Variance Regularizing Adversarial Learning

Karan Grewal, Devon Hjelm, Yoshua Bengio

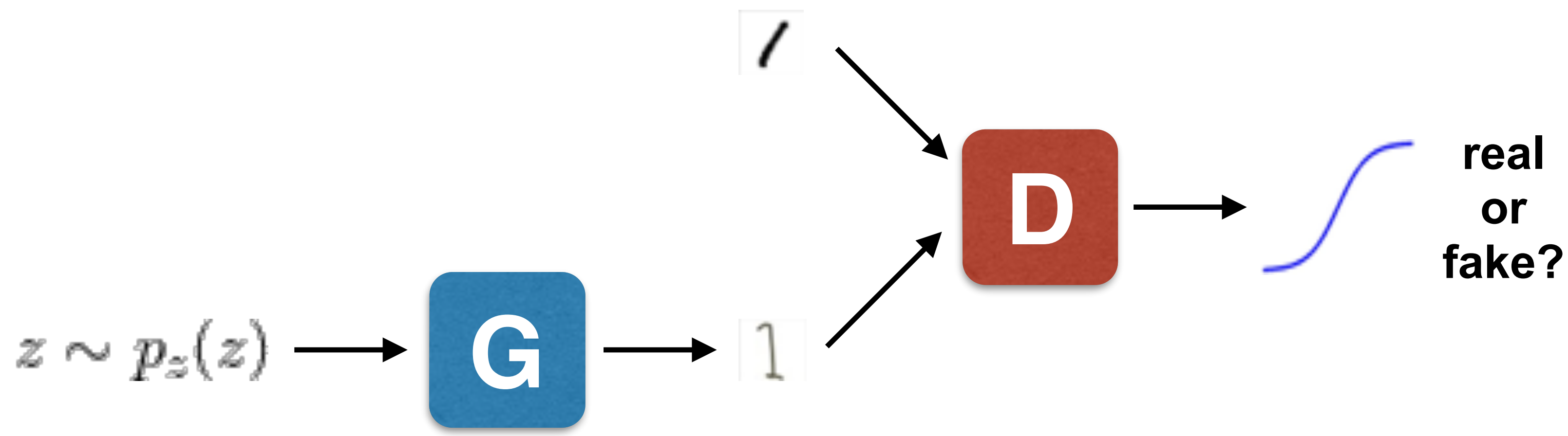
MILA

Overview

1. GANs
2. Problems with training GANs
3. Lipschitz Discriminators & VRAL
4. Empirical Results

Generative Adversarial Networks

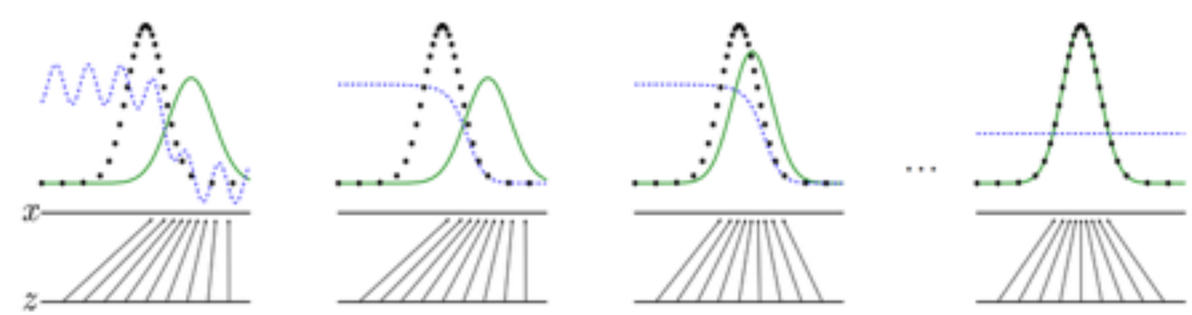
Architecture:



Value Function:

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

What's really happening:



