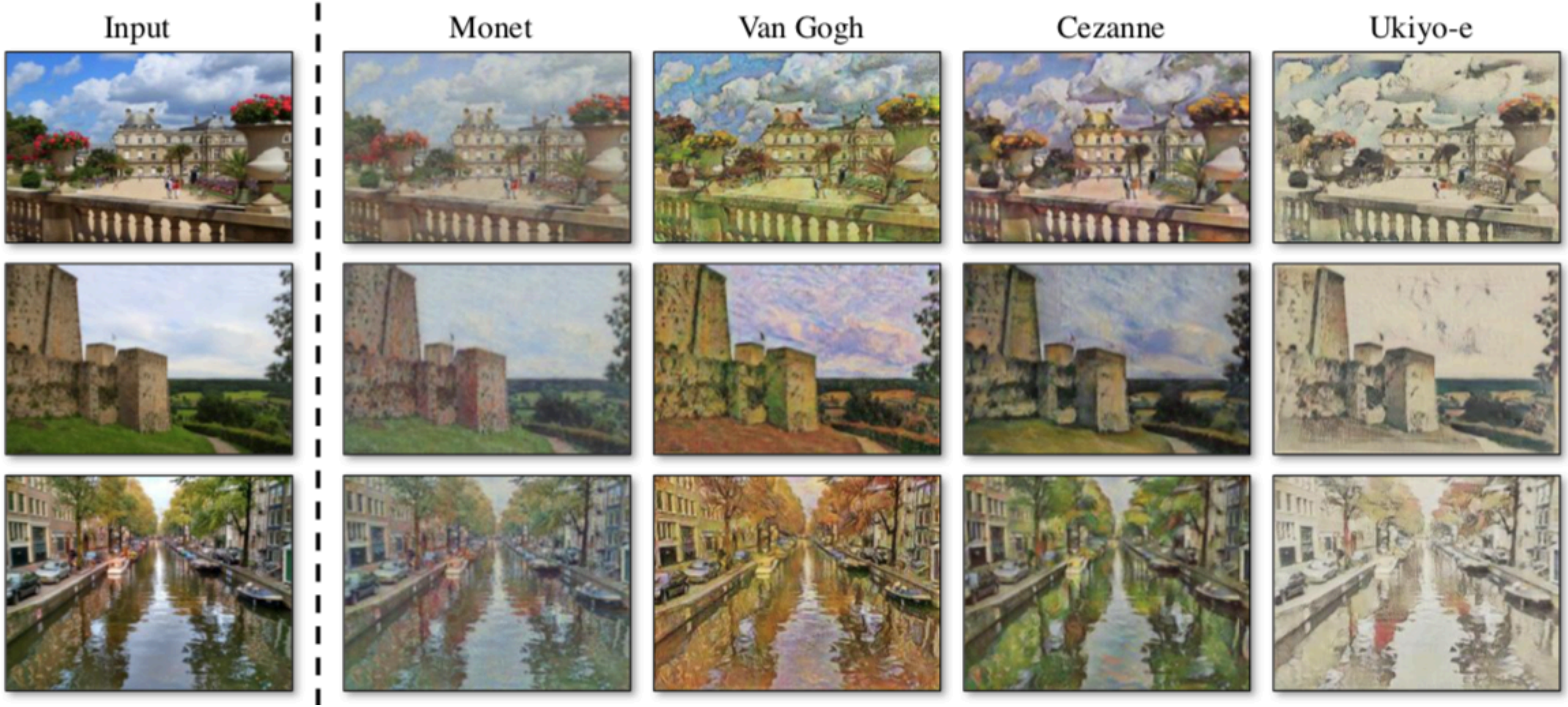
where $\hat{\boldsymbol{x}}$ ***sampled from*** $\tilde{x}$ ***and*** x ***with t uniformly sampled between 0 and 1***

$$\hat{\boldsymbol{x}} = t\tilde{\boldsymbol{x}} + (1-t)\boldsymbol{x} \text{ with } 0 \leq t \leq 1$$