Goodfellow et. al, 2014

Generative Adversarial Networks

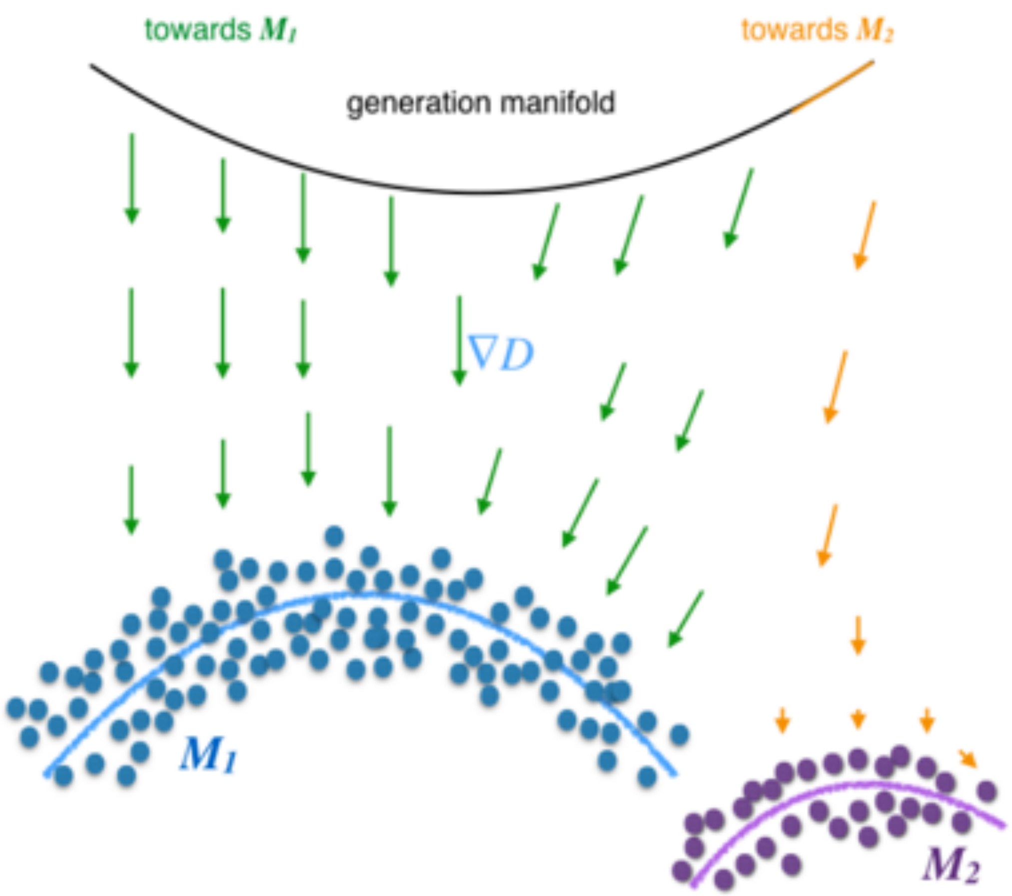
Original

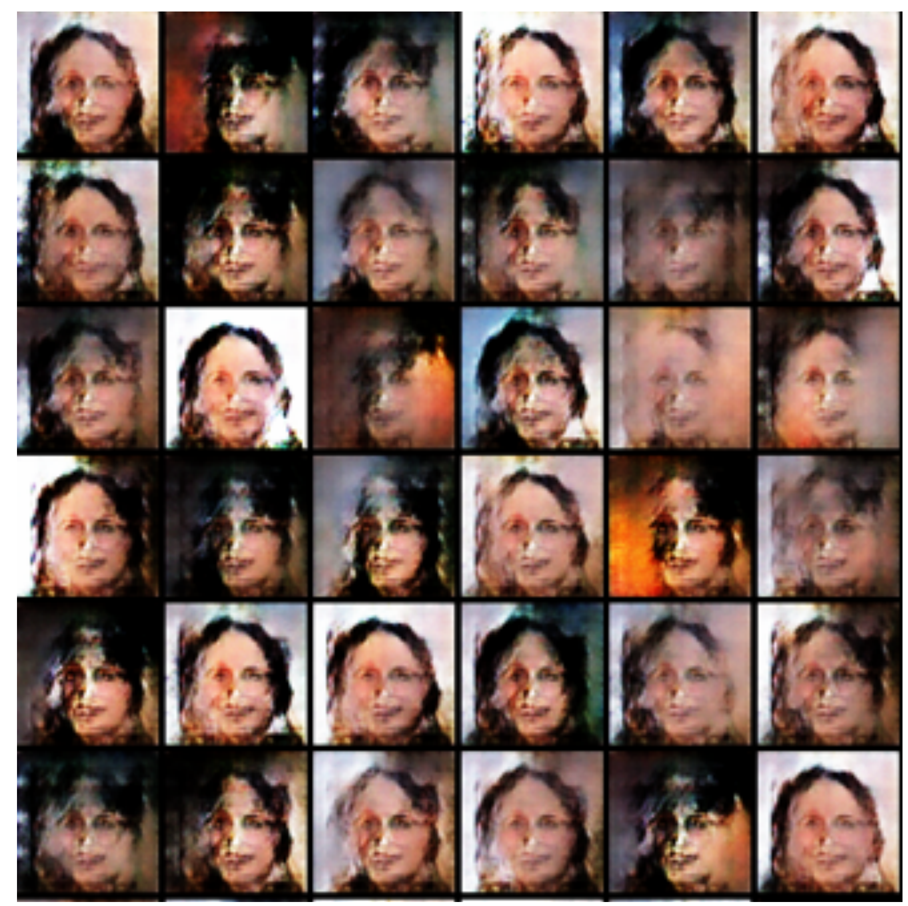
Reconstructions

Dumoulin et. al, 2016

Problems with Training GANs

1. Mode Collapse

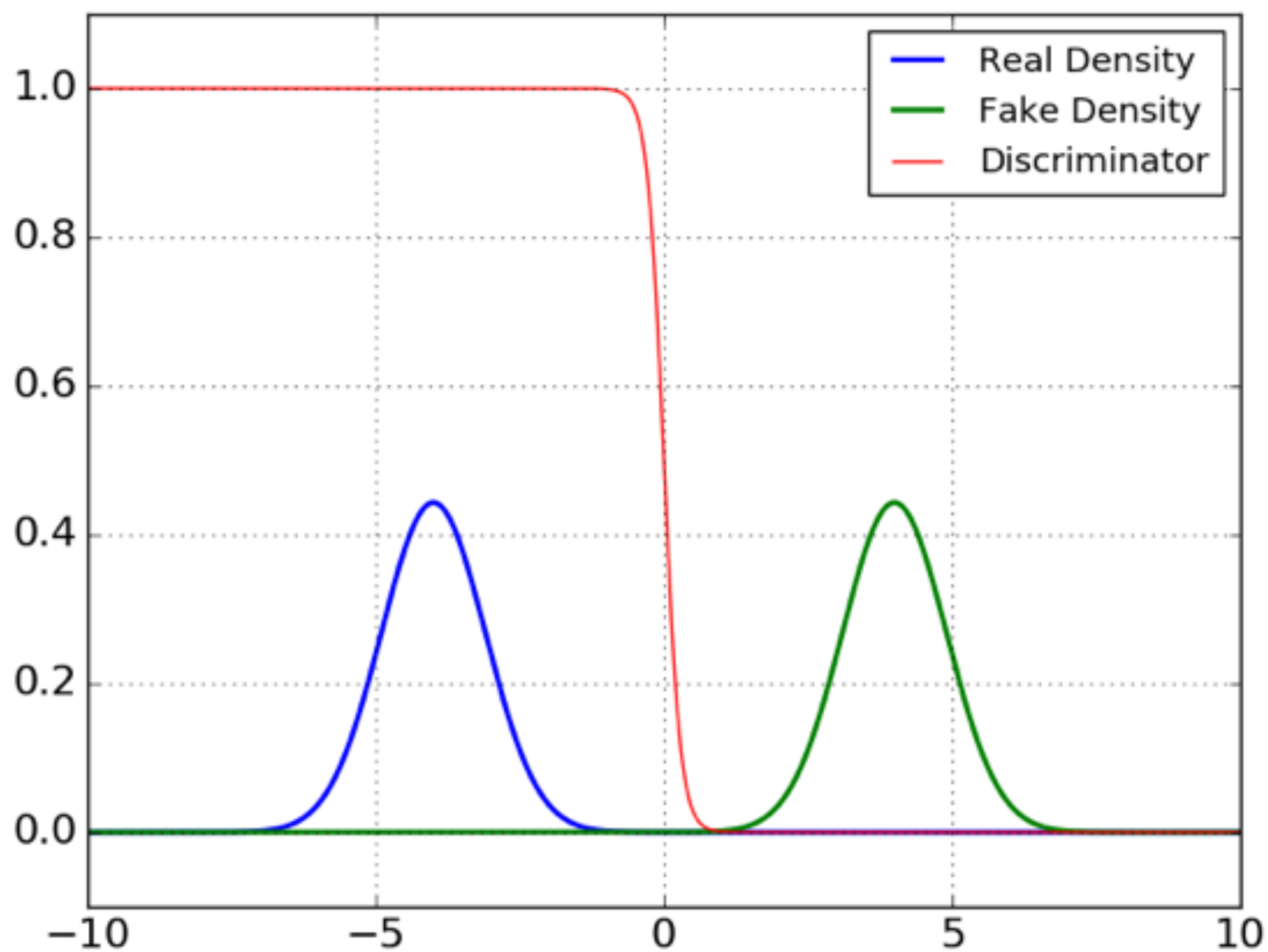




Salimans et. al, 2016; Che et. al, 2016

Problems with Training GANs

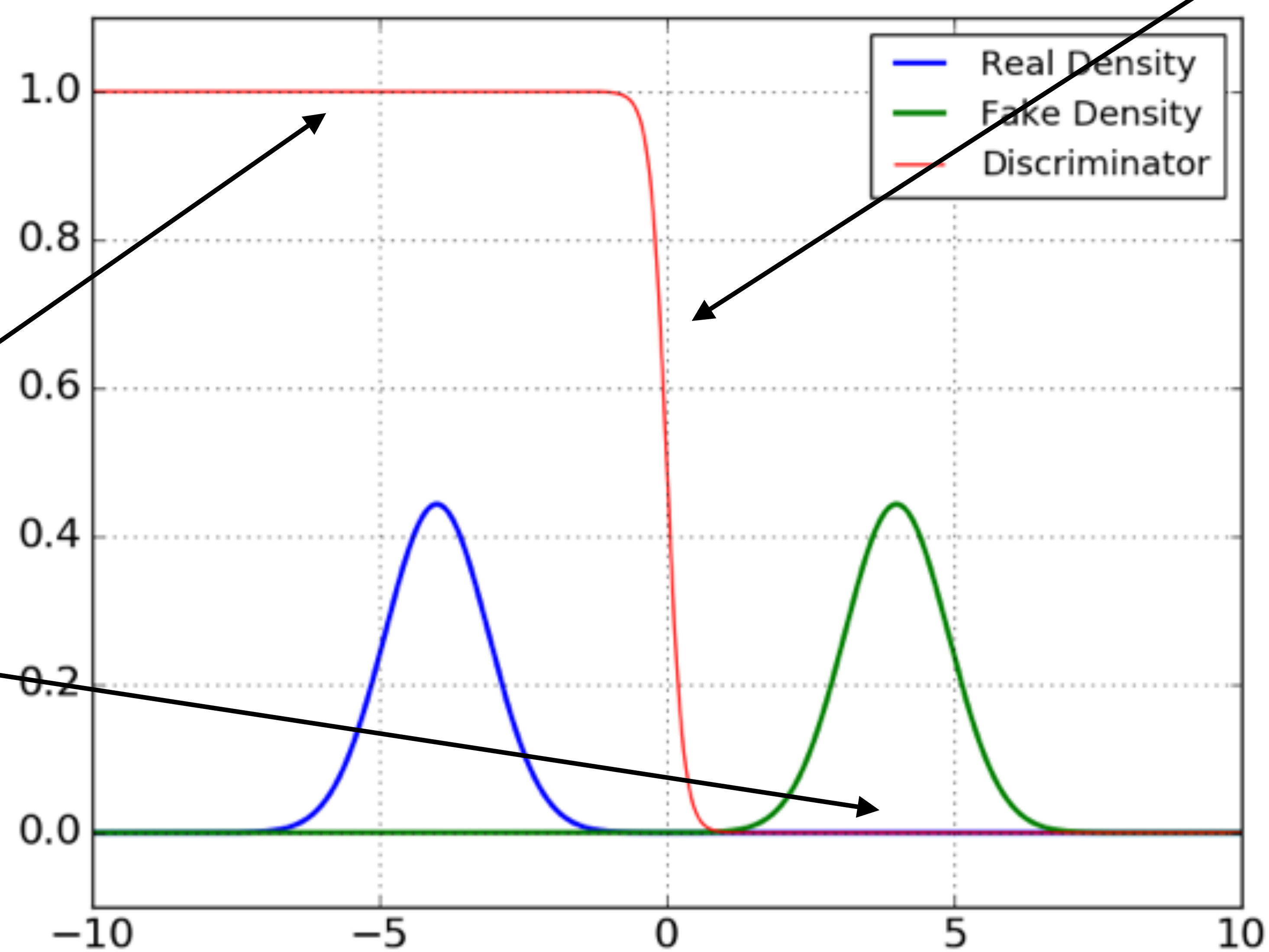
2. Problematic Gradients



Problems with Training GANs

2. Problematic Gradients

$||\nabla D||$ explodes!



$||\nabla D||$ vanishes!

The Lipschitz Constraint

A function $f : X \to \mathbb{R}$ is K-Lipschitz if for every $x_1, x_2 \in X$, f satisfies

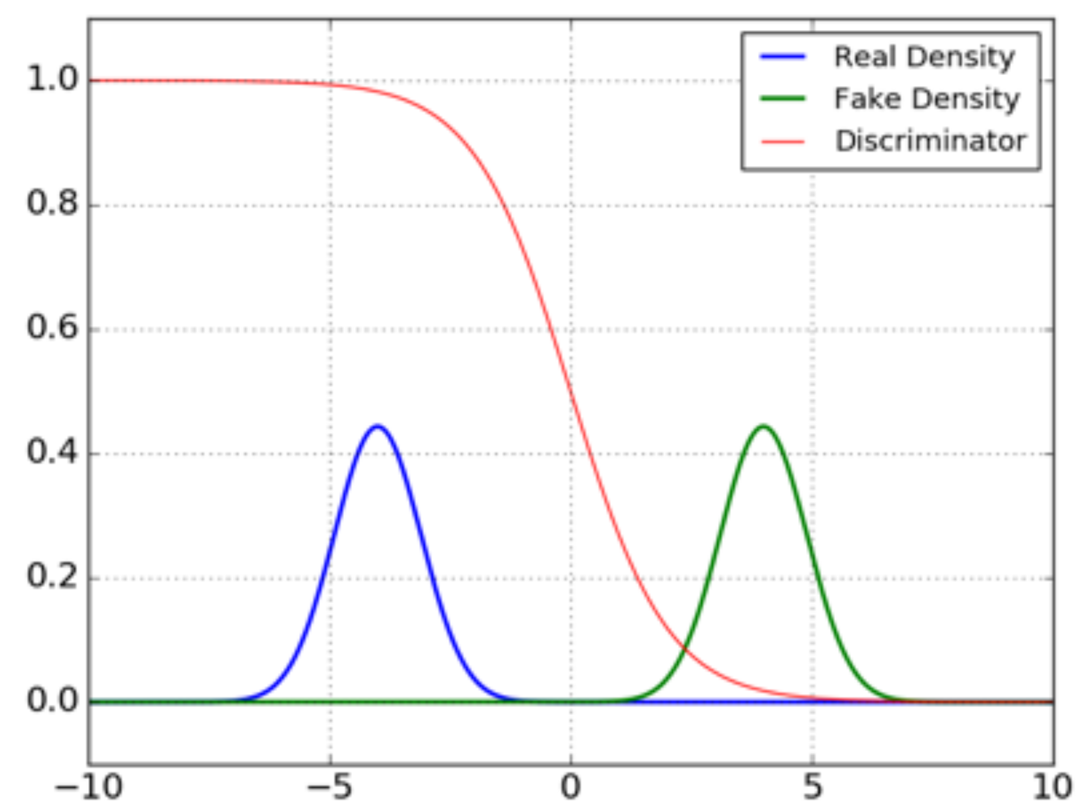
$$\frac{||f(x_1) - f(x_2)||}{||x_1 - x_2||} \leq K$$

The Lipschitz Constraint

A function $f : X \to \mathbb{R}$ is K-Lipschitz if for every $x_1, x_2 \in X$, f satisfies