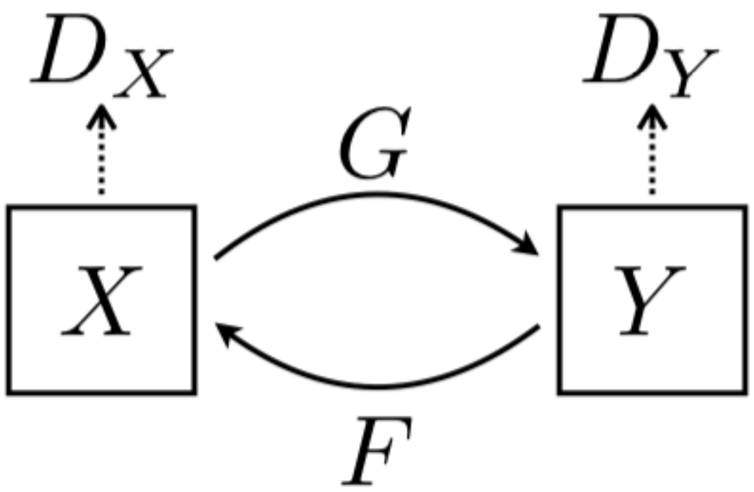
Progressive Growing of GANs



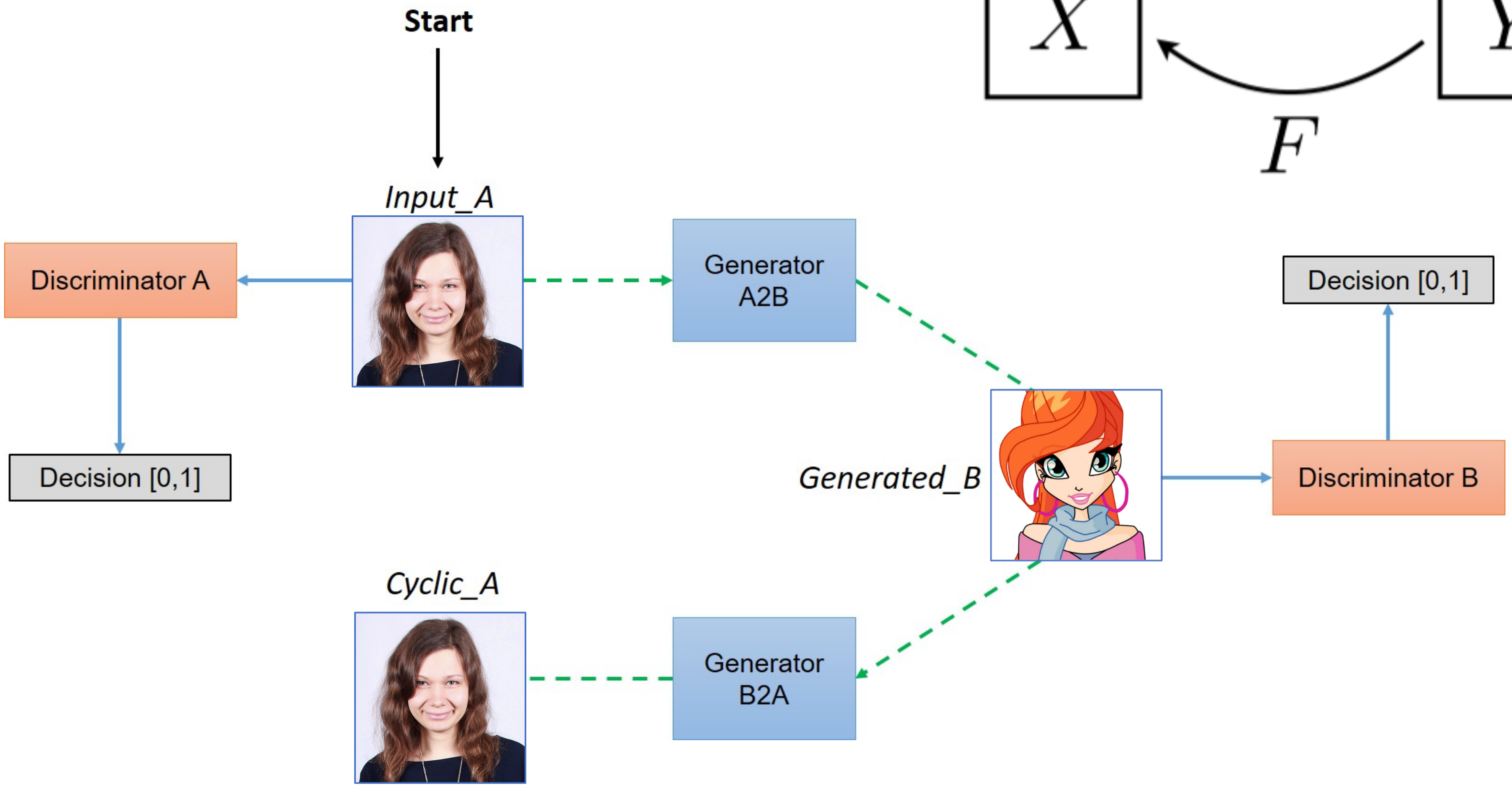
CycleGAN



CycleGAN



D_X
D_Y
G
X
Y
F



Start
Input_A
Discriminator A
Generator A2B
Decision [0,1]
Decision [0,1]
Generated_B
Discriminator B
Cyclic_A
Generator B2A

Questions

1) Wasserstein Distance description, what kind of problem is that supposed to overcome?
2) What is Gradient Penalty and for what should we use it?
3) The main idea of Progressive Growing of GANs
4) The main idea of CycleGAN

Bibliography

Generative Adversarial Networks (2 - 9). https://arxiv.org/pdf/1406.2661.pdf

Wasserstein GAN (10 - 14). https://arxiv.org/pdf/1701.07875.pdf

Improved Training of Wasserstein GANs (15 - 16). https://arxiv.org/pdf/1704.00028.pdf

Progressive Growing of GANs (17). https://arxiv.org/pdf/1710.10196.pdf

CycleGAN (18 - 19). https://arxiv.org/pdf/1703.10593.pdf

Main (2 - 16). https://medium.com/@jonathan_hui/gan-wasserstein-gan-wgan-gp-6a1a2aa1b490