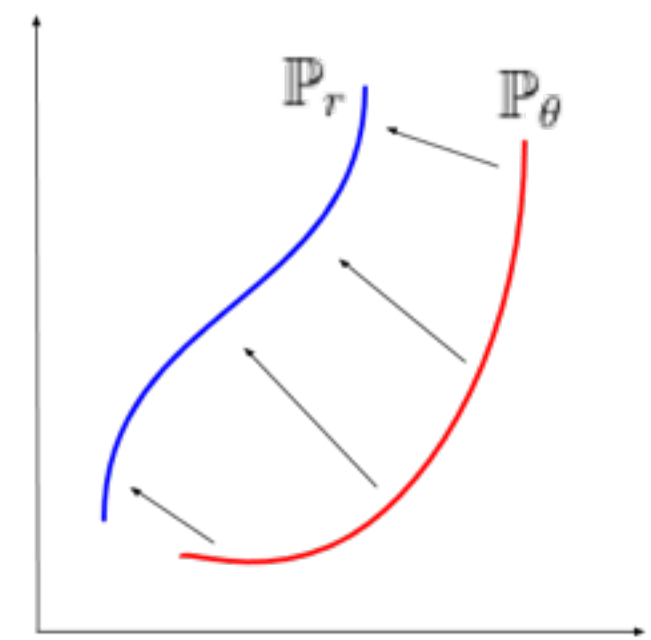
$$\frac{||f(x_1) - f(x_2)||}{||x_1 - x_2||} \leq K$$

1-Lipschitz Discriminator:



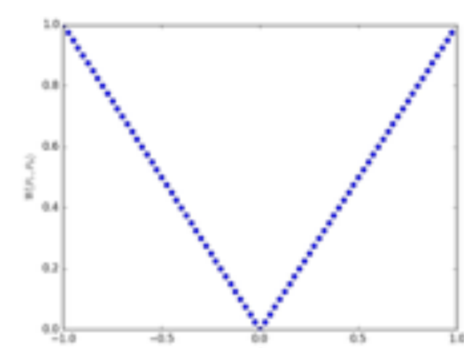
Wasserstein GAN

- Real data lies on a low-dimensional manifold in a high-dimensional space, $\mathbb{P}_r$

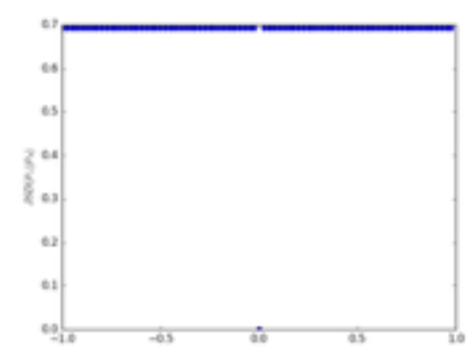


- Jenson-Shannon and KL divergences are not meaningful

- Use Wasserstein-1, or “earth-mover’s” objective instead



Wasserstein-1



JS divergence

Arjovsky et. al, 2017

Wasserstein GAN

- Objective: $$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim \mathbb{P}_r}[f(x)] - \mathbb{E}_{x \sim \mathbb{P}_\theta}[f(x)]$$
- Use weight clipping to enforce Lipschitz constraint

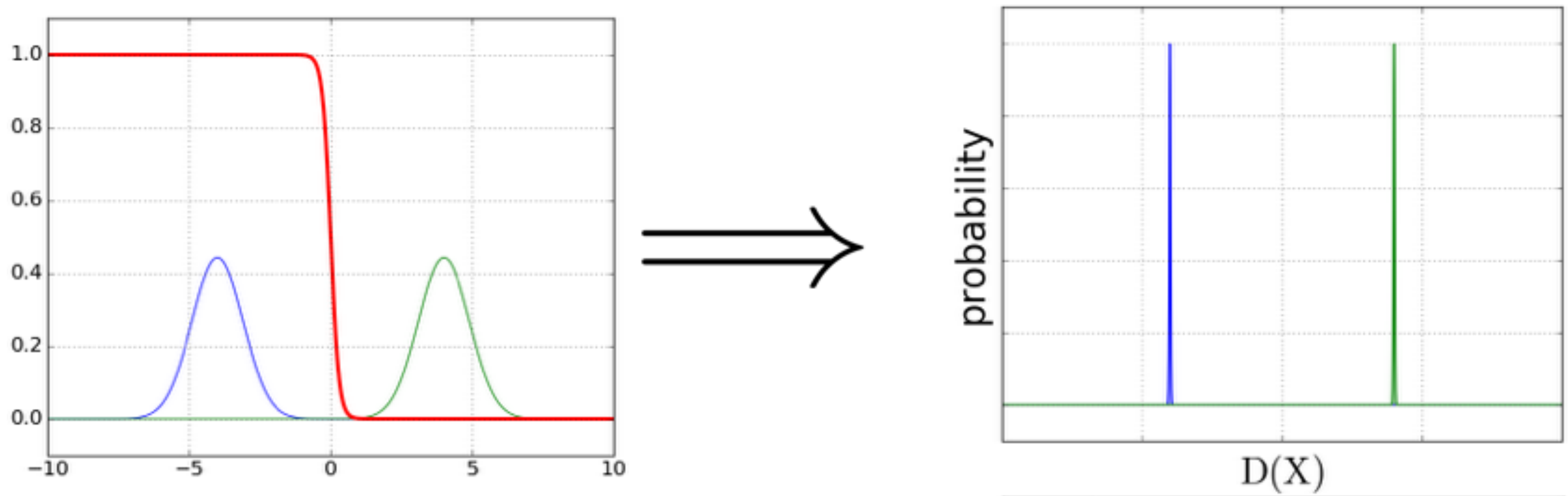
Wasserstein GAN

Standard GAN

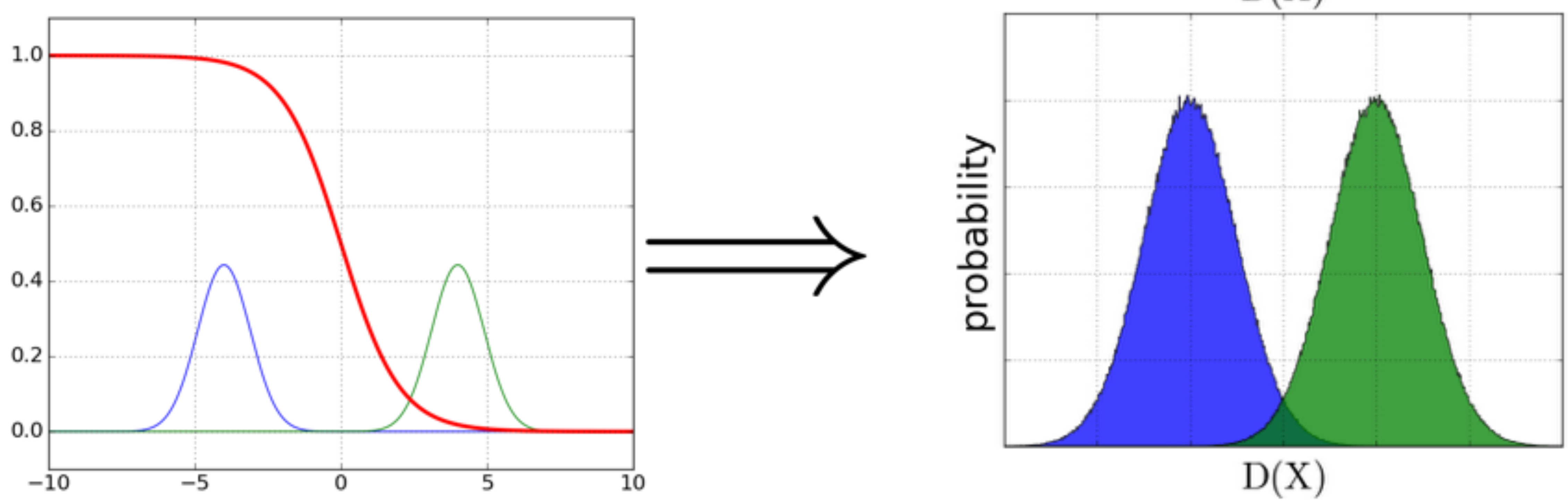
Arjovsky et. al, 2017

Another Interpretation: Variance

non-Lipschitz:

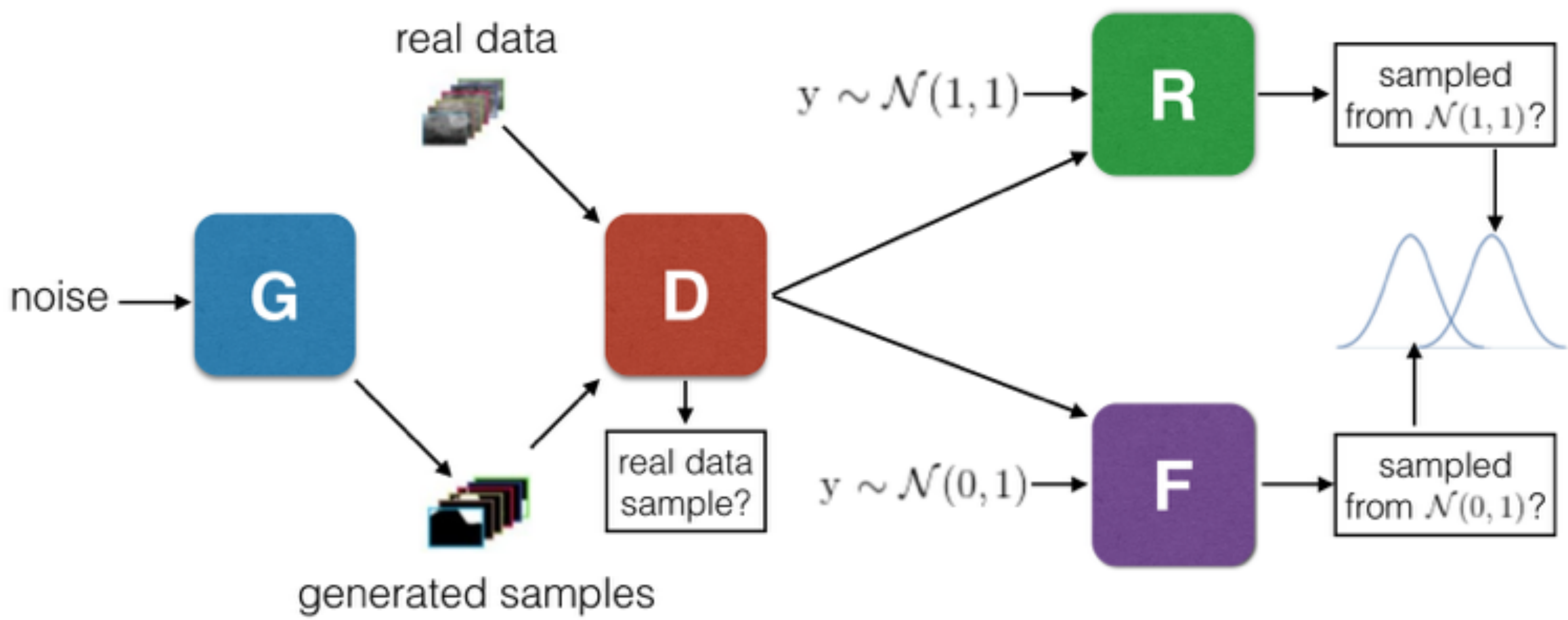


Lipschitz:



Variance regularized GANs: meta-discriminators

- 3 adversarial games:
    1. ***G*** tries to fool ***D*** by creating real-looking samples
    2. ***D*** tries to fool ***R*** by mimicking $\mathcal{N}(1,1)$ for real samples
    3. ***D*** tries to fool ***F*** by mimicking $\mathcal{N}(0,1)$ for fake samples



Variance regularized GANs: meta-discriminators

Objective #1:

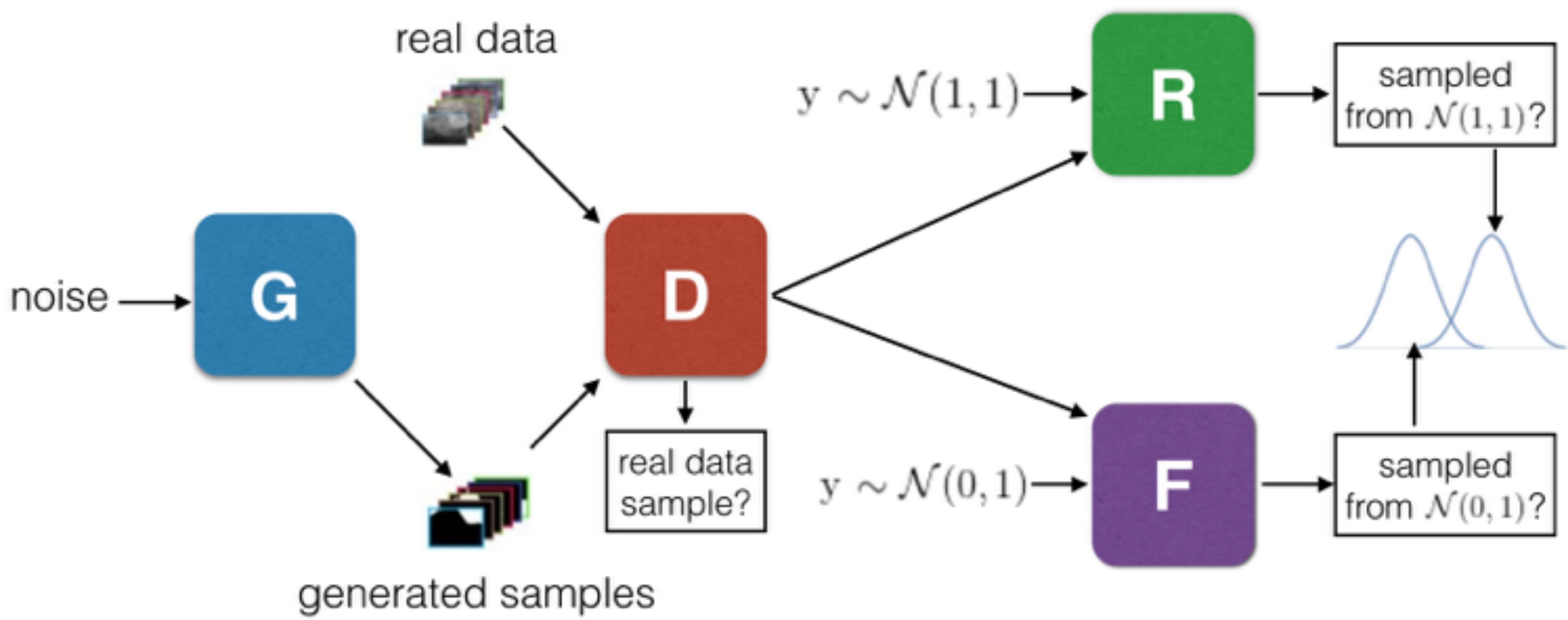
$$\min_G \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}[(D(G(\mathbf{z}))-1)^2] \quad \text{or} \quad \min_G \left(\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}[D(G(\mathbf{z}))] - \mathbb{E}_{\mathbf{x}\sim p_{data}}[D(\mathbf{x})]\right)^2$$

Objective #2:

$$\min_D \max_F V_F(F, D, G) = \mathbb{E}_{y\sim\mathcal{N}(0,1)}[\log F(y)] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}[\log(1 - F(D(G(\mathbf{z}))))]$$

Objective #3:

$$\min_D \max_R V_R(R, D) = \mathbb{E}_{y\sim\mathcal{N}(1,1)}[\log R(y)] + \mathbb{E}_{\mathbf{x}\sim p_{data}}[\log(1 - R(D(\mathbf{x})))]$$



Variance regularized GANs: density estimators

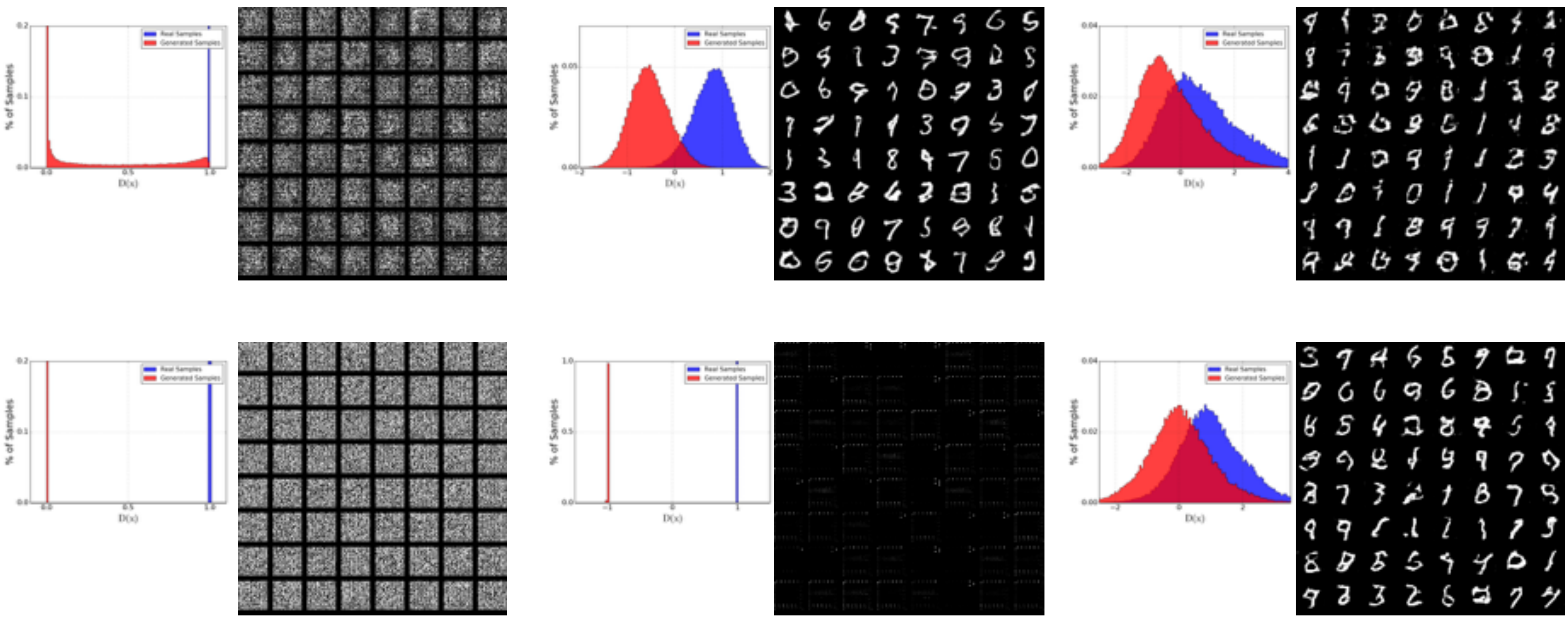
- Minimize the KL-divergence between ***D***'s normalized output distribution and $\mathcal{N}(0,1)$ or $\mathcal{N}(1,1)$

- Use a parzen-window density estimator to approximate ***D***'s normalized output distribution, $\tilde{p}_D$

$$\mathcal{L}_D = \underbrace{\mathbb{E}_{\mathbf{z}\sim p_z}[KL(\mathcal{N}(0,1)||\tilde{p}_D(G(\mathbf{z})))]}_{\text{fit } D\text{'s output given fake samples to a gaussian}} + \underbrace{\mathbb{E}_{\mathbf{x}\sim p_{data}}[KL(\mathcal{N}(1,1)||\tilde{p}_D(\mathbf{x}))]}_{\text{fit } D\text{'s output given real samples to a gaussian}}$$

Well-trained Discriminators

	Standard GAN	Least-Squares GAN	VRAL, meta-disc.
1:1			
50:1			



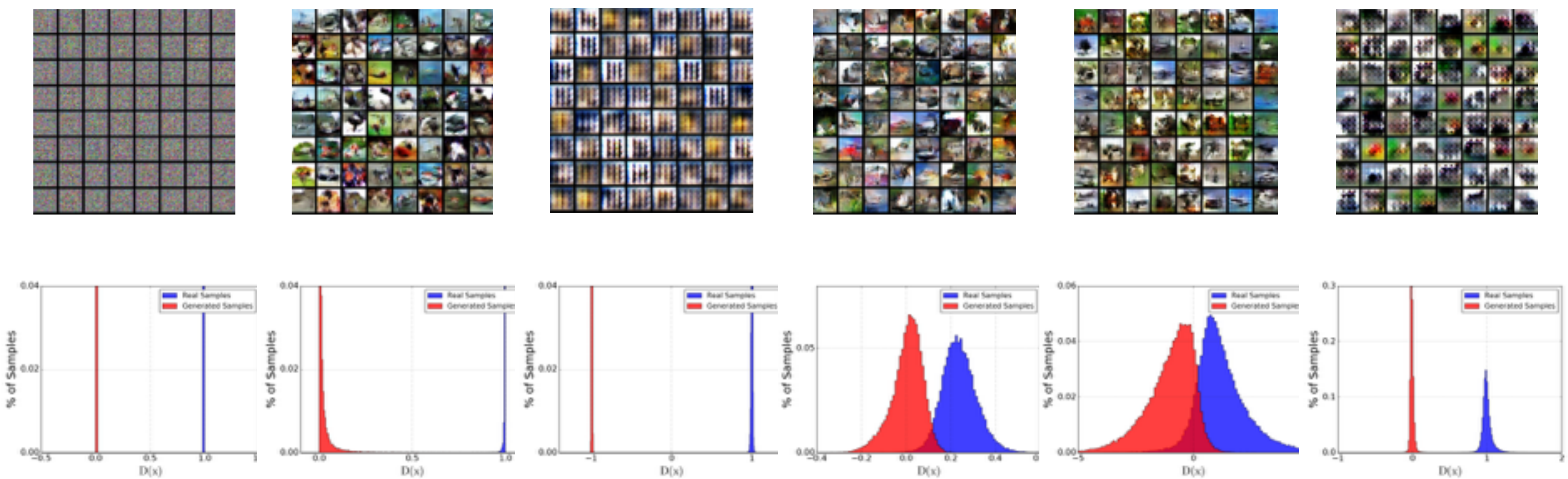
Why Large Training Ratios

- The more the discriminator is trained, the more reliable the learning signal
- If the discriminator becomes too strong, the generator may not learn at all
- **Goal:** ensure training methods are robust against large training ratios (e.g. 50 discriminator updates per generator update)

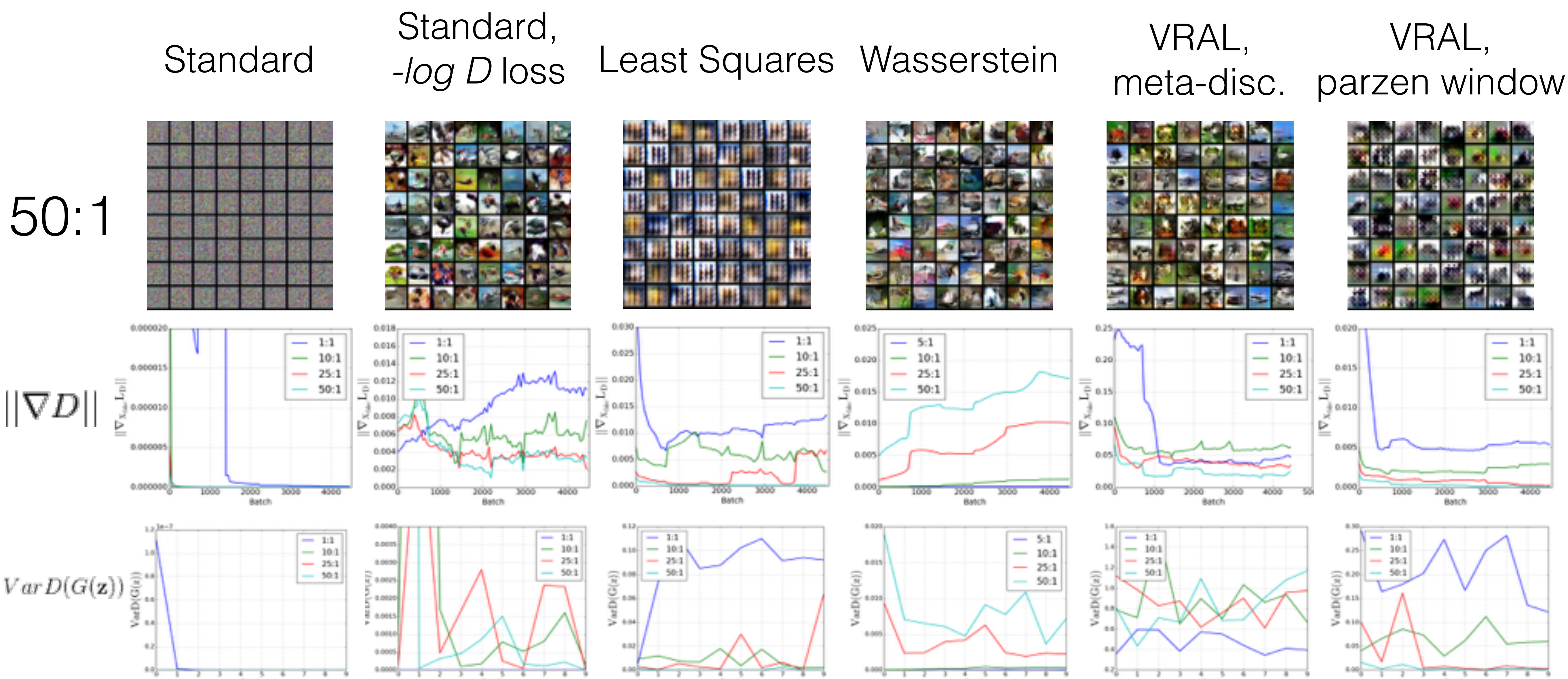
Learning & ***D*** output

Standard | Standard, *-log D* loss | Least Squares | Wasserstein | VRAL, meta-disc. | VRAL, parzen window

50:1



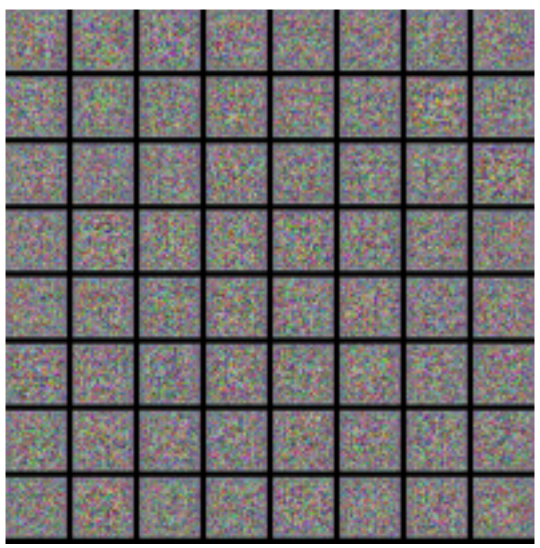
Learning & Gradients

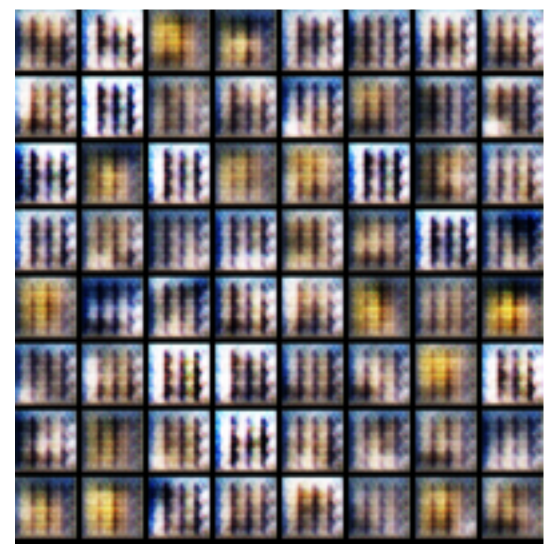


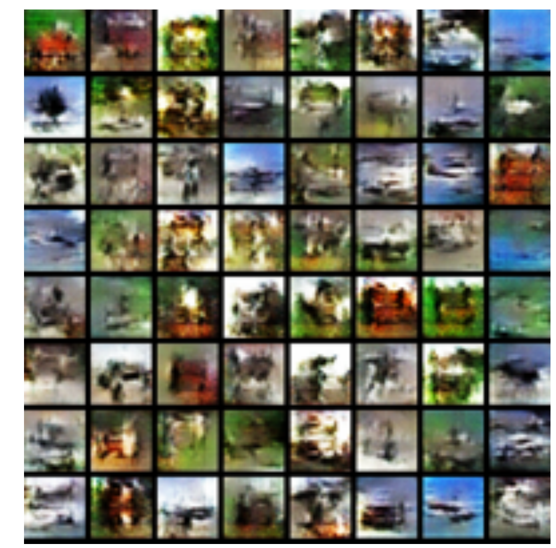
Learning & Gradients

Standard | Least Squares | VRAL, meta-disc.

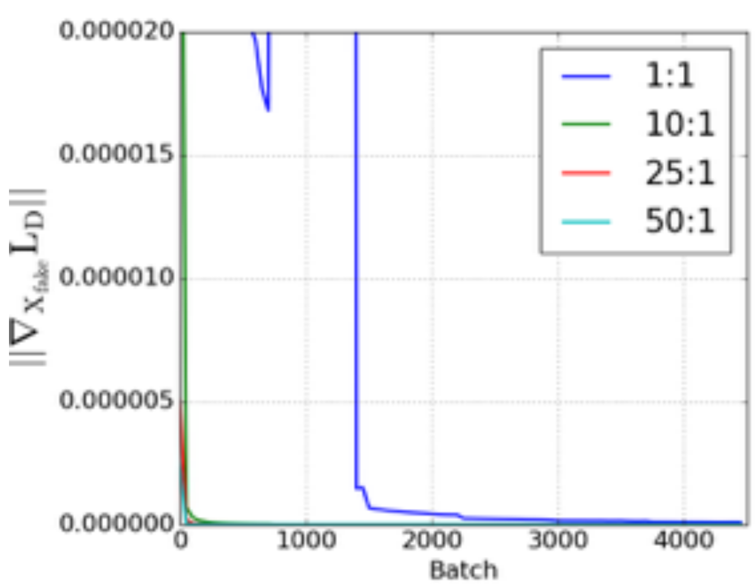
50:1

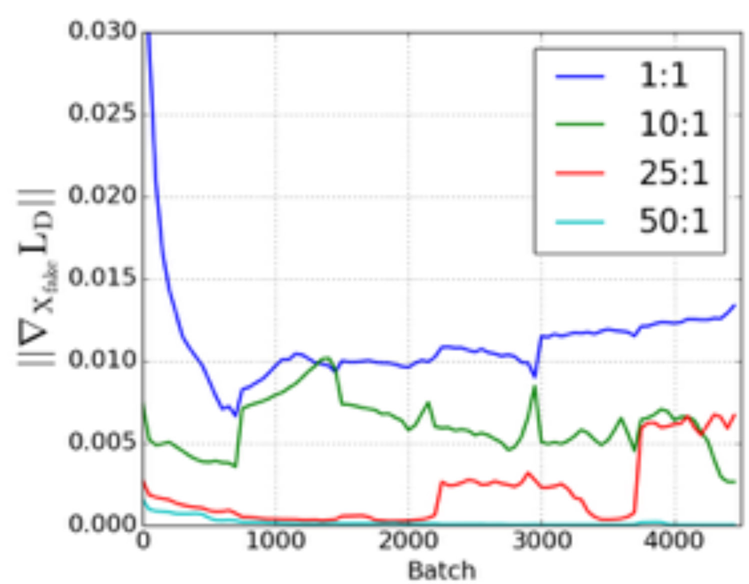


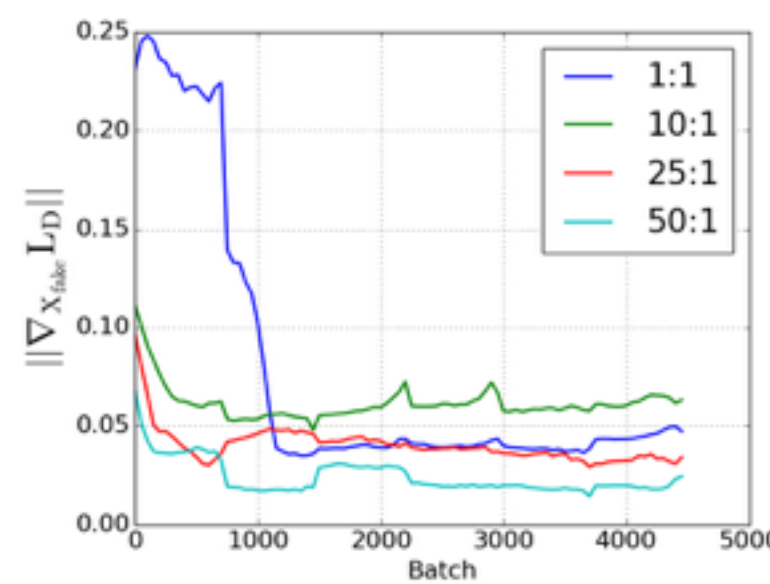




$|\nabla D||$







Alternatives to Weight Clipping

- Add noise to intermediate layers of the discriminator to promote variance

- Weight clipping leads to unstable gradient norms, instead use the following gradient penalty:

$$\mathbb{E}_{\mathbf{x}\sim\hat{p}_{\mathbf{x}}}[(||\nabla_x D(\mathbf{x})|| - 1)^2],$$

$$\hat{p}_{\mathbf{x}}(\mathbf{x}) = (1-t)\cdot p_{data}(\mathbf{x}) + t \cdot p_g(\mathbf{x})$$

Salimans et. al, 2016; Gulrajani et. al, 2017

Conclusion

1. Overly strong discriminators emit vanishing gradients, making it difficult for the generator to learn

2. The Lipschitz constraint can be interpreted as forcing the discriminator to have variance in its output

3. Mode-matching allows a generator to learn the data distribution/manifold in the presence of a well-trained discriminator

Future Work

1. Are we actually enforcing a Lipschitz function by regularizing the discriminator?

2. Why our proposed method fails under certain choices of hyperparameters

Collaborators

Devon Hjelm

Yoshua Bengio